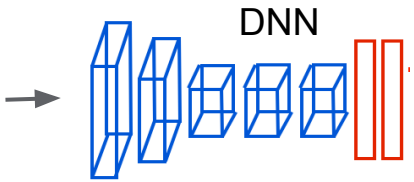
Regression Methods for Localization

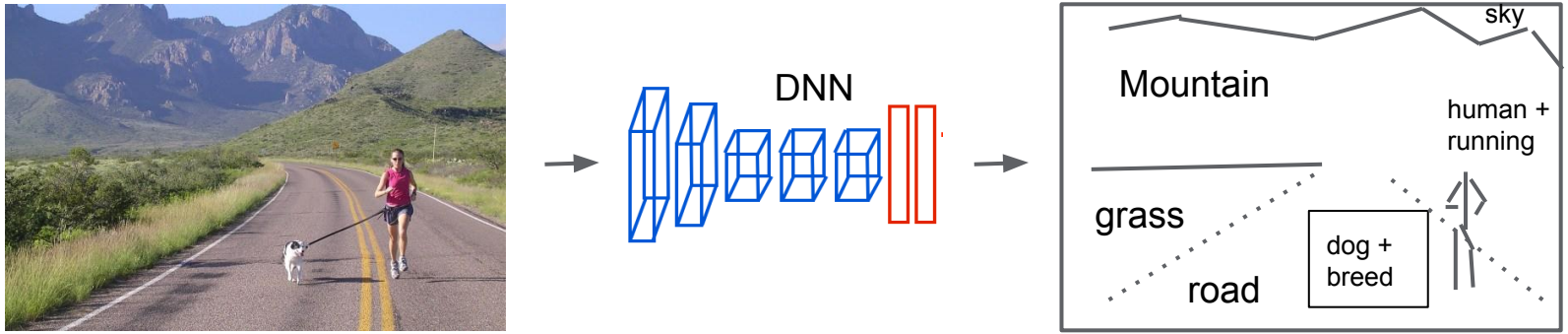
Alexander Toshev

Global Image Labeling



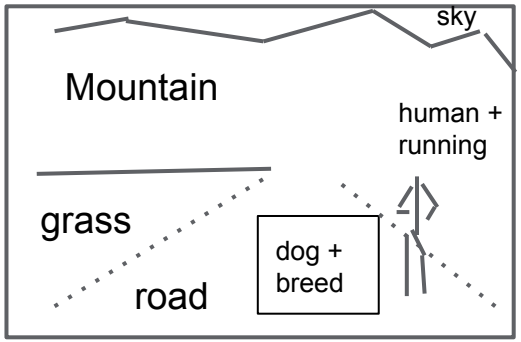
Mountain
Road
Runner (?)

Towards Image Understanding



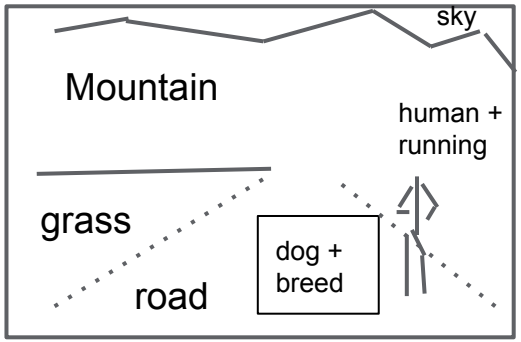
Towards Image Understanding

- Object localization
- Object segmentation
- Human pose estimation



Towards Image Understanding

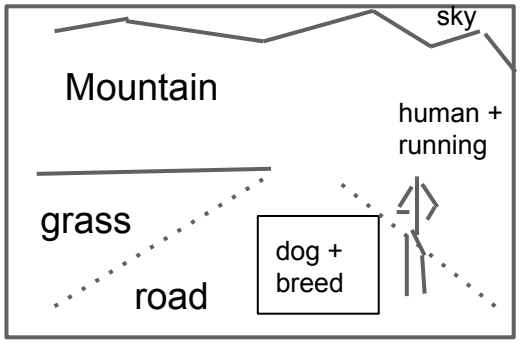
- Object localization
- Object segmentation
- Human pose estimation



Formulations as DNN-based Regression

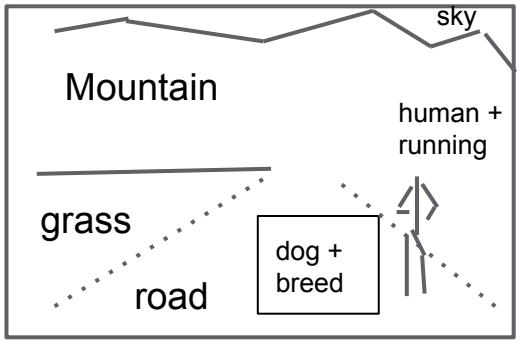
Outline

- Deep Neural Net-based Regression
- Object Mask Regression
- Object Bounding Box Regression
- Human Pose Estimation



Outline

- **Deep Neural Net-based Regression**
- Object Mask Regression
- Object Bounding Box Regression
- Human Pose Estimation



DNN-based Regression

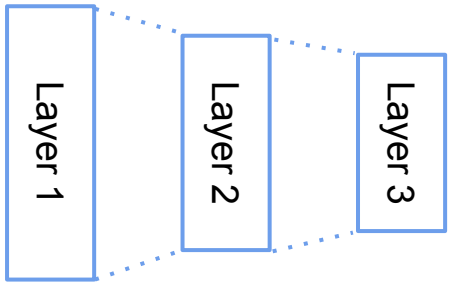
Represent localization information as a real valued vector **p**:

- **pixel association** with objects -> object segmentation
- landmark **coordinates** -> object boxes, human body joints

Predict:

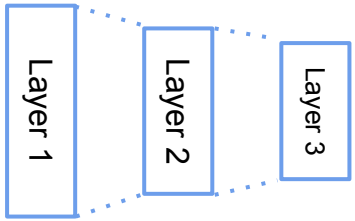
$$p = DNN(image)$$

where DNN =



Choices to be Made

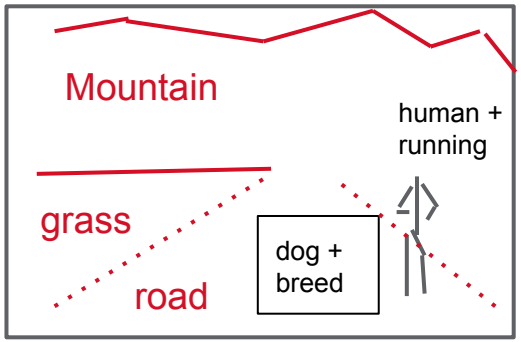
1. Problem parameterization as regression vector **p**.
2. Architecture:



3. Training loss: L_2, L_1, logistic, ...

Outline

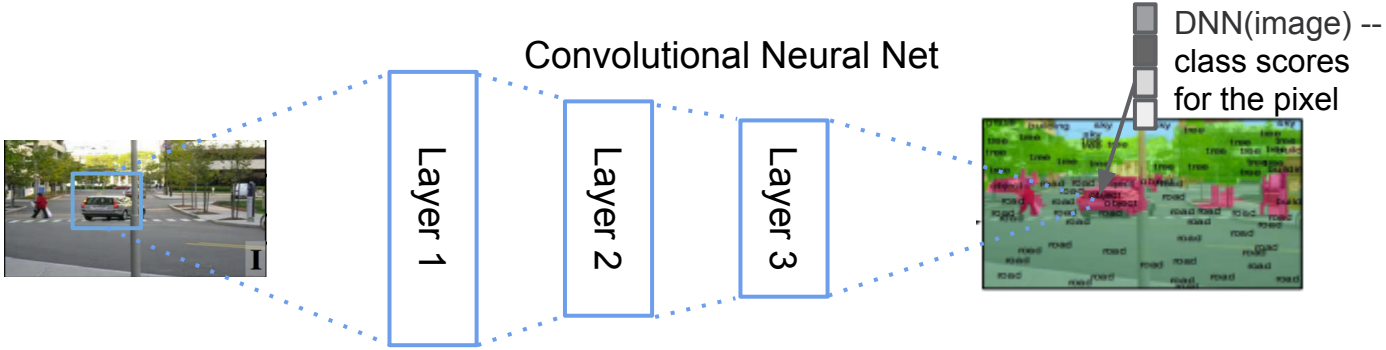
- Deep Neural Net-based Regression
- **Object Mask Regression**
- Object Bounding Box Regression
- Human Pose Estimation



Object Mask

- Generate object mask by **sliding a neural net classifier**
- Object mask directly as **neural net output**

Sliding Window Classification



- Sliding the net produces labeling for each pixel
- Loss is a classifier, e.g. **Cross Entropy:**

$$- \Sigma_c \, q(c) \log DNN_c(image)$$

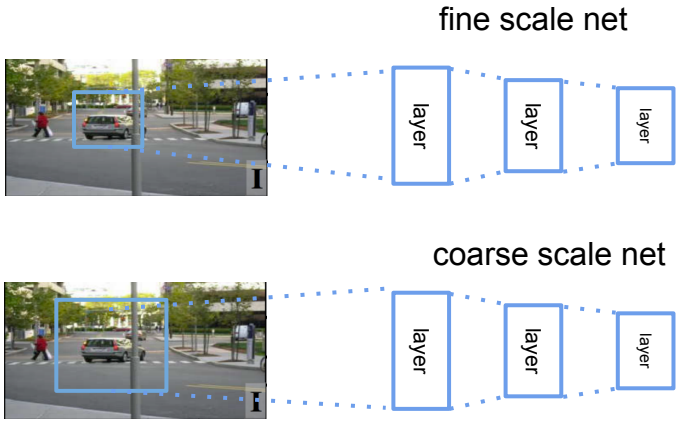
Larger Context vs Precision

Contradicting requirements:

- Large input context
- High input resolution

Proposed solutions:

- Multi-scale nets



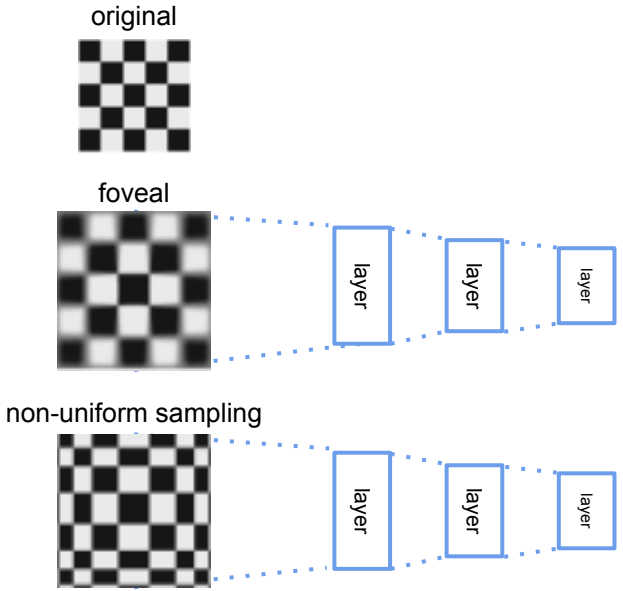
Larger Context vs Precision

Contradicting requirements:

- Large input context
- High input resolution

Proposed solutions:

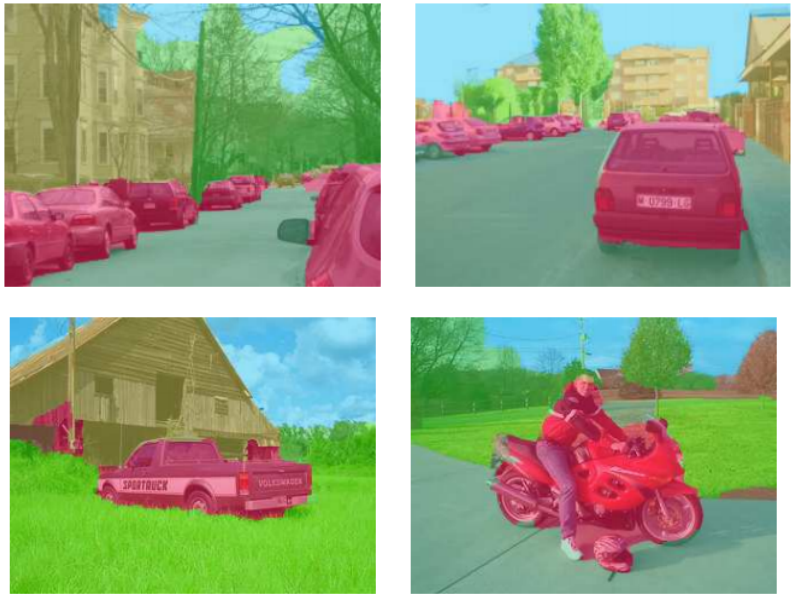
- Multi-scale nets
- Foveal net input
- Non-uniformly sampled input



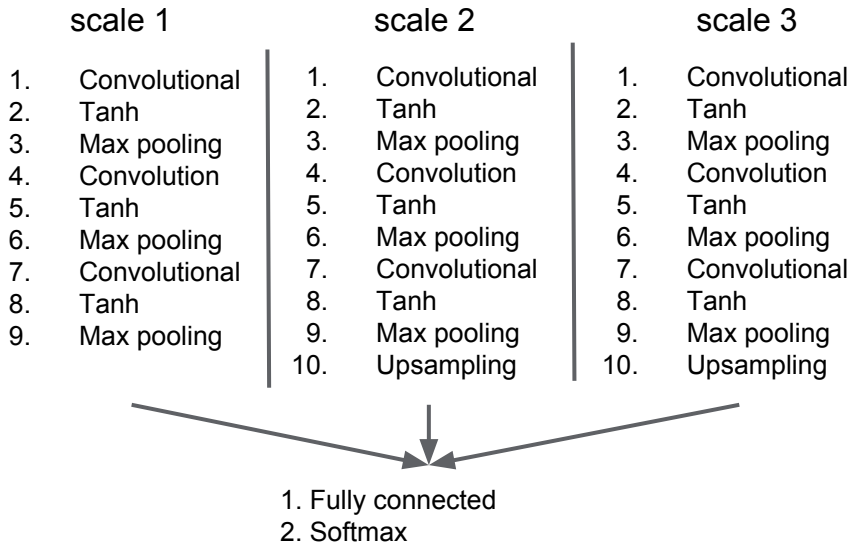
Street Image Segmentation [Farabet et al., 2013]

scale 1	scale 2	scale 3
1. Convolutional	1. Convolutional	1. Convolutional
2. Tanh	2. Tanh	2. Tanh
3. Max pooling	3. Max pooling	3. Max pooling
4. Convolution	4. Convolution	4. Convolution
5. Tanh	5. Tanh	5. Tanh
6. Max pooling	6. Max pooling	6. Max pooling
7. Convolutional	7. Convolutional	7. Convolutional
8. Tanh	8. Tanh	8. Tanh
9. Max pooling	9. Max pooling	9. Max pooling
	10. Upsampling	10. Upsampling

1. Fully connected
2. Softmax



Street Image Segmentation [Farabet et al., 2013]



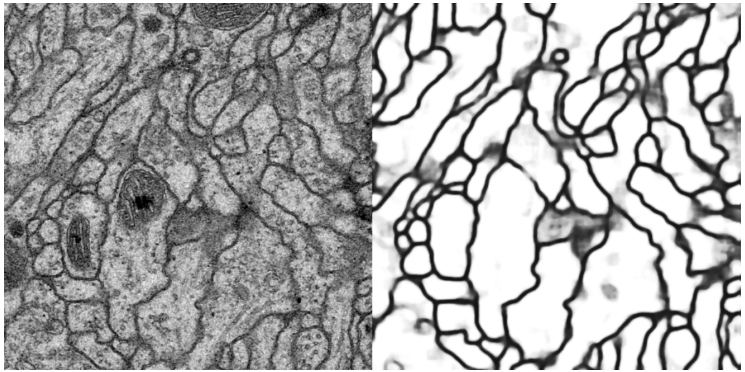
Graphical model [Lempitsky et al., 09]	72.4%
Multiscale CNN	72.4%
Multiscale CNN + Superpixel smoothing	74.56%

Average per-class accuracy on Stanford Benchmark Dataset

Neuronal Membranes in Electron Microscopy Images [Ciresan et al. 2012]

1. Foveal + non-uniform sampling of input
2. Convolutional + non-linearity
3. Max pooling
4. Convolutional + non-linearity
5. Max pooling
6. Convolutional + non-linearity
7. Max pooling
8. Convolutional + non-linearity
9. Max pooling
10. Fully connected
11. Fully connected

Winner of ISBI'12 Challenge on segmenting neuronal structures

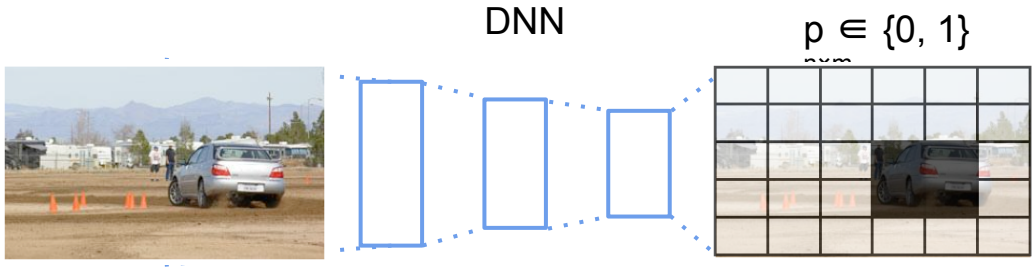


Object Mask

- Generate object mask by sliding a neural net classifier
- **Object mask directly as neural net output**

Direct Object Mask Regression [Szegedy, Toshev, Erhan, 2013]

- Apply net on full image
- Output object mask **p:**
  - p(x,y) = 1 if pixel is object, otherwise 0
  - typical output size: 30 x 30.



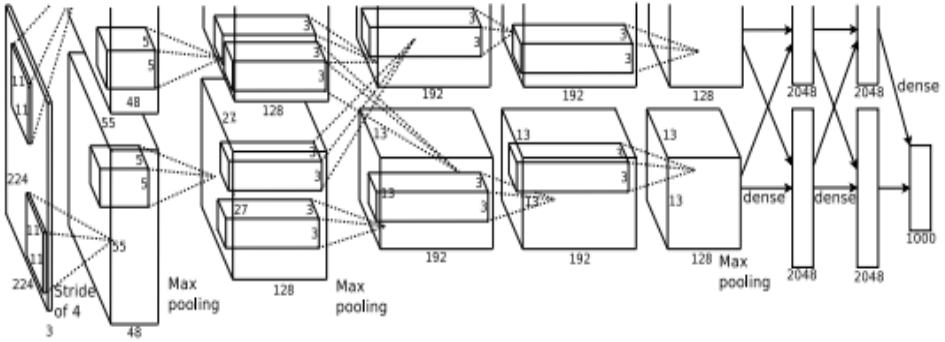
Architecture for Object Mask Regression

Adopted from [Krizhevsky et al., 2012]

<u>Differences to sliding window nets:</u>

- Larger fully connected layers
- Larger number of fully connected layers

DNN



Training Object Mask Regression

Loss:

$$\min ||\text{DNN}(\text{image}) - \mathbf{q}||_2$$

for **q** the true mask.

Data augmentation:

- Large number of random image crops.
- Every object to be present at large number of locations.
- For VOC 2011: 11000 training images → 30M crops.

Larger Context vs Precision

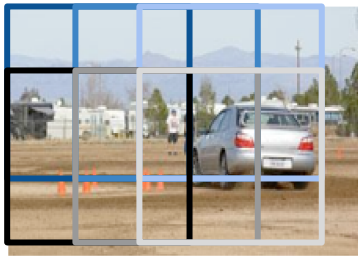
Properties:

+ large context -- full image
- limited input resolution -- size of input layer

large scale

Multi-resolution Refinement:

1. Apply DNN regressor on several large subimages
2. Refine detections -- apply regressors on each detection



mid scale

Example Detections on VOC 2007 Test

Results on VOC 2007 test

class	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
DetectorNet[1]	.292	.352	**.194**	.167	.037	**.532**	.502	**.272**	.102	**.348**
Sliding windows[1]	.213	.190	.068	.120	.058	.294	.237	.101	.059	.131
3-layer model [19]	.294	.558	.094	.143	**.286**	.440	.513	.213	.200	.193
Felz. et al. [9]	**.328**	.568	.025	**.168**	.285	.397	.516	.213	.179	.185
Girshick et al. [11]	.324	**.577**	.107	.157	.253	.513	**.542**	.179	**.210**	.240
class	table	dog	horse	m-bike	person	plant	sheep	sofa	train	tv
DetectorNet[1]	**.302**	**.282**	.466	.417	.262	.103	**.328**	.268	.398	**.470**
Sliding windows[1]	.110	.134	.220	.243	.173	.070	.118	.166	.240	.119
3-layer model [19]	.252	.125	.504	.384	.366	**.151**	.197	.251	.368	.393
Felz. et al. [9]	.259	.088	.492	.412	.368	.146	.162	.244	.392	.391
Girshick et al. [11]	.257	.116	**.556**	**.475**	**.435**	.145	.226	**.342**	**.442**	.413

Results on VOC 2007

animals

class	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
DetectorNet[1]	.292	.352	**.194**	.167	.037	**.532**	.502	**.272**	.102	**.348**
Sliding windows[1]	.213	.190	.068	.120	.058	.294	.237	.101	.059	.131
3-layer model [19]	.294	.558	.094	.143	**.286**	.440	.513	.213	.200	.193
Felz. et al. [9]	**.328**	.568	.025	**.168**	.285	.397	.516	.213	.179	.185
Girshick et al. [11]	.324	**.577**	.107	.157	.253	.513	**.542**	.179	**.210**	.240

class	table	dog	horse	m-bike	person	plant	sheep	sofa	train	tv
DetectorNet[1]	**.302**	**.282**	.466	.417	.262	.103	**.328**	.268	.398	**.470**
Sliding windows[1]	.110	.134	.220	.243	.173	.070	.118	.166	.240	.119
3-layer model [19]	.252	.125	.504	.384	.366	**.151**	.197	.251	.368	.393
Felz. et al. [9]	.259	.088	.492	.412	.368	.146	.162	.244	.392	.391
Girshick et al. [11]	.257	.116	**.556**	**.475**	**.435**	.145	.226	**.342**	**.442**	.413

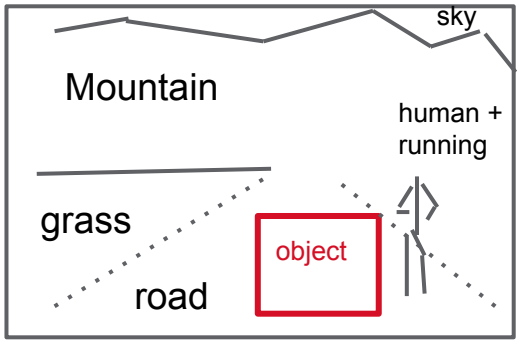
Results on VOC 2007

vehicles

class	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
DetectorNet[1]	.292	.352	**.194**	.167	.037	**.532**	.502	**.272**	.102	**.348**
Sliding windows[1]	.213	.190	.068	.120	.058	.294	.237	.101	.059	.131
3-layer model [19]	.294	.558	.094	.143	**.286**	.440	.513	.213	.200	.193
Felz. et al. [9]	**.328**	.568	.025	**.168**	.285	.397	.516	.213	.179	.185
Girshick et al. [11]	.324	**.577**	.107	.157	.253	.513	**.542**	.179	**.210**	.240
class	table	dog	horse	m-bike	person	plant	sheep	sofa	train	tv
DetectorNet[1]	**.302**	**.282**	.466	.417	.262	.103	**.328**	.268	.398	**.470**
Sliding windows[1]	.110	.134	.220	.243	.173	.070	.118	.166	.240	.119
3-layer model [19]	.252	.125	.504	.384	.366	**.151**	.197	.251	.368	.393
Felz. et al. [9]	.259	.088	.492	.412	.368	.146	.162	.244	.392	.391
Girshick et al. [11]	.257	.116	**.556**	**.475**	**.435**	.145	.226	**.342**	**.442**	.413

Outline

- Deep Neural Net-based Regression
- Object Mask Regression
- **Object Bounding Box Regression**
- Human Pose Estimation



Bounding Box Regression

Predict *normalized bounding box coordinates*:

$$p = (x_1, y_1, x_2, y_2)$$

Examples:

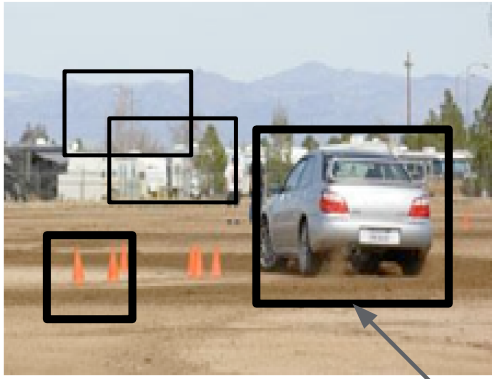
- Train a **single box regressor** per class -- ImageNet 2012 winner [Krizhevsky, Sutskever, Hinton, 2012]
- Part of a **joint localization and classification** net [Sermanet et al., 2014]
- **Multiple box regressor** in a **class agnostic** fashion [Erhan, Szegedy, Toshev, Anguelov, 2014]

Multi-Box Prediction

Decompose DNN(x) as

- n box normalized coordinate predictions: $p_i \in R^4$
- n box objectness confidence predictions: $s_i \in [0, 1]$

thickness represents confidence



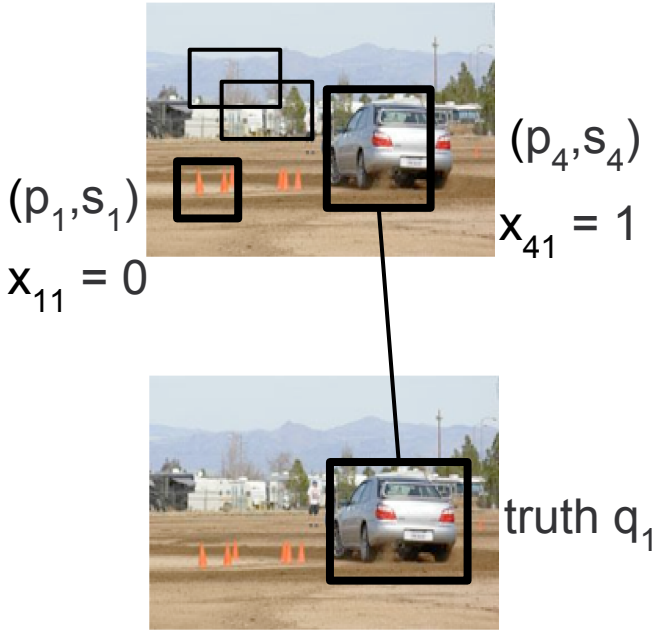
$(p_4, s_4 = 0)$ $(p_4, s_4 = 1)$

Multi-Box Prediction Training

Objective: match predicted boxes to truth and reinforce the best matches and their scores:

$$\text{Match}(x) = -\Sigma_{i,j} \exp(-||p_i - q_j||^2) s_i x_{i,j}$$

for prediction to truth assignment $x_{i,j} \in \{0, 1\}$



Multi-Object Box Prediction

- box coordinate predictions: $p_i \in R^4$
- box confidence predictions: $s_i \in R$
- class agnostic true boxes: $q_j \in R^4$

Objective:

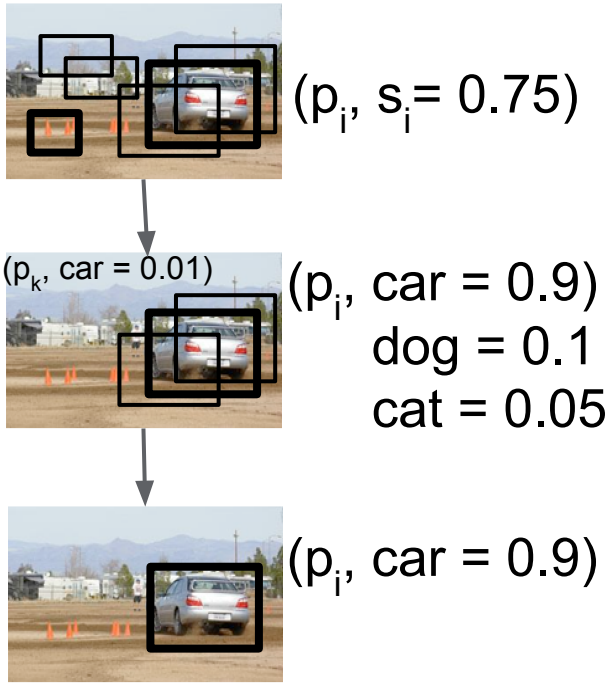
$$\text{Match}(x; \Theta) = -\Sigma_{i,j} \exp(-||p_i(\Theta) - q_j(\Theta)||^2) s_i x_{i,j}$$

Updates:

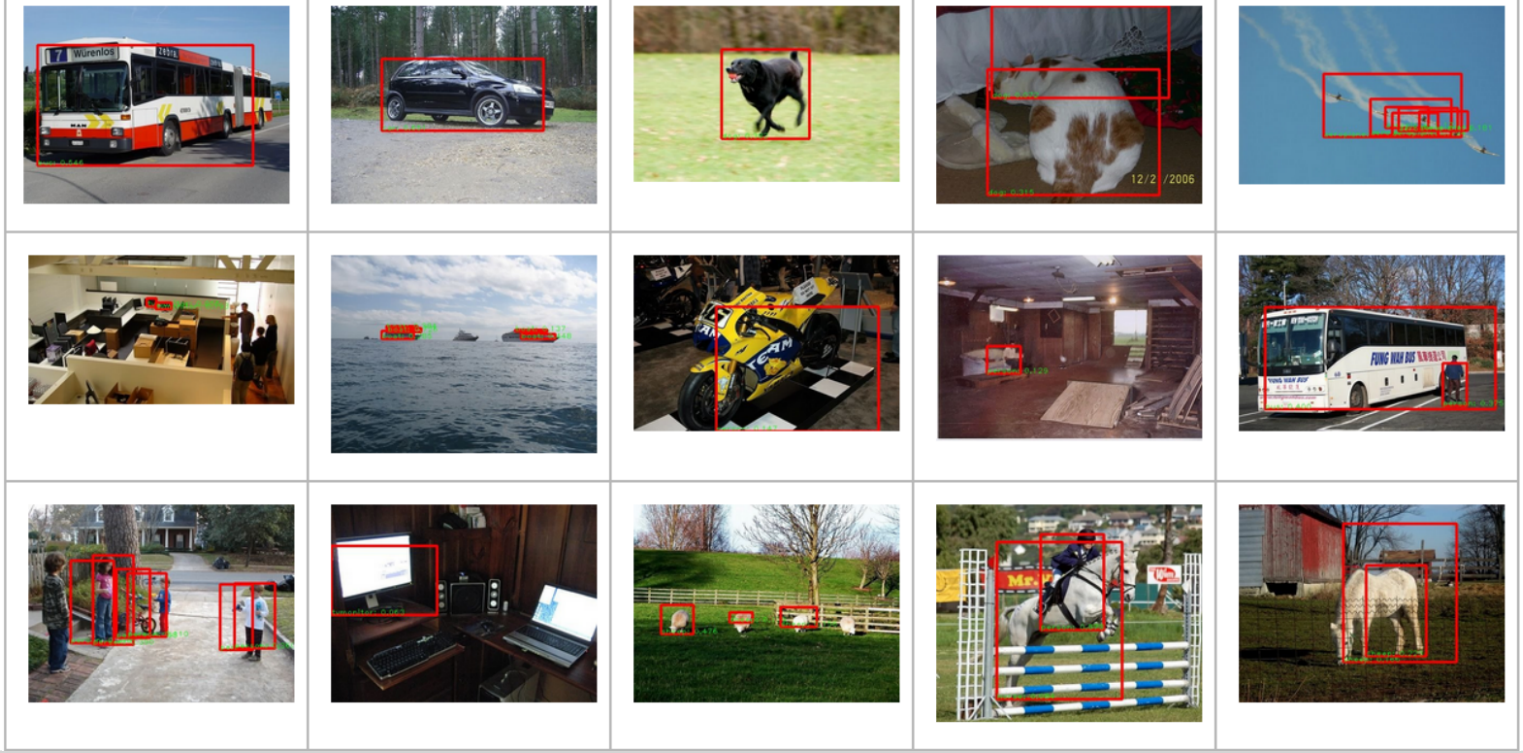
1. Compute optimal $x^* = \text{argmax Match}(x; \Theta)$
2. $\partial \text{ Match}(x; \Theta) / p_i = 2\Sigma_j \exp(-||p_i - q_j||^2) s_i x^*_{i,j} (p_i - q_j)$
3. $\partial \text{ Match}(x; \Theta) / s_i = -2\Sigma_j \exp(-||p_i - q_j||^2) x^*_{i,j}$

Object Detection via Multi-Box Prediction

1. **Predict multiple boxes** over several large crops
2. **Score** each of them with an object classifier
3. Perform **non-max suppression**

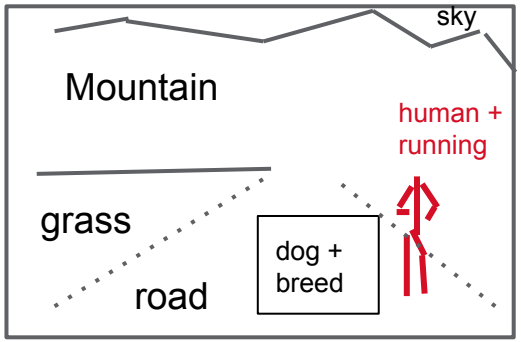


Example Detections on VOC 2007 Test



Outline

- Deep Neural Net-based Regression
- Object Mask Regression
- Object Bounding Box Regression
- **Human Pose Estimation**



Human Pose Estimation

Problem: Localization of human body parts

Importance:

- activity and gesture recognition
- human segmentation
- understanding human interactions

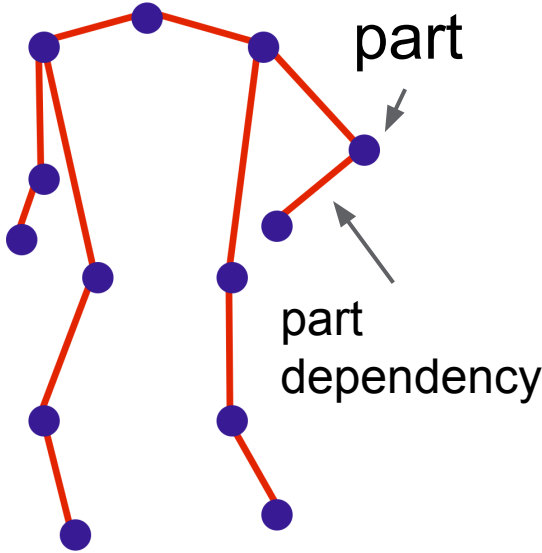
Part-based Models

Pros: handle gracefully **compositionality**:

- reason about a part in the **context** of other parts
- **tractability** under certain assumptions, e.g. tree structure

Cons:

- independence assumptions **limit expressivity**
- **handcrafted** model structure



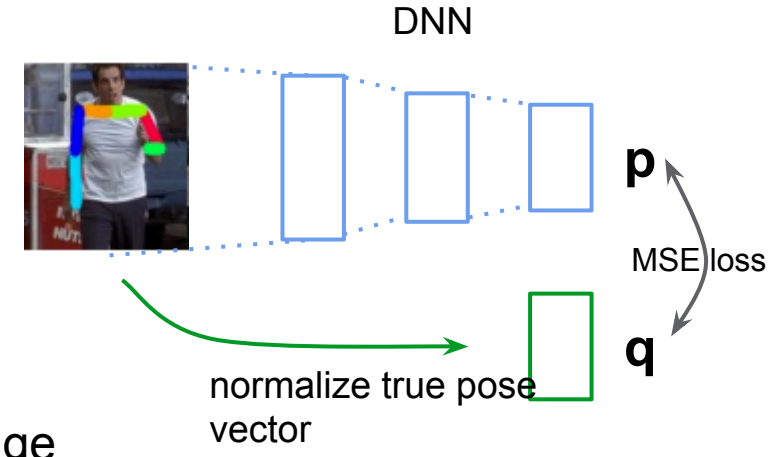
DeepPose: DNN-based Human Pose [Toshev et al., 2014]

Pose as **normalized joint coordinate vector**:

$$(x_1, y_1, \ldots, x_n, y_n) = p^T$$

Learn a DNN to predict the above vector:

$$\min ||\text{DNN}(\text{image}) - p||_2^2$$

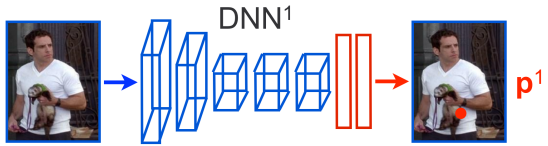


Properties:

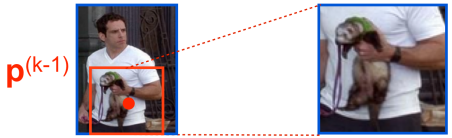
- **Context**: each regressor sees the full image
- **Simplicity:** no explicit model design

Larger Context vs Precision

Sequential pose refinement via a **cascade** of **DNN-regressors**.

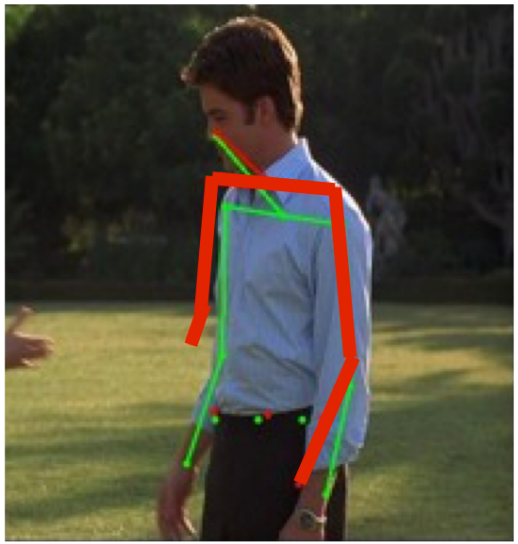


1. Estimate initial pose $\mathbf{p}^1 \leftarrow$ DNN(full image).
2. At stage **k** do:
   a. Crop large subimages at $\mathbf{p}^{(k-1)}$
   b. Predict displacement from current to true pose:
      $\mathbf{p}^k \leftarrow \text{DNN}^K(\text{subimage}) + \mathbf{p}^{(k-1)}$

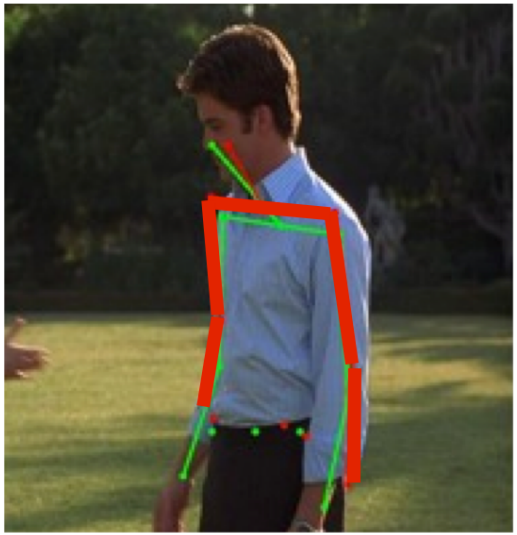


Refinement via Cascade of DNN Regressors

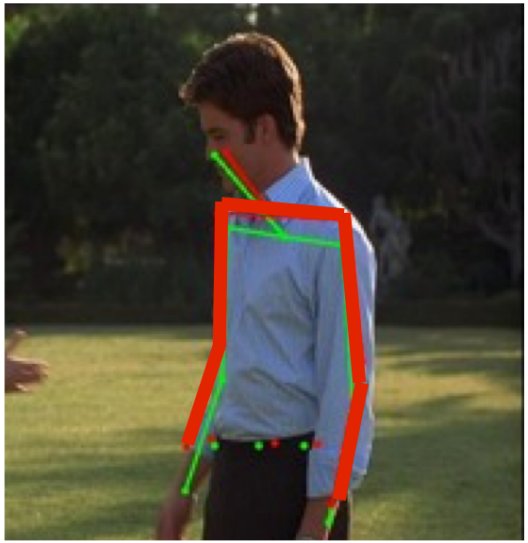
stage 1



stage 2

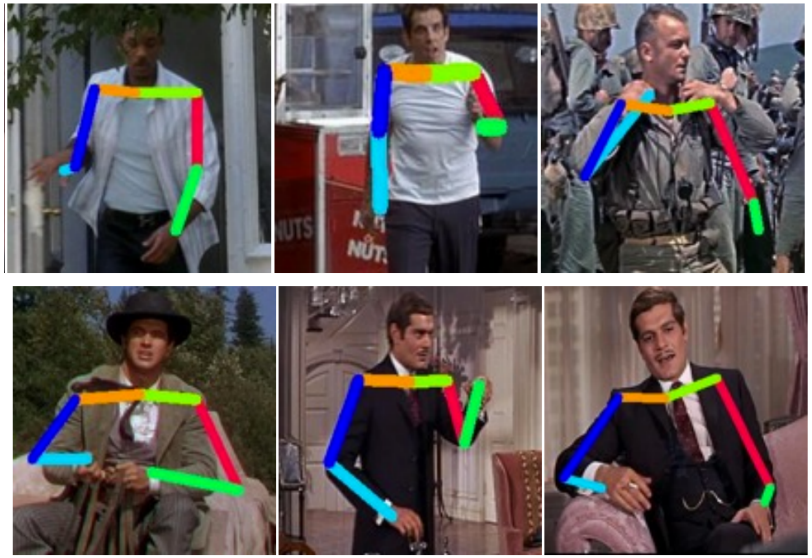


stage 3



Example Pose Estimations

Movies (FLIC)

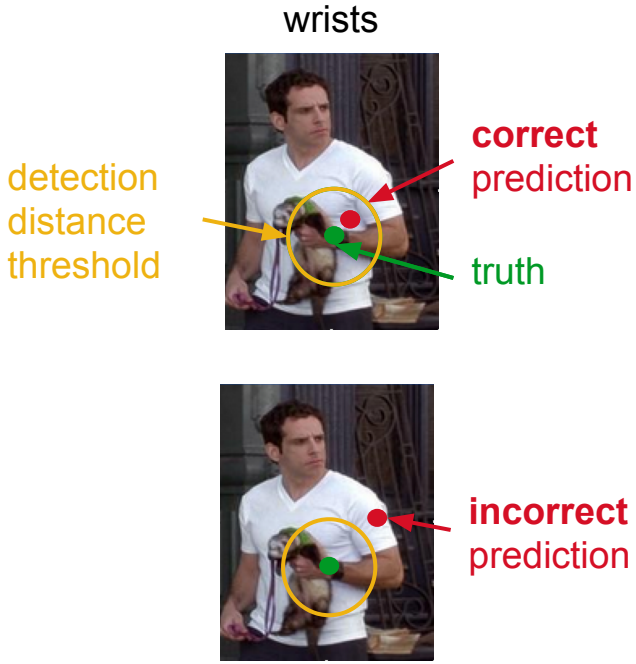


Sports (LPS)

Pose Evaluation Metric

Detection rates for each joint:

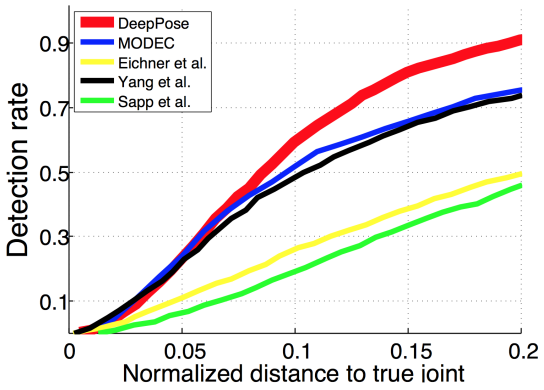
- **detection criterion** based on a **distance threshold**
- distance threshold defined as **fraction of true body** size



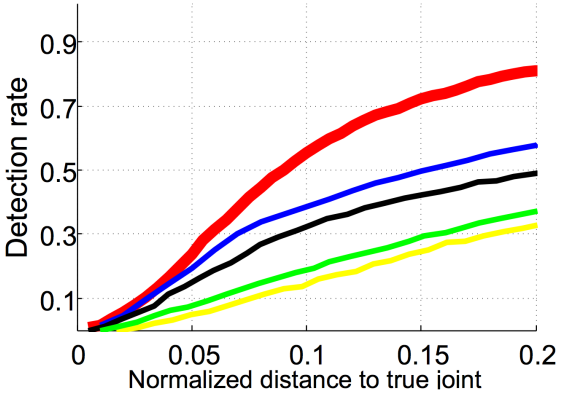
Empirical Evaluation

Movies (FLIC)

Elbows

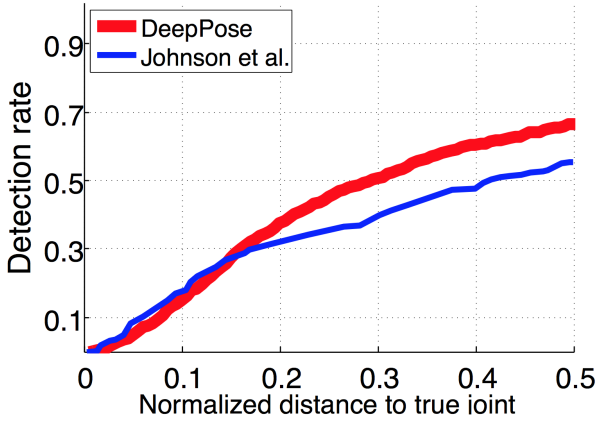


Wrists

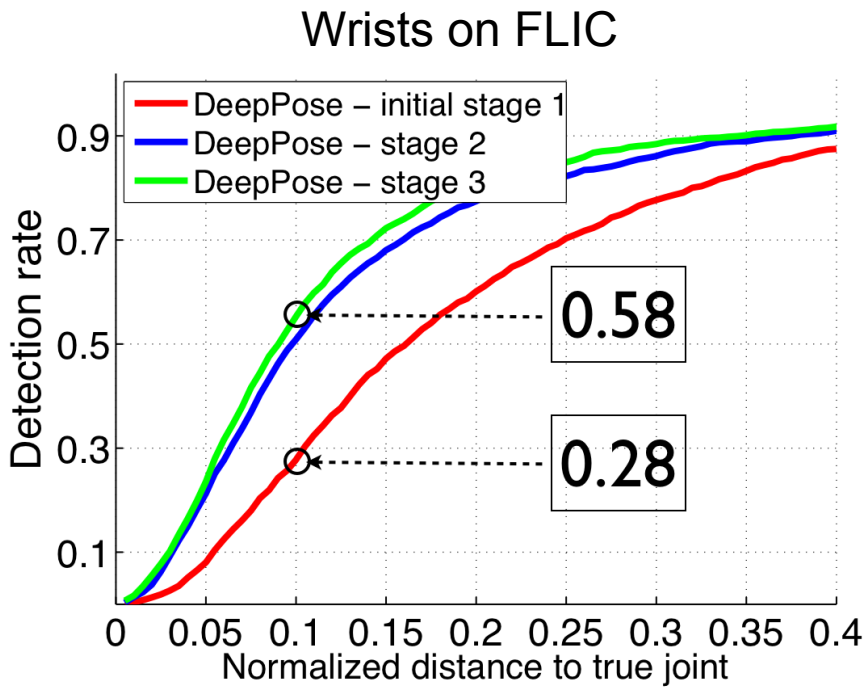


Sports (LPS)

Wrist



Importance of Cascade

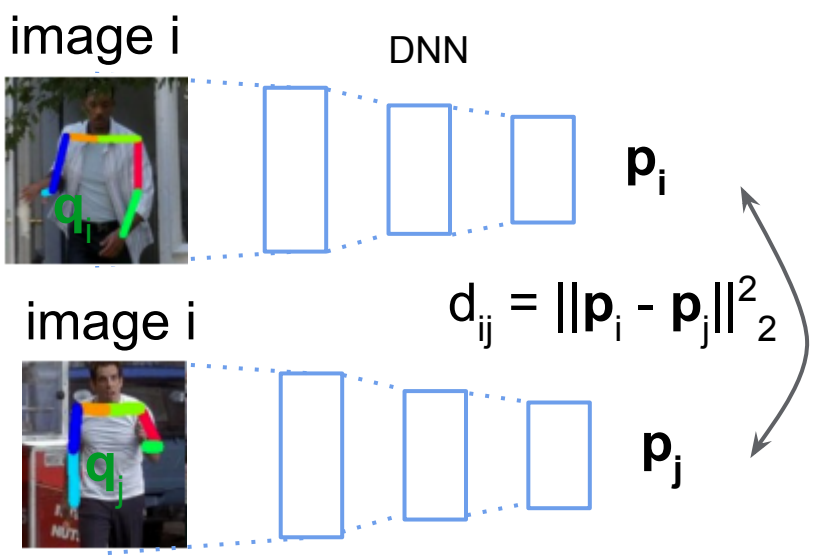


DNN-based NCA Regression

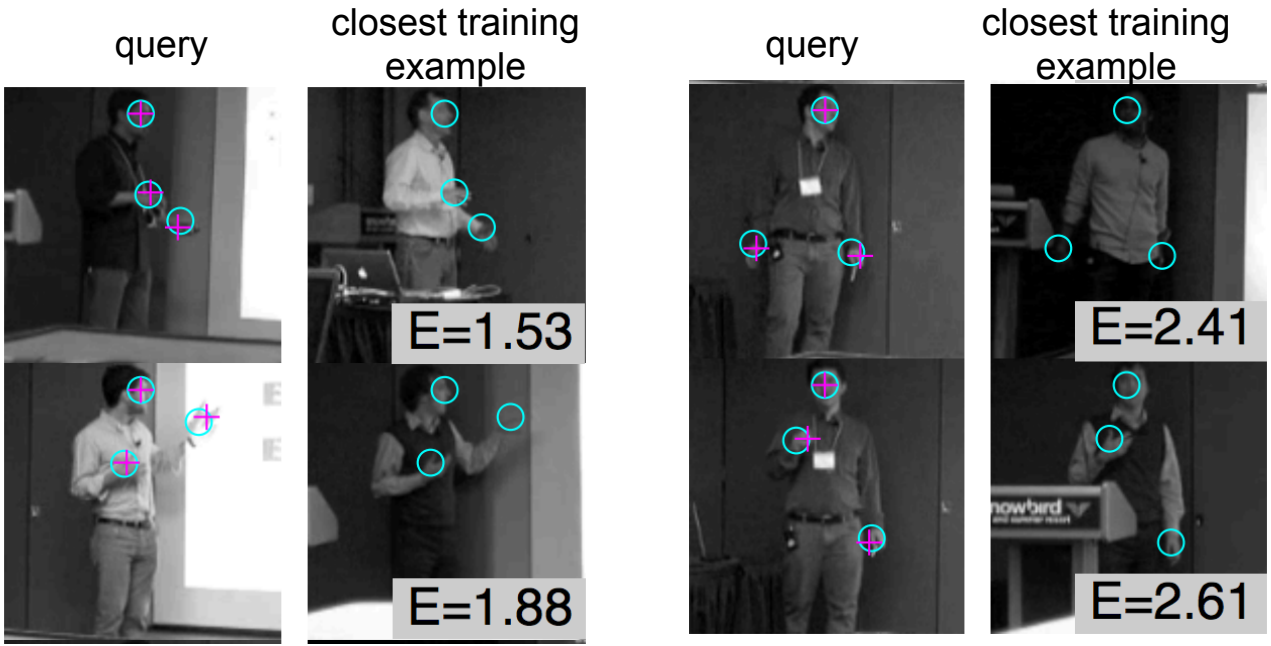
Distances between regressed vectors mimic distances between true poses:

$$\Sigma_i \Sigma_j \alpha_i \exp(-d^2_{ij}) \|\mathbf{q}_i - \mathbf{q}_j\|^2_2$$

with normalizer $\alpha_i = 1 / \Sigma_k \exp(-d^2_{ik})$



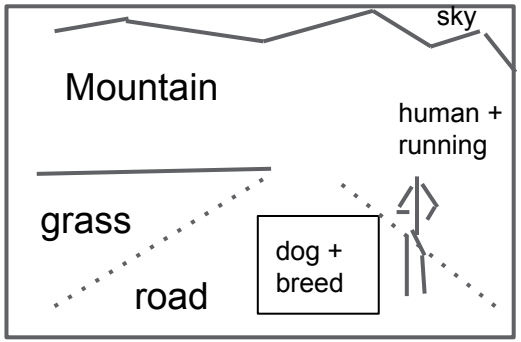
DNN-based NCA Regression



Conclusion

- Various regression formulation for localization problems: detection, segmentation, human pose estimation

- Similar DNN architectures successfull across all problems



Questions

Thanks:
Christian Szegedy, Dumitru Erhan, Drago Anguelov