Computer Vision – Lecture 16

Deep Learning Applications

11.01.2017

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

Announcements

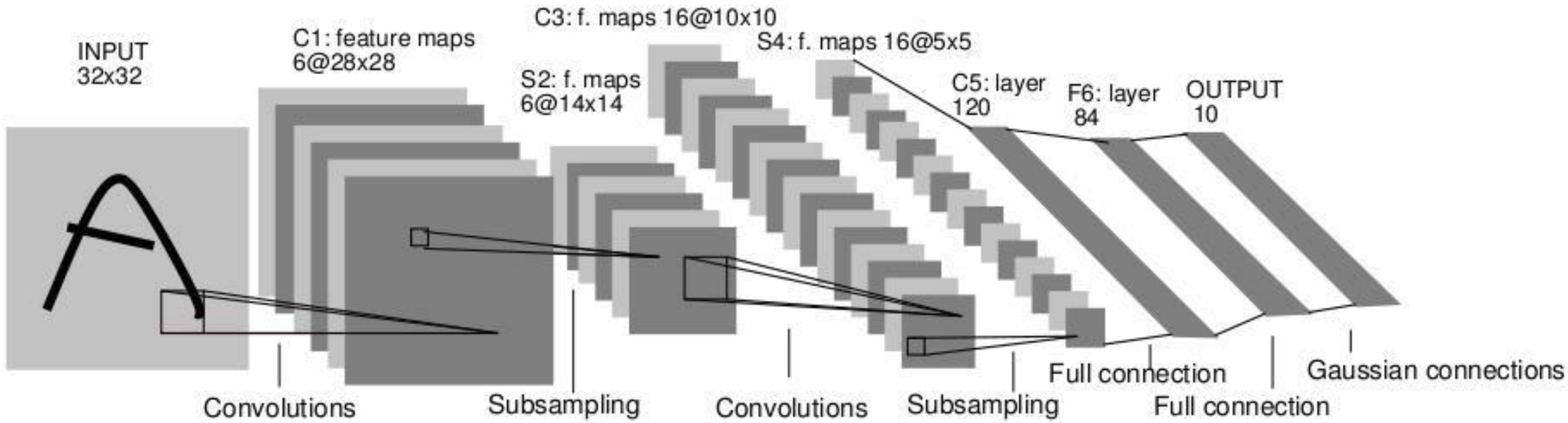
- Seminar registration period starts on Friday
  - We will offer a lab course in the summer semester "Deep Robot Learning"
  - Topic: Deep reinforcement learning for robot control
    - Either UAV or grasping robot
  - If you're interested, you can register at http://www.graphics.rwth-aachen.de/apse
  - Registration period: 13.01.2016 – 29.01.2016

  - *Quick poll: Who would be interested in that?*

Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Object Categorization I
  - Sliding Window based Object Detection
- Local Features & Matching
  - Local Features - Detection and Description
  - Recognition with Local Features
  - Indexing & Visual Vocabularies
- Object Categorization II
  - Bag-of-Words Approaches & Part-based Approaches
  - Deep Learning Methods
- 3D Reconstruction

Recap: Convolutional Neural Networks

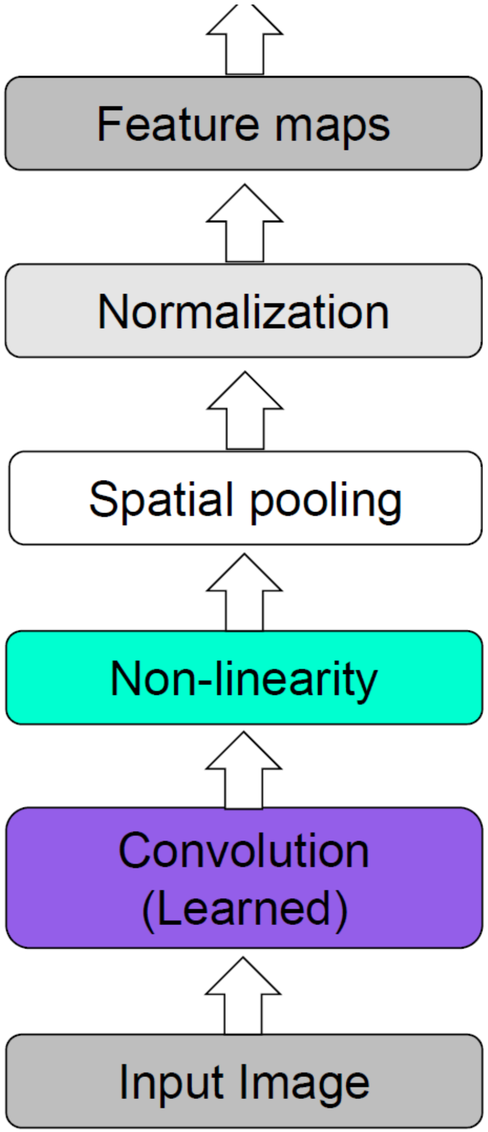


- **Neural network with specialized connectivity structure**
  - **Stack multiple stages of feature extractors**
  - **Higher stages compute more global, more invariant features**
  - **Classification layer at the end**

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, [Gradient-based learning applied to document recognition](), Proceedings of the IEEE 86(11): 2278–2324, 1998.

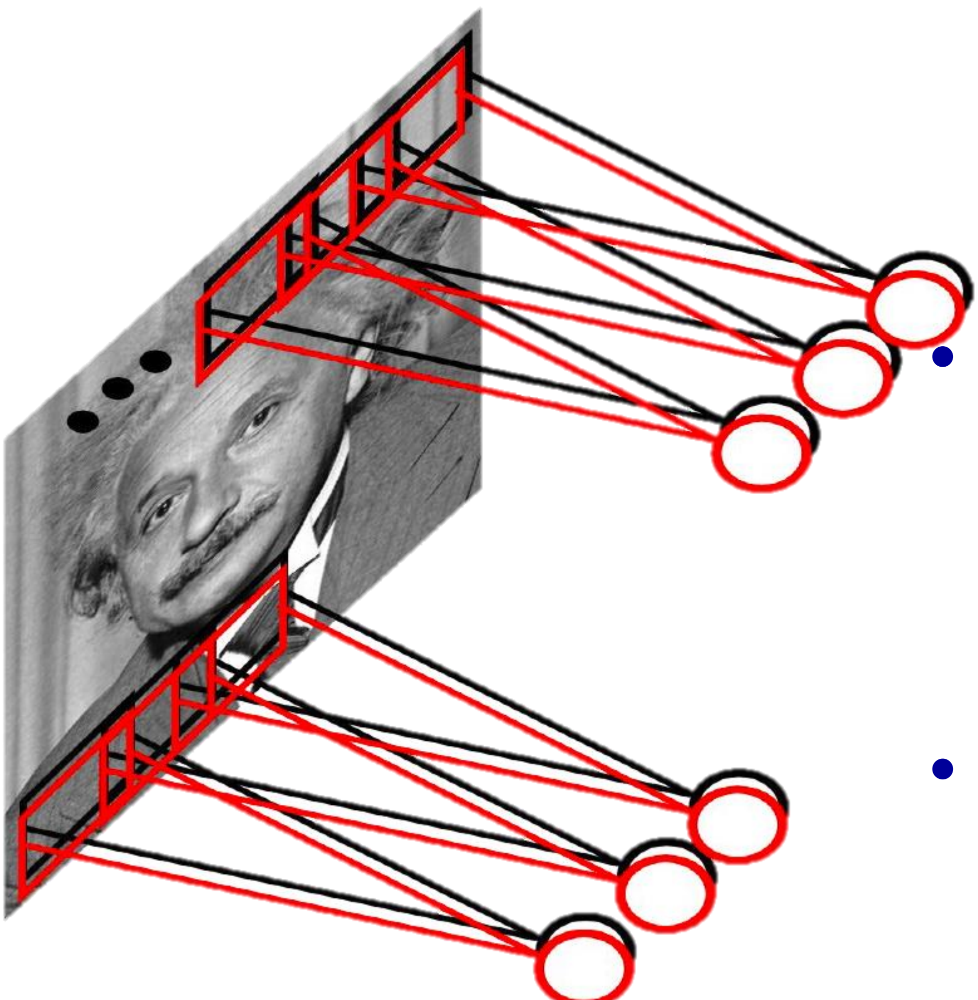
Recap: CNN Structure

- **Feed-forward feature extraction**
  1. Convolve input with learned filters
  2. Non-linearity
  3. Spatial pooling
  4. (Normalization)
- **Supervised training of convolutional filters by back-propagating classification error**



Slide credit: Svetlana Lazebnik

Recap: Intuition of CNNs

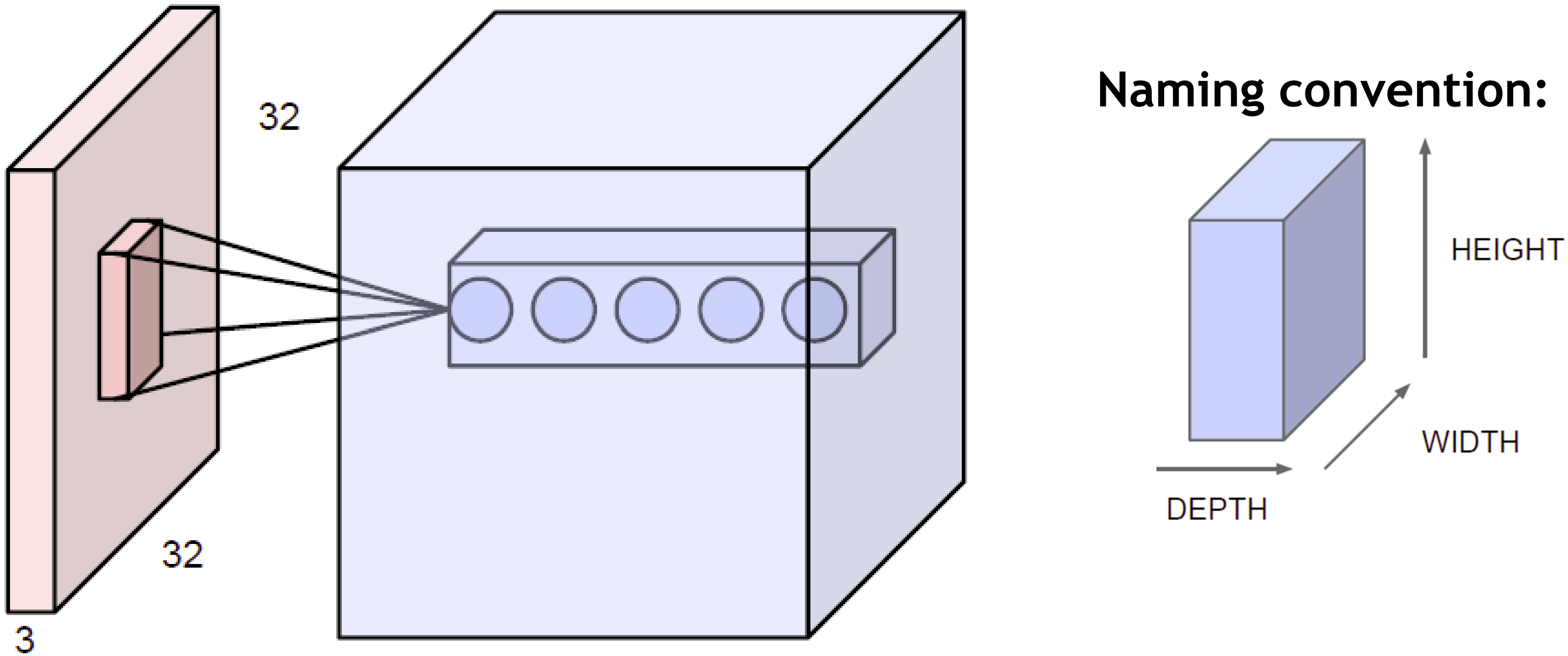


- Convolutional net
  - Share the same parameters across different locations
  - Convolutions with learned kernels
- Learn *multiple* filters
  - E.g. 1000×1000 image
    100 filters
    10×10 filter size

  ⇒ only 10k parameters
- Result: Response map
  - size: 1000×1000×100
  - Only memory, not params!

Image source: Yann LeCun

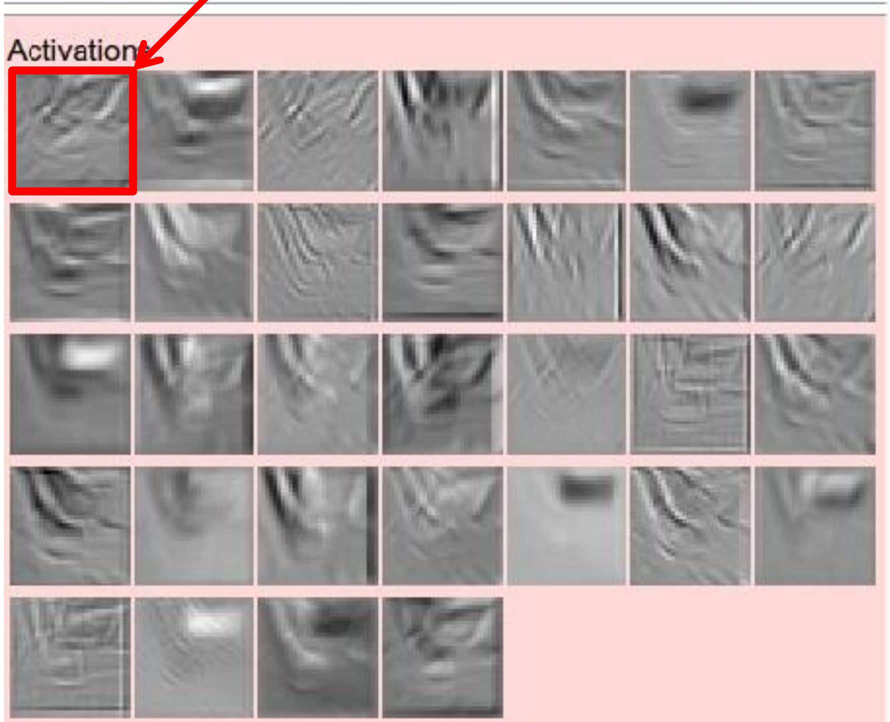
Recap: Convolution Layers



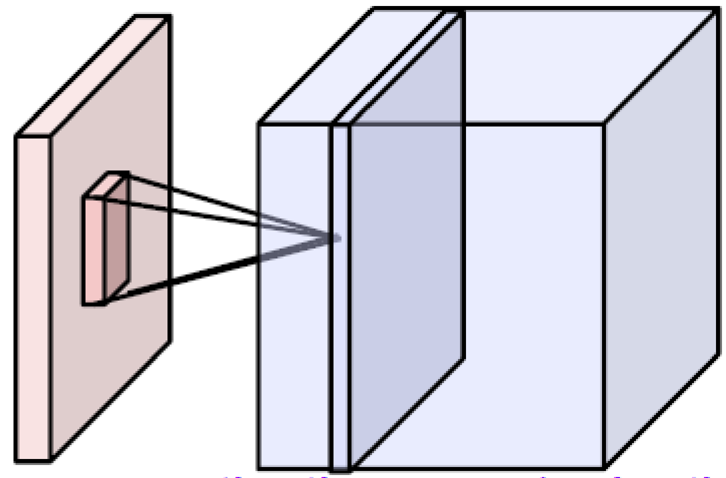
- All Neural Net activations arranged in 3 dimensions
  - Multiple neurons all looking at the same input region, stacked in depth
  - Form a single [1×1×depth] depth column in output volume.

Recap: Activation Maps

5×5 filters

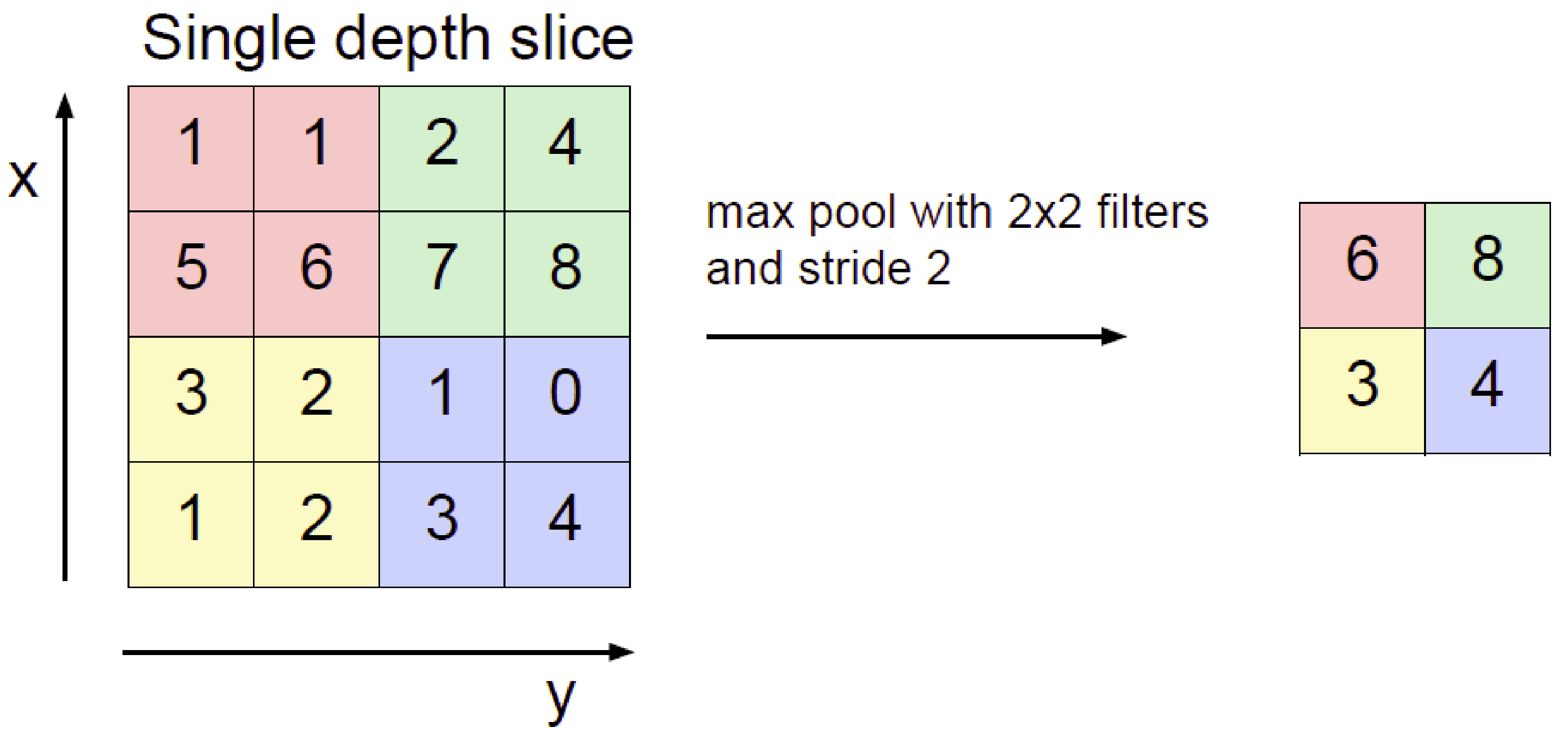


Activation maps



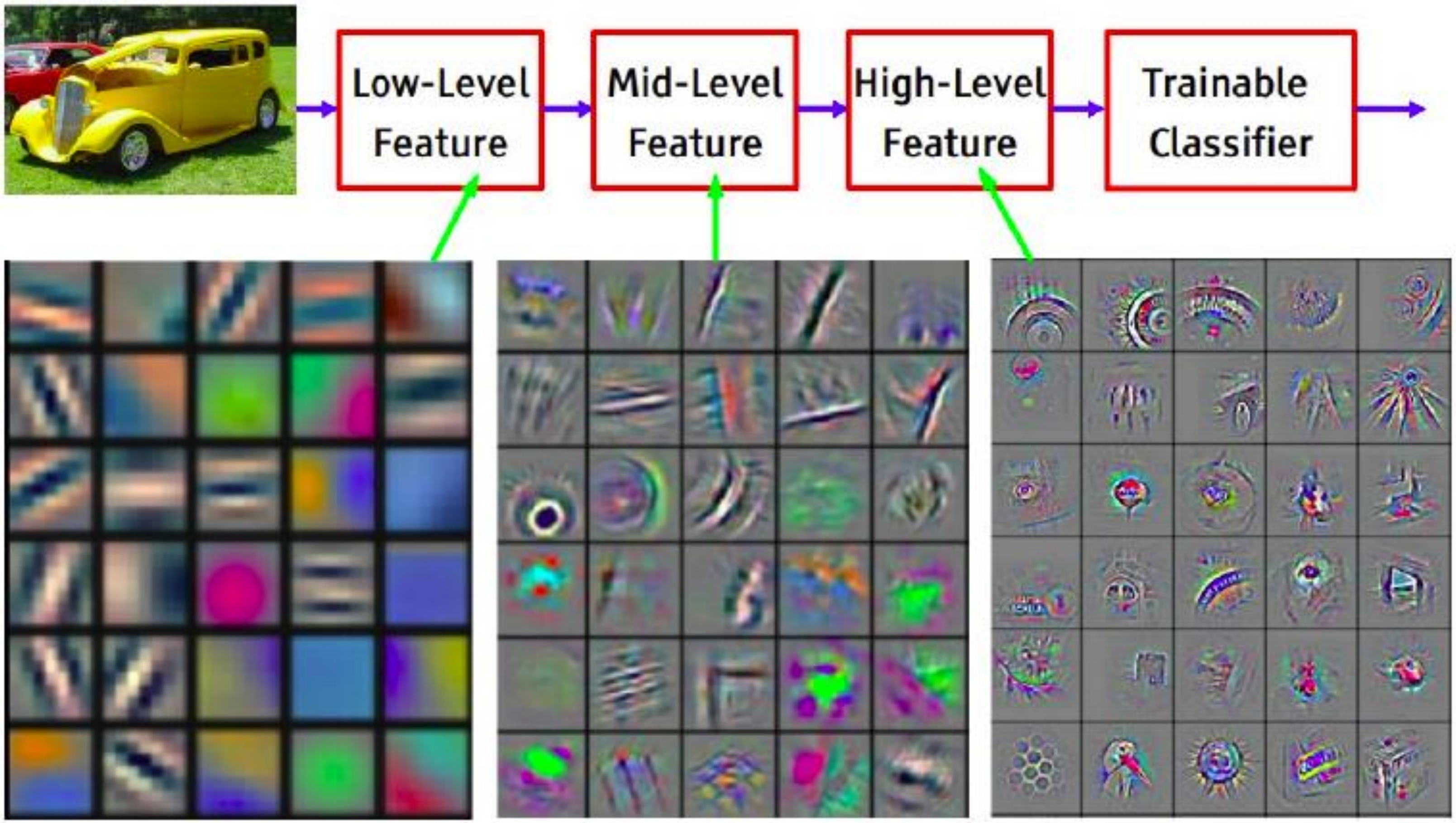
Each activation map is a depth slice through the output volume.

Recap: Pooling Layers



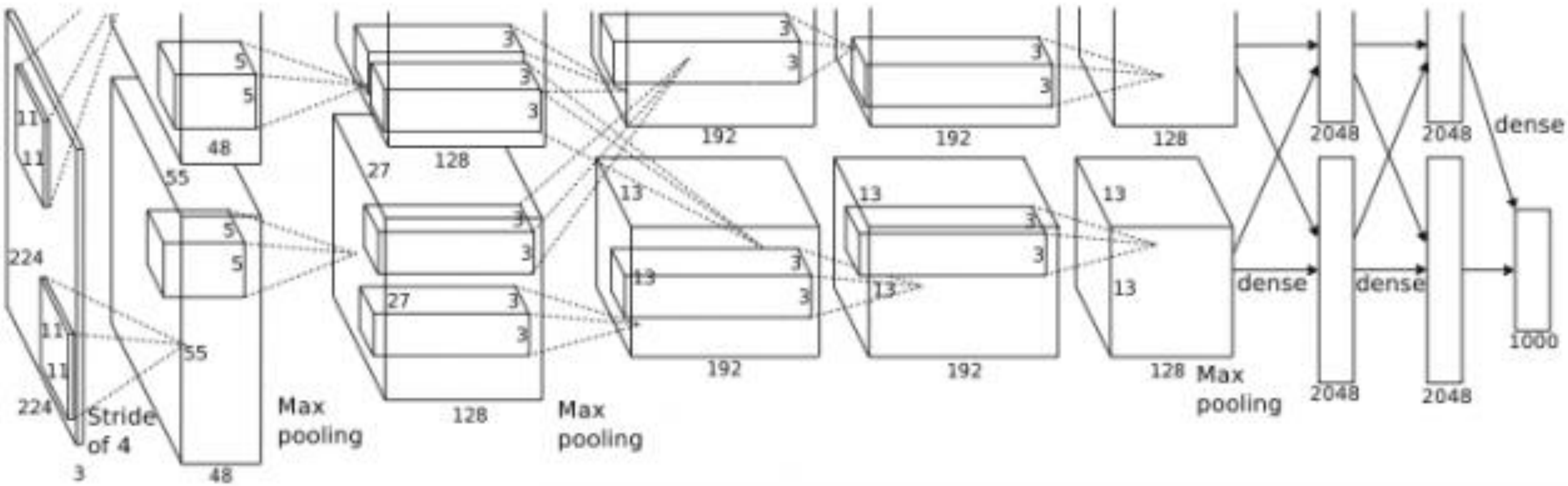
- **Effect:**
  - **Make the representation smaller without losing too much information**
  - **Achieve robustness to translations**

Recap: Effect of Multiple Convolution Layers



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Recap: AlexNet (2012)



- **Similar framework as LeNet, but**
  - **Bigger model (7 hidden layers, 650k units, 60M parameters)**
  - **More data (10^6 images instead of 10^3)**
  - **GPU implementation**
  - **Better regularization and up-to-date tricks for training (Dropout)**

A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012.

Image source: A. Krizhevsky, I. Sutskever and G.E. Hinton, NIPS 2012

Recap: VGGNet (2014/15)

- **Main ideas**
  - Deeper network
  - Stacked convolutional layers with smaller filters (+ nonlinearity)
  - Detailed evaluation of all components
- **Results**
  - Improved ILSVRC top-5 error rate to 6.7%.

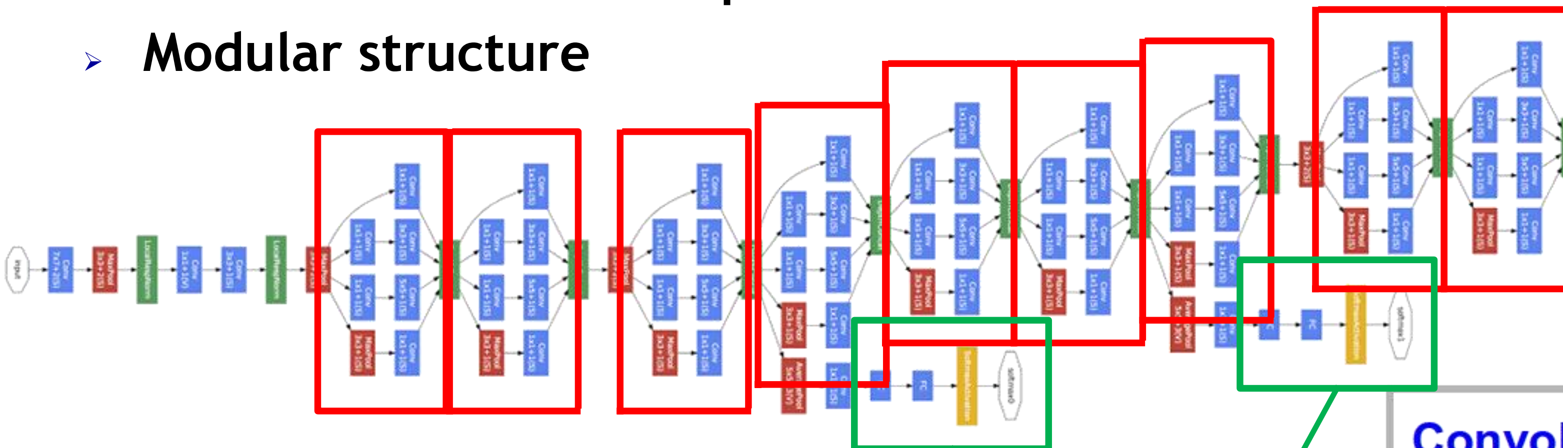
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Mainly used

Image source: Simonyan & Zisserman

Recap: GoogLeNet (2014)

- **Ideas:**
  - Learn features at multiple scales
  - Modular structure

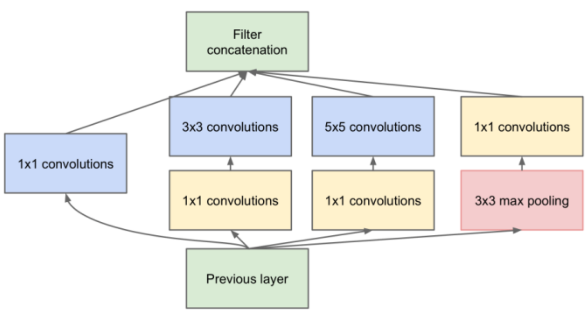


Inception module

+ copies

Auxiliary classification outputs for training the lower layers (deprecated)

Convolution
Pooling
Softmax
Other



(b) Inception module with dimension reductions

Image source: Szegedy et al.

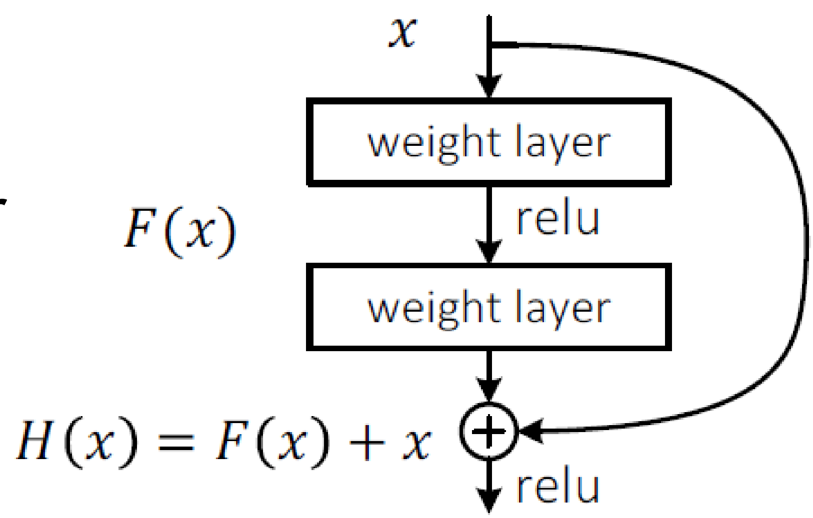
Recap: Residual Networks

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

- Core component
  - Skip connections bypassing each layer
  - Better propagation of gradients to the deeper layers
  - This makes it possible to train (much) deeper networks.



Transfer Learning with CNNs

1. Train on ImageNet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

I.e., swap the Softmax layer at the end

Transfer Learning with CNNs

1. Train on ImageNet

3. If you have medium sized dataset, "**finetune**" instead: use the old weights as initialization, train the full network or only some of the higher layers.

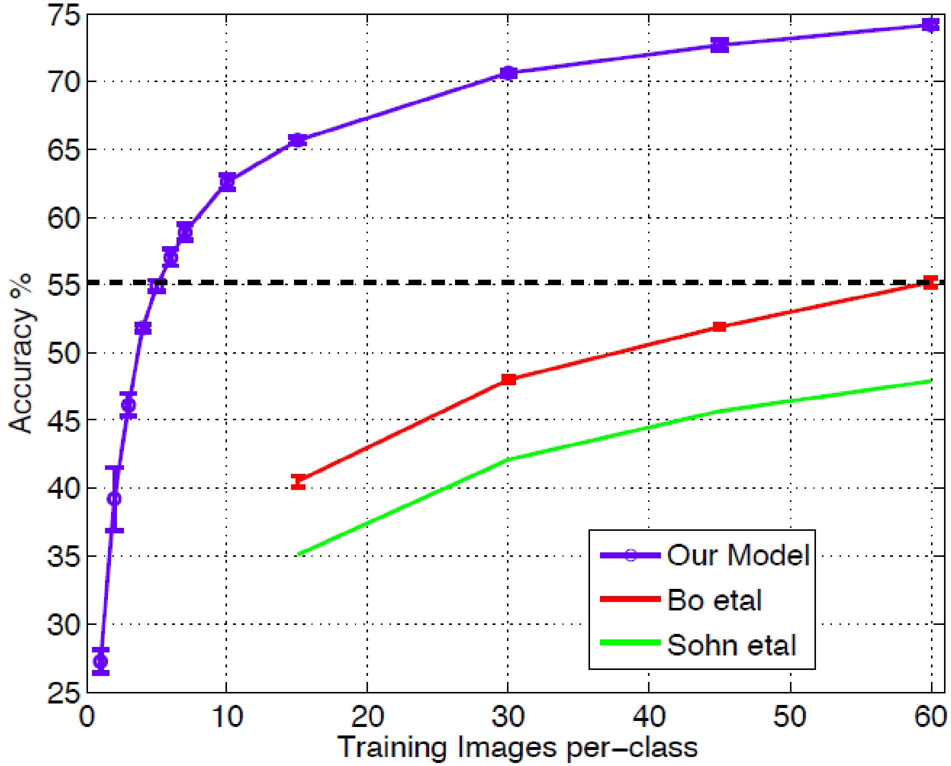
Retrain bigger portion of the network

Slide credit: Andrej Karpathy

Topics of This Lecture

- Object Detection with CNNs
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
- Semantic Image Segmentation
- Human Pose Estimation
- Face/Person Identification
  - DeepFace
  - FaceNet

The Learned Features are Generic



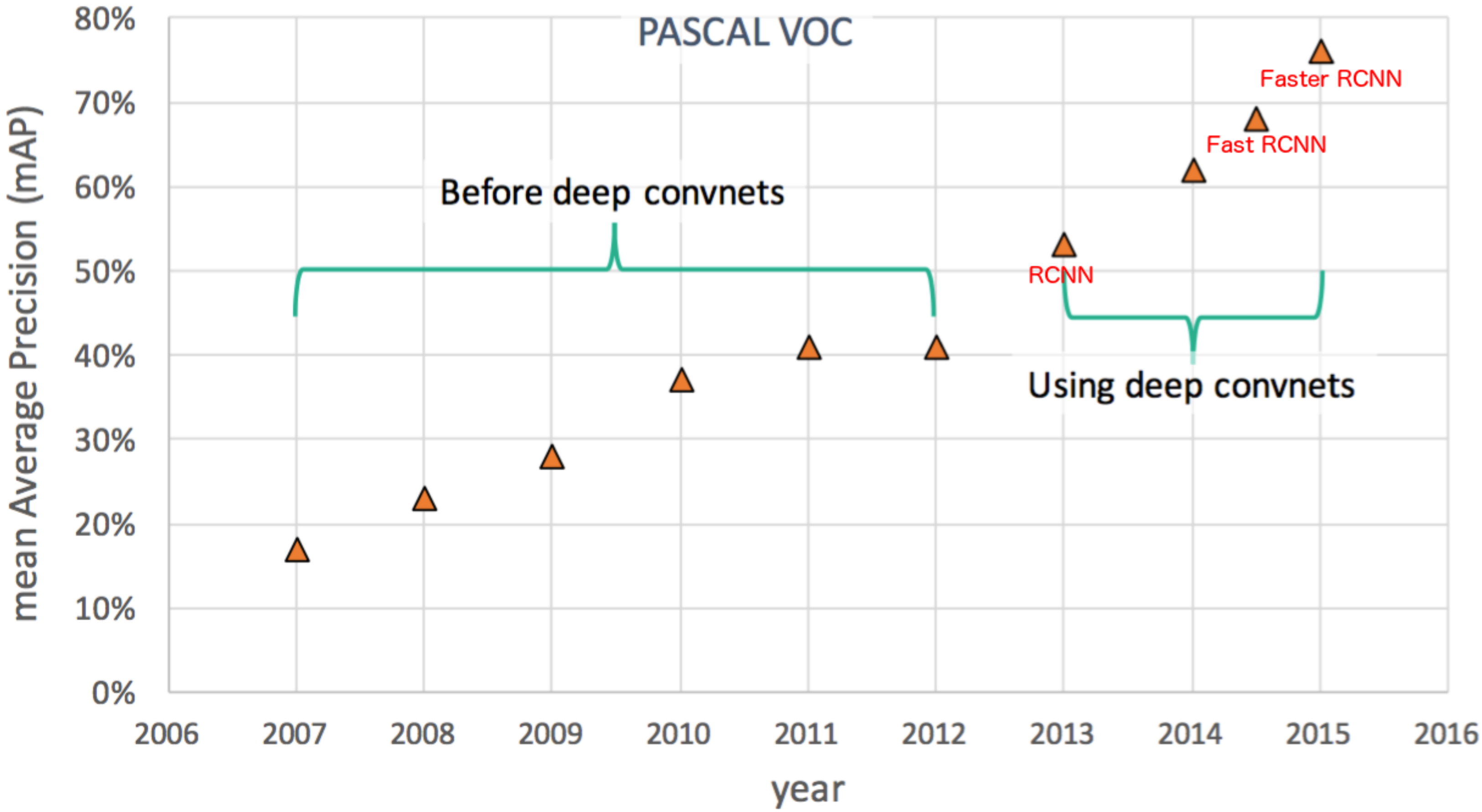
state of the art level (pre-CNN)

- **Experiment: feature transfer**
  - Train network on ImageNet
  - Chop off last layer and train classification layer on CalTech256

$\Rightarrow$ State of the art accuracy already with only 6 training images

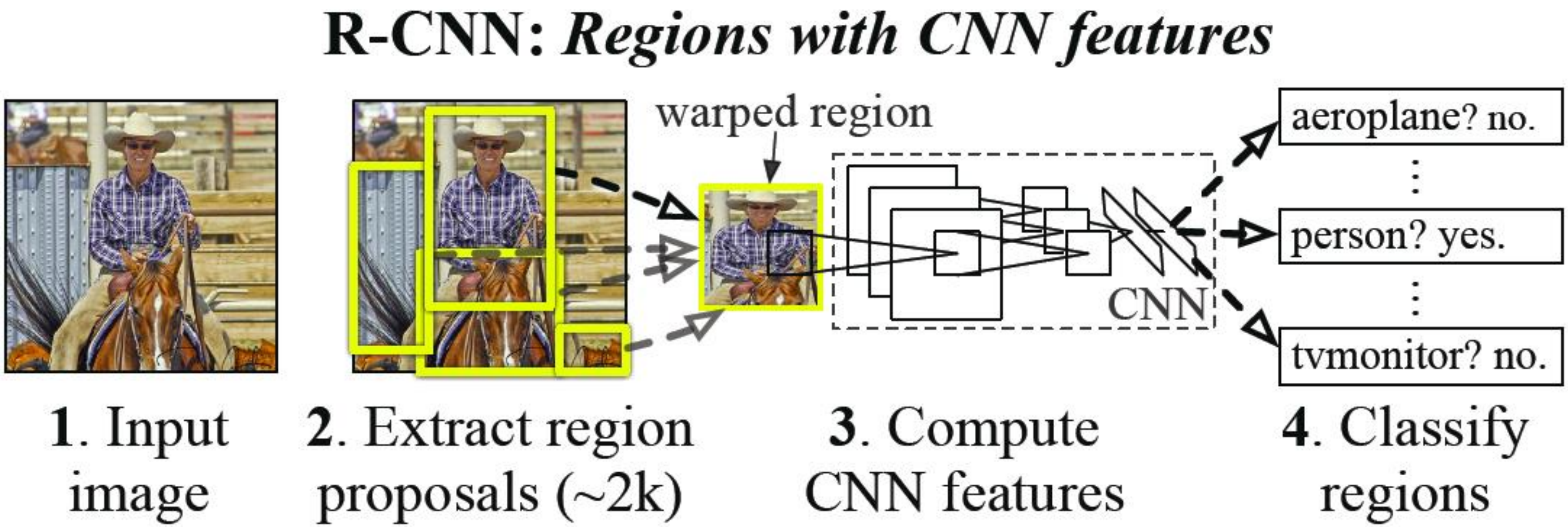
Image source: M. Zeiler, R. Fergus

Object Detection Performance



Slide credit: Ross Girshick

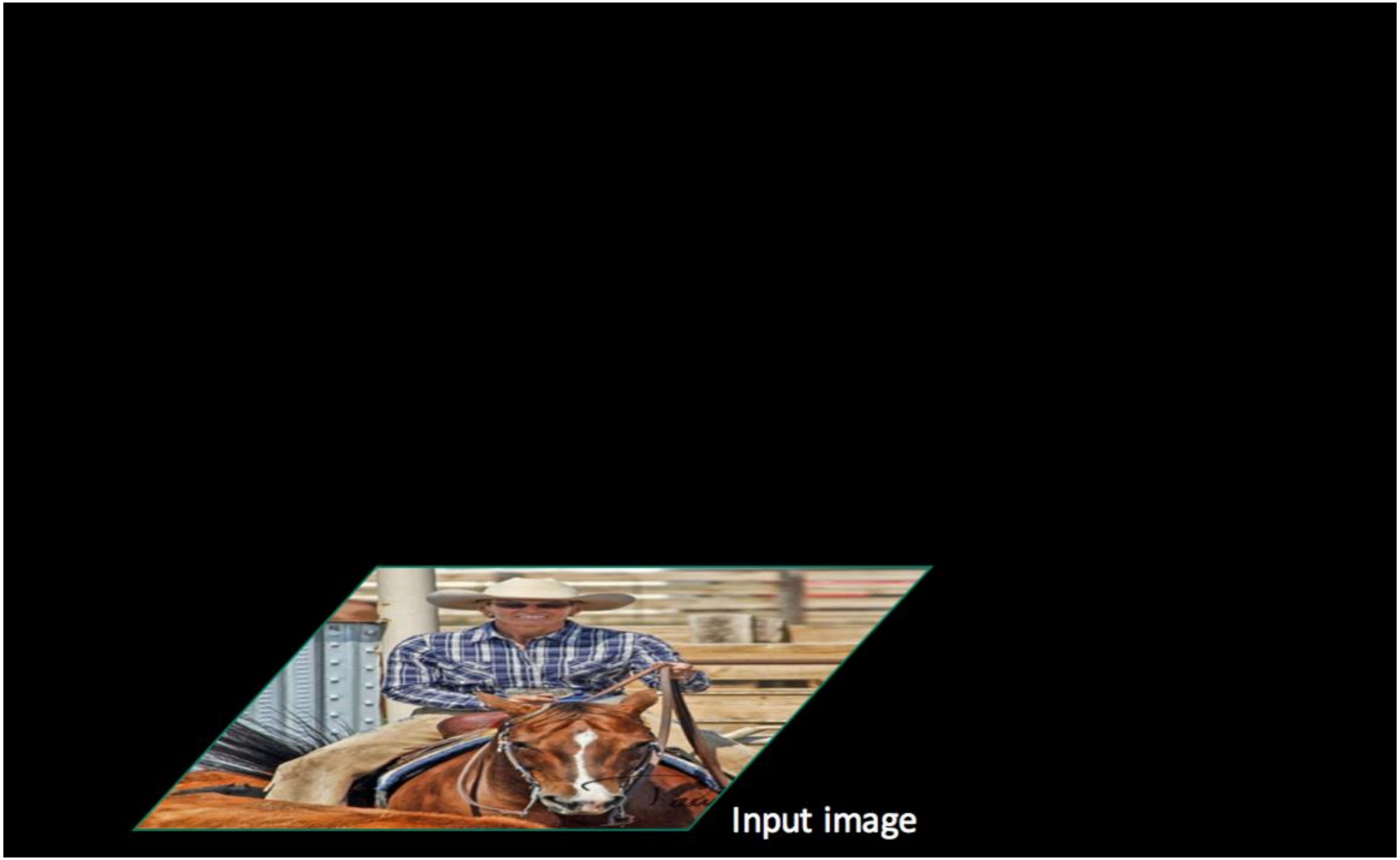
Object Detection: R-CNN



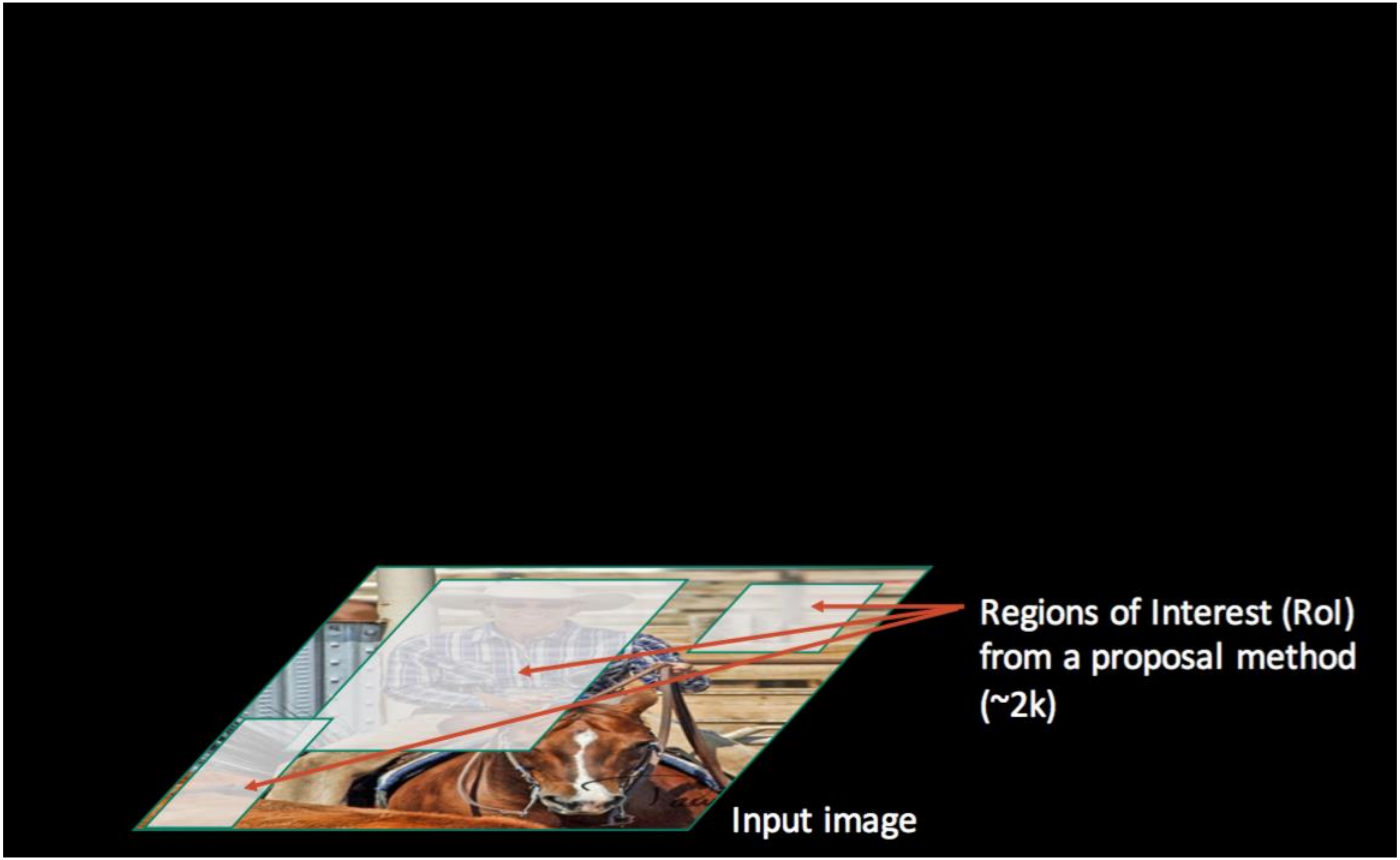
- **Key ideas**
  - **Extract region proposals (Selective Search)**
  - **Use a pre-trained/fine-tuned classification network as feature extractor (initially AlexNet, later VGGNet) on those regions**

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014

R-CNN Pipeline

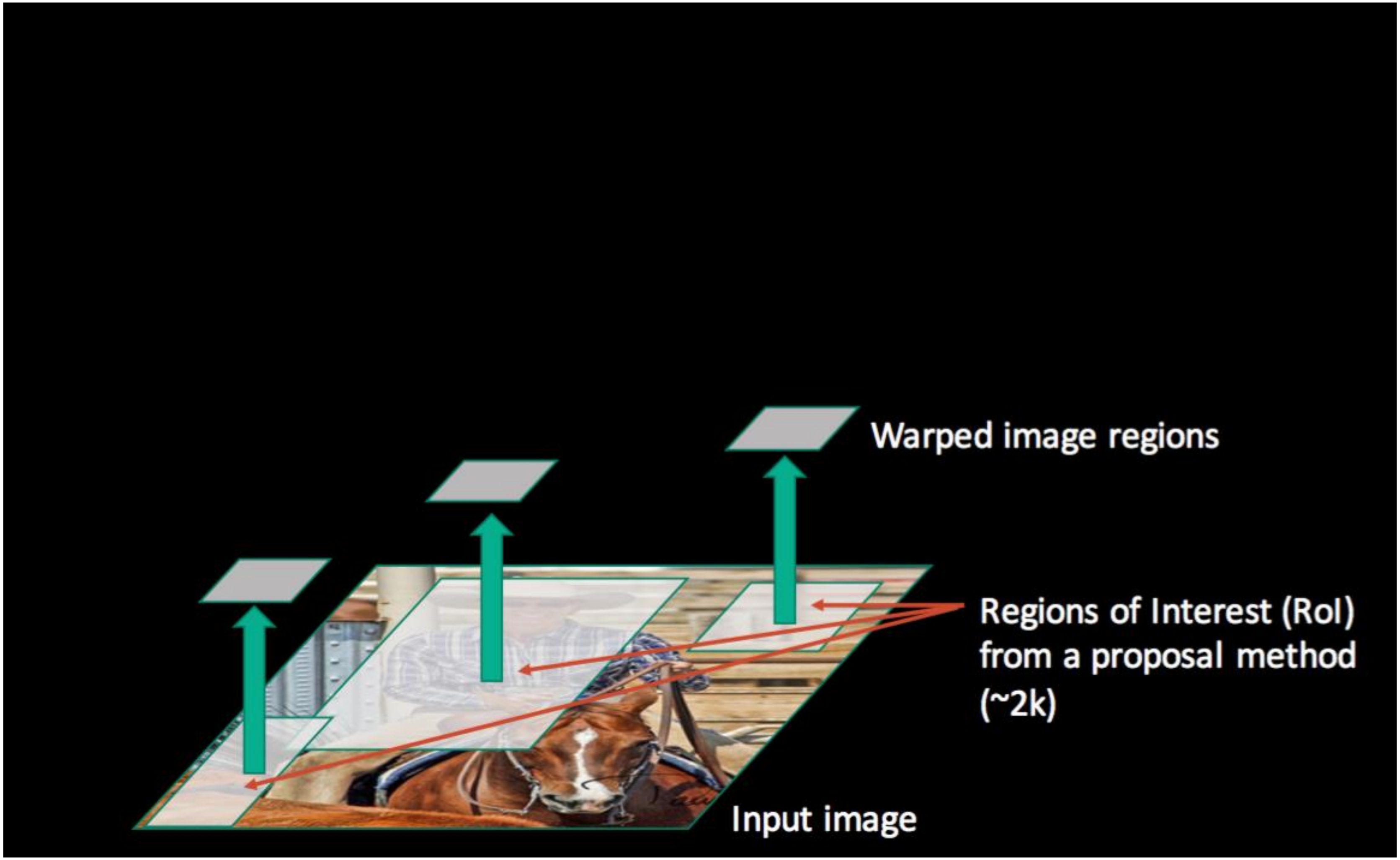


R-CNN Pipeline



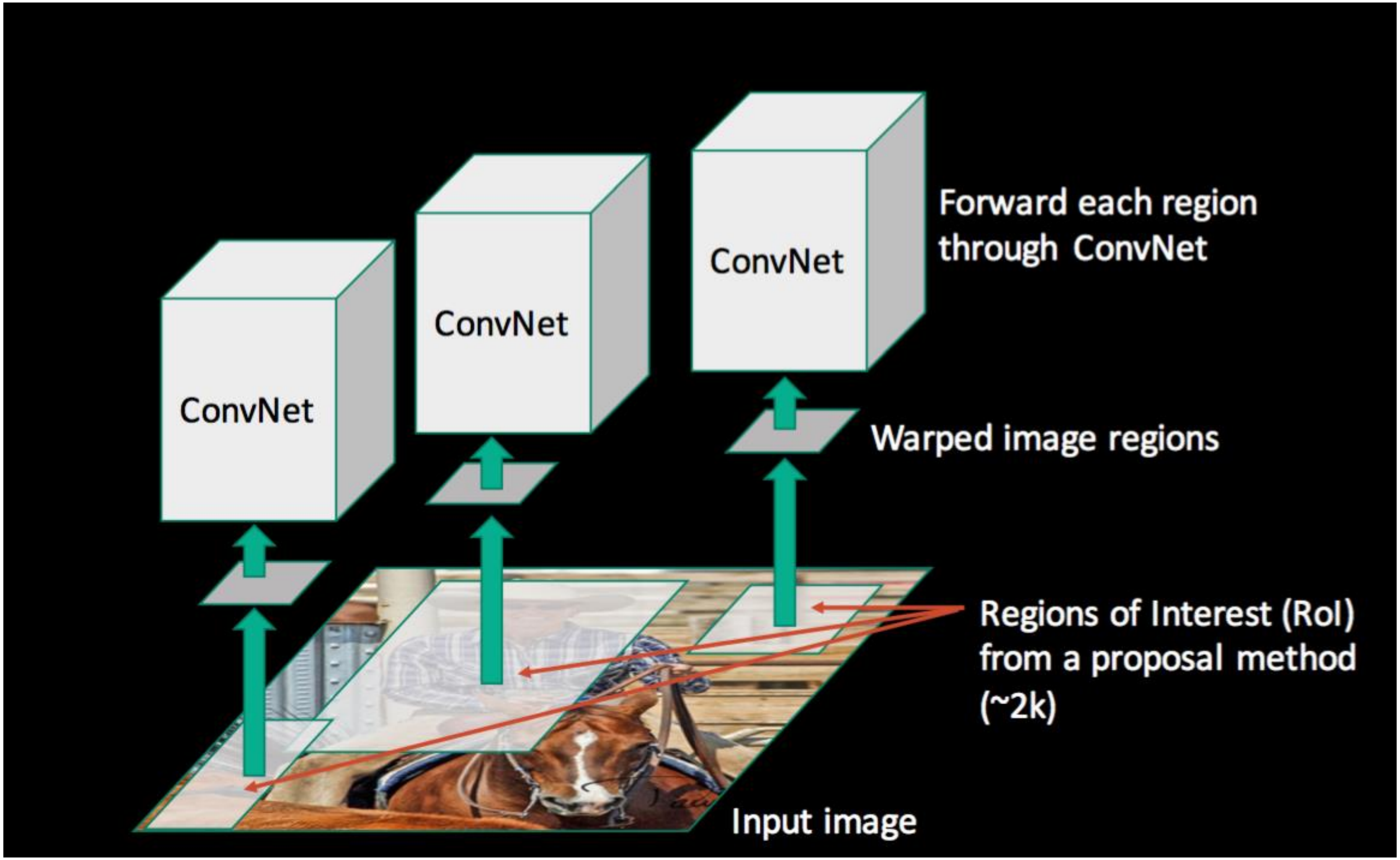
Slide credit: Ross Girshick

R-CNN Pipeline



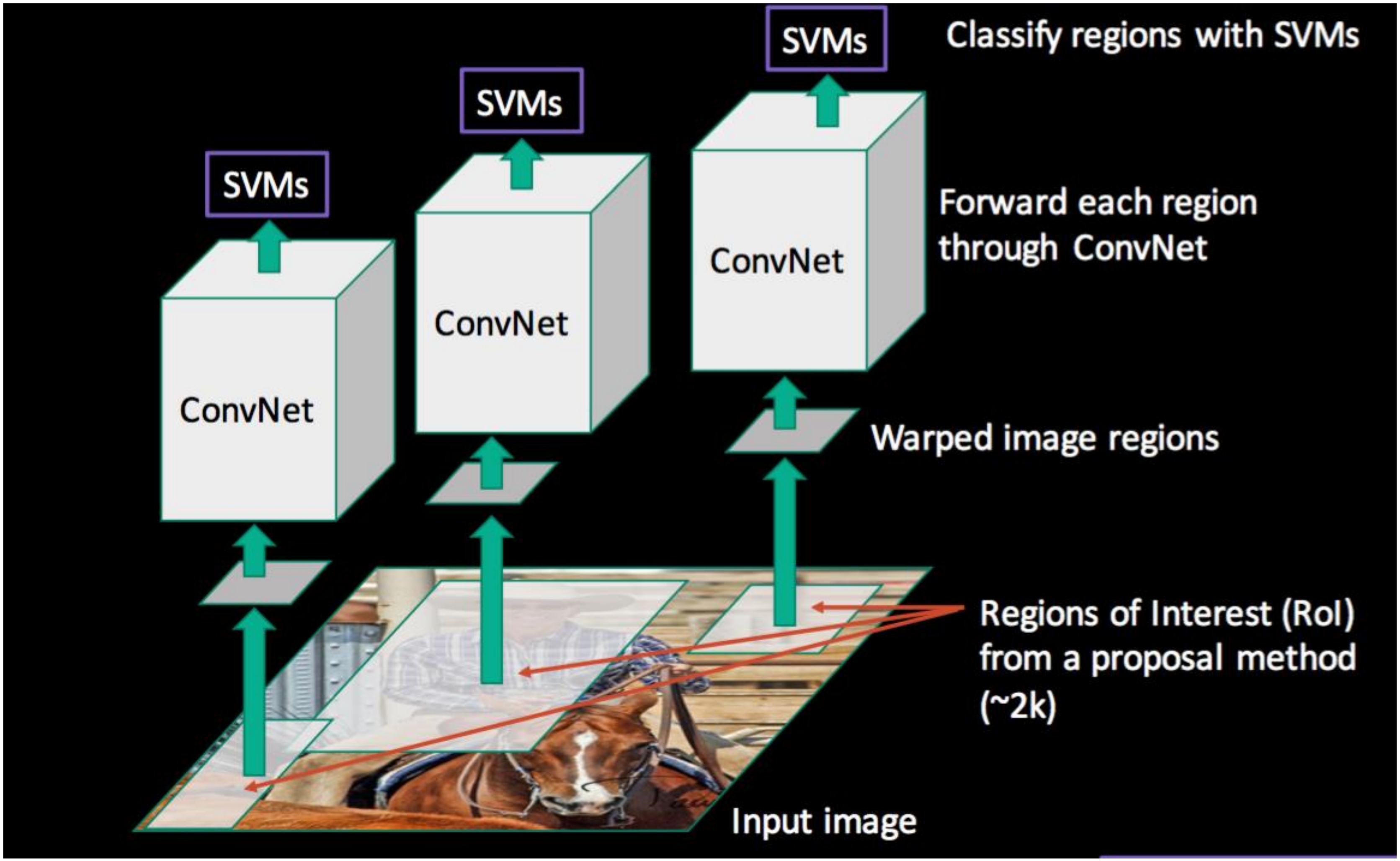
Slide credit: Ross Girshick

R-CNN Pipeline



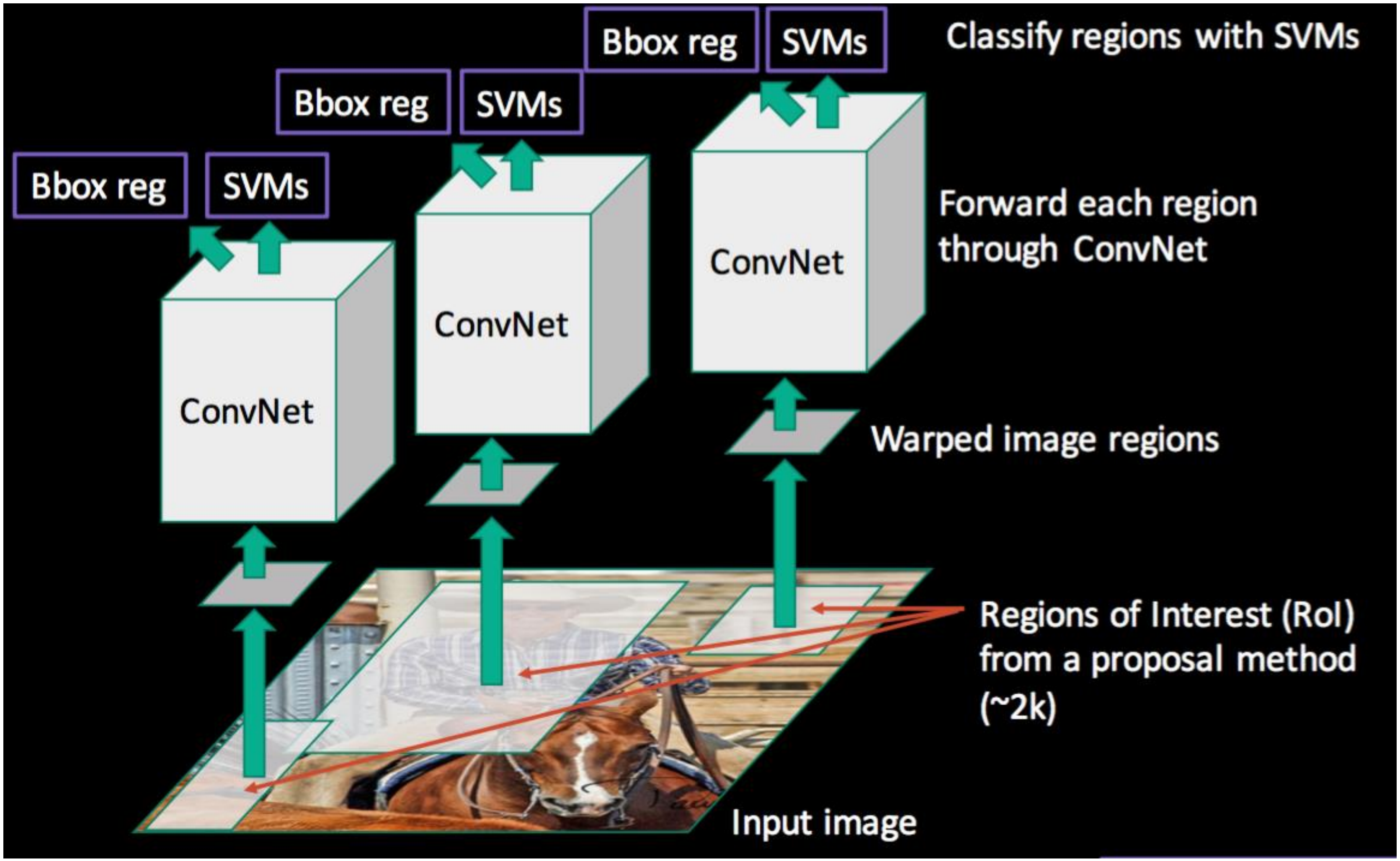
Slide credit: Ross Girshick

R-CNN Pipeline



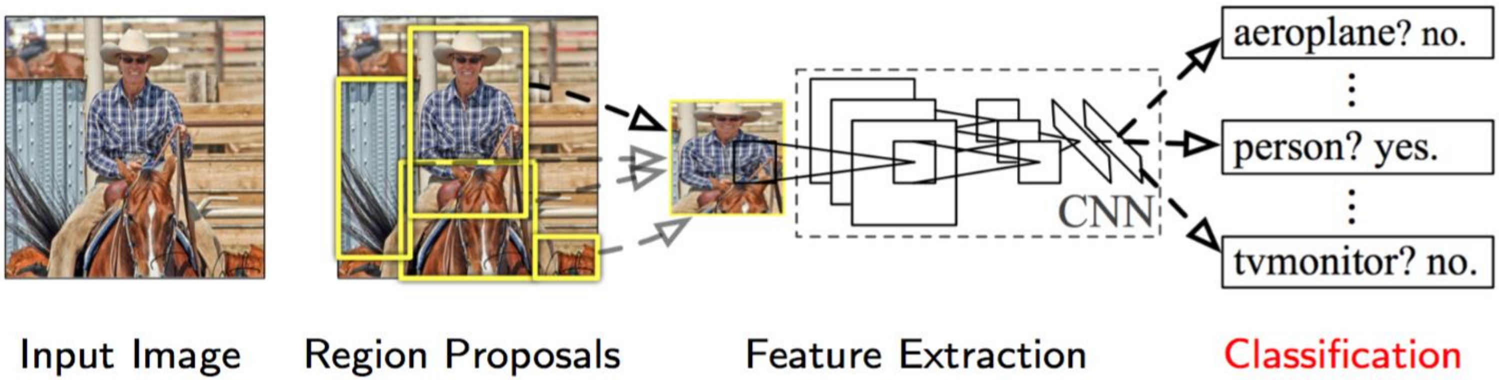
Slide credit: Ross Girshick

R-CNN Pipeline



Slide credit: Ross Girshick

Classification



- **Linear model with class-dependent weights**
  - **Linear SVM**

$$f_c(x_{fc7}) = w_c^T x_{fc7}$$

  - **where**
    - x_{fc7} = **features from the network (fully-connected layer 7)**
    - c = **object class**

Bounding Box Regressors

- **Prediction of the 2D box**
  - **Necessary, since the proposal region might not fully coincide with the (annotated) object bounding box**
  - **Perform regression for location (x^*,y^*), width w^* and height h^***

$$\frac{x^* - x}{w} = w_{c,x}^T x_{pool5}$$

$$\frac{y^* - y}{h} = w_{c,y}^T x_{pool5}$$

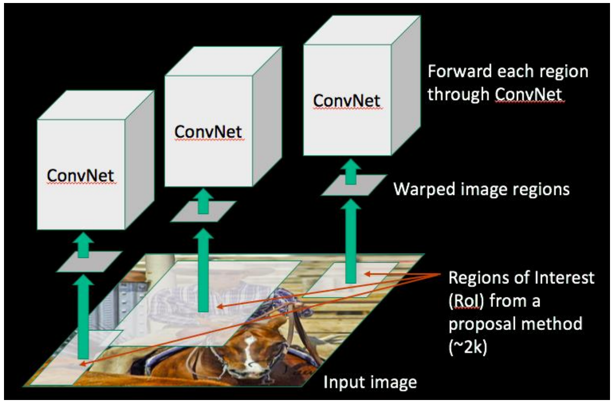
$$\ln \frac{w^*}{w} = w_{c,w}^T x_{pool5}$$

$$\ln \frac{h^*}{h} = w_{c,w}^T x_{pool5}$$

  - **Where x_{pool5} are the features from the pool5 layer of the network.**

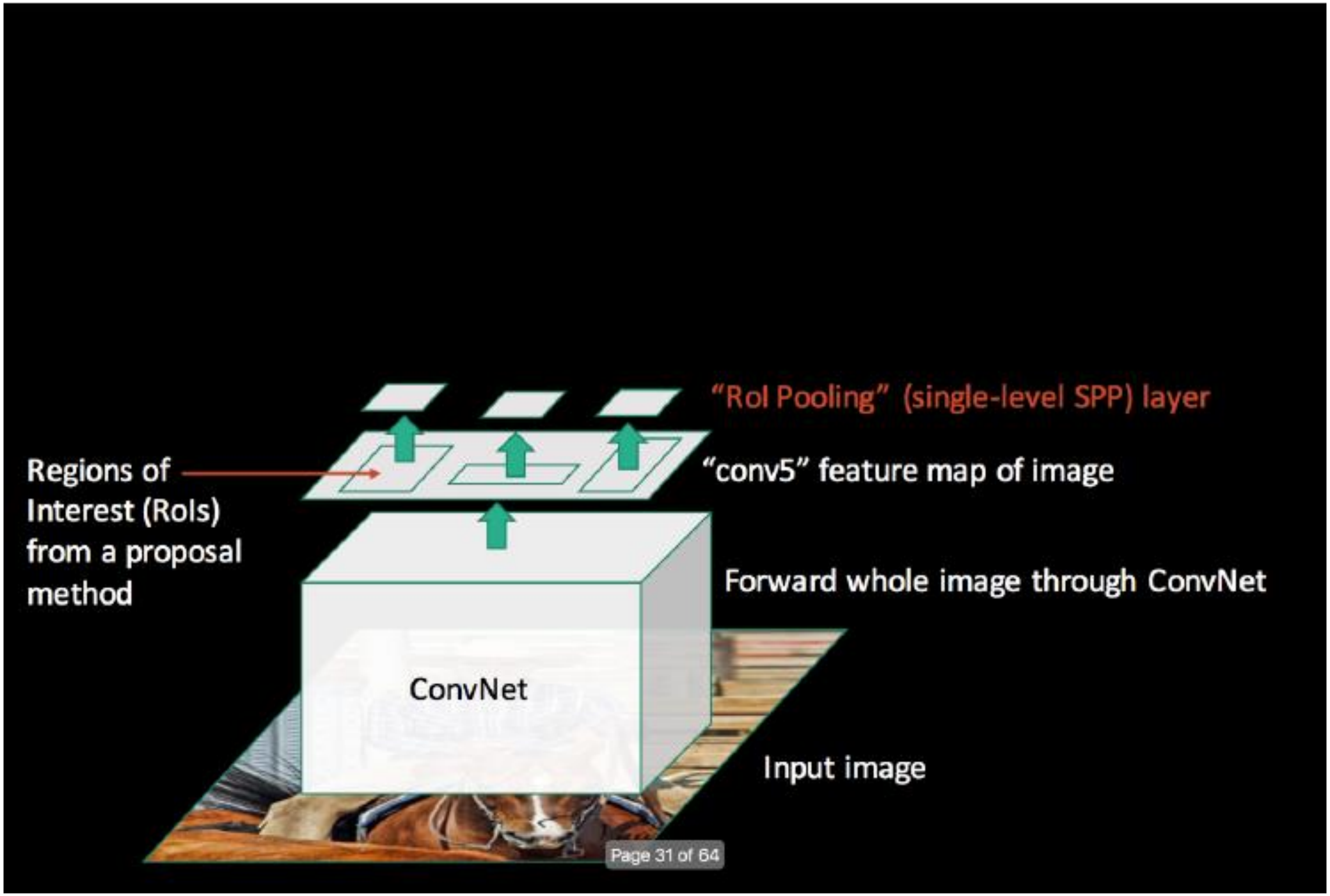
Problems with R-CNN

- Ad hoc training objectives
  - Fine tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressors (squared loss)
- Training (3 days) and testing (47s per image) is slow.
  - Many separate applications of region CNNs
- Takes a lot of disk space
  - Need to store all precomputed CNN features for training the classifiers
  - Easily 200GB of data



Fast R-CNN

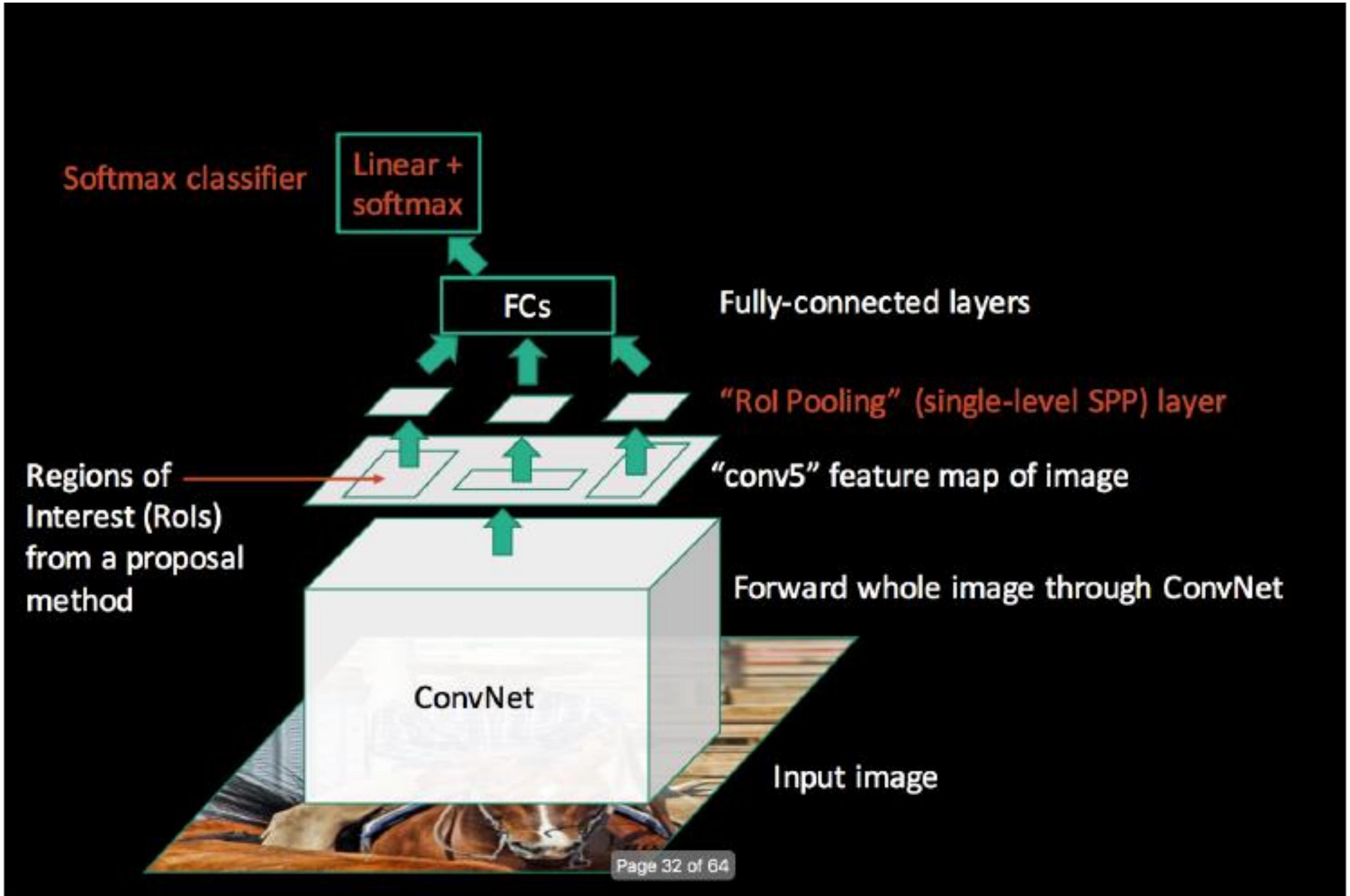
- **Forward Pass**



Slide credit: Ross Girshick

Fast R-CNN

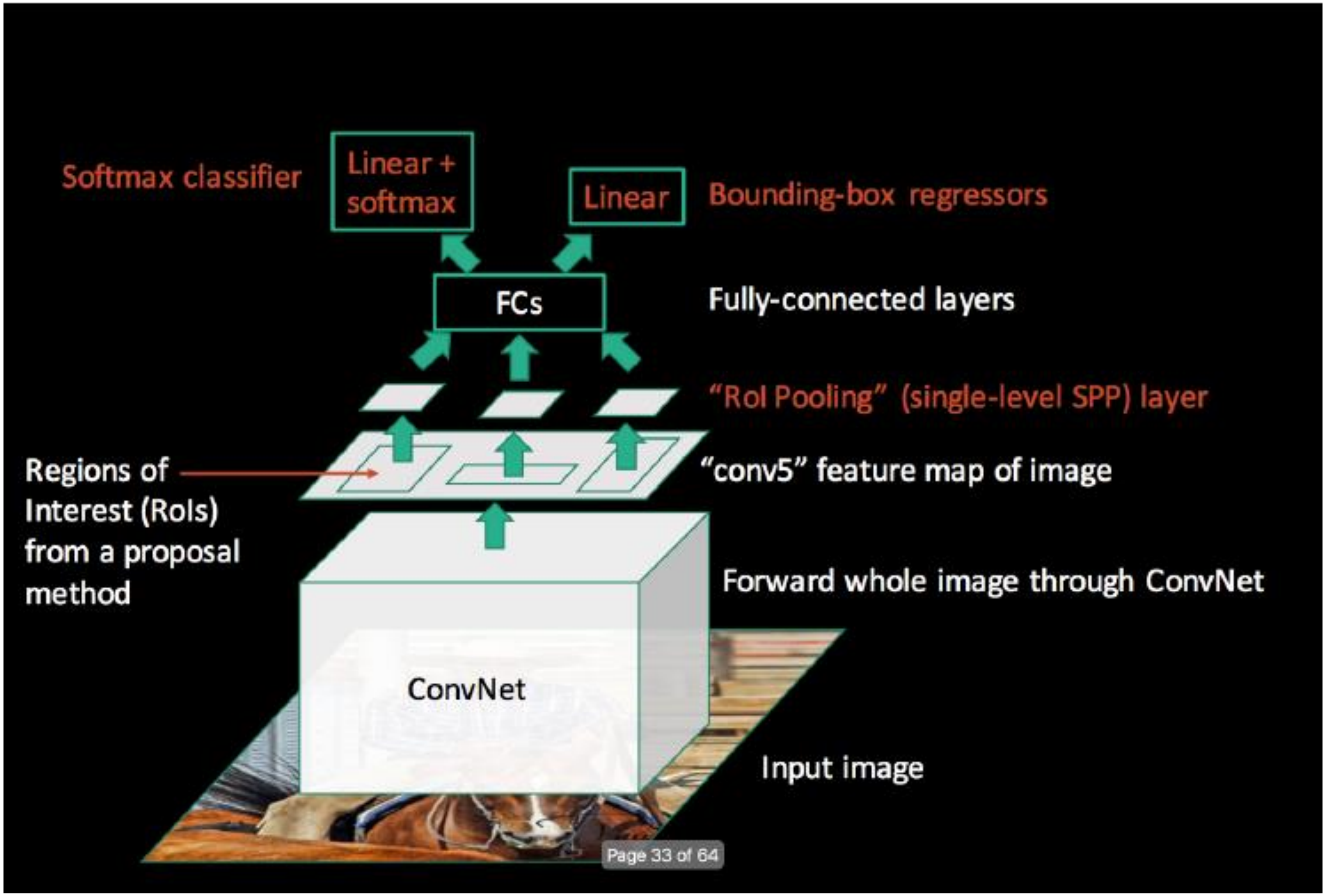
- Forward Pass



Slide credit: Ross Girshick

Fast R-CNN

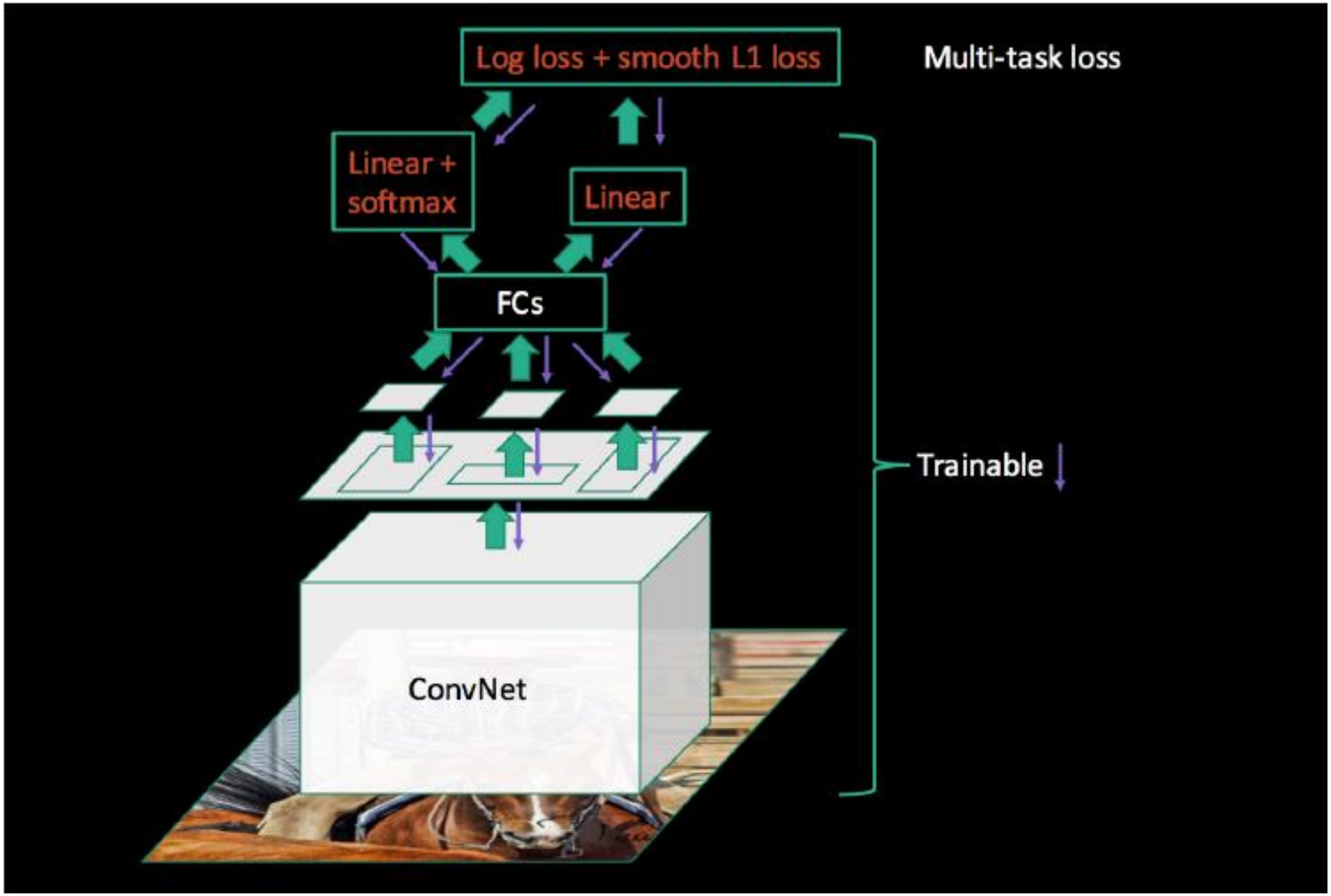
- Forward Pass



RWTH AACHEN UNIVERSITY

Fast R-CNN Training

- **Backward Pass**



Slide credit: Ross Girshick

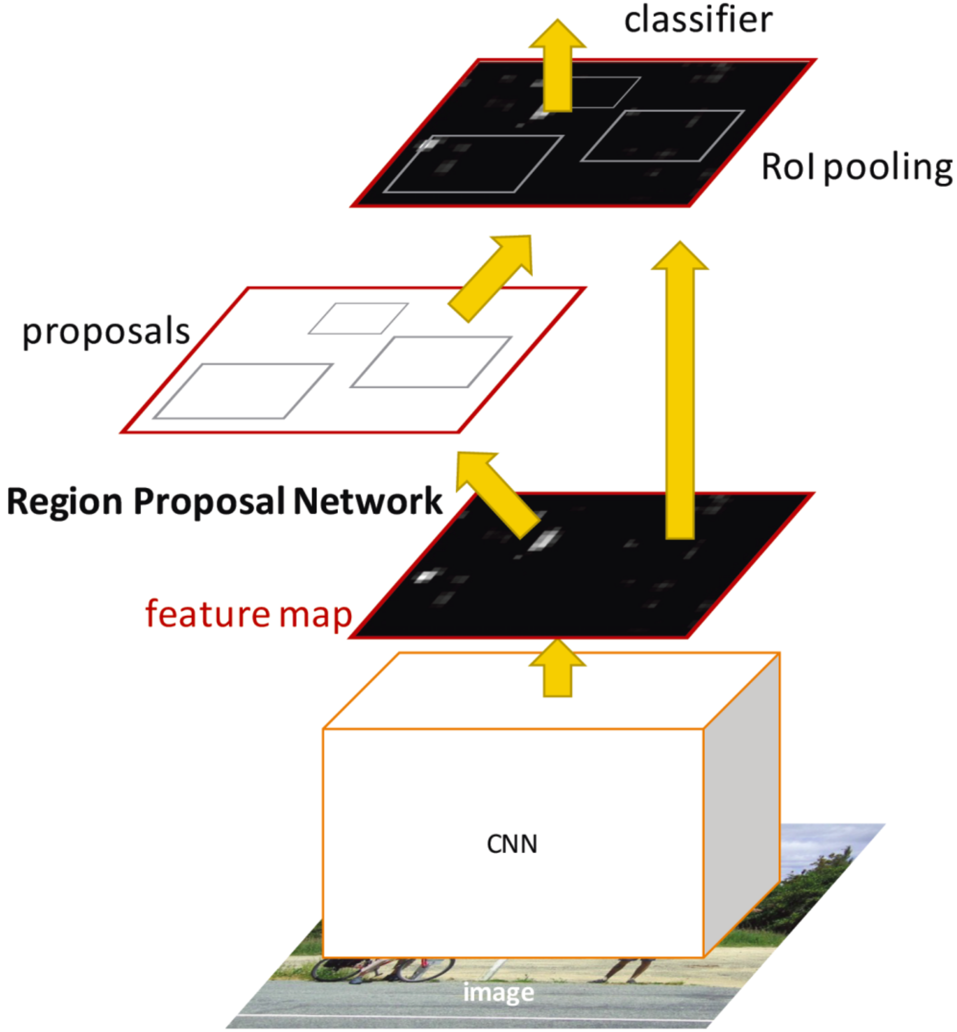
Region Proposal Networks (RPN)

- **Idea**
  - Remove dependence on external region proposal algorithm.
  - Instead, infer region proposals from same CNN.

  ⇒ Feature sharing

  ⇒ Object detection in a single pass becomes possible.

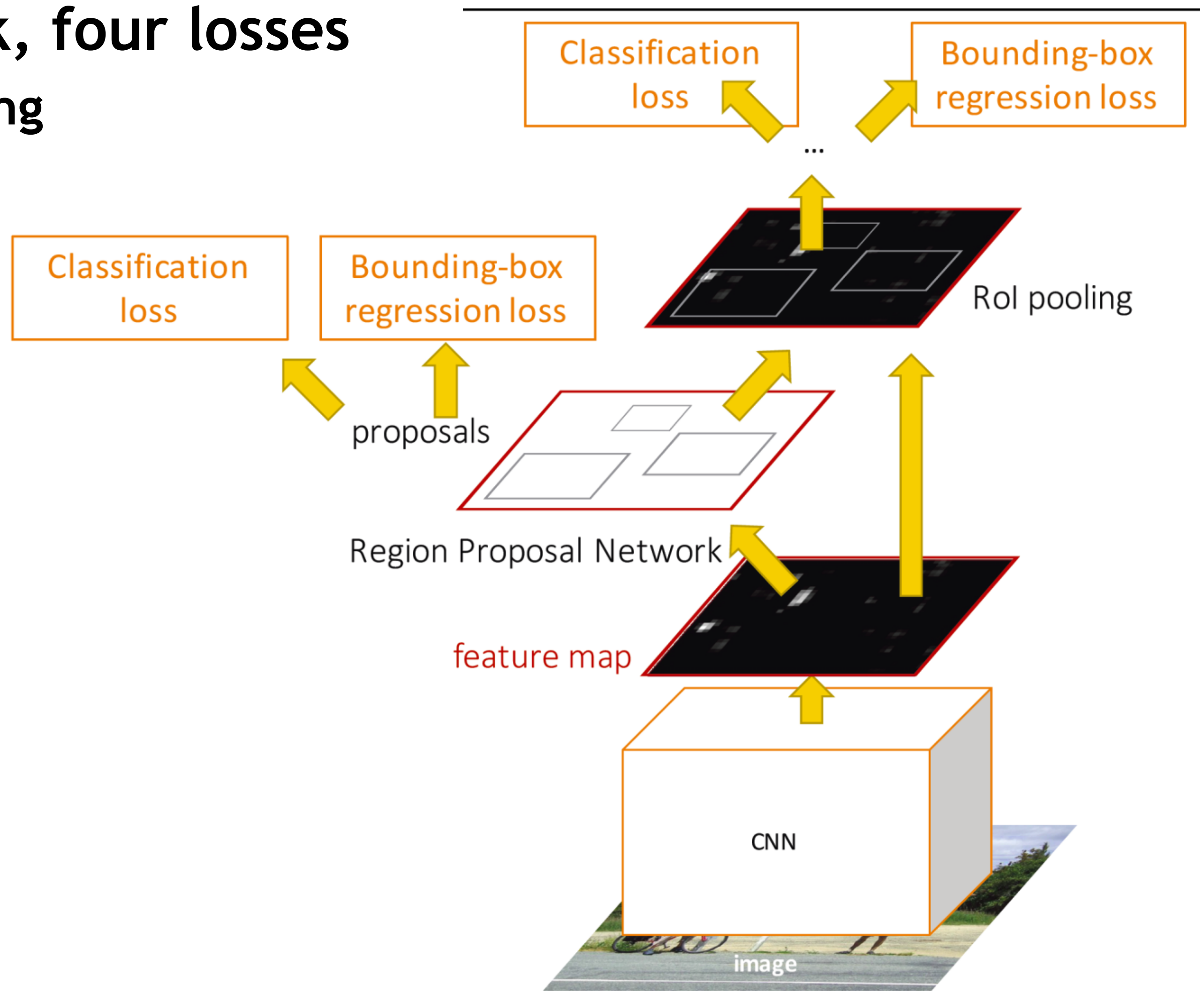
- **Faster R-CNN = Fast R-CNN + RPN**



Slide credit: Ross Girshick

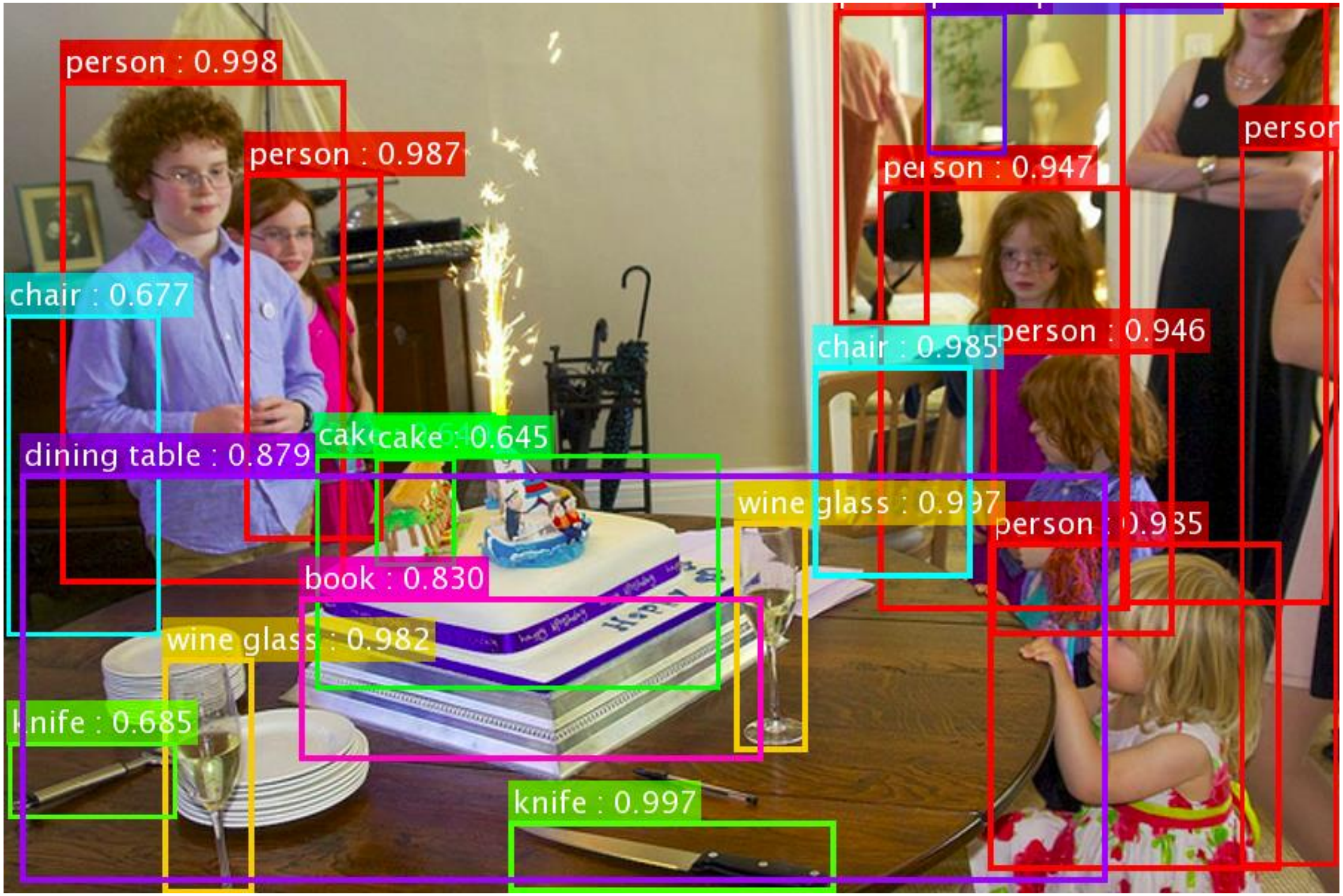
Faster R-CNN

- **One network, four losses**
  - **Joint training**



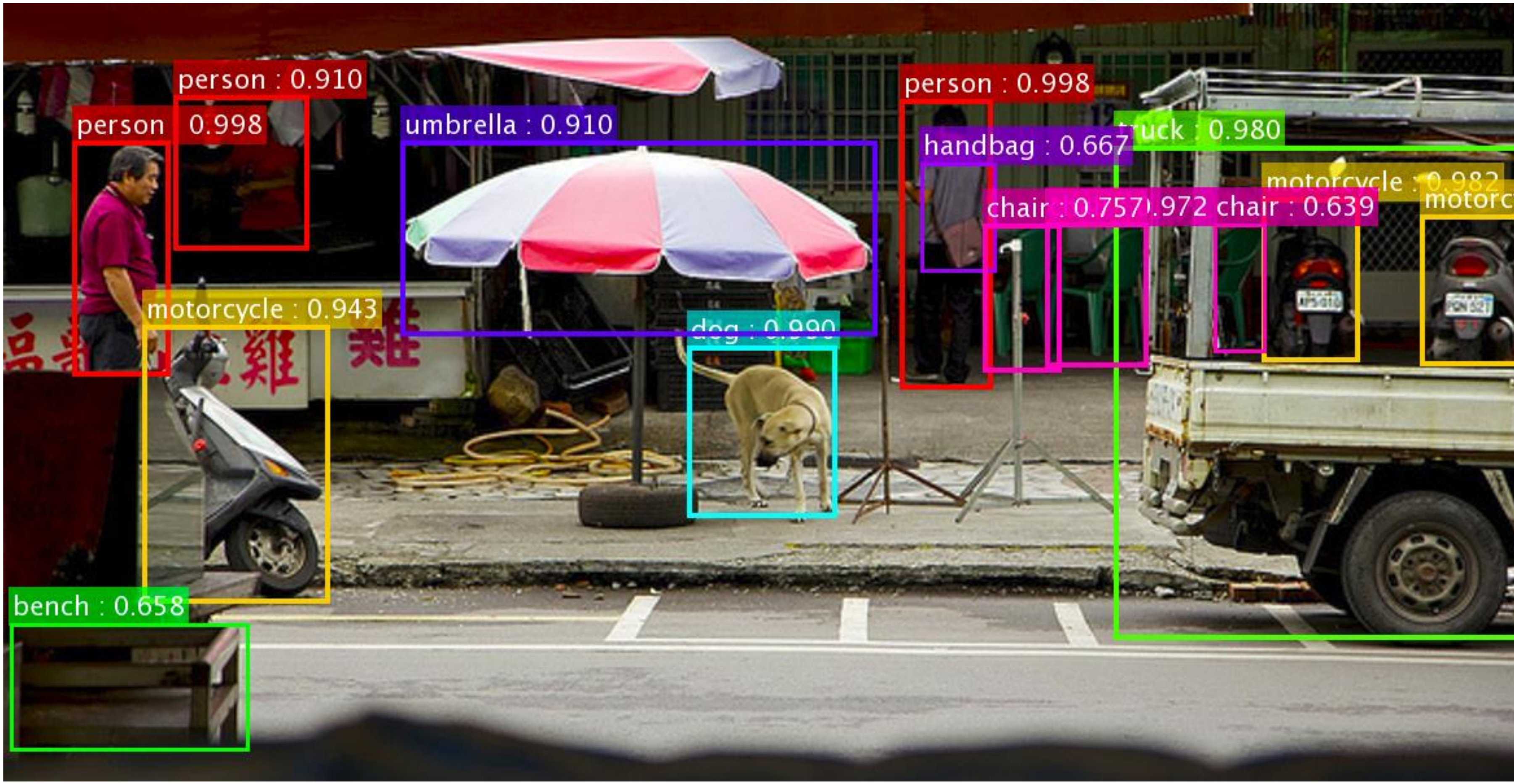
Slide credit: Ross Girshick

Faster R-CNN (based on ResNets)



K. He, X. Zhang, S. Ren, J. Sun, [Deep Residual Learning for Image Recognition](), CVPR 2016.

Faster R-CNN (based on ResNets)



K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, CVPR 2016.

Summary

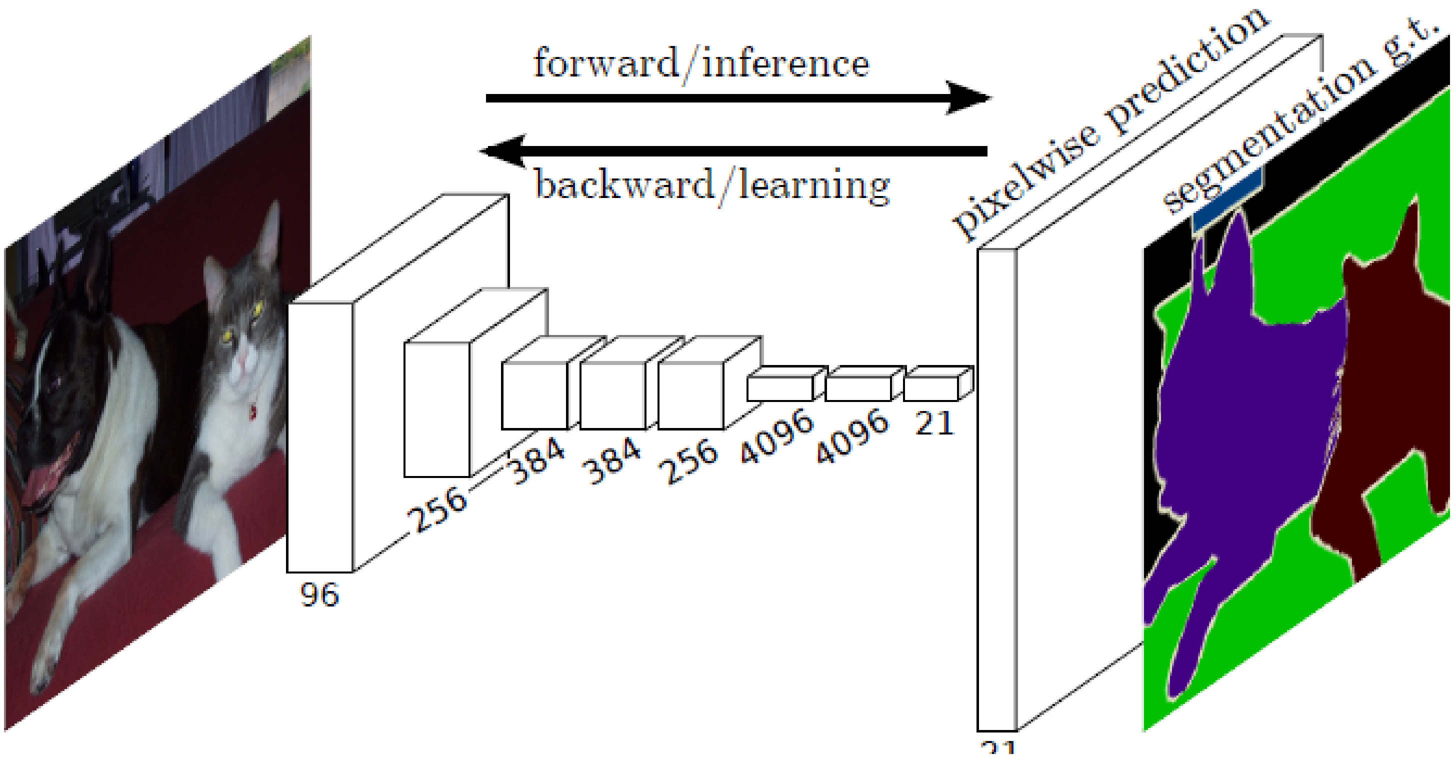
- **Object Detection**
  - Find a variable number of objects by classifying image regions
  - Before CNNs: dense multiscale sliding window (HoG, DPM)
  - Avoid dense sliding window with region proposals
  - R-CNN: Selective Search + CNN classification / regression
  - Fast R-CNN: Swap order of convolutions and region extraction
  - Faster R-CNN: Compute region proposals within the network
  - Deeper networks do better

Topics of This Lecture

- Object Detection with CNNs
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
- **Semantic Image Segmentation**
- Human Pose Estimation
- Face/Person Identification
  - DeepFace
  - FaceNet

RWTH AACHEN UNIVERSITY

Semantic Image Segmentation

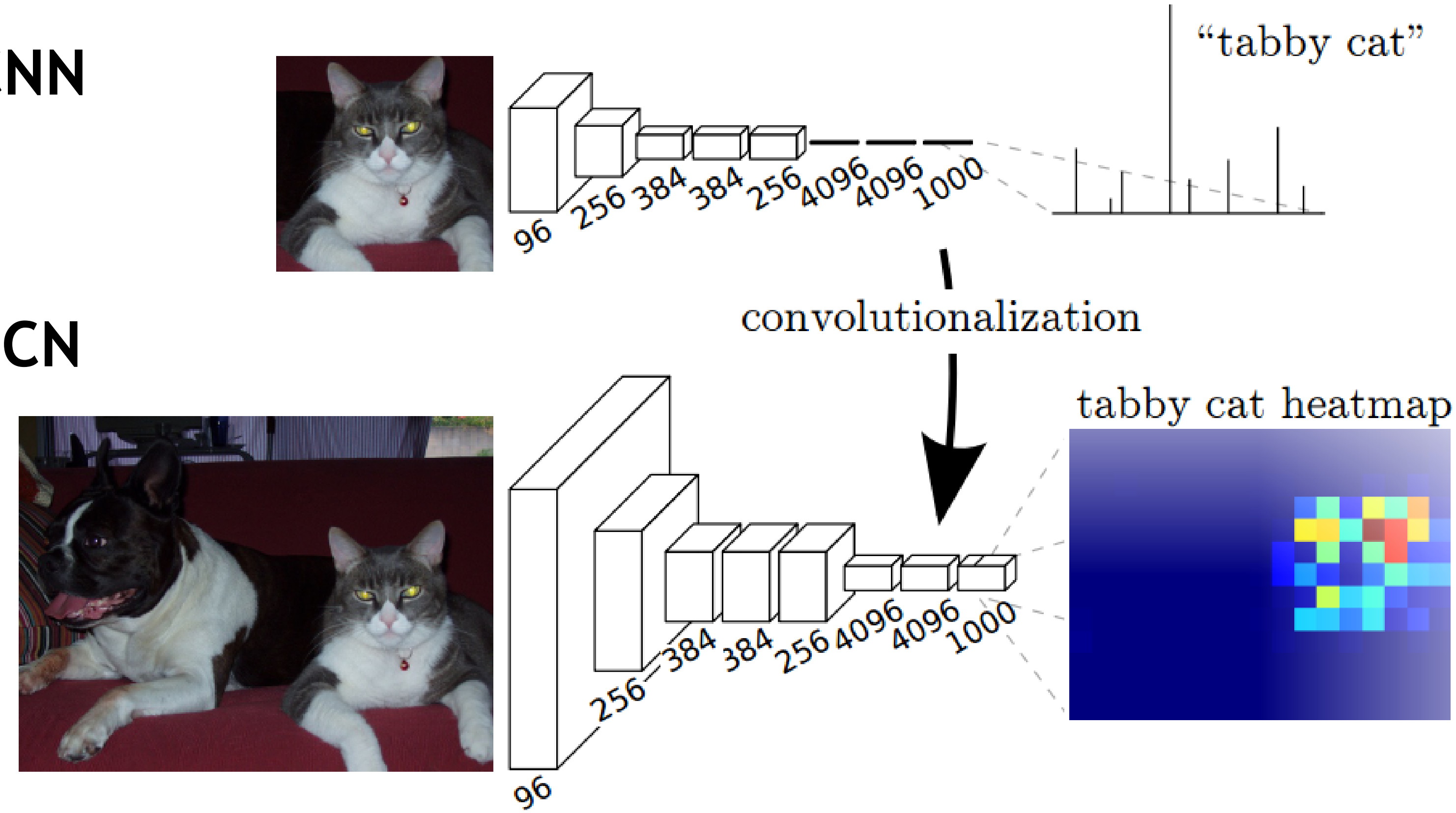


- Perform pixel-wise prediction task
  - Usually done using Fully Convolutional Networks (FCNs)
    - All operations formulated as convolutions
    - Advantage: can process arbitrarily sized images

Image source: Long, Shelhamer, Darrell

CNNs vs. FCNs

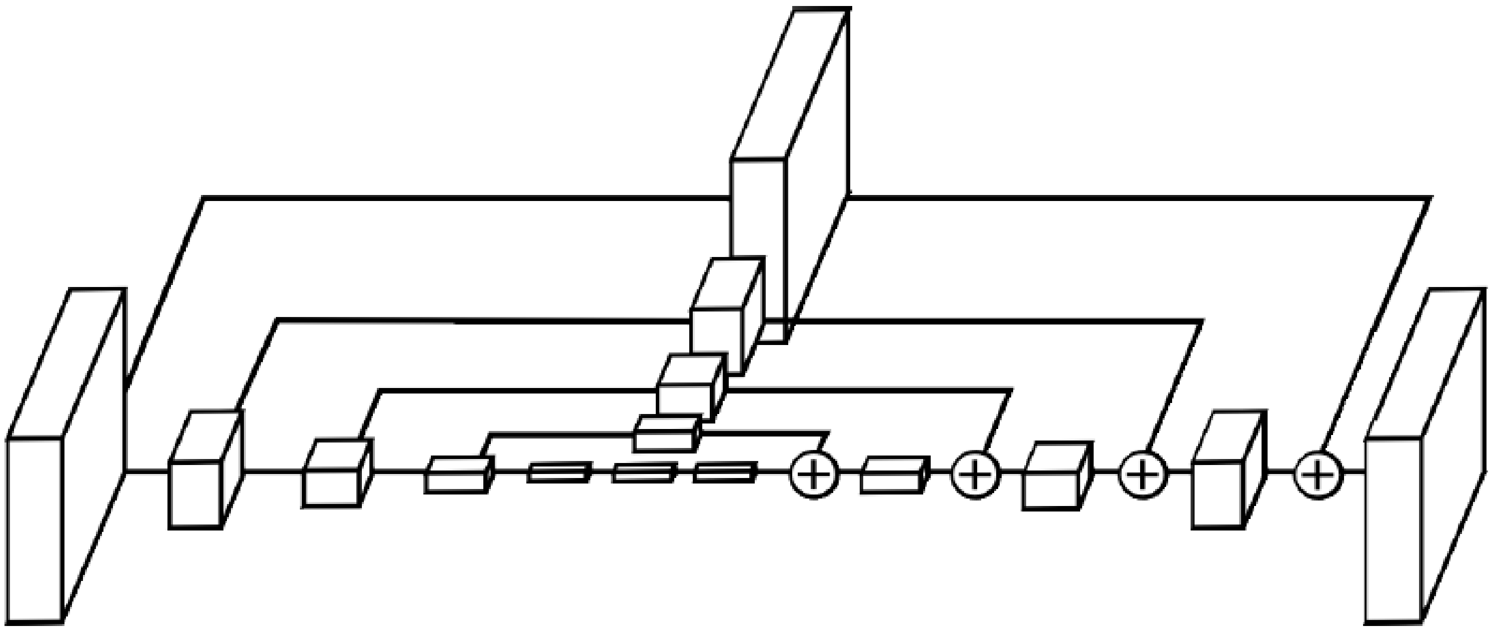
- CNN
- FCN



- Intuition
  - Think of FCNs as performing a sliding-window classification, producing a heatmap of output scores for each class

Image source: Long, Shelhamer, Darrell

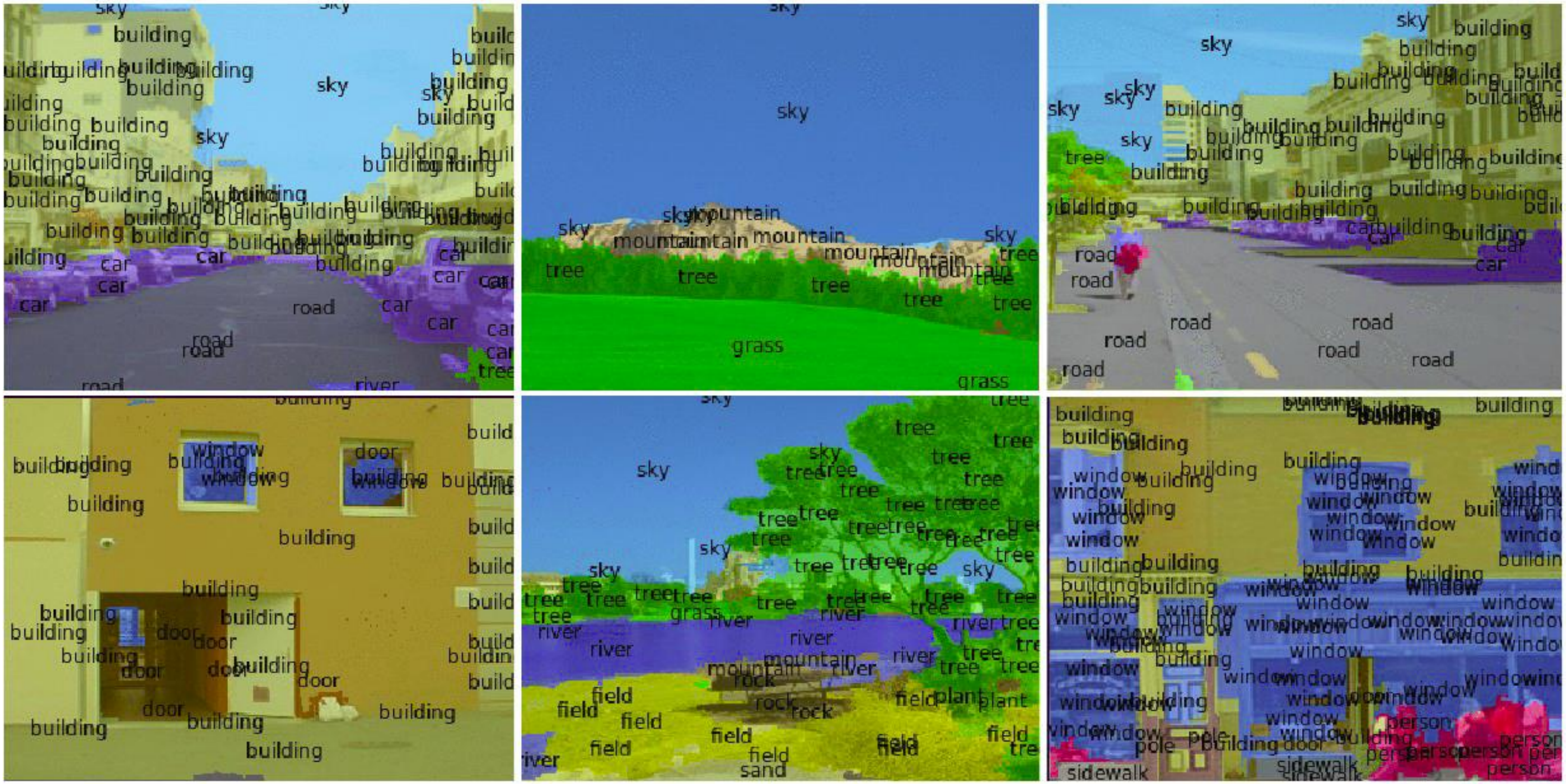
Semantic Image Segmentation



- **Encoder-Decoder Architecture**
  - Problem: FCN output has low resolution
  - Solution: perform upsampling to get back to desired resolution
  - Use skip connections to preserve higher-resolution information

Image source: Newell et al.

Other Tasks: Semantic Segmentation



[Farabet et al. ICML 2012, PAMI 2013]

Semantic Segmentation

[Pohlen, Hermans, Mathias, Leibe, arXiv 2016]

- **More recent results**
  - **Based on an extension of ResNets**

Topics of This Lecture

- Object Detection with CNNs
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
- Semantic Image Segmentation
- **Human Pose Estimation**
- Face/Person Identification
  - DeepFace
  - FaceNet

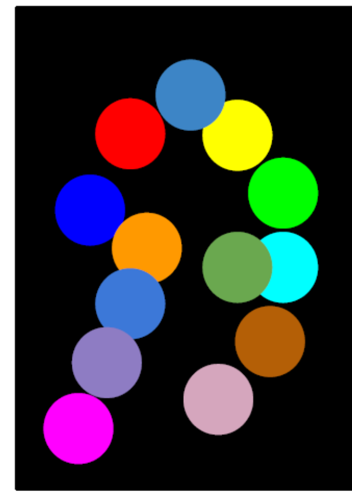
FCNs for Human Pose Estimation

- **Input data**

Image

Keypoints

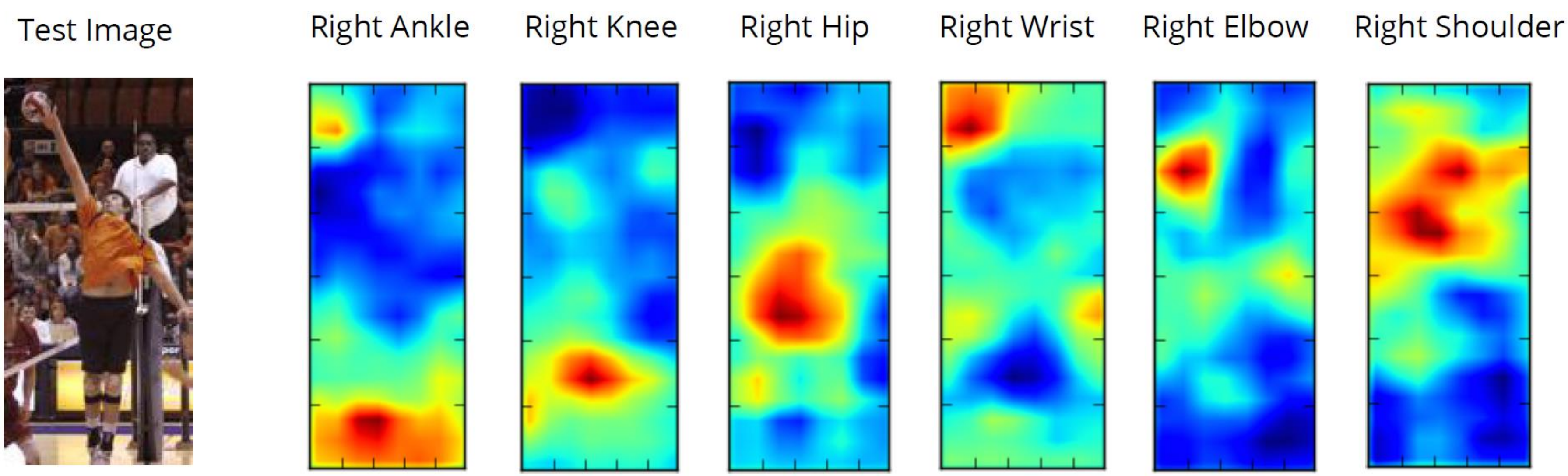
Labels



- **Task setup**
  - **Annotate images with keypoints for skeleton joints**
  - **Define a target disk around each keypoint with radius r**
  - **Set the ground-truth label to 1 within each such disk**
  - **Infer heatmaps for the joints as in semantic segmentation**

Slide adapted from Georgia Gkioxari

Heat Map Predictions from FCN



Slide adapted from Georgia Gkioxari

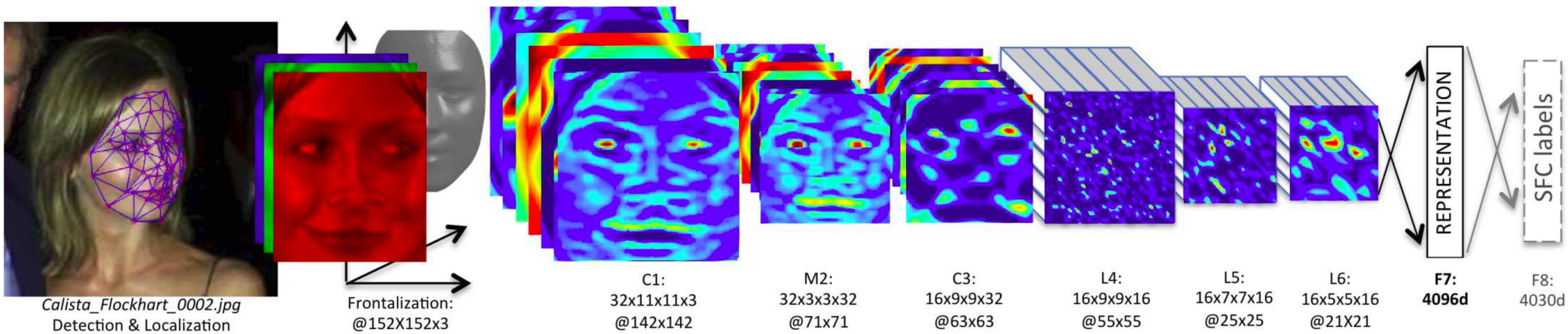
Example Results: Human Pose Estimation

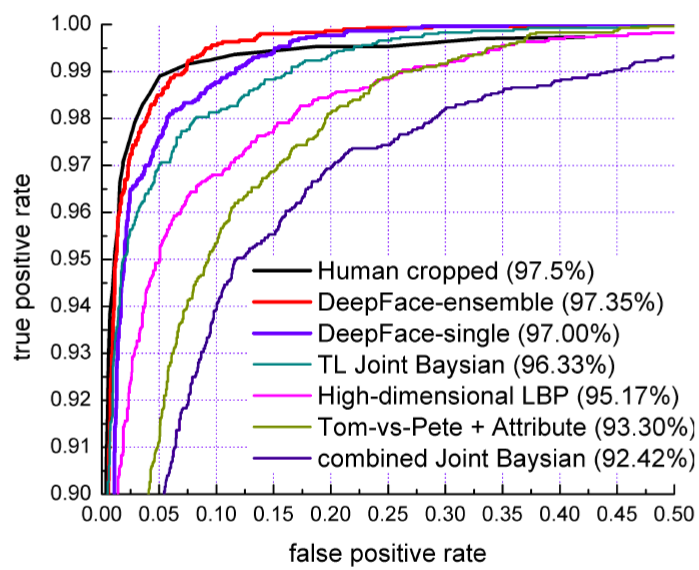
[Rafi, Gall, Leibe, BMVC 2016]

Topics of This Lecture

- Object Detection with CNNs
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
- Semantic Image Segmentation
- Human Pose Estimation
- **Face/Person Identification**
  - **DeepFace**
  - **FaceNet**

Other Tasks: Face Verification



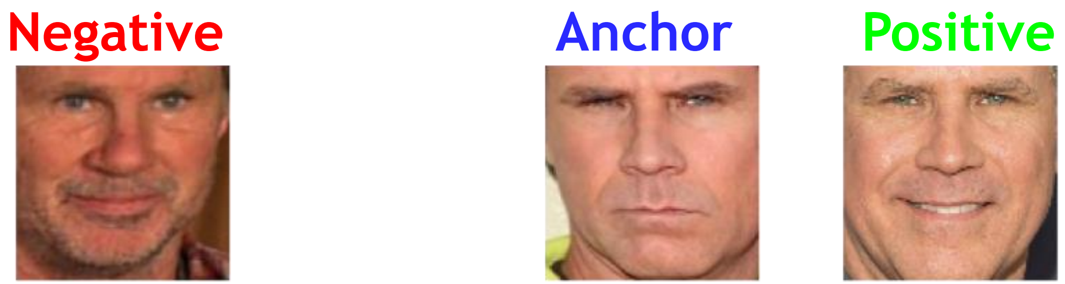


Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR 2014

Slide credit: Svetlana Lazebnik

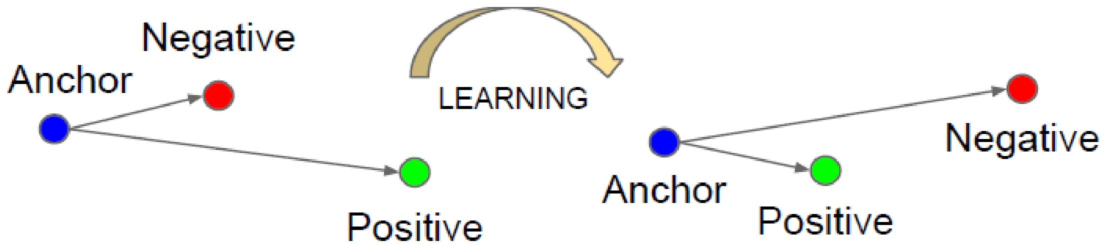
Discriminative Face Embeddings

- **Learning an embedding using a Triplet Loss Network**
  - **Present the network with triplets of examples**



  - **Apply triplet loss to learn an embedding $f(\cdot)$ that groups the positive example closer to the anchor than the negative one.**

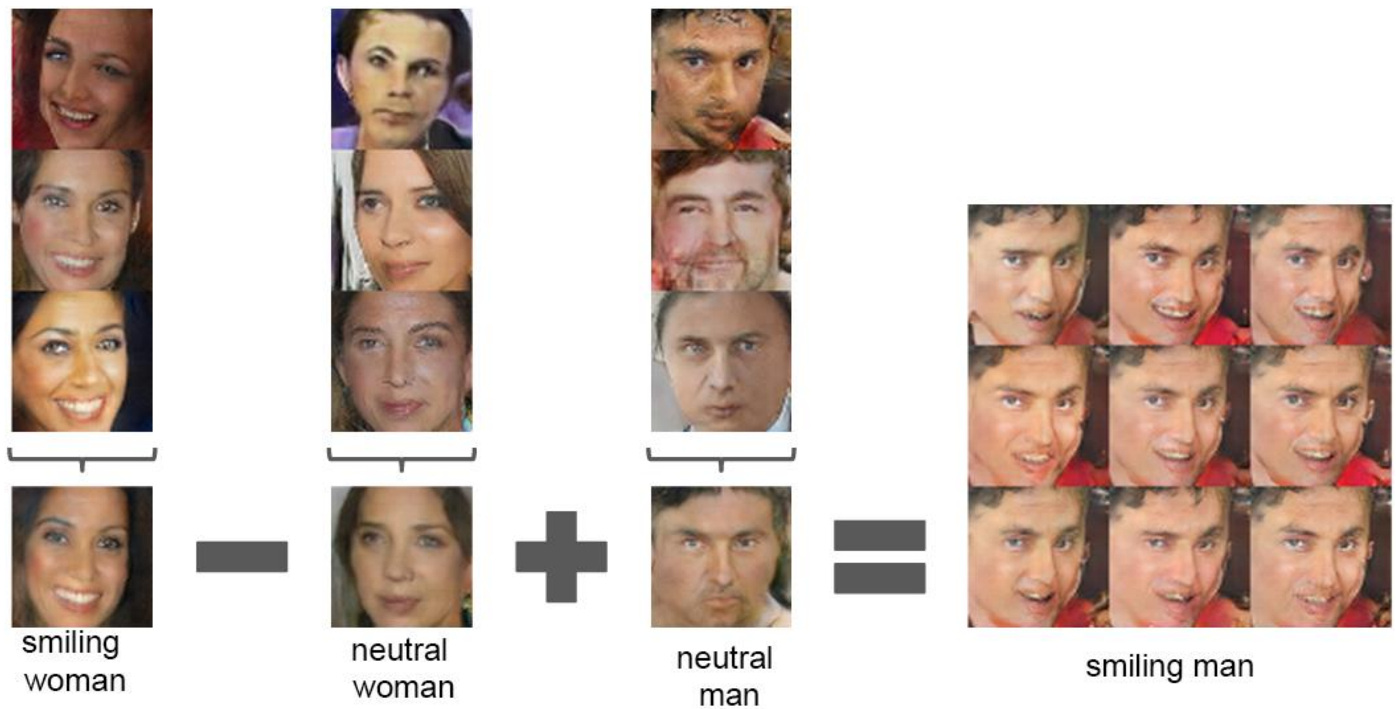
$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$$



$\Rightarrow$ **Used with great success in Google's FaceNet face recognition**

Vector Arithmetics in Embedding Space

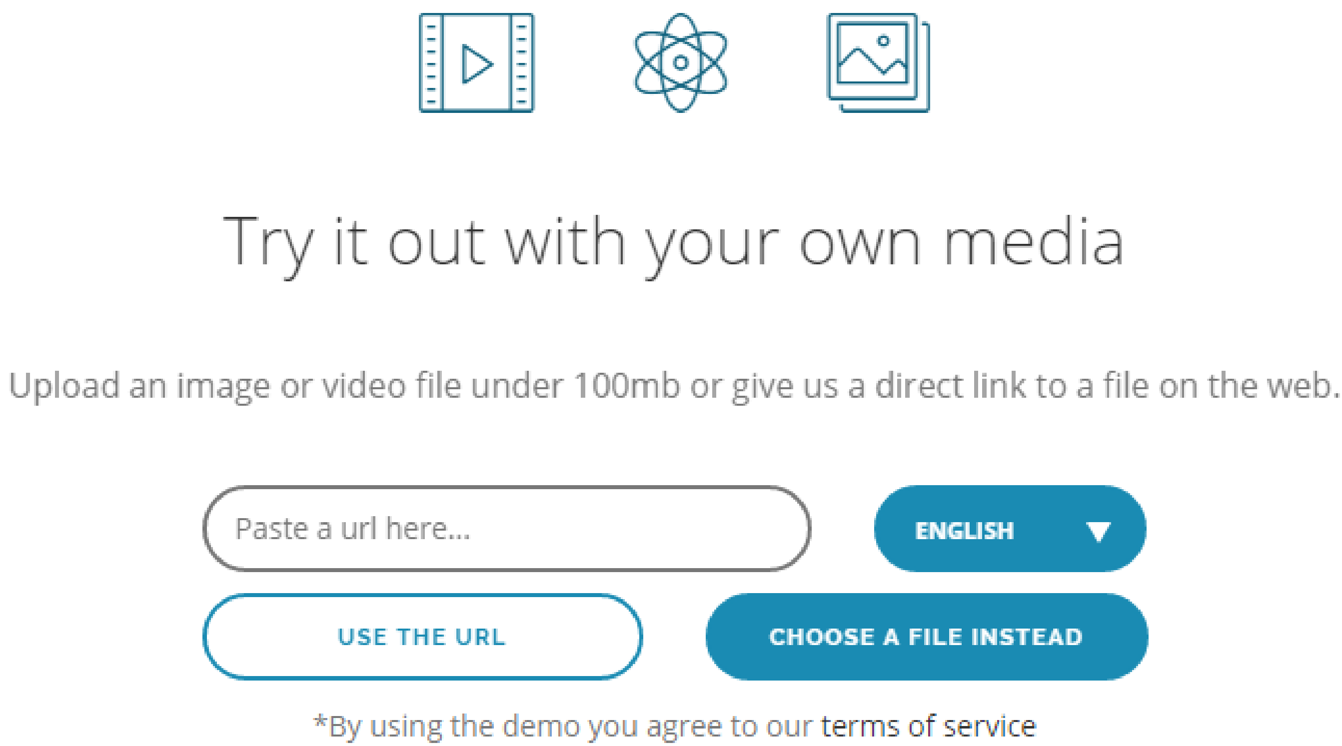
- **Learned embeddings often preserve linear regularities between concepts**
  - Analogy questions can be answered through simple algebraic operations with the vector representation of words.
  - E.g., vec("King") - vec("Man") + vec("Woman") $\approx$ vec("Queen")
  - E.g.,



[Mikolov, NIPS 2013], [Radford, ICLR 2016]

Commercial Recognition Services

- E.g., clarifai



- **Be careful when taking test images from Google Search**
  - **Chances are they may have been seen in the training set...**

Image source: clarifai.com

Commercial Recognition Services

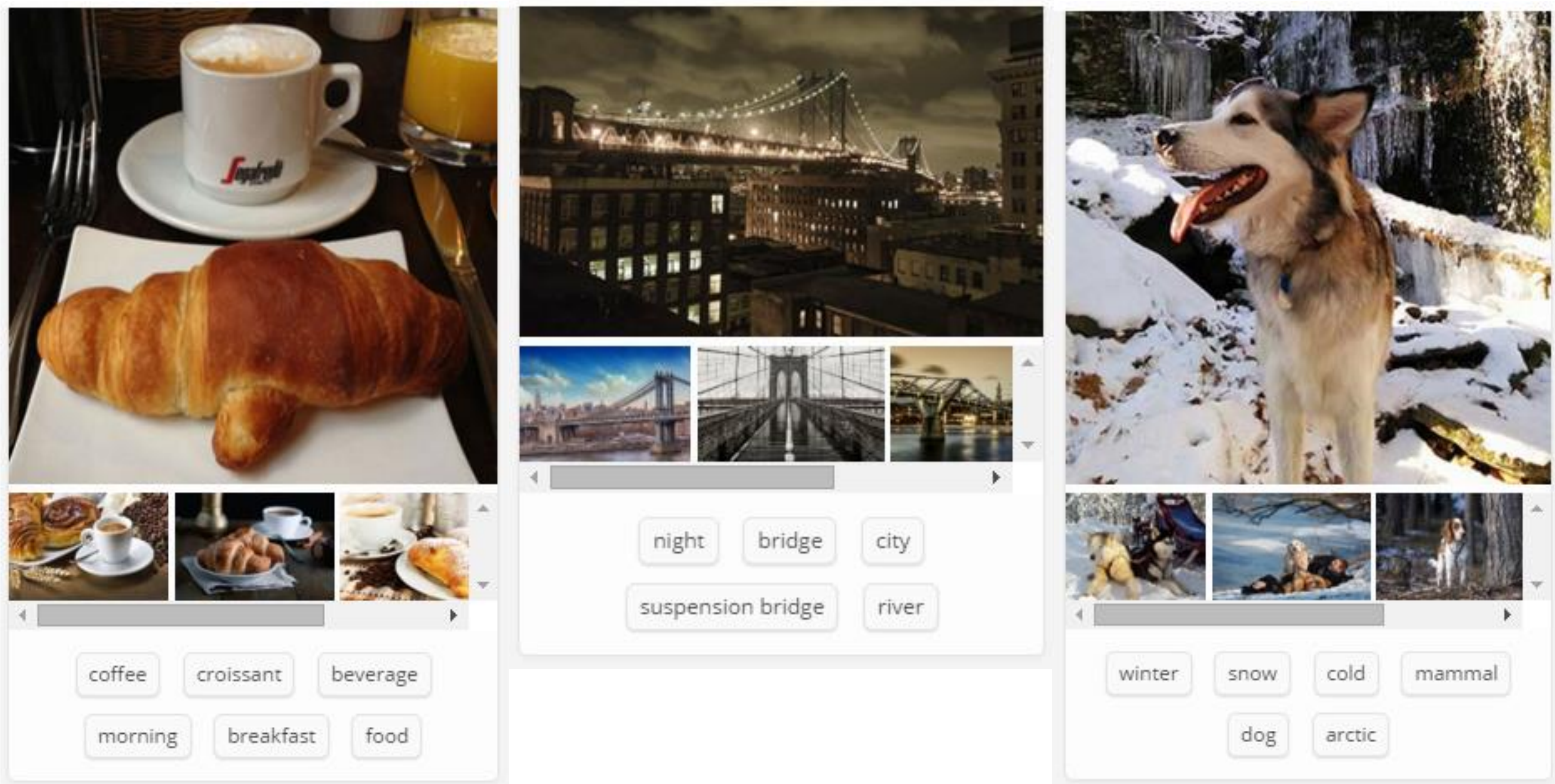


Image source: clarifai.com

RWTH AACHEN UNIVERSITY

References and Further Reading

- RCNN and related ideas:
  - Girshick et al., Region-based Convolutional Networks for Accurate Object Detection and Semantic Segmentation, PAMI, 2014.
  - Zhu et al., segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection, 2015.
- Fast RCNN and related ideas:
  - He et al., Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, 2014.
  - Girshick, Ross, Fast R-CNN, 2015.
- Faster RCNN and related ideas:
  - Szegedy et al., Scalable, High-Quality Object Detection, 2014.
  - Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2015.