Part Models and Pixel Labeling

Computer Vision

CS 543 / ECE 549

University of Illinois

Derek Hoiem

Influential Works in Detection

- Sung-Poggio (1994, 1998) : ~2412 citations
  - Basic idea of statistical template detection, bootstrapping to get "face-like" negative examples, multiple whole-face prototypes (in 1994)
- Rowley-Baluja-Kanade (1996-1998) : ~4953
  - "Parts" at fixed position, neural network based detector, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Viola-Jones (2001, 2004) : ~27,000
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : ~18000
  - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-Huttenlocher (2000): ~2100
  - Efficient way to solve part-based detectors
- Felzenszwalb-McAllester-Ramanan DPM (2008,2010): ~7200
  - Excellent template/parts-based blend
- Girshick-Donahue-Darrell-Malik R-CNN (2014-): ~4700
  - Region proposals + fine-tuned CNN features (marks significant advance in accuracy over hog-based methods)

CNN (Deep Learning) for Classification

- Operations
  - Convolution with learned filters, ReLU
    - Note: after first layer, filters operate on high dimensional feature maps, not intensities
  - Spatial pooling and downsample 2x
  - "Bottlenecks" to limit parameters
  - "Skip connections" to simplify optimization, enable ensemble behavior
  - Classification (with multilabel or softmax loss)

- Tricks to train
  - Batchnorm
  - ADAM / momentum
  - Data augmentation
  - Set parameters appropriately (weight initialization, learning rate schedule, momentum, weight decay)

Each filter: 3x3x512

Final feature map: 7x7x512

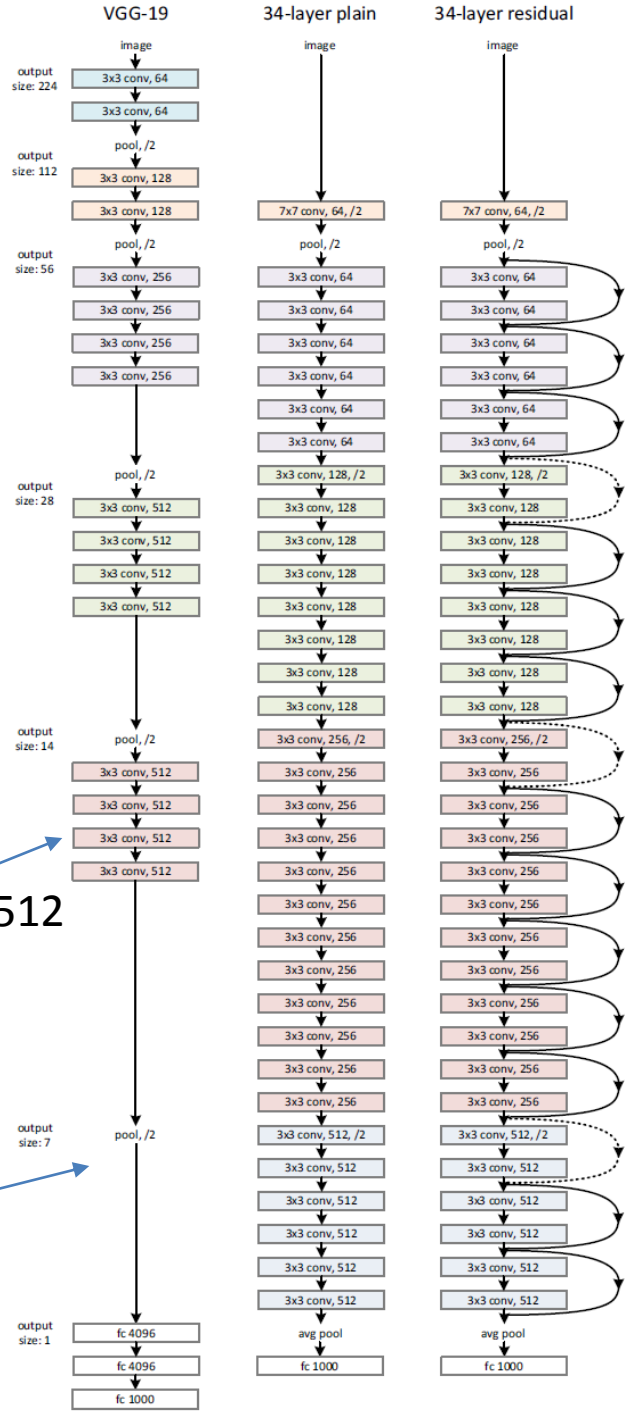
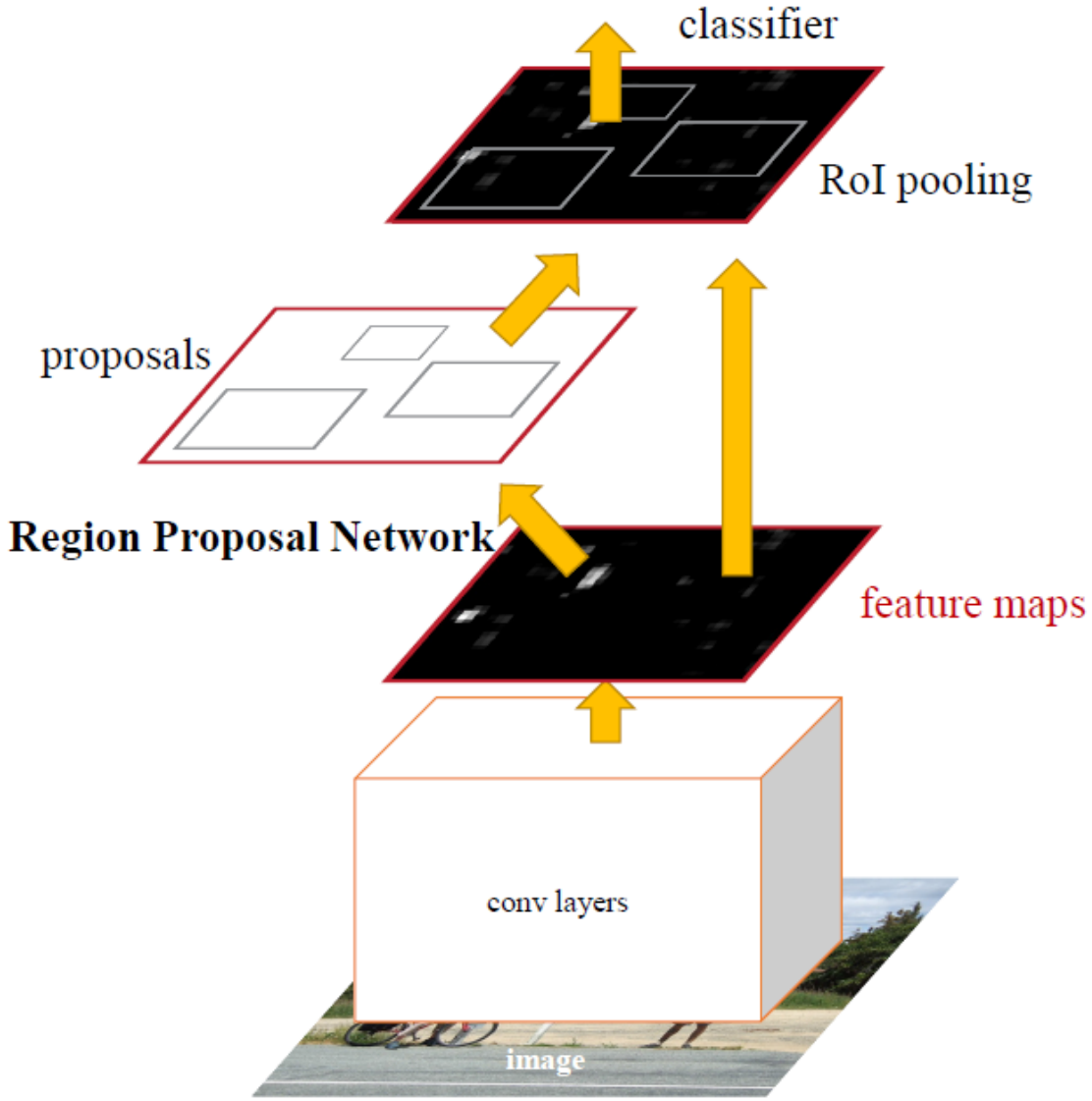


Fig: Deep Residual Learning for Image Recognition
He et al. CVPR 2006

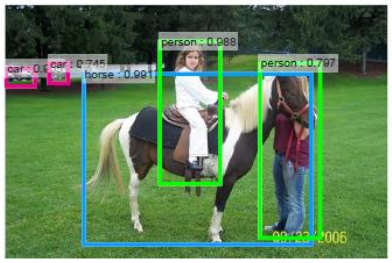
CNN for Detection

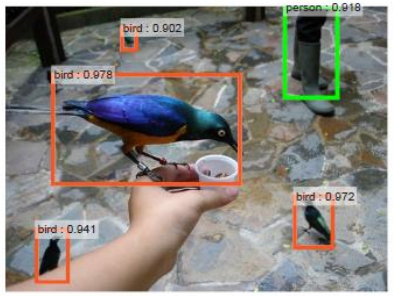
- Classifier network produces a set of feature maps
- Each cell proposes bounding boxes that might be objects
- Features are pooled into bbox regions and classified into object categories or background

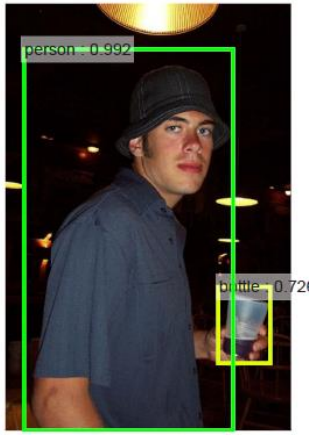


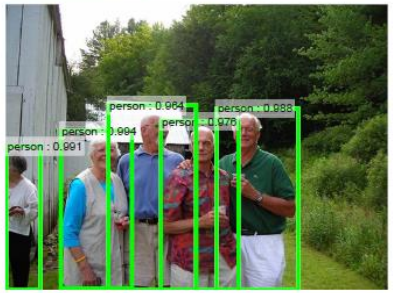
Faster R-CNN (Ren et al. 2016)

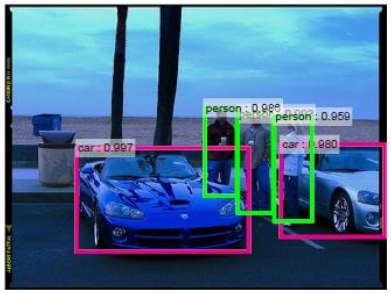
Object bounding box detections

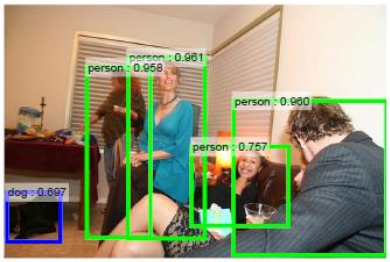


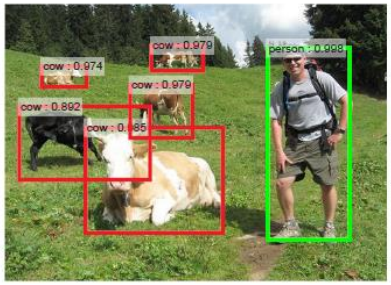












Faster RCNN detections

Today’s class

- Object part models

- Pixel labeling

Part/keypoint Prediction

http://mscoco.org/dataset/#keypoints-challenge2016

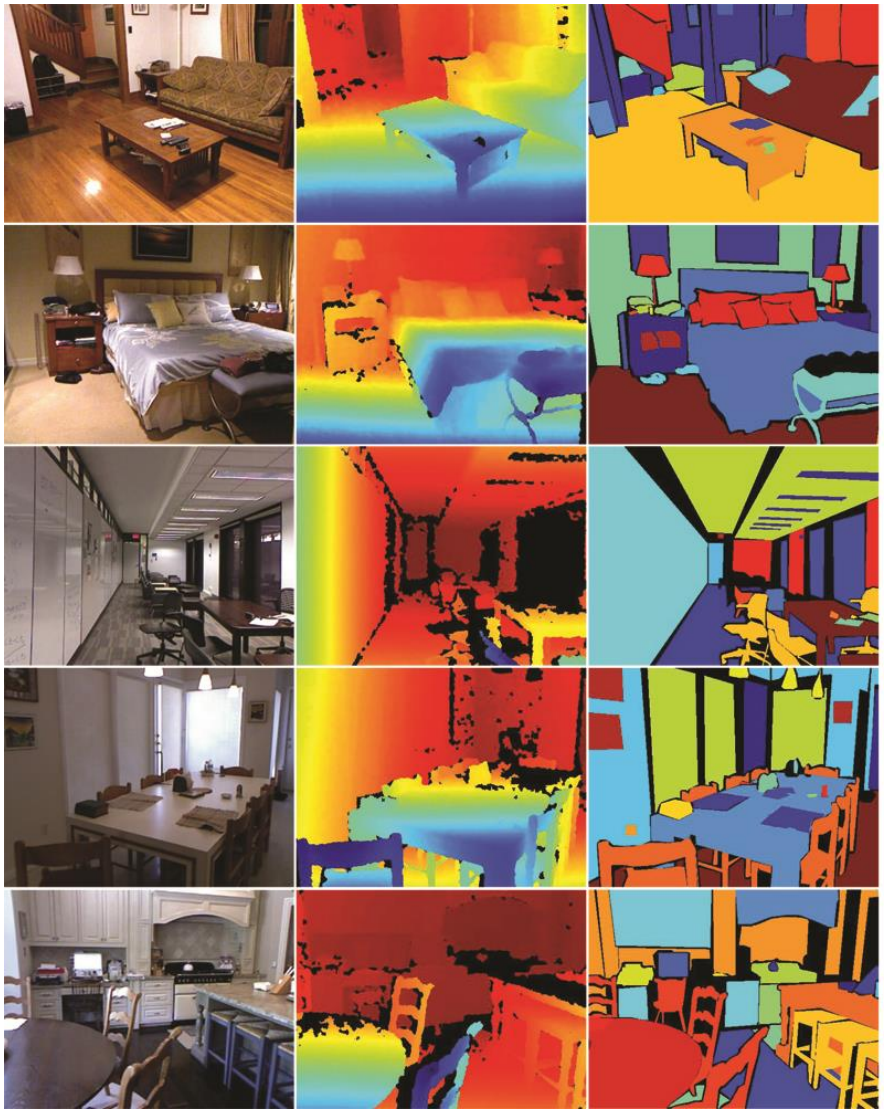
Semantic Segmentation

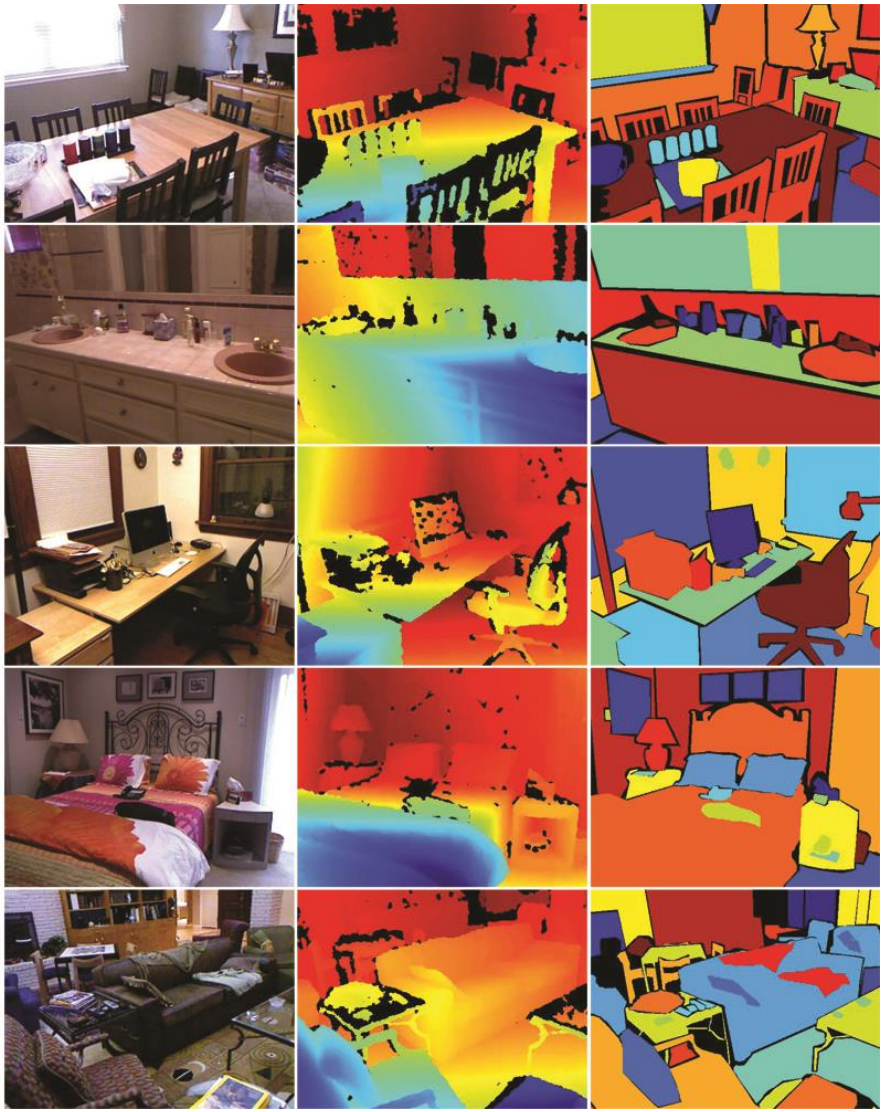


http://mscoco.org/dataset/#detections-challenge2016

https://www.cityscapes-dataset.com/examples/#fine-annotations

Semantic Segmentation





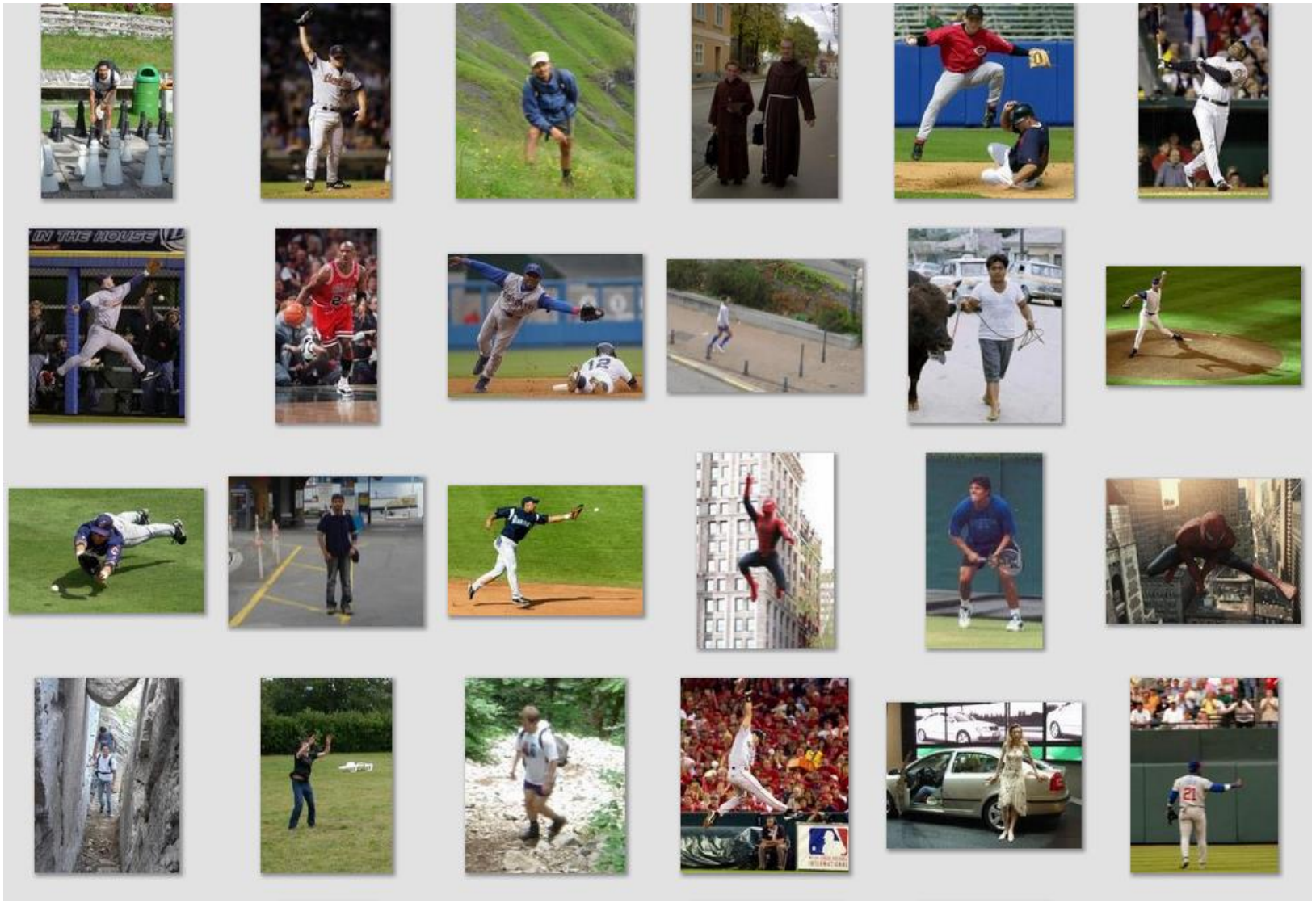
http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Deformable objects

Images from Caltech-256

Slide Credit: Duan Tran

Deformable objects



Images from D. Ramanan's dataset

Slide Credit: Duan Tran

Compositional objects

Parts-based Models

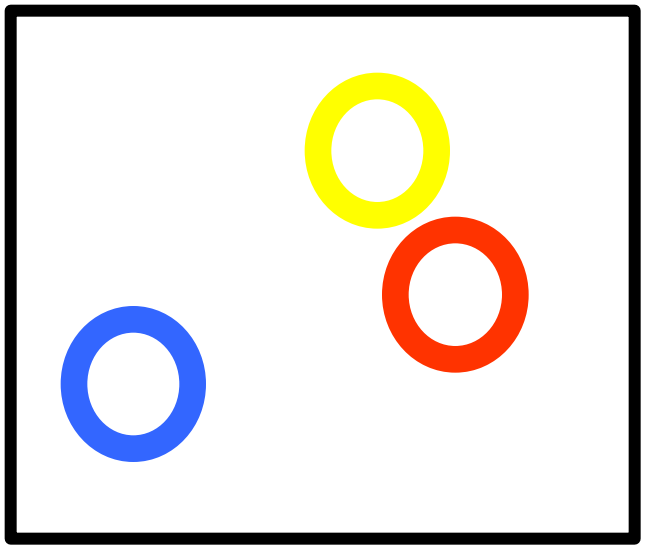
Define object by collection of parts modeled by

1. Appearance
2. Spatial configuration

Slide credit: Rob Fergus

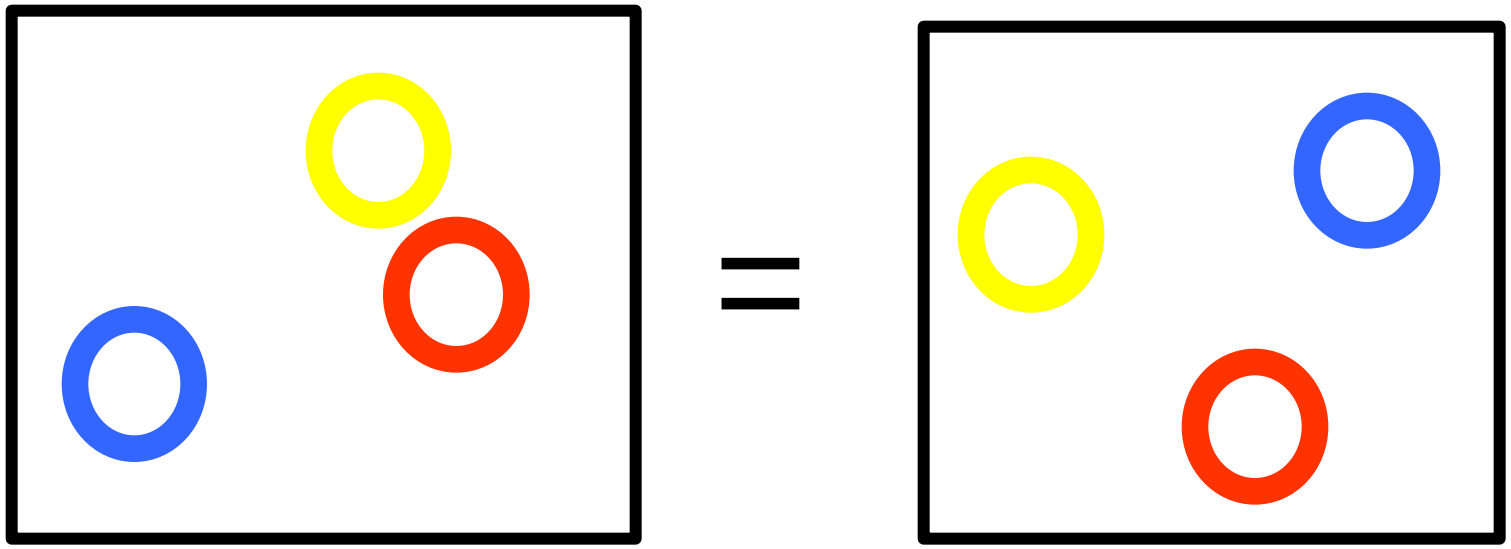
How to model spatial relations?

- One extreme: fixed template



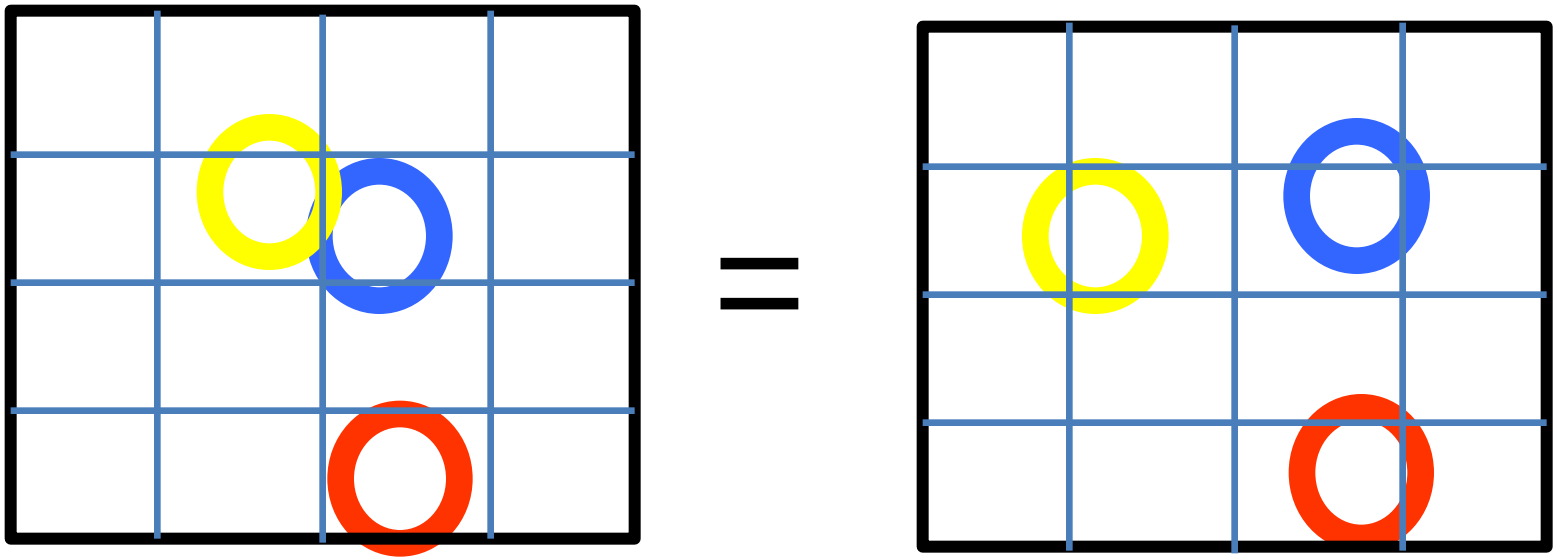
How to model spatial relations?

- Another extreme: bag of words



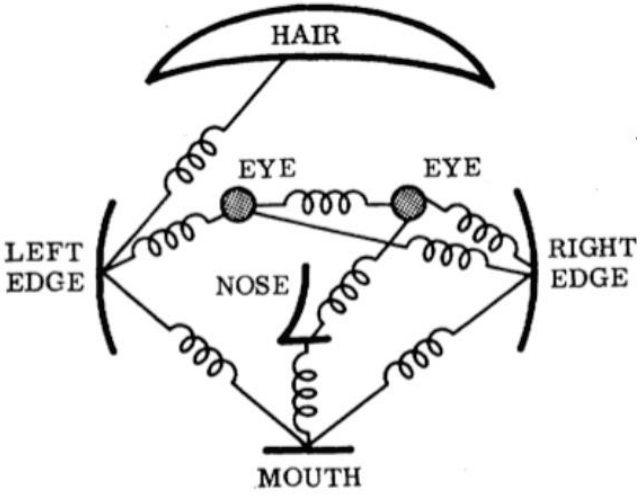
How to model spatial relations?

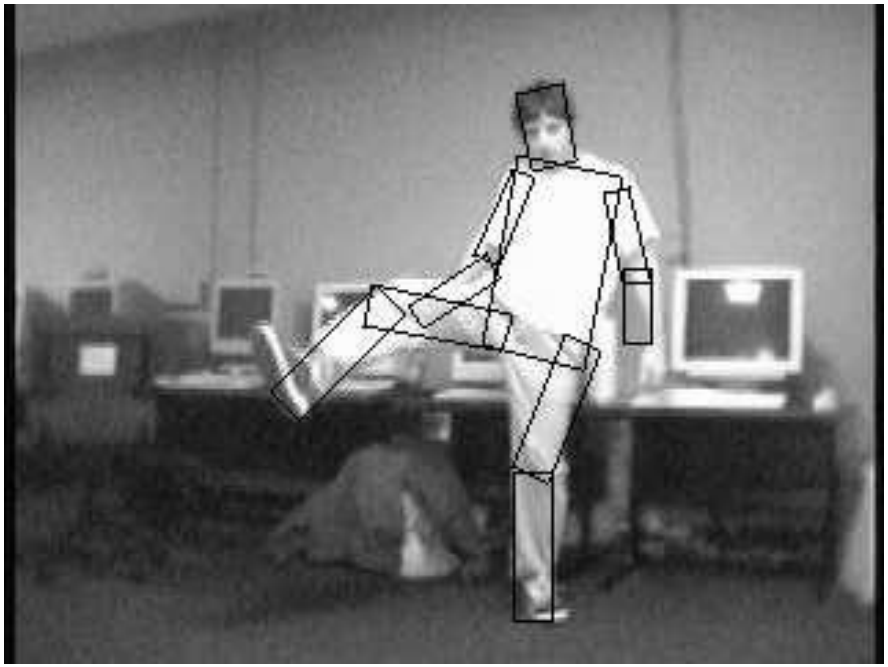
- CNNs have flexible models through spatial pooling



How to model spatial relations?

- Articulated parts model
  - Object is configuration of parts
  - Each part is detectable

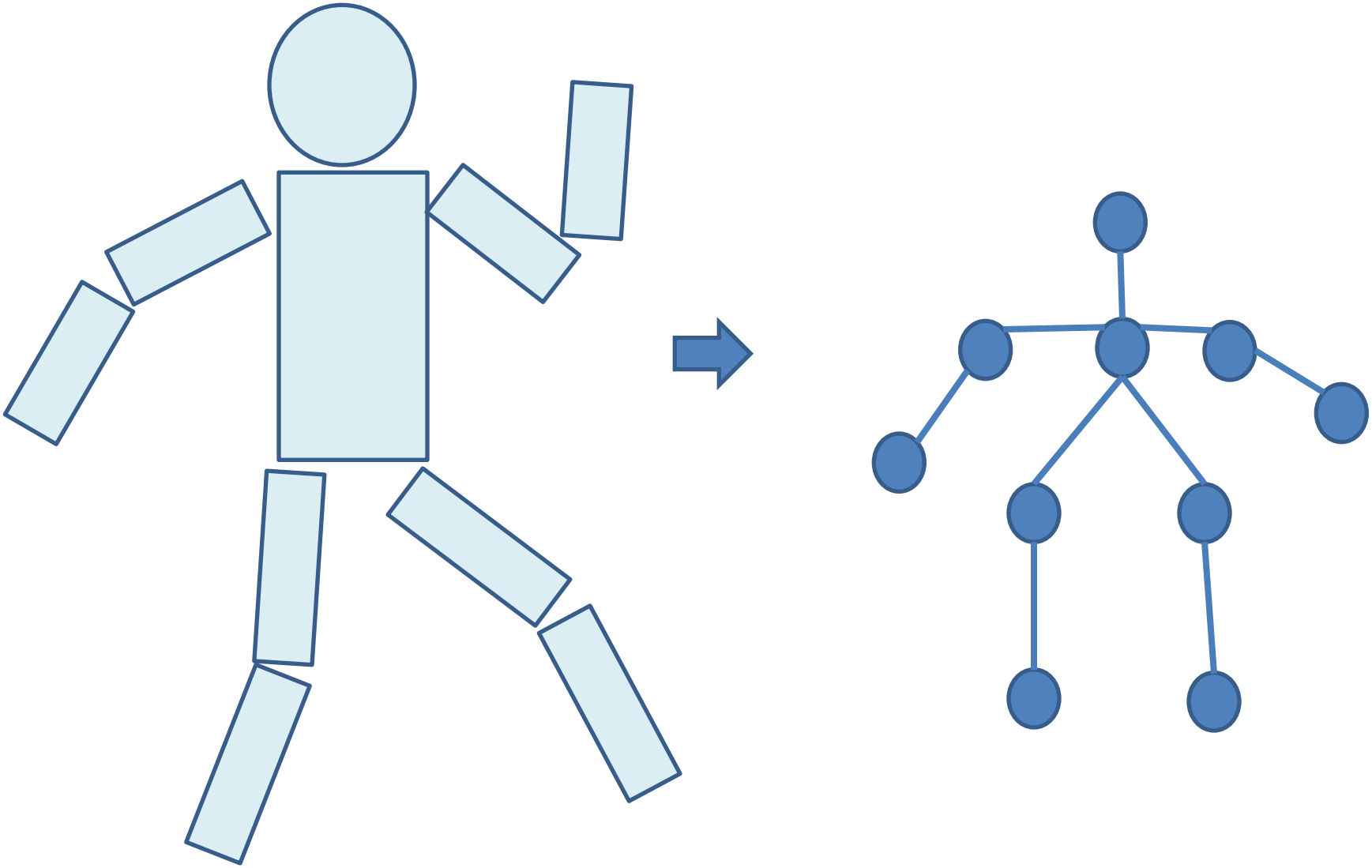




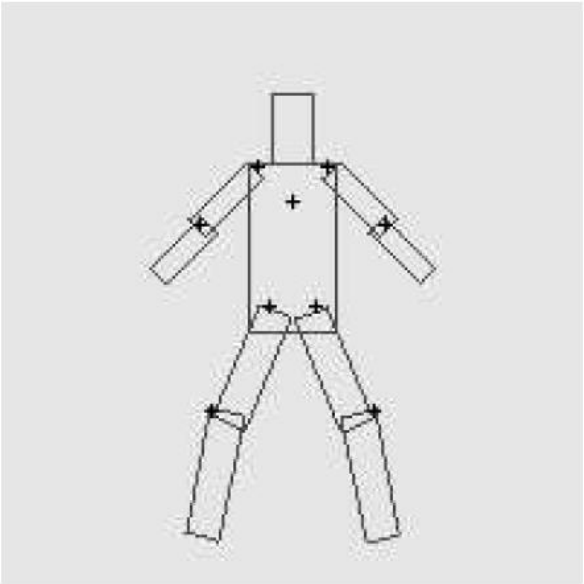
Images from Felzenszwalb

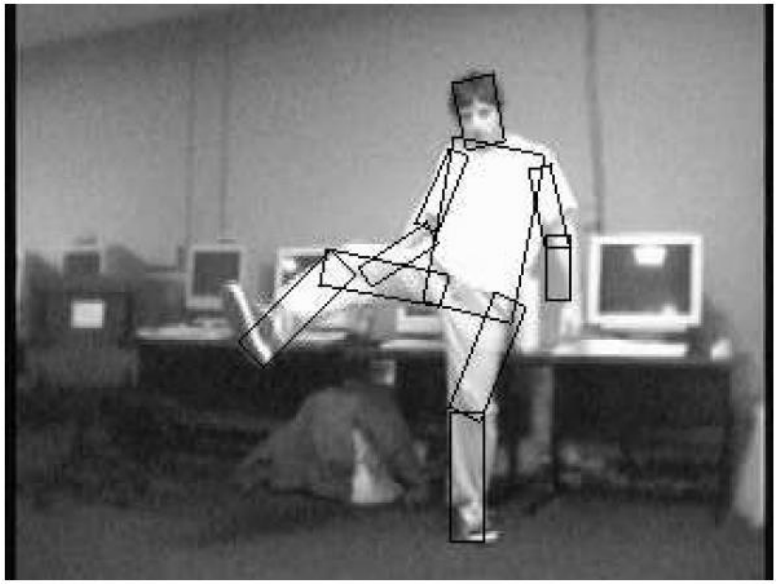
How to model spatial relations?

- Tree-shaped model



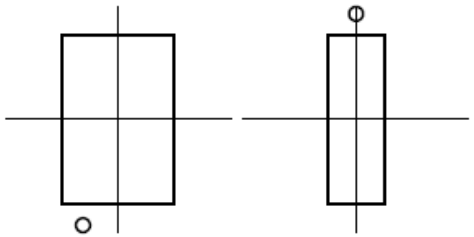
Pictorial Structures Model



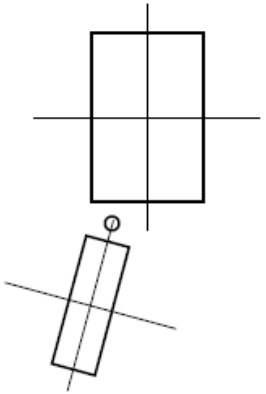


Part = oriented rectangle

Spatial model = relative size/orientation



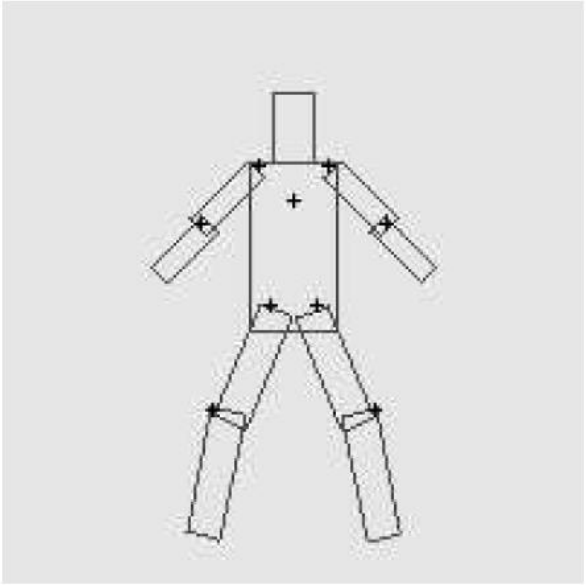
a



b

Felzenszwalb and Huttenlocher 2005

Pictorial Structures Model



$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \right)$$

Appearance likelihood

Geometry likelihood

Modeling the Appearance

- Any appearance model could be used
  - HOG Templates, etc.
  - Here: rectangles fit to background subtracted binary map

- Can train appearance models independently (easy, not as good) or jointly (more complicated but better)

$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \right)$$

Appearance likelihood

Geometry likelihood

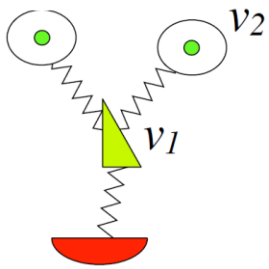
Part representation

- Background subtraction

Pictorial structures model

Optimization is tricky but can be efficient

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$



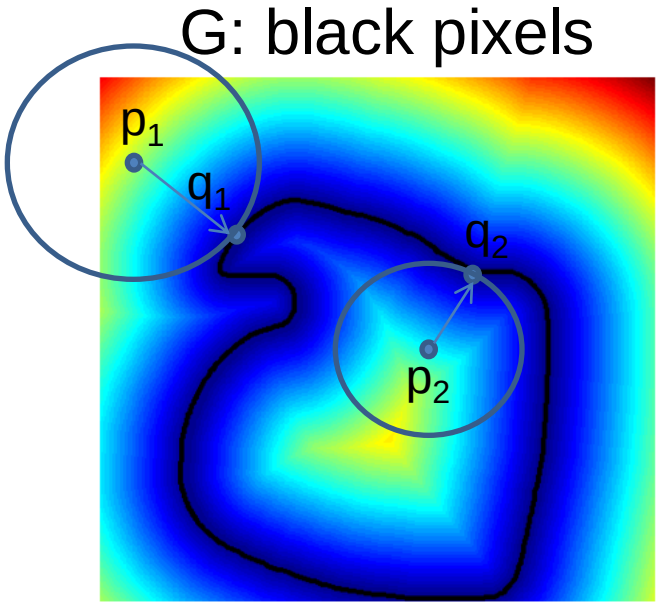
- For each l_1, find best l_2:

$$\text{Best}_2(l_1) = \min_{l_2} \left[m_2(l_2) + d_{12}(l_1, l_2) \right]$$

- Remove v_2, and repeat with smaller tree, until only a single part
- For k parts, n locations per part, this has complexity of $O(kn^2)$, but can be solved in ~$O(kn)$ using generalized distance transform

Distance Transform

- For each pixel p, how far away is the nearest pixel q of set G
  - $f(p) = \min_{q \in G} \; d(p, q)$
  - G is often the set of edge pixels



Distance Transform - Applications

- Set distances – e.g. Hausdorff Distance
- Image processing – e.g. Blurring
- Robotics – Motion Planning
- Alignment
  - Edge images
  - Motion tracks
  - Audio warping
- Deformable Part Models

Generalized Distance Transform

- Original form: $f(p) = \min_{q \in G} \; d(p,q)$
- General form: $f(p) = \min_{q \in [1,N]} \; m(q) + d(p,q)$

- For many deformation costs, $O(N^2) \rightarrow O(N)$

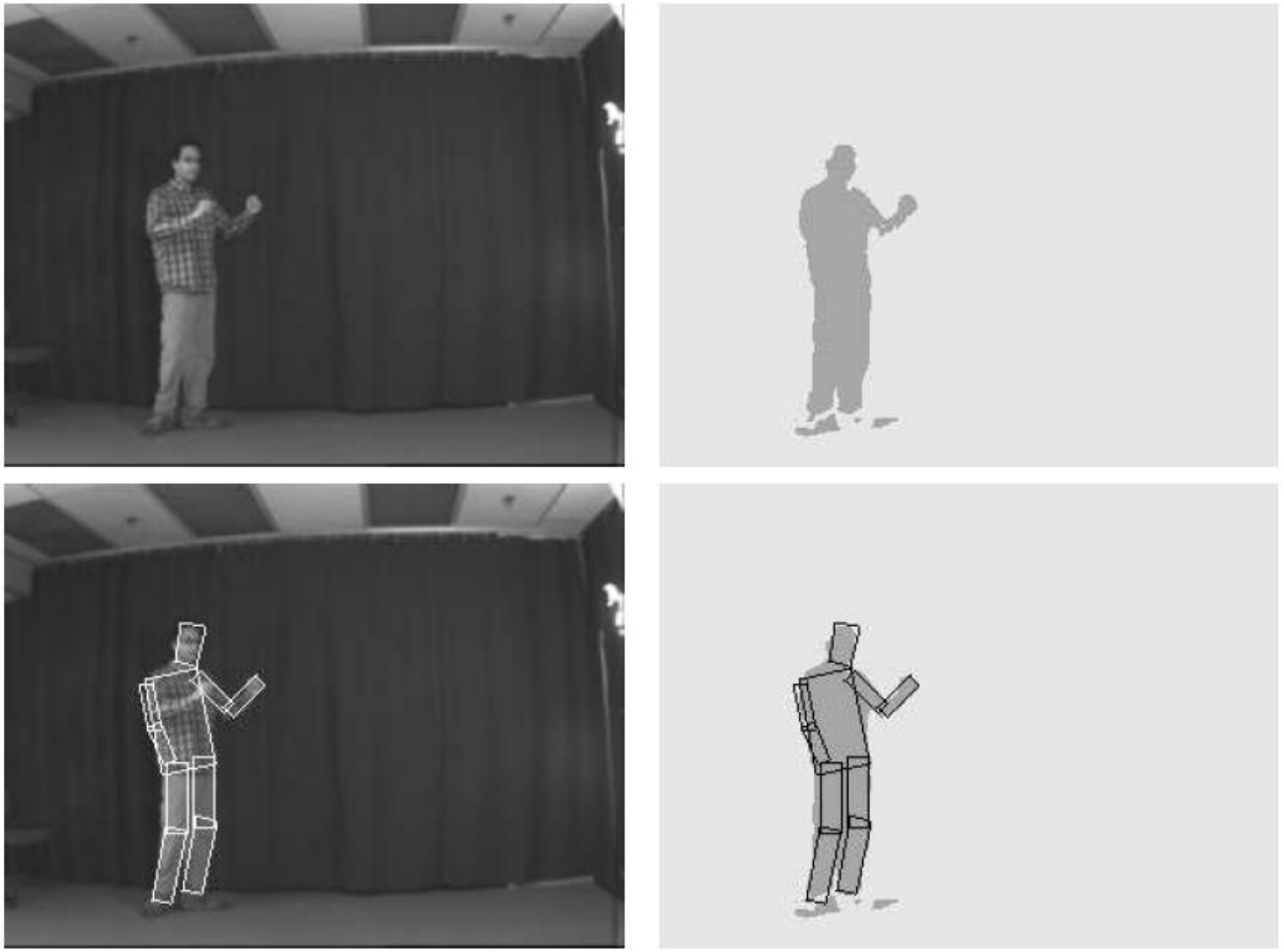
Quadratic $d(p,q) = \alpha(p-q)^2 + \beta(p-q)$

Abs Diff $d(p,q) = \alpha|p-q|$

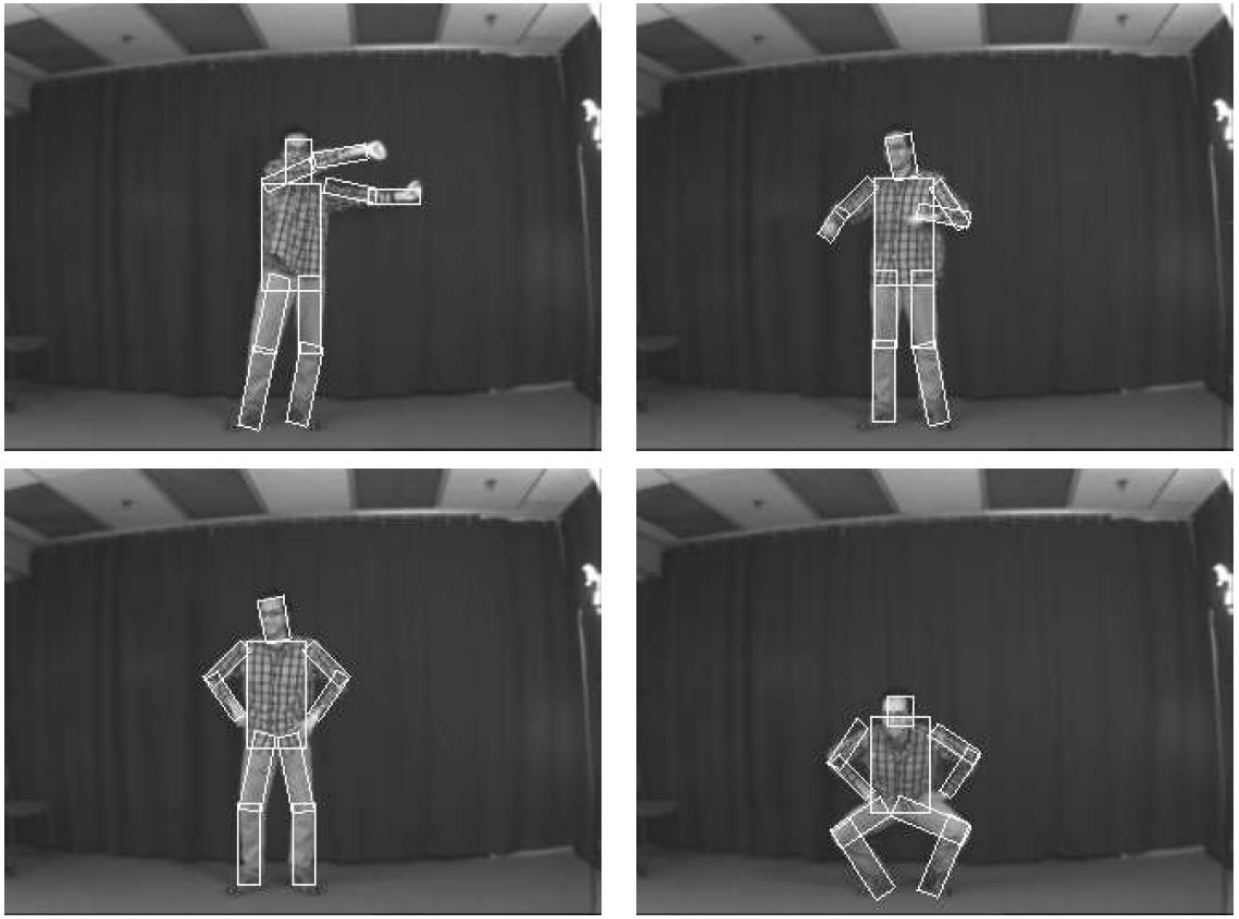
Min Composition $d(p,q) = \min(d_1(p,q), d_2(p,q))$

Bounded $d_\tau(p,q) = \begin{cases} d(p,q) & : |p-q| < \tau \\ \infty & : |p-q| \geq \tau \end{cases}$

Results for person matching



Results for person matching



Enhanced pictorial structures

- Learn spatial prior
- Color models from soft segmentation (initialized by location priors of each part)

BMVC 2009

Parts can be hard to find on their own

Which patch corresponds to a body part?

Example from Ramakrishna

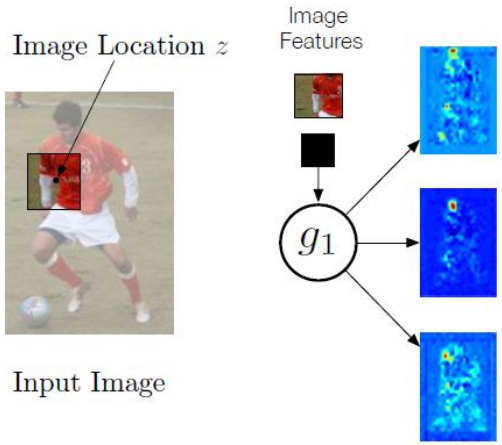
Sequential structured prediction

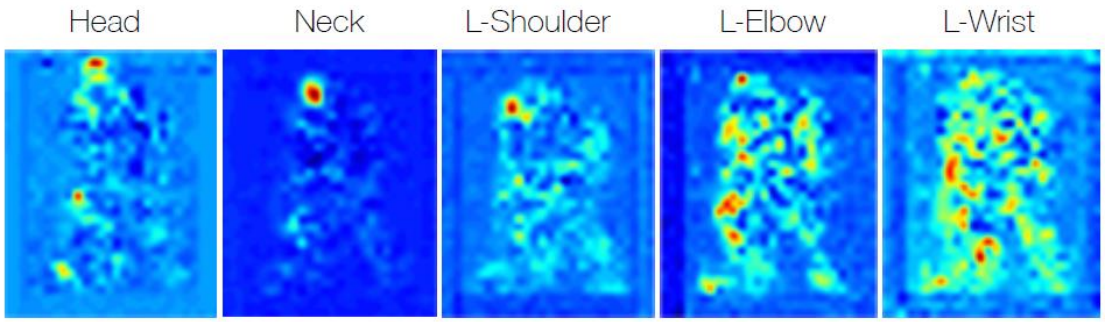
- Can consider pose estimation as predicting a set of related variables (called structured prediction)
  - Some parts easy to find (head), some are hard (wrists)

- One solution: jointly solve for most likely variables (DPM, pictorial structures)

- Another solution: iteratively predict each variable based in part on previous predictions

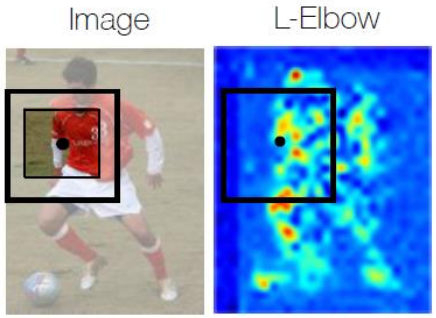
Pose machines

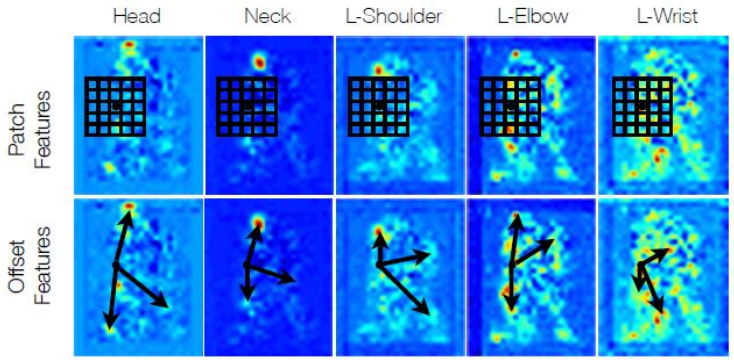




Local image evidence is weak

Certain parts are easier to detect than others





Ramakrishna et al. ECCV 2014

Example results

General principle

- "Auto-context" (Tu CVPR 2008): instead of fancy graphical models, create feature from past predictions and repredict

- Can view this as an "unrolled belief propagation" (Ross et al. 2011)

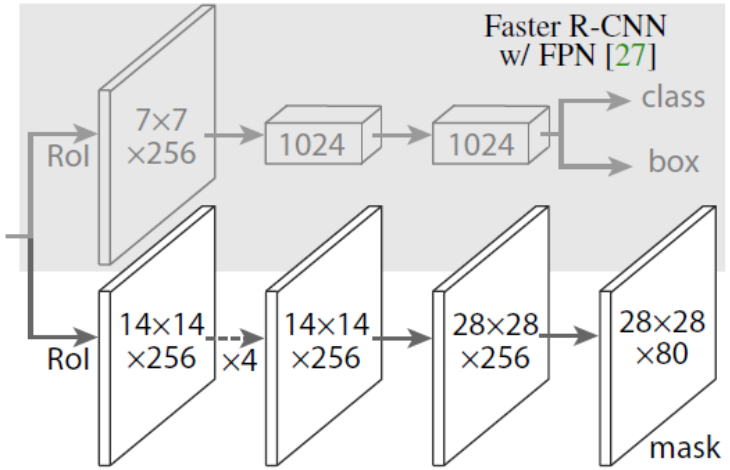
Tu Bai 2010: Auto-context
Ross Munoz Hebert Bagnell 2011: Message-Passing Inference Machines

One more approach: parallel structured prediction

- Back to CNNs
  - CNN model is a sequence of iterative feature processing
  - Last feature layer stores features that encode key information for all predictions
  - *In parallel*, predict bounding boxes, category, parts, and keypoints from last feature layer

Mask R-CNN – He Gxioxari Dollar Girshick

- Same network as Faster R-CNN, except
  - Bilinearly interpolate when extracting 7x7 cells of ROI features for better alignment of features to image
  - Instance segmentation: produce a 28x28 mask for each object category
  - Keypoint prediction: produce a 56x56 mask for each keypoint (aim is to label single pixel as correct keypoint)



Example ROI and predicted mask

Example ROI and predicted mask and keypoints

Top performing object detector, keypoint segmenter, instance segmenter

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [37]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [36]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

Table 3. **Object detection** *single-model* results (bounding box AP), *vs.* state-of-the-art on `test-dev`. Mask R-CNN usir

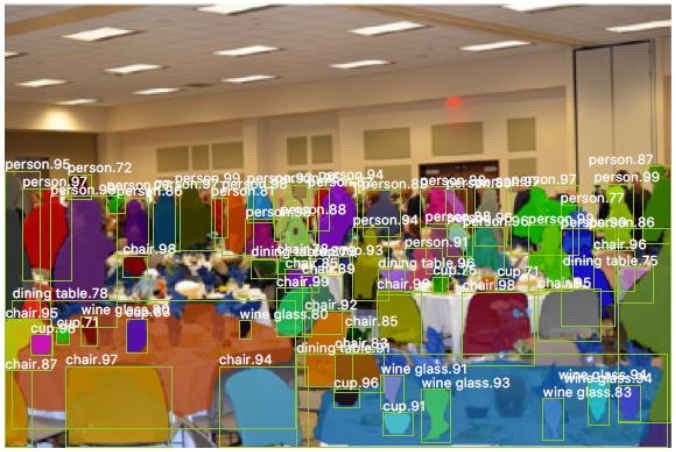
	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Table 1. **Instance segmentation** *mask* AP on COCO `test-dev`. MNC [10] and FCIS [26] are the winners of the COCO 2015 and 2016

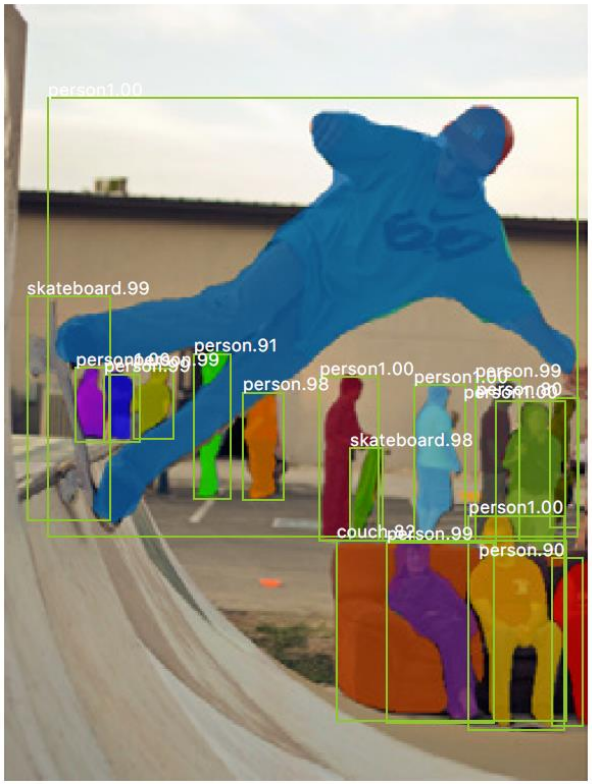
	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
CMU-Pose+++ [6]	61.8	84.9	67.5	57.1	68.2
G-RMI [31]†	62.4	84.0	68.5	**59.1**	68.1
Mask R-CNN, keypoint-only	62.7	87.0	68.4	57.4	71.1
Mask R-CNN, keypoint & mask	**63.1**	**87.3**	**68.7**	57.8	**71.4**

Table 4. **Keypoint detection** AP on COCO `test-dev`. Ours

Example detections and instance segmentations



Example detections and instance segmentations



Example keypoint detections

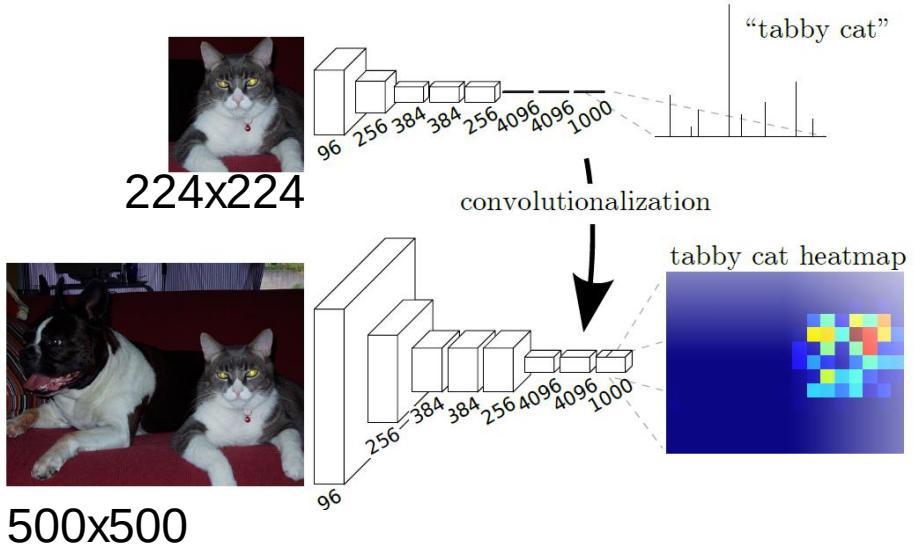
What if you want to label every pixel?

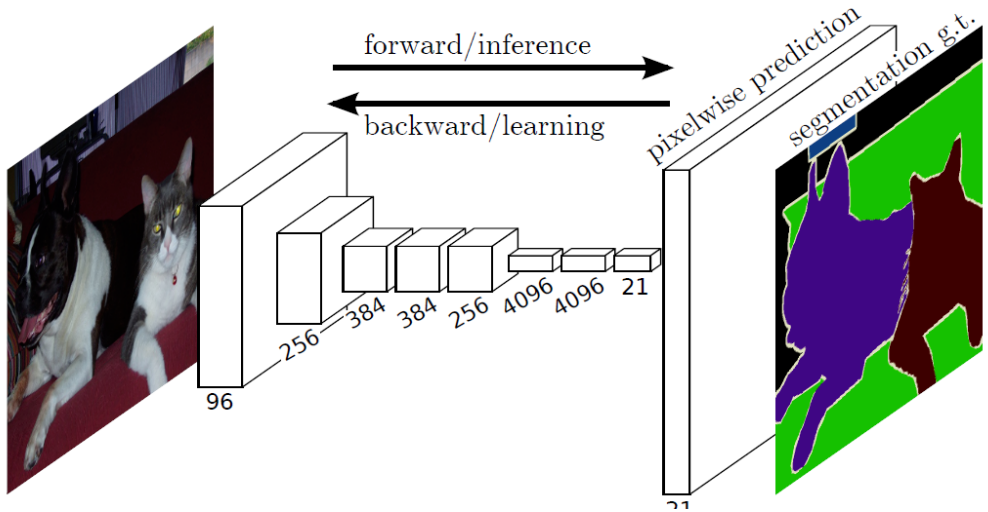
"Stuff" can be hard to capture with bounding boxes

https://www.cityscapes-dataset.com/examples/#fine-annotations

Fully convolutional networks for semantic segmentation – Long Shelhamer Darrel 2015

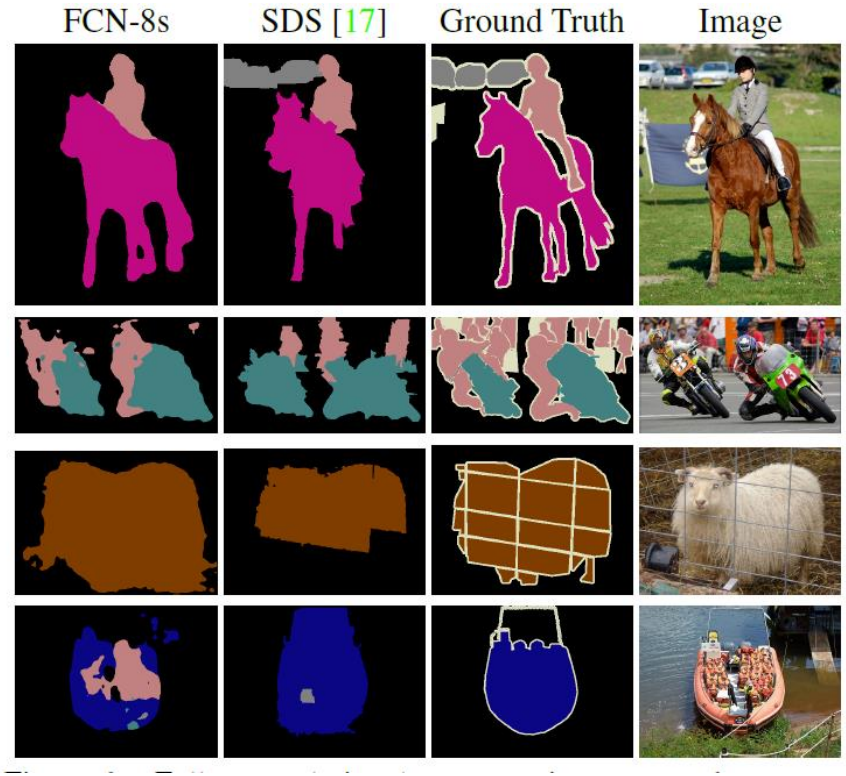
- Use network trained for classification as pre-trained network for pixel labeling
- Convert fully connected layers into convolutions
- Add features from earlier conv layers to improve resolution
- Fine-tune for pixel labeling





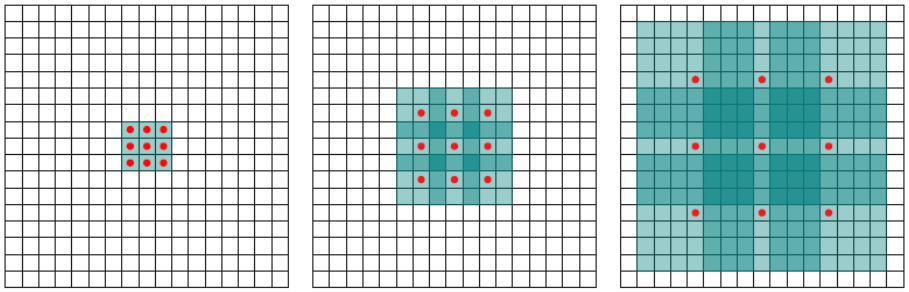
“Fully convolutional” results

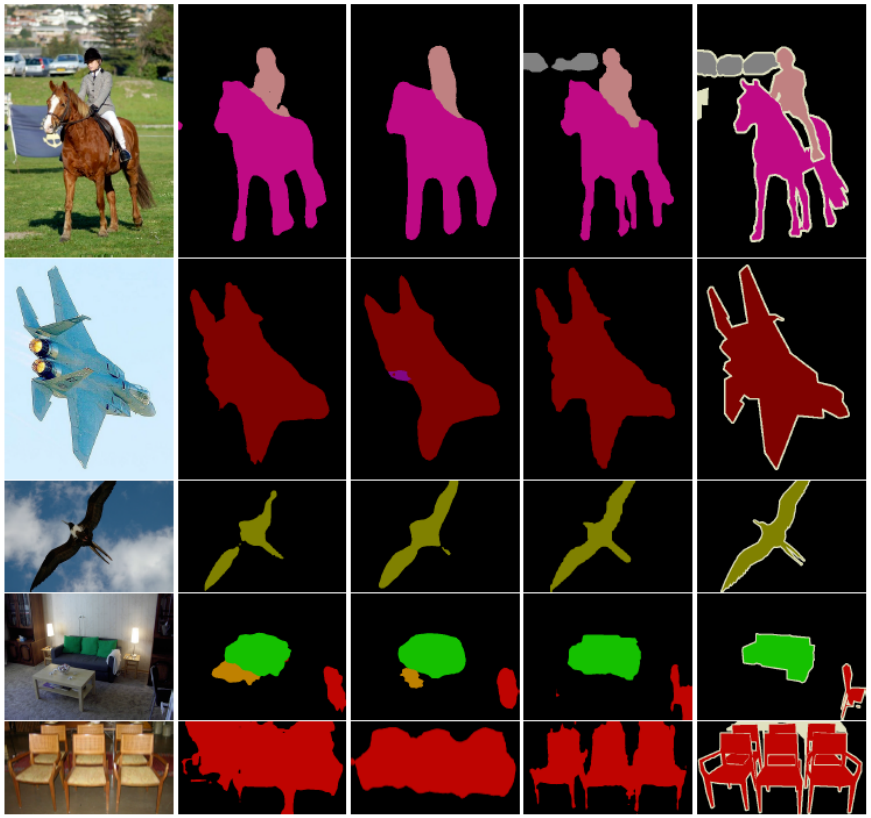
- Takes advantage of pre-training from classification
- Applied to objects and scenes (NYUd v2)
- But feature pooling reduces spatial sensitivity and resolution



Dilated Convolutions – Yu Kolton 2016

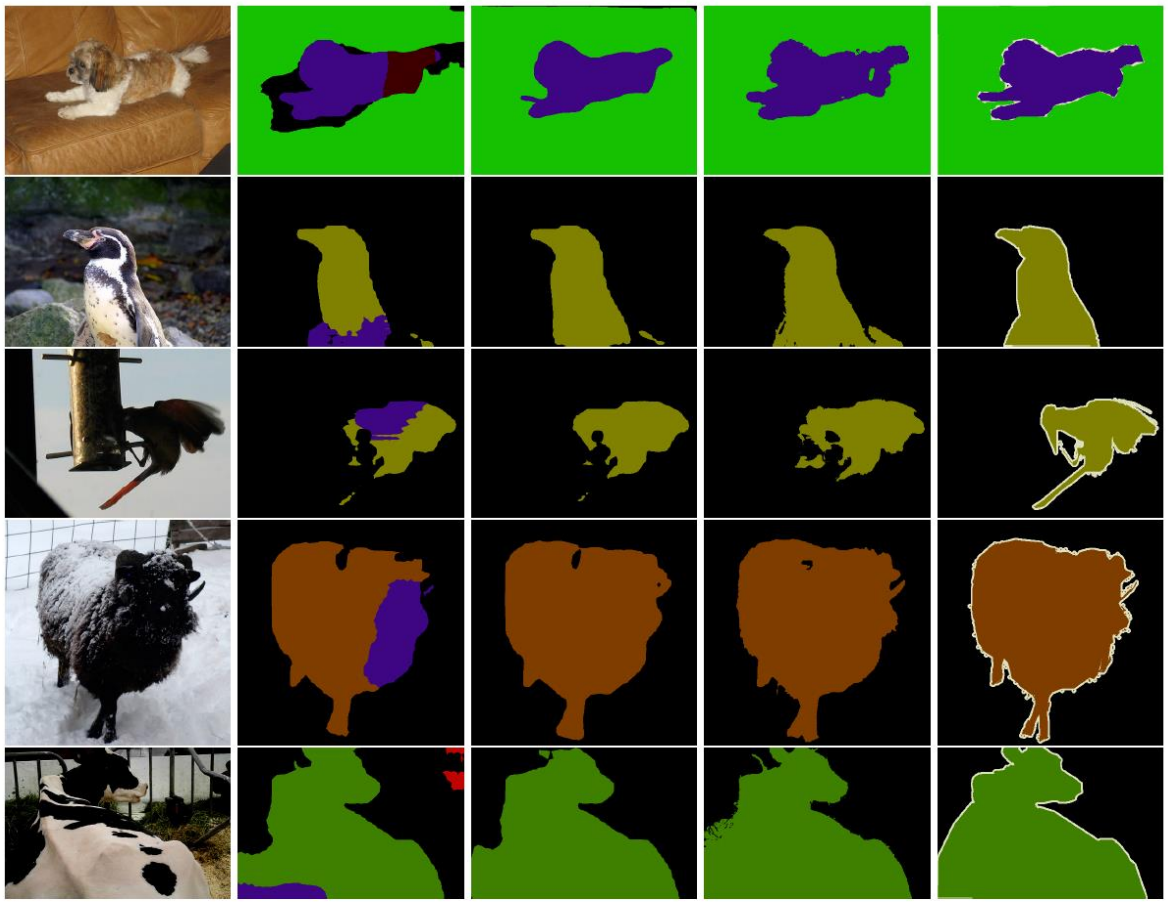
- Replacing last two pooling layers with “dilated convolution” that filters a sparse 3x3 grid of pixels
- Enables large receptive field with few parameters
- Improves resolution





(a) Image (b) FCN-8s (c) DeepLab (d) Our front end (e) Ground truth

Dilated Convolutions results



(a) Image (b) Front end (c) + Context (d) + CRF-RNN (e) Ground truth

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
DeepLab++	89.1	38.3	88.1	63.3	69.7	87.1	**83.1**	85	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab-MSc++	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	73.9
CRF-RNN	90.4	**55.3**	88.7	**68.4**	69.8	88.3	82.4	85.1	32.6	78.5	**64.4**	79.6	81.9	**86.4**	81.8	**58.6**	82.4	53.5	77.4	**70.1**	74.7
Front end	86.6	37.3	84.9	62.4	67.3	86.2	81.2	82.1	32.6	77.4	58.3	75.9	81	83.6	82.3	54.2	81.5	50.1	77.5	63	71.3
Context	89.1	39.1	86.8	62.6	68.9	88.2	82.6	87.7	33.8	81.2	59.2	81.8	87.2	83.3	83.6	53.6	84.9	53.7	80.5	62.9	73.5
Context + CRF	91.3	39.9	**88.9**	64.3	69.8	88.9	82.6	89.7	34.7	82.7	59.5	83	88.4	84.2	85	55.3	86.7	54.4	**81.9**	63.6	74.7
Context + CRF-RNN	**91.7**	39.6	87.8	63.1	**71.8**	**89.7**	82.9	**89.8**	**37.2**	**84**	63	**83.3**	**89**	83.8	**85.1**	56.8	**87.6**	**56**	80.2	64.7	**75.3**

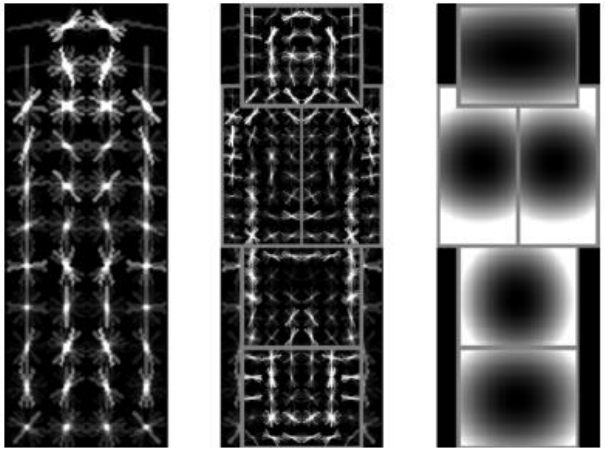
Graphical models vs. sequential/parallel prediction

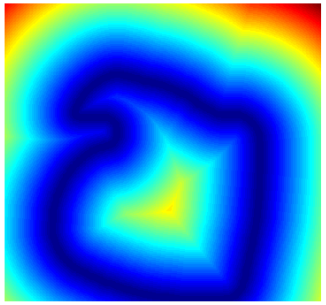
- Advantages of BP/graphcut/etc
  - Elegant
  - Relations are explicitly modeled
  - Exact inference in some cases

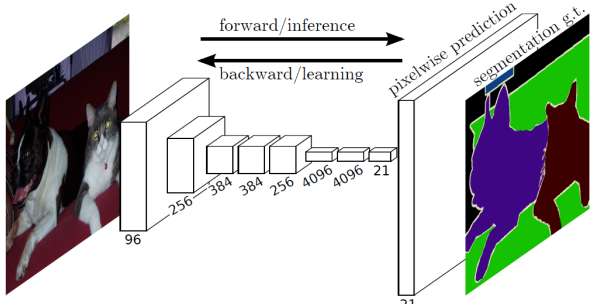
- Advantages of sequential/parallel prediction
  - Simple procedures for training and inference
  - Learns how much to rely on each prediction
  - Can model very complex relations

Things to remember

- Models can be broken down into part appearance and spatial configuration
  - Wide variety of models
- Efficient optimization can be tricky but usually possible
  - Generalized distance transform is a useful trick
- Rather than explicitly modeling contextual relations, can encode through features/classifiers







Next classes

- Tues: Object tracking with Kalman Filters
- Thurs: Action Recognition