Image Processing and Computer Graphics

Image Processing

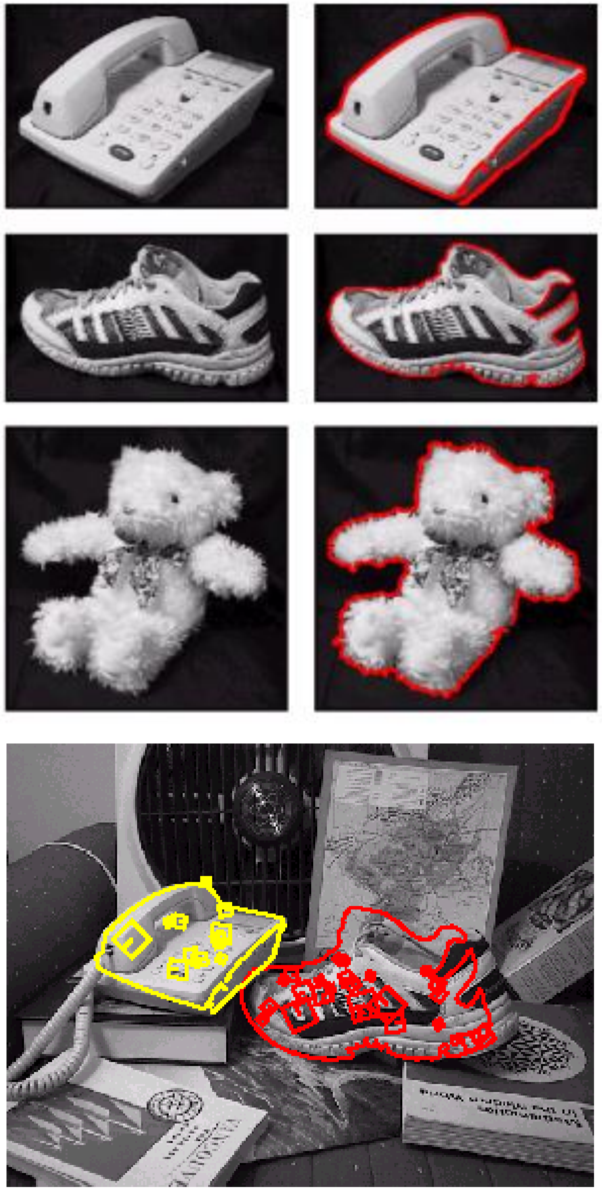
Class 10

Object recognition and deep learning

© 2015-2016 Thomas Brox

What is object recognition?

- **Instance recognition** seeks to detect the presence of a known object in a new image.
- Often the object should also be localized in the image (detection).
- Major problem: the object can look quite different in the new image due to different viewpoint, lighting, occlusions.
- Important difference to **object class recognition** or object class localization: the same instance of an object class is seen in the two images.
- There can be large differences between two instances of an object class (e.g. two dogs).



Author: David Lowe

© 2015-2016 Thomas Brox

A classification problem

- Object recognition consists of two main parts:
  1. A set of **features** that describe the object
  2. A **classifier** that separates objects/object classes

- Classical approach:
  Use a handcrafted feature representation and learn the classifier

- Deep learning:
  Learn the feature representation and the classifier

- Typical classifiers (see also Statistical Pattern Recognition):
  - **Support vector machine** (two-class)
  - **Logistic regression** (multi-class, used in deep networks)
  - Nearest neighbor classifier (multi-class)

© 2015-2016 Thomas Brox

Support vector machine

- Basic idea: learn a decision function with maximum margin (distance to most critical training points).

- Decision function modeled as a linear combination of features:

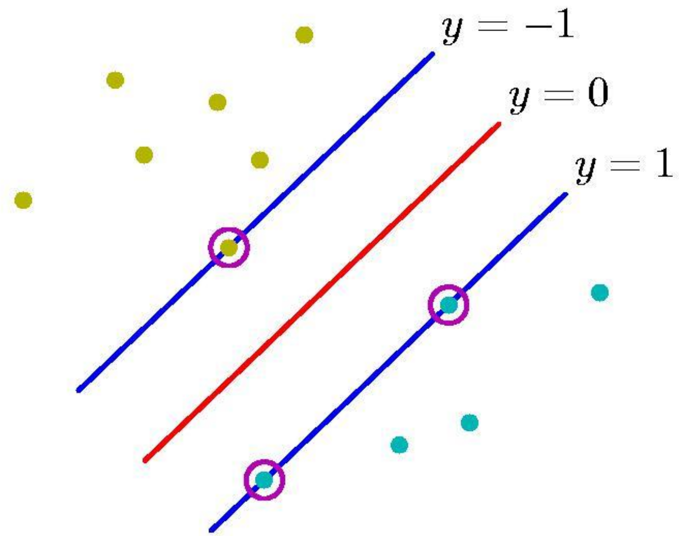
$$y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$$

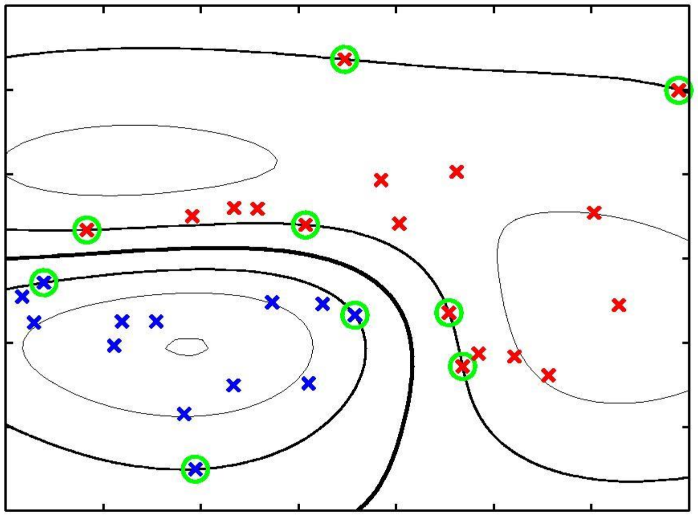
- Large margin concept leads to a convex optimization problem:

$$\mathrm{argmin}_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to $t_n(\mathbf{w}^\top \phi(\mathbf{x}_n) + b) \geq 1$

- Efficient code is publicly available (e.g. libSVM, liblinear)





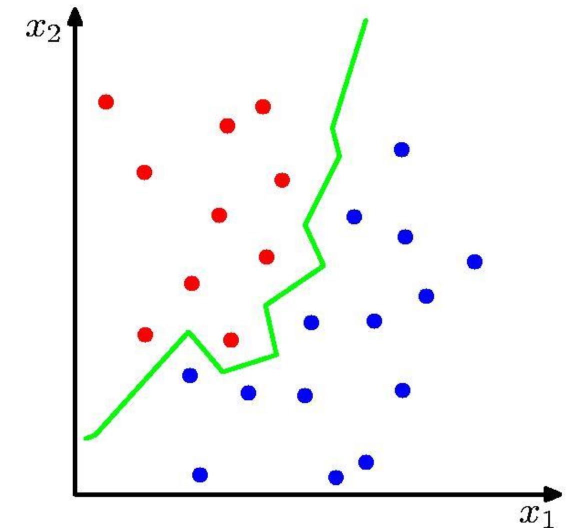
Author: Christopher Bishop

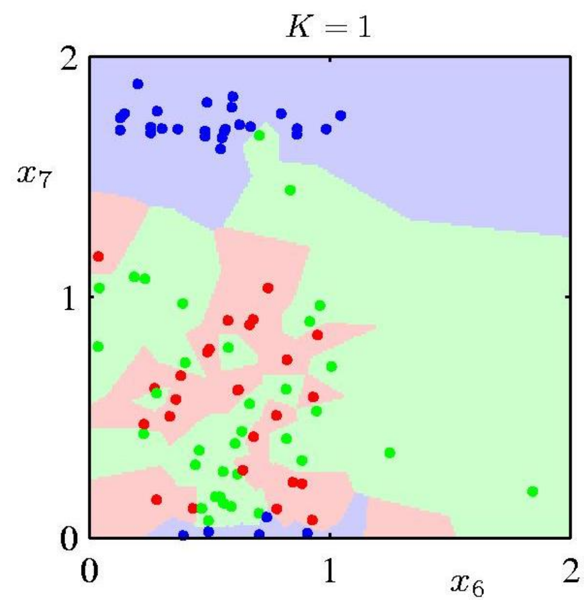
Nearest neighbor classifier

- Simple idea: assign the class label of the most similar training point.

- Advantages:
  - Simple concept
  - Works for multiple classes

- Drawbacks:
  - All training samples must be stored.
  - Search for most similar training sample may consume too much time
  - Does not generalize well





Author: Christopher Bishop

Feature representation

- Color
  - Discriminative capabilities very restricted (tomato = red car)

- Local descriptors (e.g. SIFT)
  - Discriminative properties good (if sufficiently textured)
  - Easy to extract from input images
  - Describe the object **locally**
    → robust to occlusions, local variation
  - Fixed level of abstraction
    → problems with complex variations

- Deep representations
  - Multiple levels of abstraction
  - Learned from training data

© 2015-2016 Thomas Brox

Invariance requirements in object recognition

- Aim: the object should be recognized even if its appearance has changed due to some typical transformations.

- Then the descriptors and classifiers are called **invariant** with respect to this transformation.

- There is a tradeoff between invariance and discriminative power of a descriptor

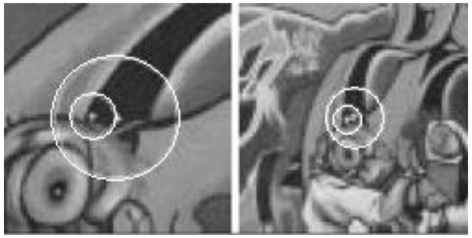
- Instance recognition: invariance needed with respect to
  - Viewpoint (includes translation, scaling, rotation, perspective distortion)
  - Background
  - Lighting
  - Partial occlusion

- Class recognition: needs complex invariant features learned from training examples

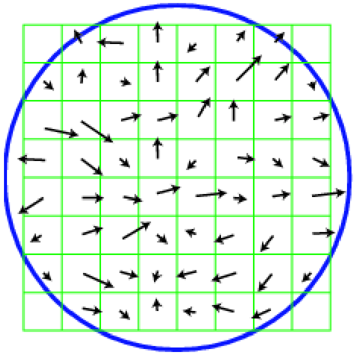
© 2015-2016 Thomas Brox

Some popular local descriptors

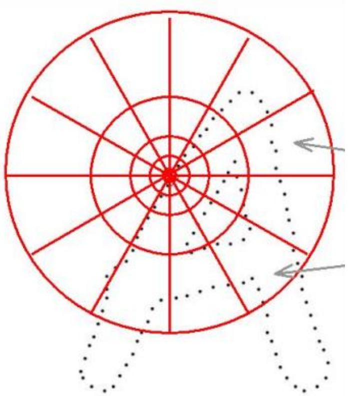
- Distortion corrected patches
  - distortion correction provides affine invariance



- SIFT/HOG (Lowe 2004, Dalal-Triggs 2005)
  - based on gradient orientation histograms
  - can be made invariant to scaling, rotation, and brightness/contrast change



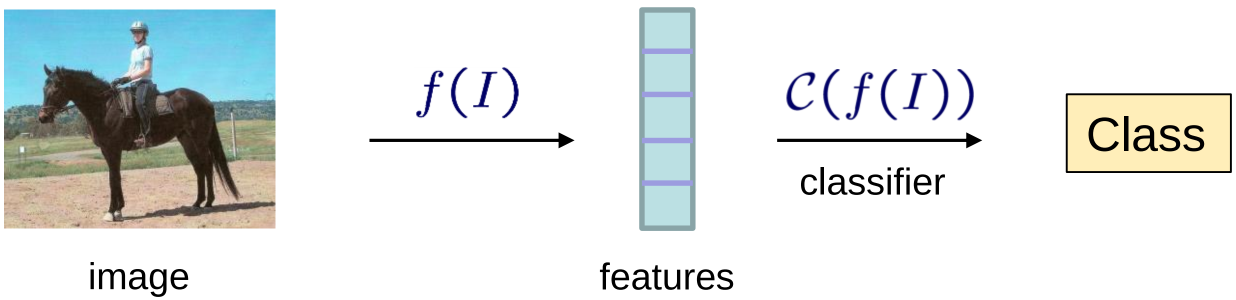
- Shape context (Belongie-Malik 2002)
  - based on object contours or edges
  - histogram of relative position of other contour points in the local neighborhood



Feature learning

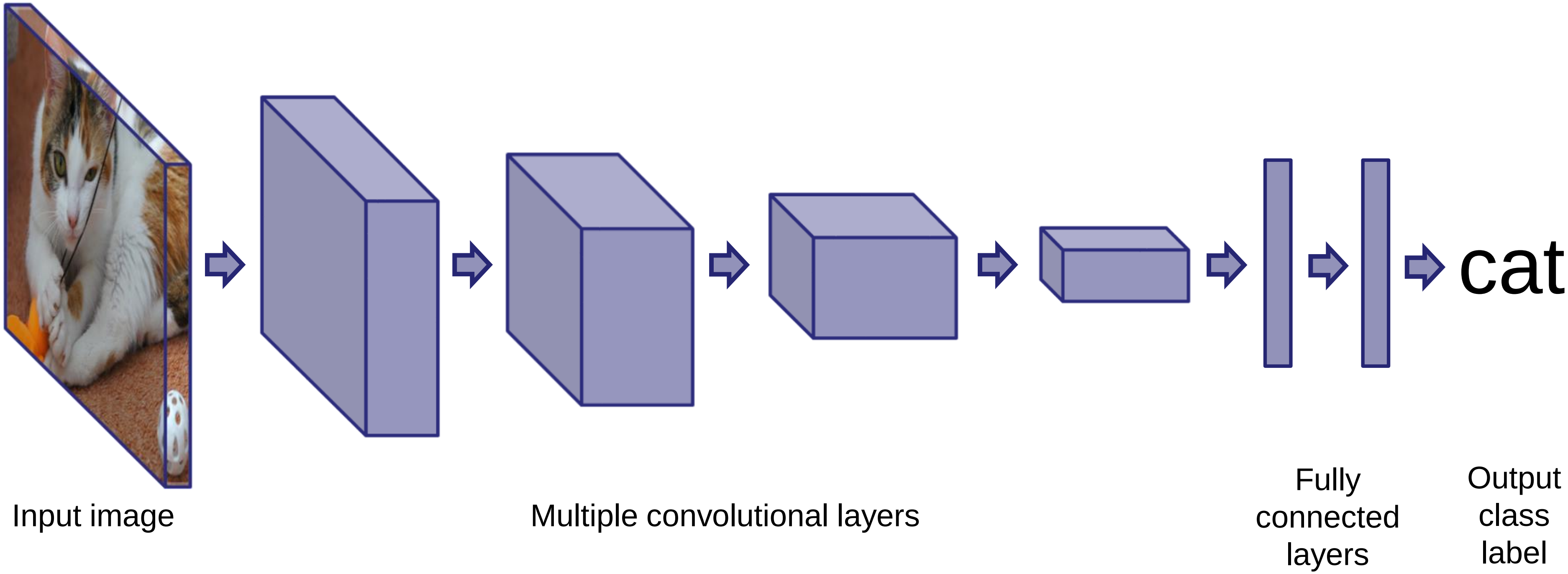
- Instead of manual descriptor design, let the computer find the optimal descriptor for a task defined by a training set

- Task: object classification
→ training set consists of images and their class labels



- Shallow modeling of the function $f(I)$ is not efficient to cover all the variation that appears in an object class
→ hierarchy of functions, "deep" representation

Typical deep network for image classification

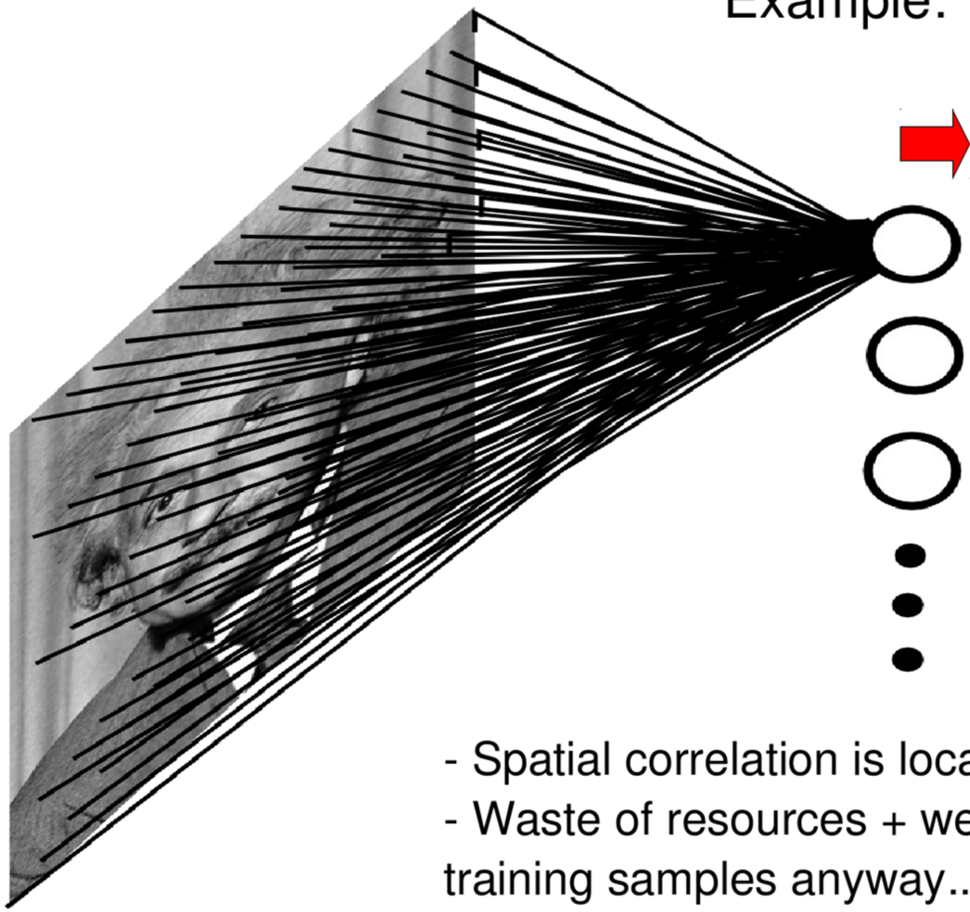


- Each layer produces a more abstract representation of the input
- Layers are similar in structure
- Weights of connections between layers (filters) learned from training data

© 2015-2016 Thomas Brox

Convolutional network vs. fully connected network

Fully Connected Layer



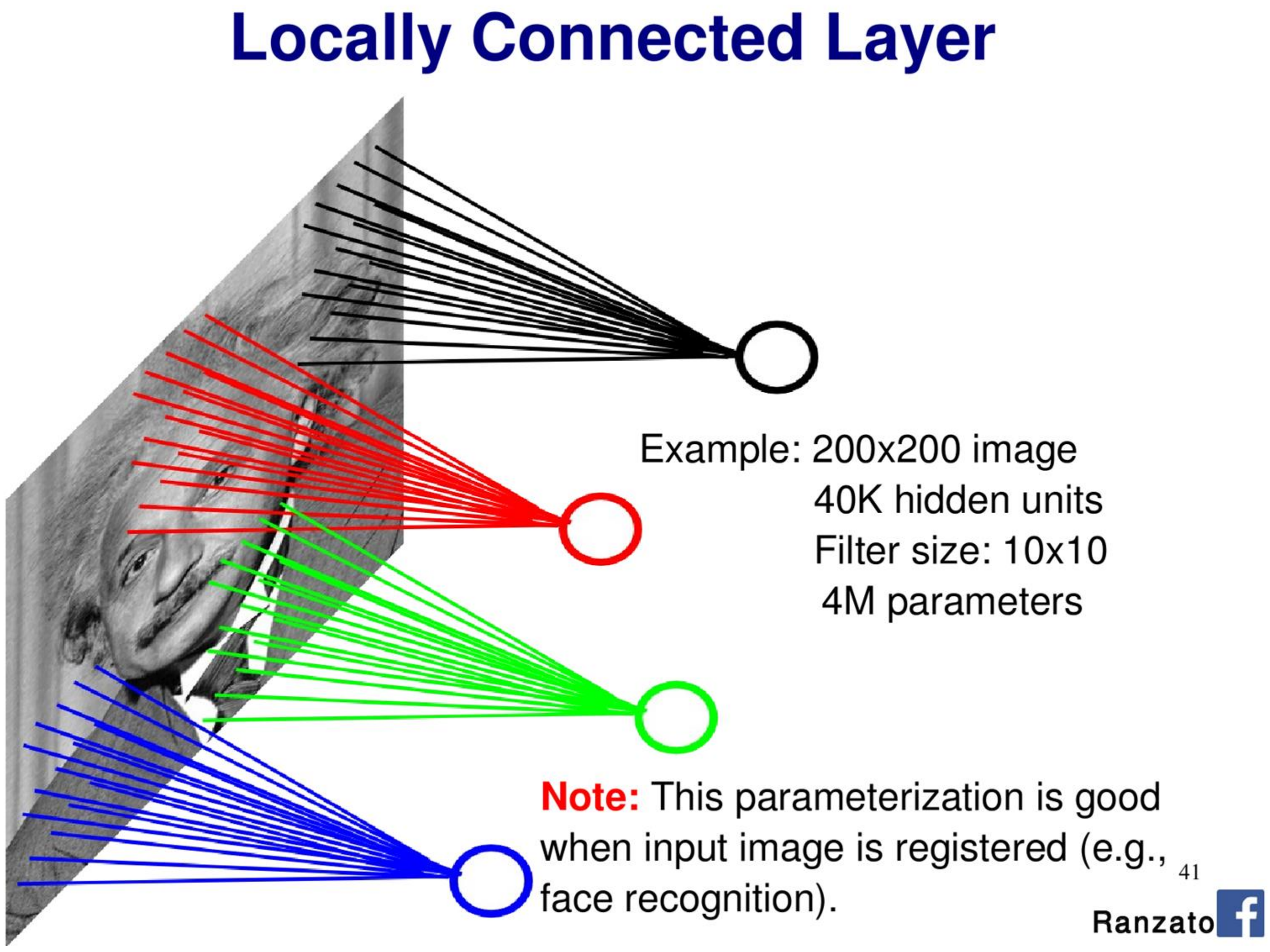
Example: 200x200 image
40K hidden units
~2B parameters!!!

- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

40

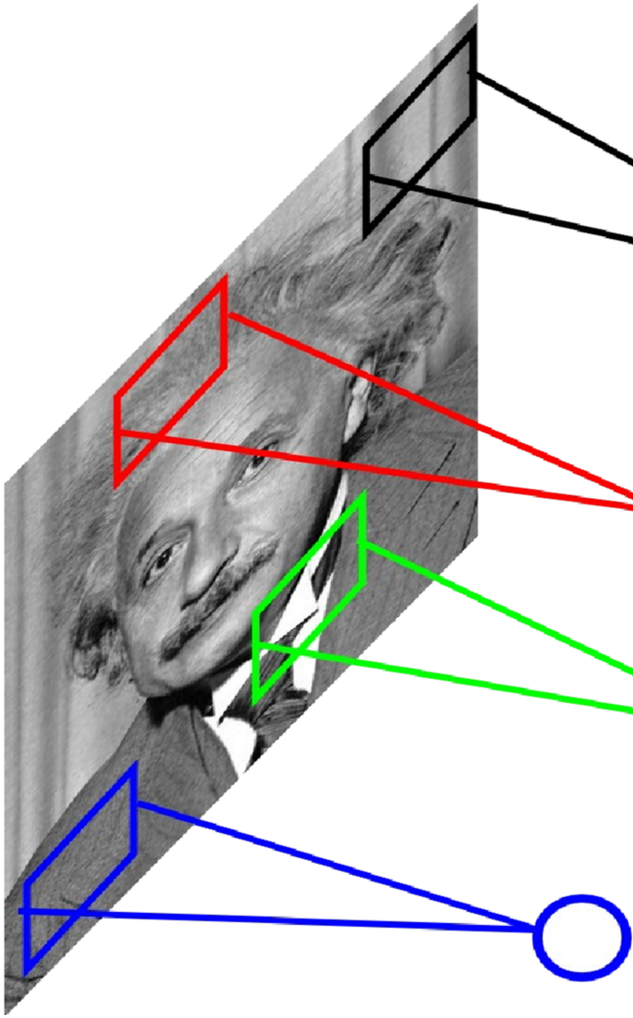
Ranzato

Convolutional network vs. fully connected network



Convolutional network vs. fully connected network

Locally Connected Layer



STATIONARITY? Statistics is similar at different locations

Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

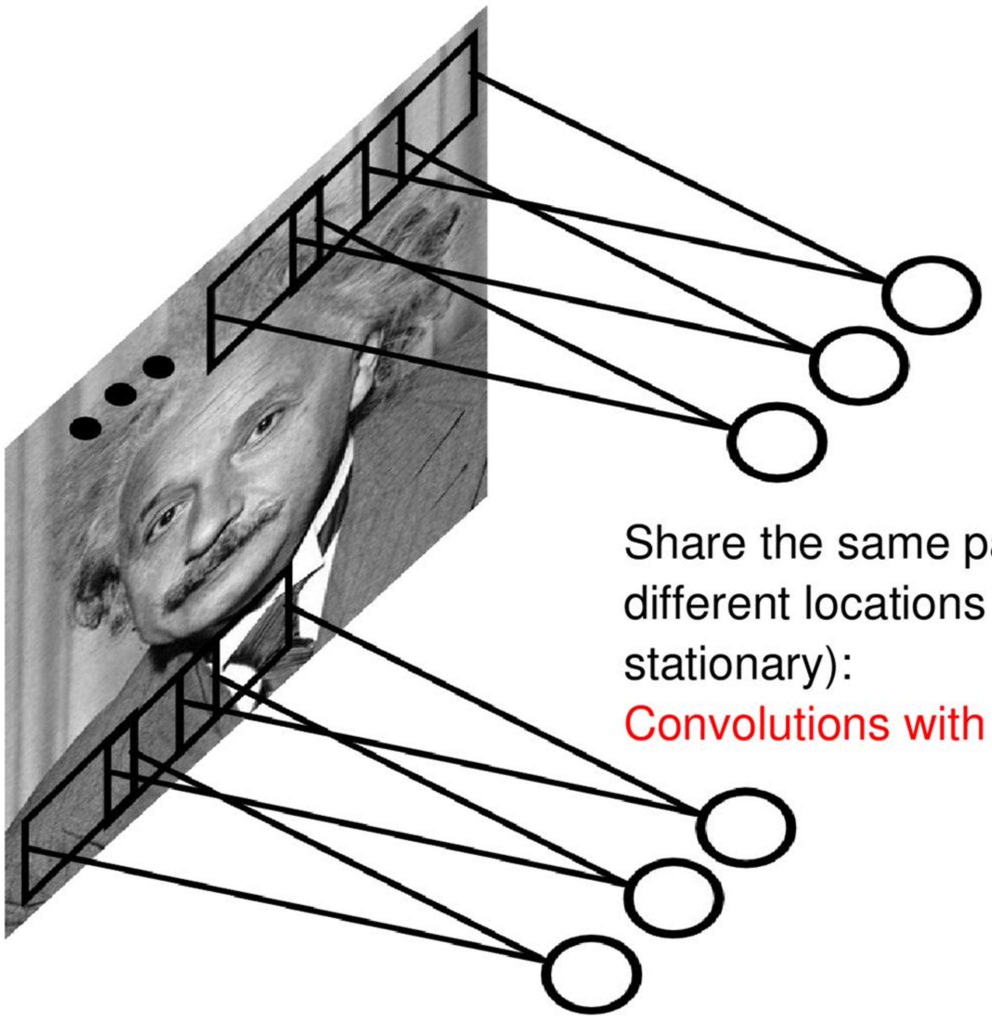
Note: This parameterization is good when input image is registered (e.g., face recognition).

42

Ranzato

Convolutional network vs. fully connected network

Convolutional Layer

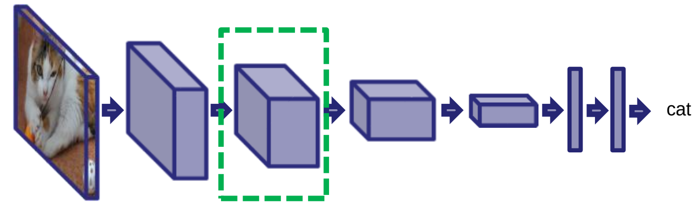


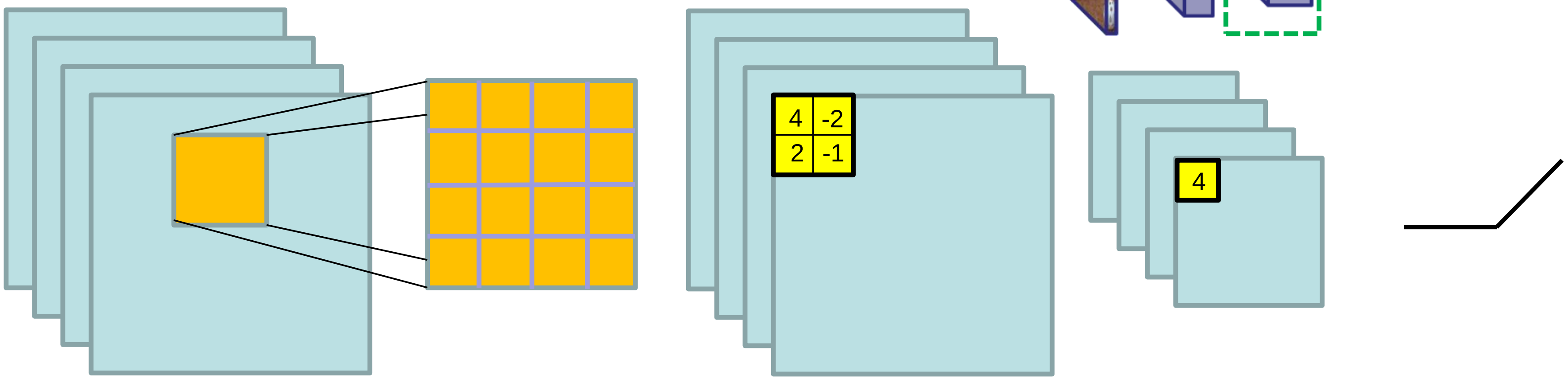
Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

43

Ranzato

Single layer module of a convolutional network





Input (image or feature map from previous layer) | m filter masks (learned) | Feature map (m filter outputs) | Max-pooling | Nonlinearity ReLU

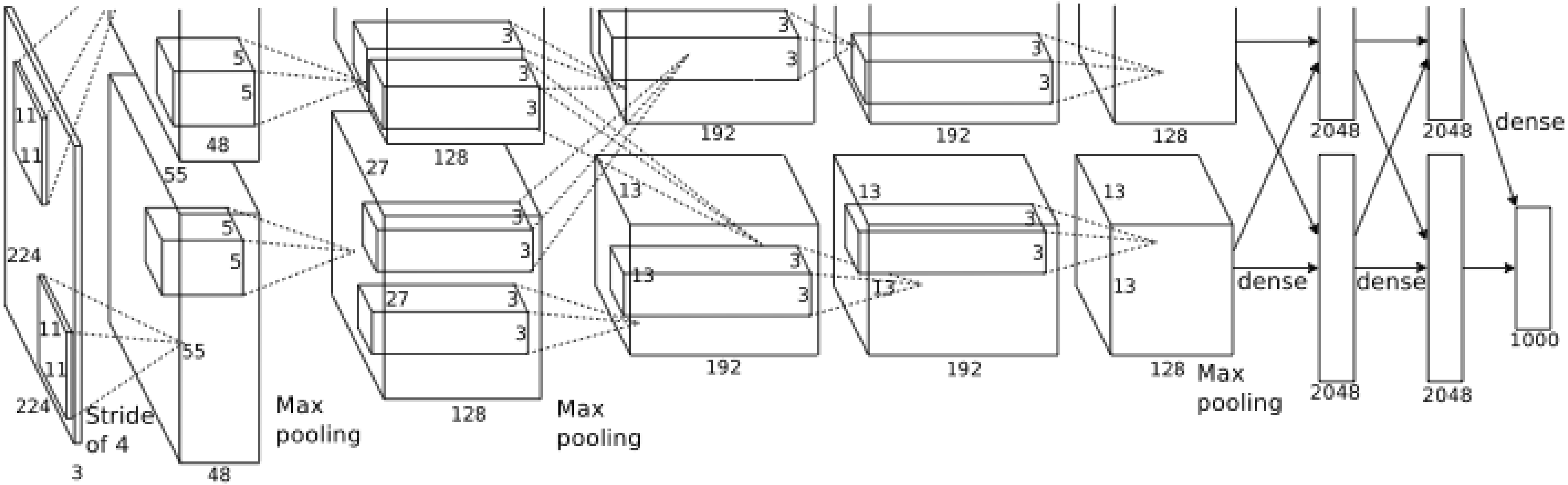
- **Max-pooling** replaces a small local area of the feature map by the maximum value in that area (= downsampling)
  → data reduction, loss of localization accuracy
  → allows for larger receptive fields in the next layer

- **ReLU nonlinearity** sets negative values to 0 (no activation)

© 2015-2016 Thomas Brox

Large ConvNet

Author: Alex Krizhevsky



- Modern networks have 30 layers or more

- Cost function for training the weights: cross entropy on training set (in principle the classification error, but better)

$$\mathbf{w}^* = \mathrm{argmin}_{\mathbf{w}} - \sum_i \sum_c y_i^c \log(f_{\mathbf{w}}^c(x_i))$$

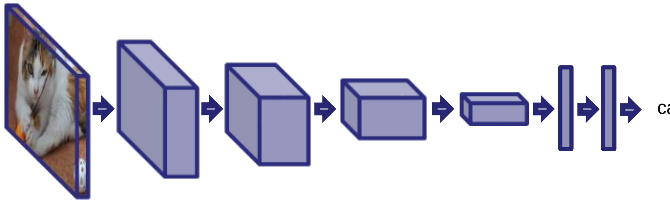
- Optimization by gradient descent (**back-propagation**)

Back-propagation

Optimization problem:

$$L(\mathbf{w}) = -\sum_i y_i^\top \log(f_{\mathbf{w}}(x_i)) \qquad \mathbf{w}^* = \text{argmin}_{\mathbf{w}} L(\mathbf{w})$$

1. Initialize the weights (starting point for gradient descent)
2. Compute $f_{\mathbf{w}}(x_i)$ by forward-propagating a sample through the network
3. Gradient with regard to a weight in a certain layer: use **chain rule**



$$\frac{dL}{df} = \sum_i (f_{\mathbf{w}}(x_i) - y_i)$$

Error

$$\frac{dL}{dw_{l_{\max}}} = \boxed{\frac{dL}{df}} \frac{df}{dw_{l_{\max}}}$$

Error propagated to the last layer

$$\frac{dL}{dw_{l_{\max-1}}} = \boxed{\frac{dL}{df}\frac{df}{dh_{l_{\max-1}}}} \frac{dh_{l_{\max-1}}}{dw_{l_{\max-1}}}$$

Error propagated to the second last layer

and so on

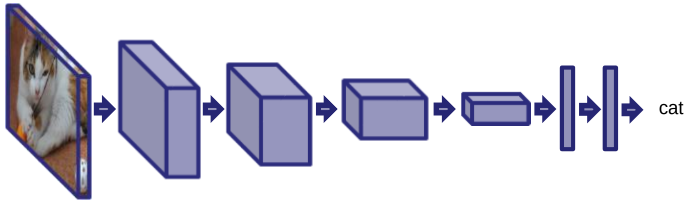
4. Update the weights

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \tau \nabla L$$

© 2015-2016 Thomas Brox

Nice properties of CNNs

- Filters of different layers capture different scales due to max-pooling
  - Filters at low layers are well localized and consider only a small image area (small **receptive field**).
  - Filters at high layers are badly localized and capture the context of a large image area (large receptive field).
- Feature sharing: the same features obtained at low layers can be useful for assembling various complex features at higher layers



- Successive abstraction

  Features close to the image input resemble simple edge filters

  Features close to the class output are invariant to the typical appearance variations

© 2015-2016 Thomas Brox

Localization tasks

- Image classification only provides a class label per image

- **Object localization**: provide a bounding box of the object

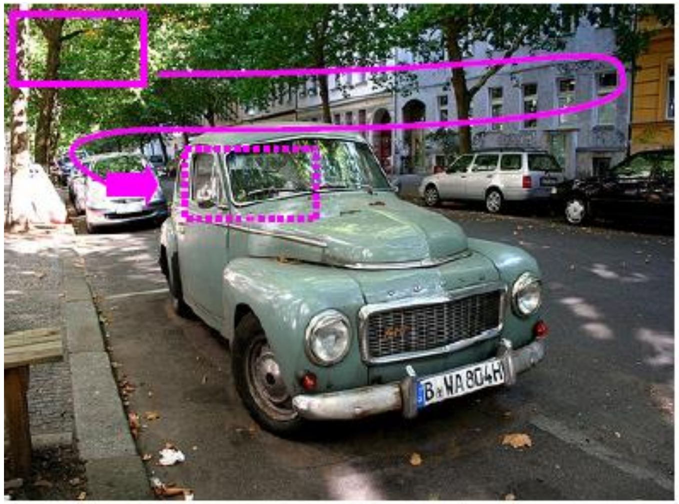
- **Object detection**: bounding boxes for potentially many object instances in the image

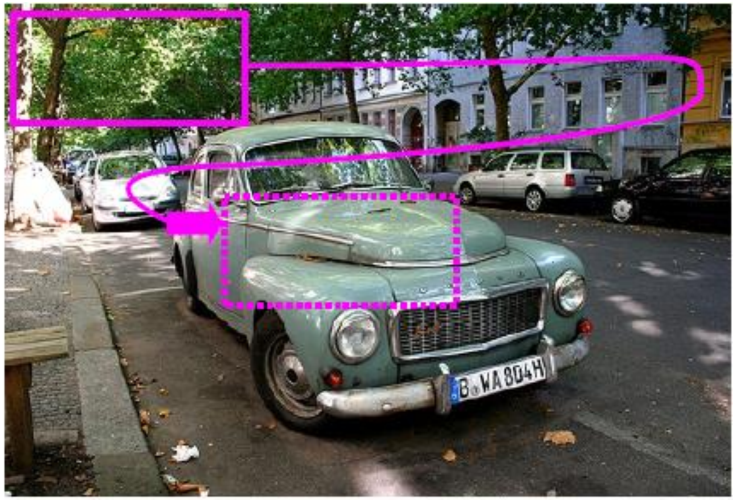
- **Semantic segmentation**: say for each pixel to which object class it belongs

- **Instance segmentation**: additionally separates different class instances

Sliding window approach (Viola-Jones 2001)

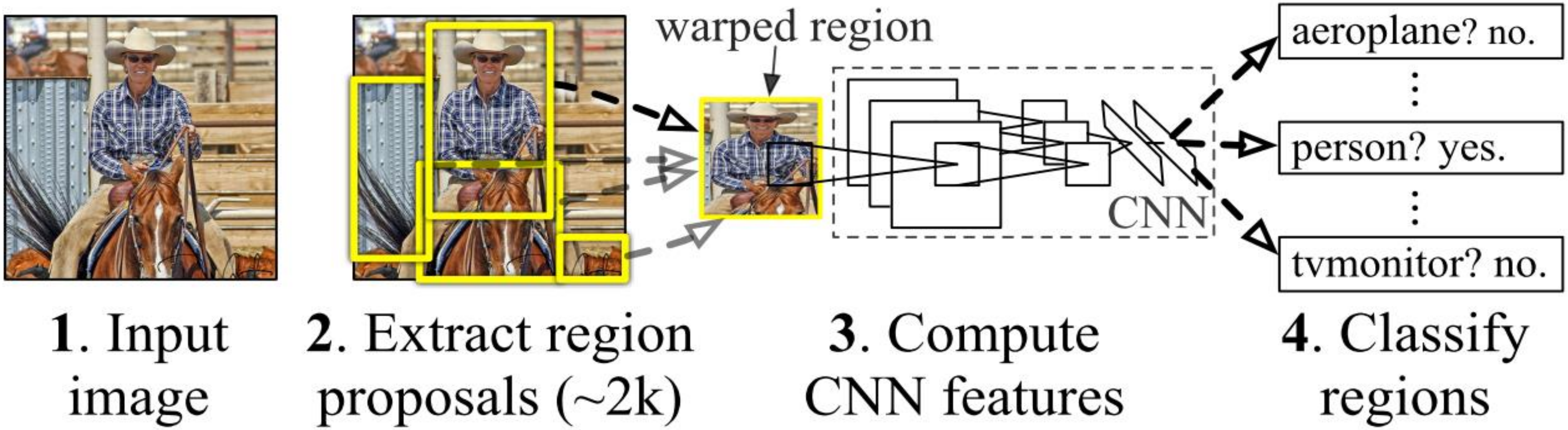
- Define features in a local window
- Consider many (or all) positions and scales and make a binary decision: Is this window a car or not?





- Non-maximum suppression: keep only the local maxima
- Convolutional networks and the sliding window concept are redundant → exploit for larger efficiency (Sermanet et al. 2014)

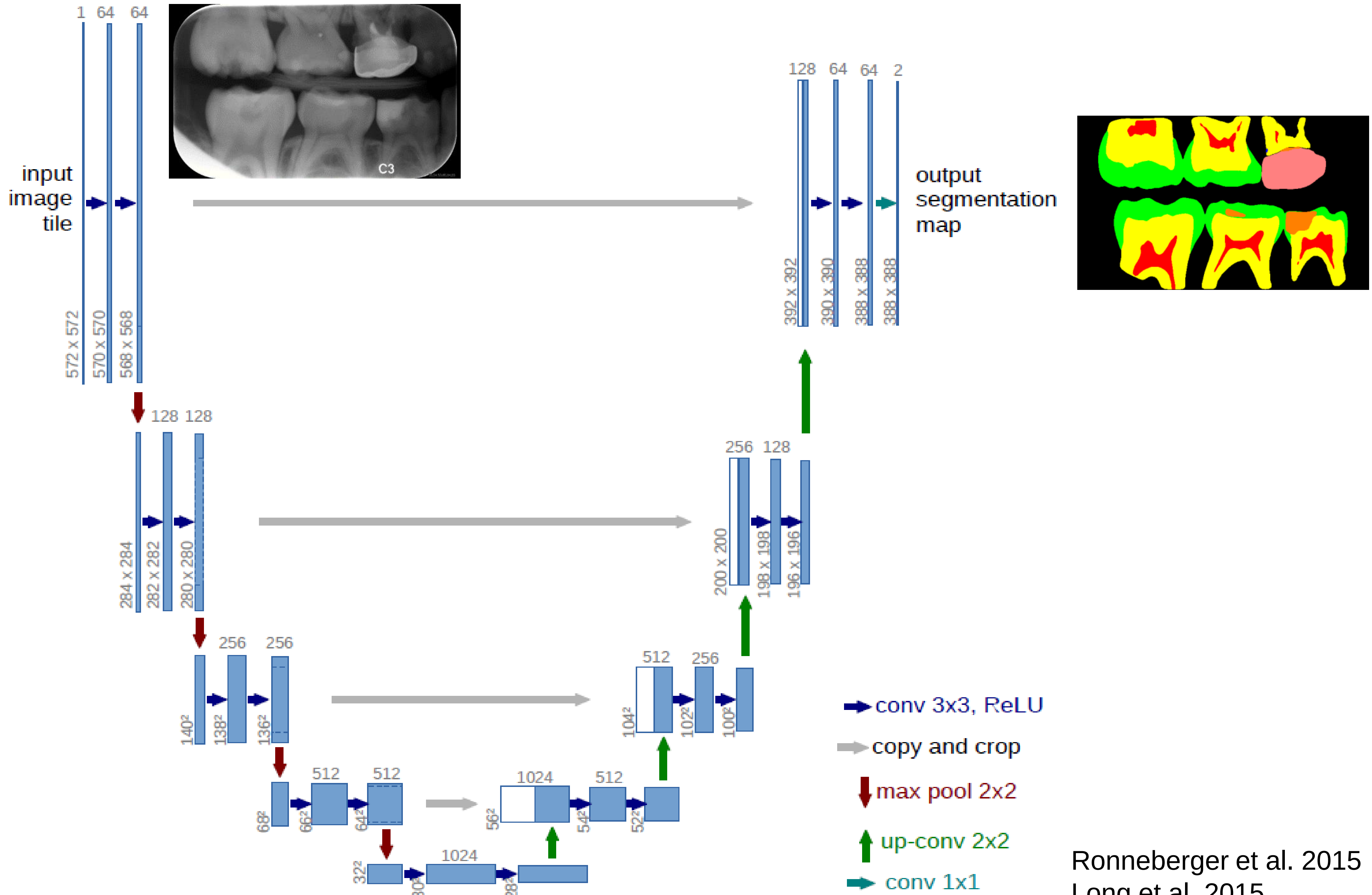
Deep network classification on object window proposals (R-CNN)



Girshick et al. 2014

- Instead of all windows, only ~1000 region proposals are considered
- Faster implementations first compute the ConvNet activations of the whole image and use them to classify the proposal windows

Semantic segmentation with a deep network



Ronneberger et al. 2015
Long et al. 2015

Object recognition benchmarks

- For comparing the performance of different recognition techniques, a common benchmark is needed.
- There are several public benchmarks:
  - ImageNet (localization with 1000 classes, detection): http://www.image-net.org/
  - PASCAL Visual Object Classes (classification, detection, segmentation, …) http://pascallin.ecs.soton.ac.uk/challenges/VOC/
  - Microsoft CoCo (detection, image caption generation) http://mscoco.org/
- Benchmarks come with a **training set** and a **test set** (both annotated)
  - Training set: train classifiers and optimize parameters
  - Test set: run the final method and measure performance
- Detection performance is assessed by precision-recall curves

© 2015-2016 Thomas Brox

Precision-recall curves

- **Precision**: which part of the detected objects is correct?

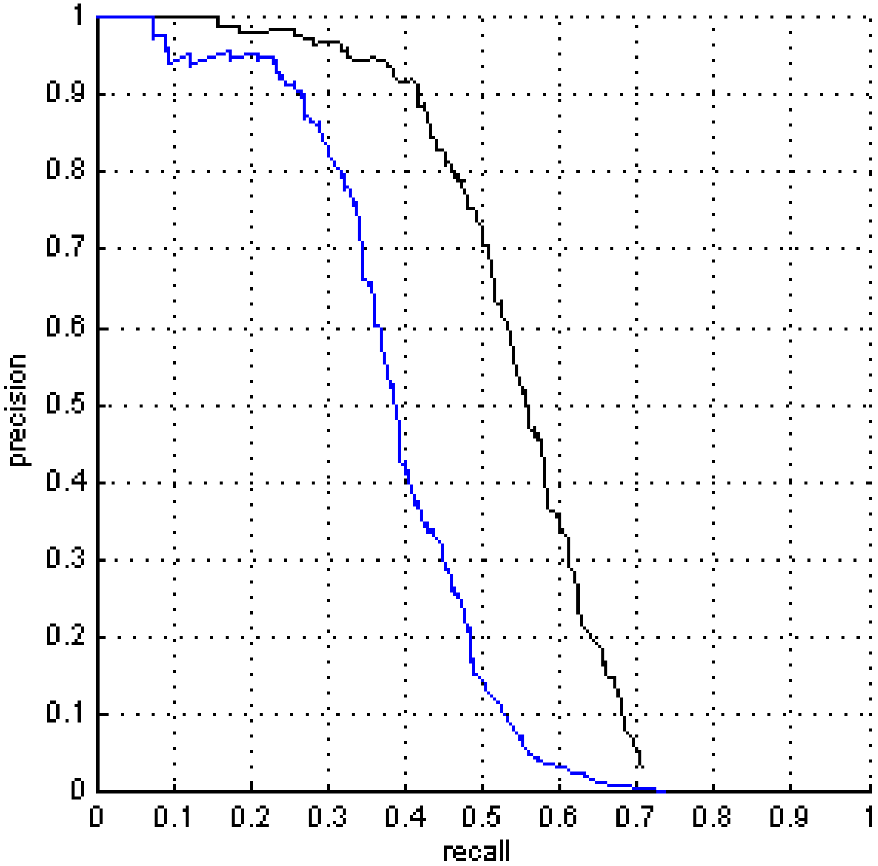
$$\frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

- **Recall**: which percentage of objects is detected?

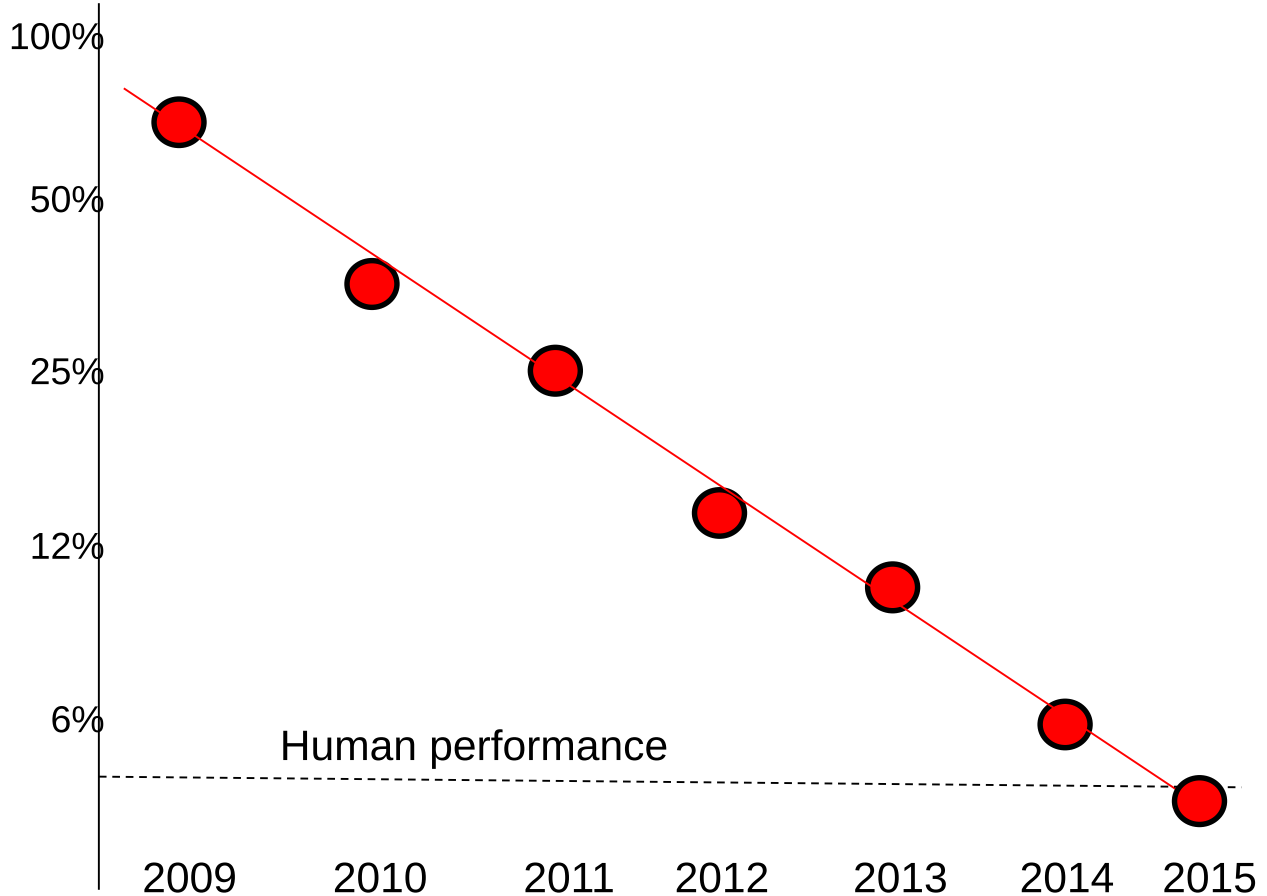
$$\frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

- Precision-recall curves obtained by different thresholds on the classification score

- Single number: **average precision** (area under the curve)



Progress on image classification



Classification error on ImageNet

© 2015-2016 Thomas Brox

Deep learning formulations for other computer vision tasks

- Image enhancement
- Superresolution

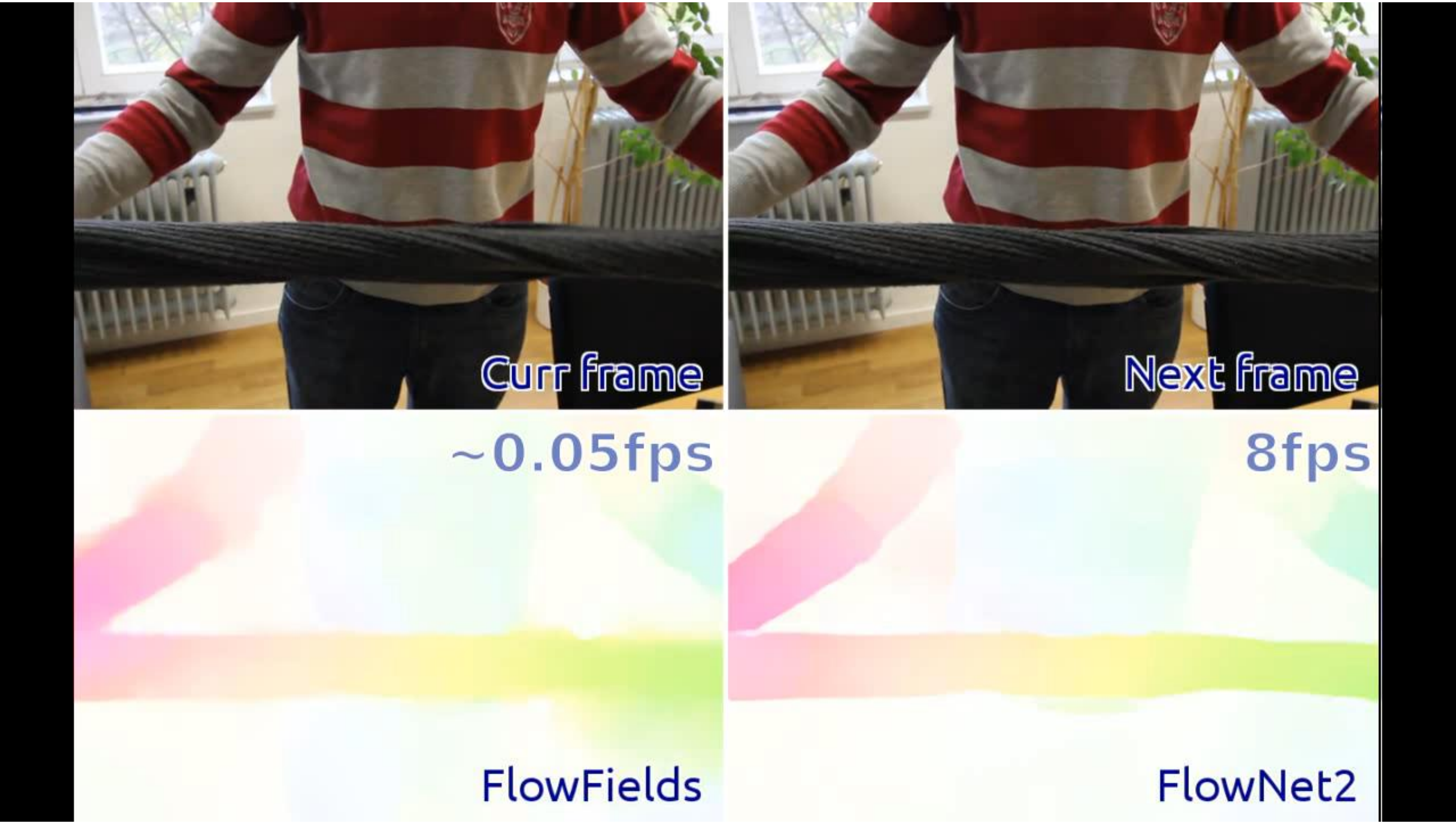
- Body part and pose estimation
- Action recognition

- Depth from single image
- Optical flow estimation
- Disparity estimation
- Structure from motion

- Image based control
- Image based planning

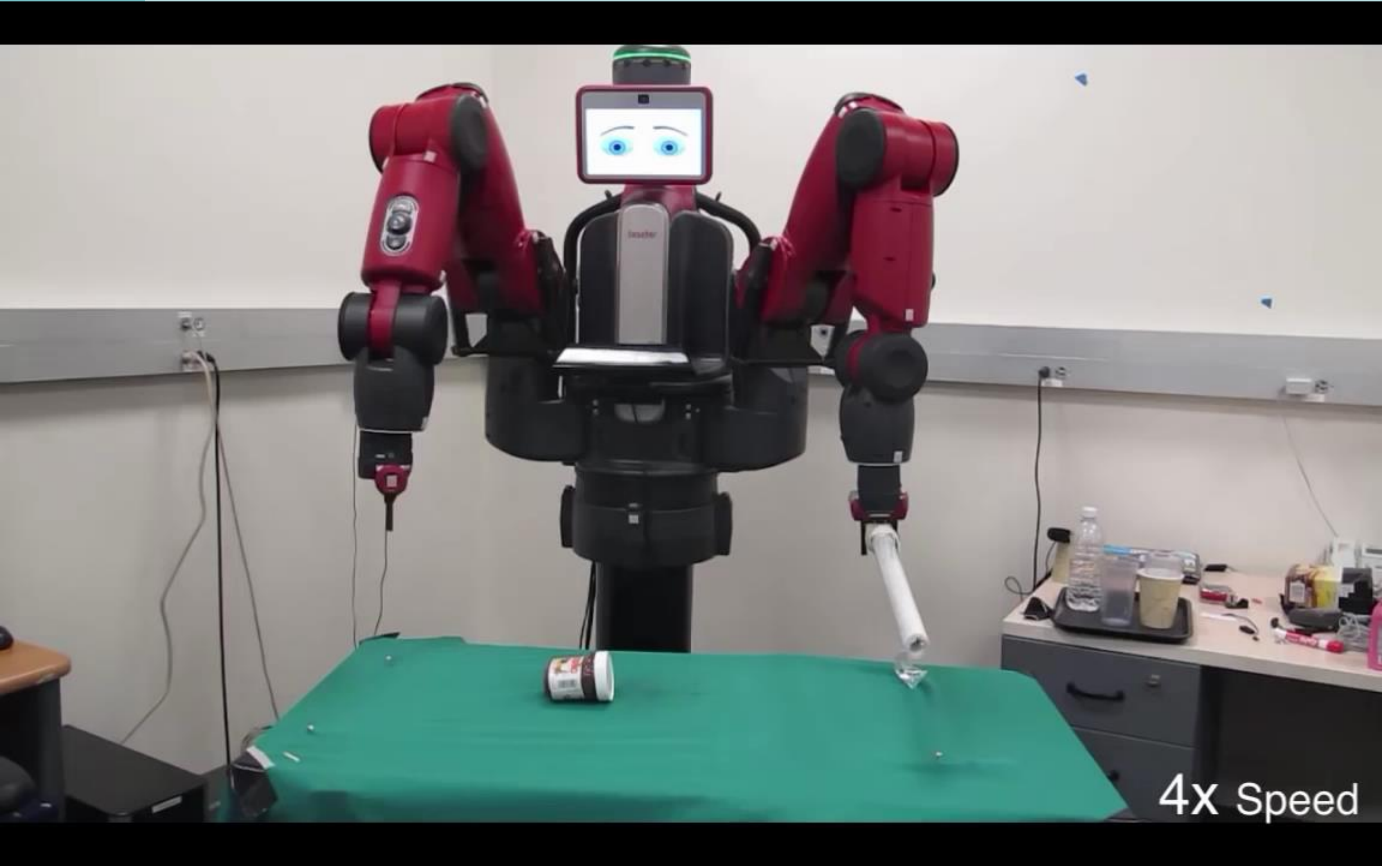
© 2015-2016 Thomas Brox

FlowNet 2.0



© 2015-2016 Thomas Brox

Image based control



Agrawal et al.: Learning to poke by poking

© 2015-2016 Thomas Brox

Summary

- Recognition tasks consist of a feature representation and a classifier
- Deep learning learns a hierarchical feature representation that is usually more powerful than hand-crafted features.
- Deep learning comes down to optimizing a simple (but highly nonlinear and non-convex) loss function with gradient descent
- Convolutional networks use localized filters that are shared across the whole image → vast reduction in the number of parameters
- Object detection and object segmentation require localization of the object (also possible with special convolutional network architectures)
- More or less all tasks can be formulated as learning problems

© 2015-2016 Thomas Brox

References

- A. Krizhevsky, I. Sutskever, G. Hinton: Imagenet classification with deep convolutional neural networks, *Neural Information Processing Systems (NIPS)*, 2012.

- P. Viola, M. J. Jones: Robust real-time object detection. *International Journal of Computer Vision,* 57(2):137-154, 2004.

- P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun: Overfeat: Integrated recognition, localization and detection using convolutional networks, *International Conference on Learning Representations (ICLR)*, 2014.

- R. Girshick, J. Donahue, T. Darrell, J. Malik: Rich feature hierarchies for accurate object detection and semantic segmentation, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

- J. Long, E. Shelhamer, T. Darrell: Fully convolutional networks for semantic segmentation, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

- O. Ronneberger, P. Fischer, T. Brox: U-Net: convolutional networks for biomedical image segmentation, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

© 2015-2016 Thomas Brox

Upcoming classes and courses

- Next week: Computer Graphics

- If you liked the field of image processing and computer vision:
  - Statistical Pattern Recognition (summer, 2+2)
  - Computer Vision (winter, 2+2)
  - Seminar (winter and summer)
  - Deep Learning lab course (winter)
  - GPU Programming lab course (summer)
  - Projects, theses

© 2015-2016 Thomas Brox