(Somewhat) Advanced Convolutional Neural Networks

Andrea Vedaldi

Medical Imaging Summer School

August 2016

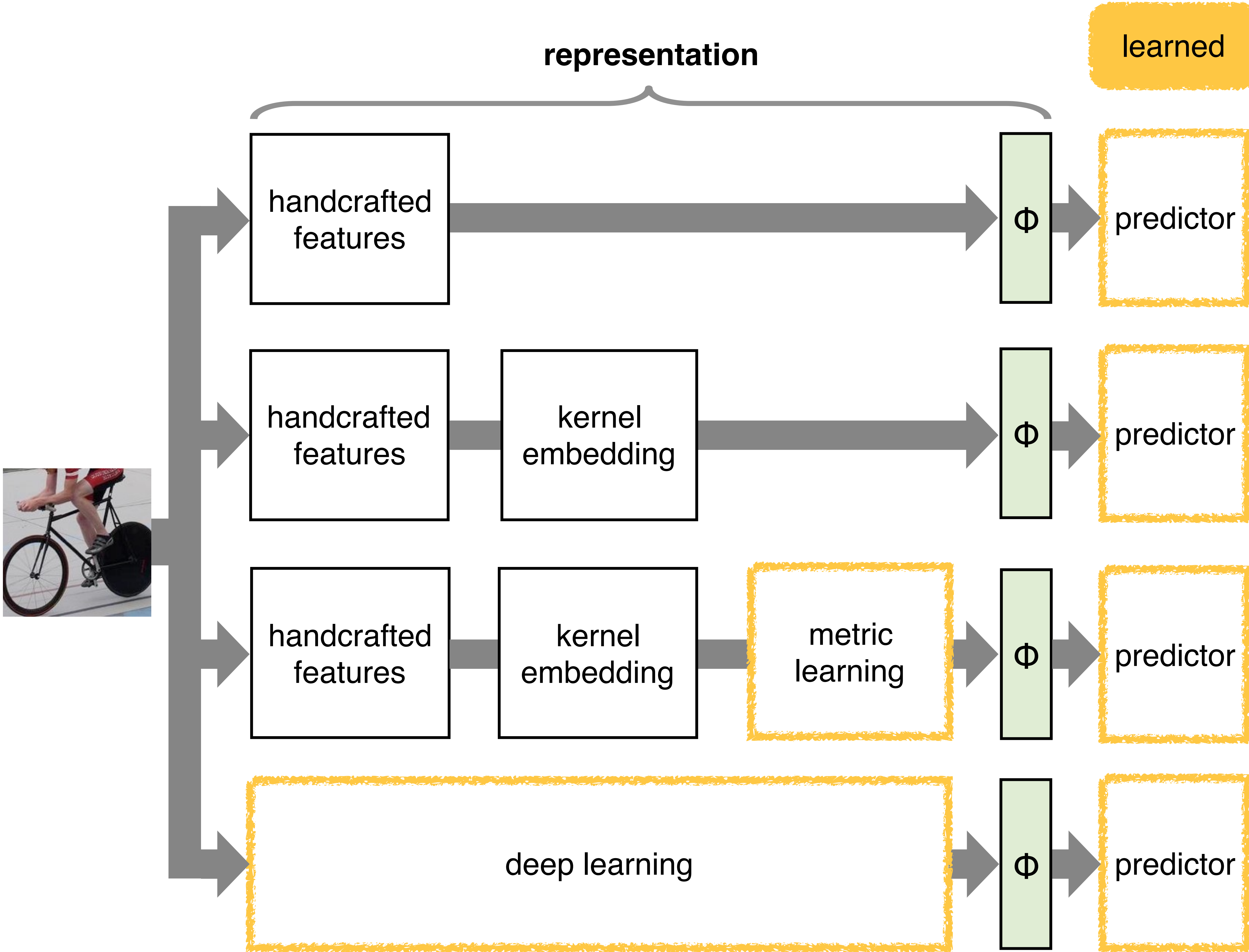
UNIVERSITY OF OXFORD

Image representations

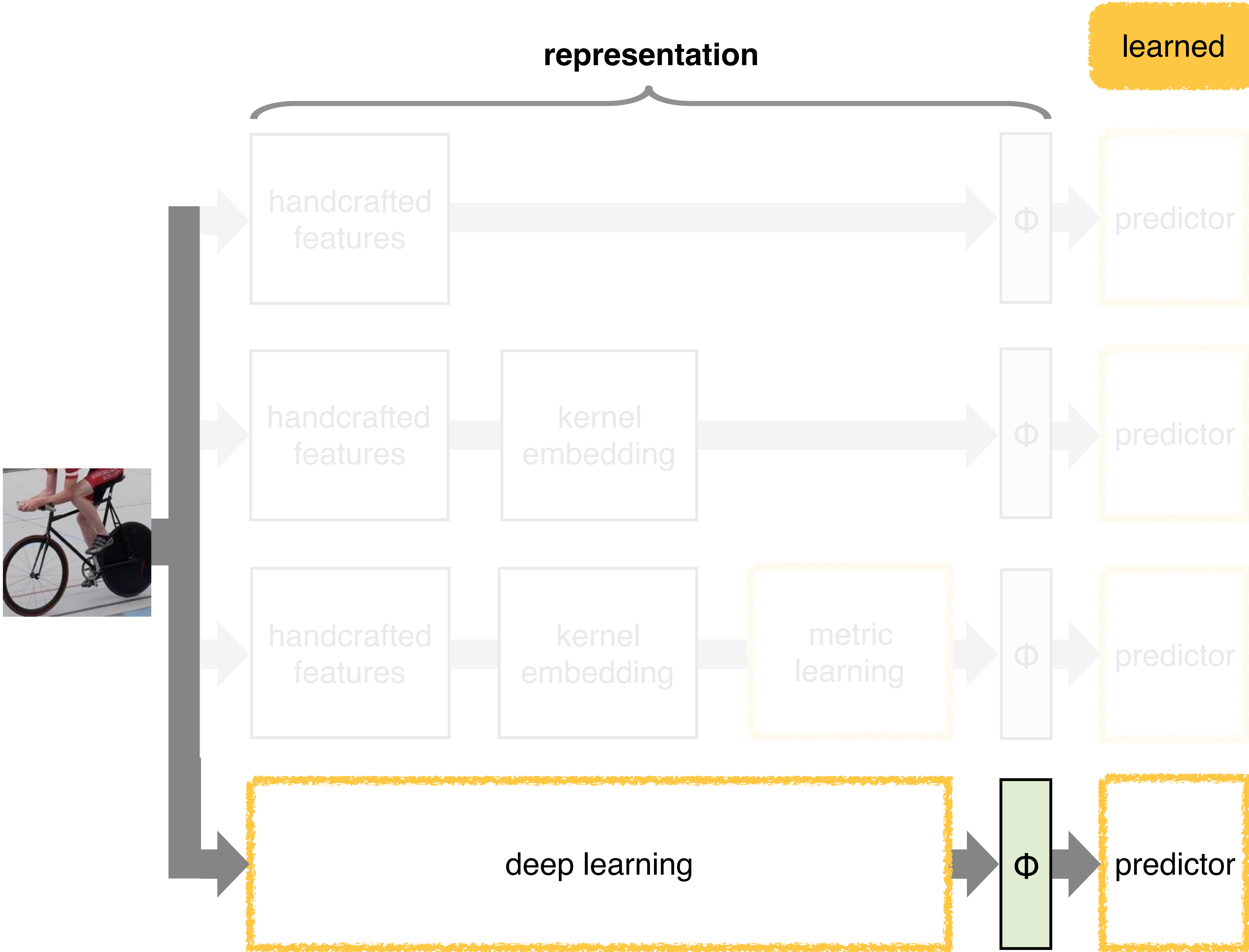
$\mathbf{x}$  $\xrightarrow{\text{encoder } \Phi}$ representation $\Phi(\mathbf{x}) \in \mathbb{R}^d$

An **encoder** maps the data into a **vectorial representation**

Facilitate labelling of images, text, sound, videos, …



learned
representation
handcrafted features
Φ
predictor
handcrafted features
kernel embedding
Φ
predictor
handcrafted features
kernel embedding
metric learning
Φ
predictor
deep learning
Φ
predictor


representation
learned
handcrafted features
Φ
predictor
handcrafted features
kernel embedding
Φ
predictor
handcrafted features
kernel embedding
metric learning
Φ
predictor
deep learning
Φ
predictor

Understanding visual representations

Modern convolutional neural networks

Applications

Segmentation: “fully convolutional” networks

Object detection: R-CNN and weak supervision

Modern convolutional neural networks

Applications

Segmentation: “fully convolutional” networks

Object detection: R-CNN and weak supervision

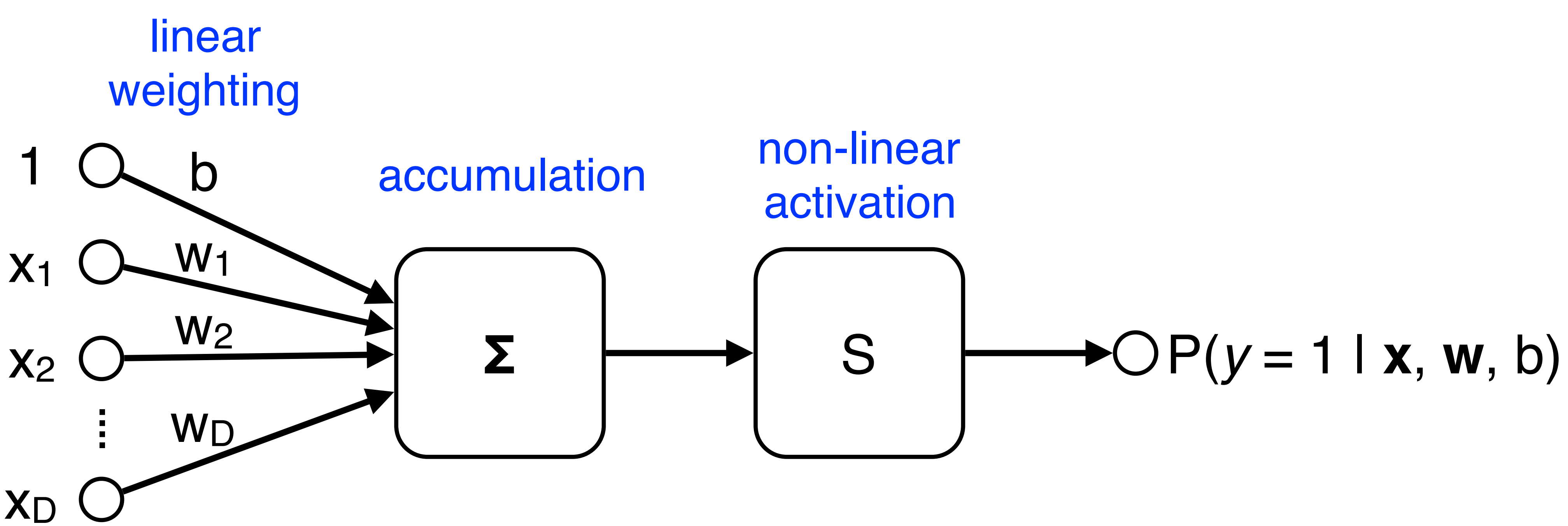
Convolutional neural networks

Origin (1950-60)

Perceptron

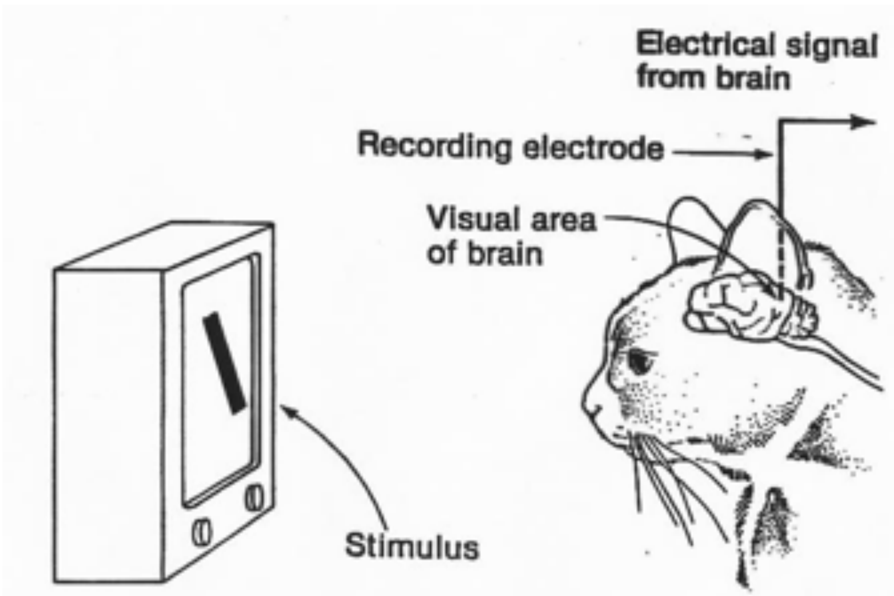
[Rosenblatt 57]

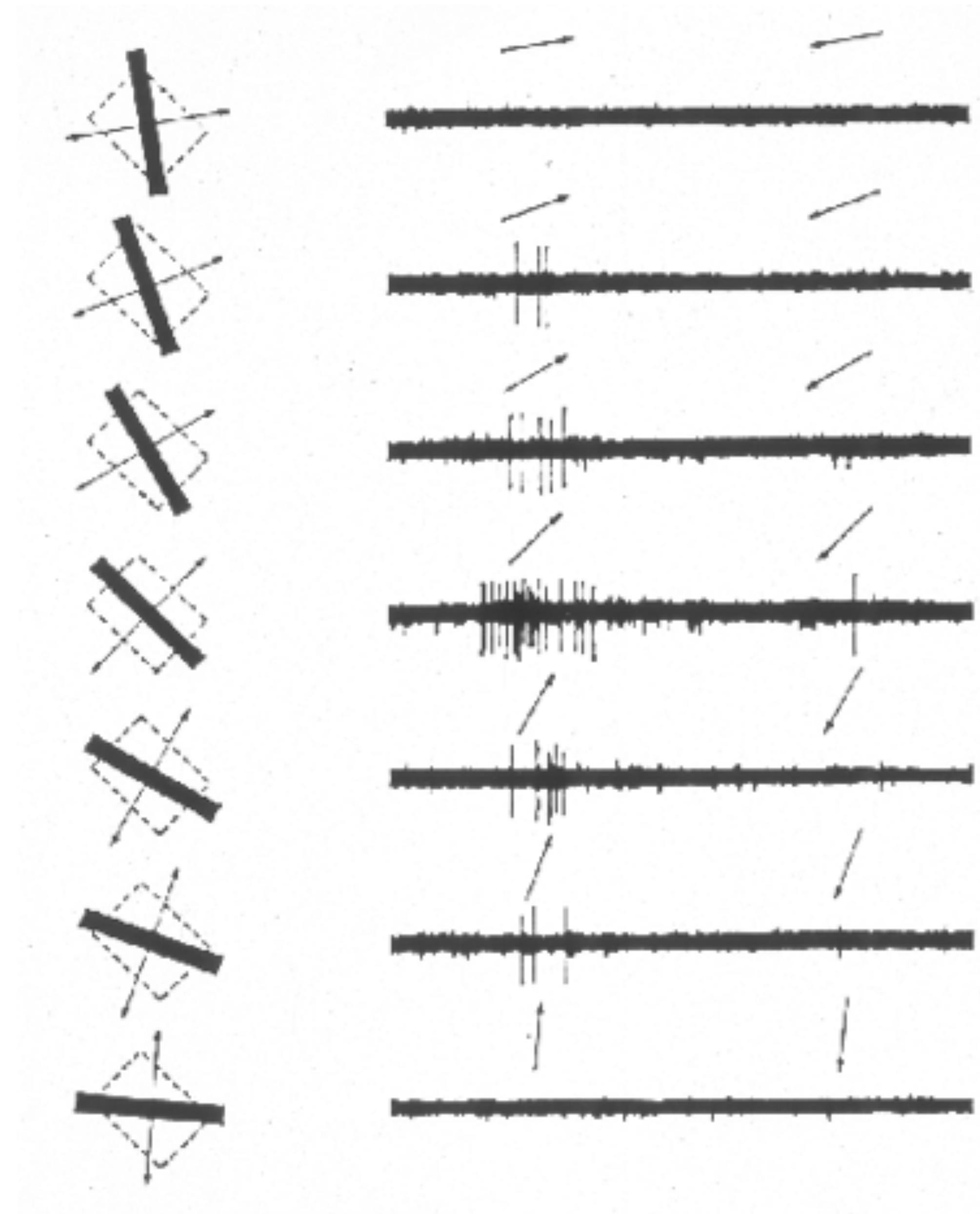
The goal is estimating the posterior probability of the binary label *y* of a vector **x**:

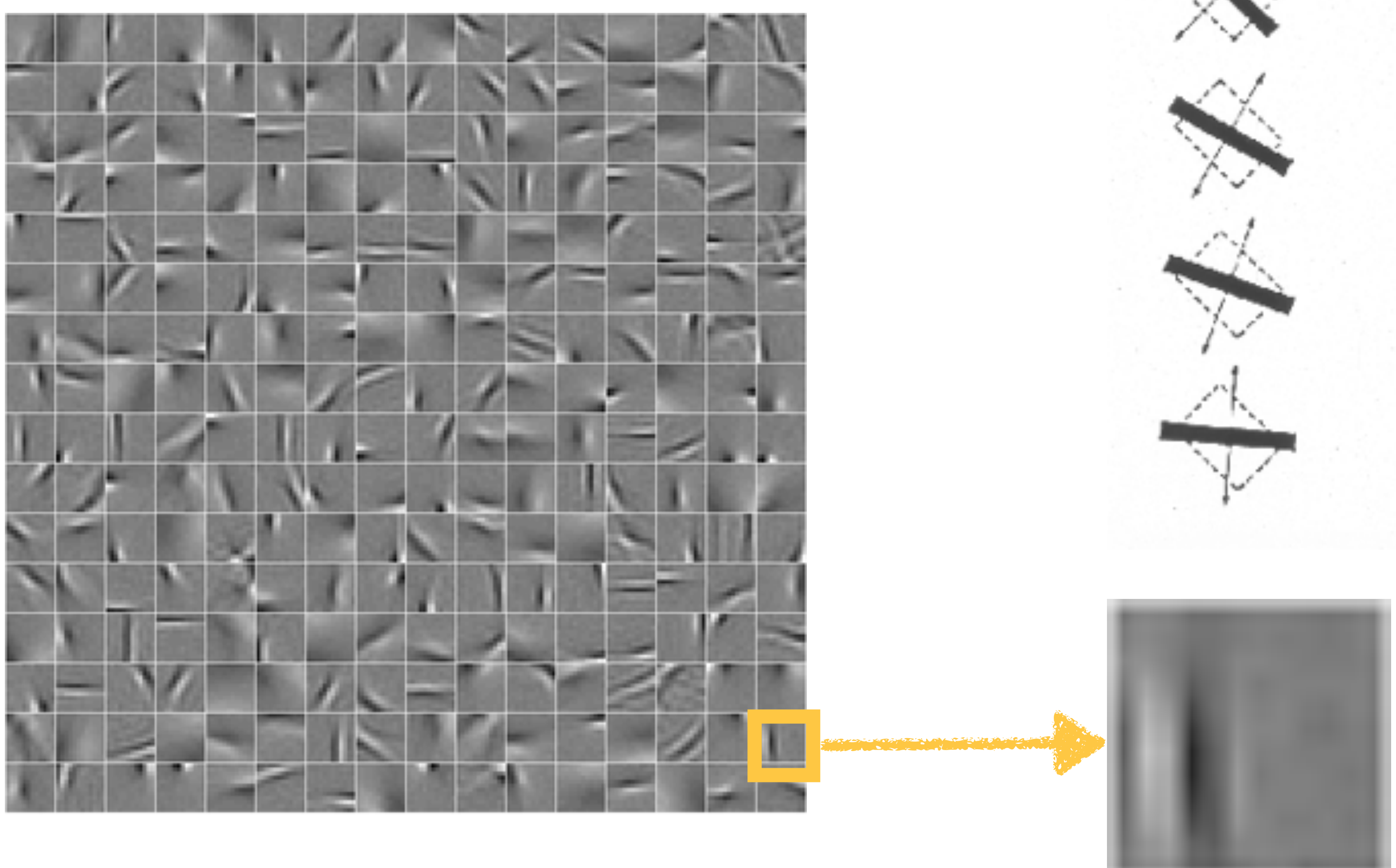


Discovery of oriented cells in the visual cortex

[Hubel and Wiesel 59]



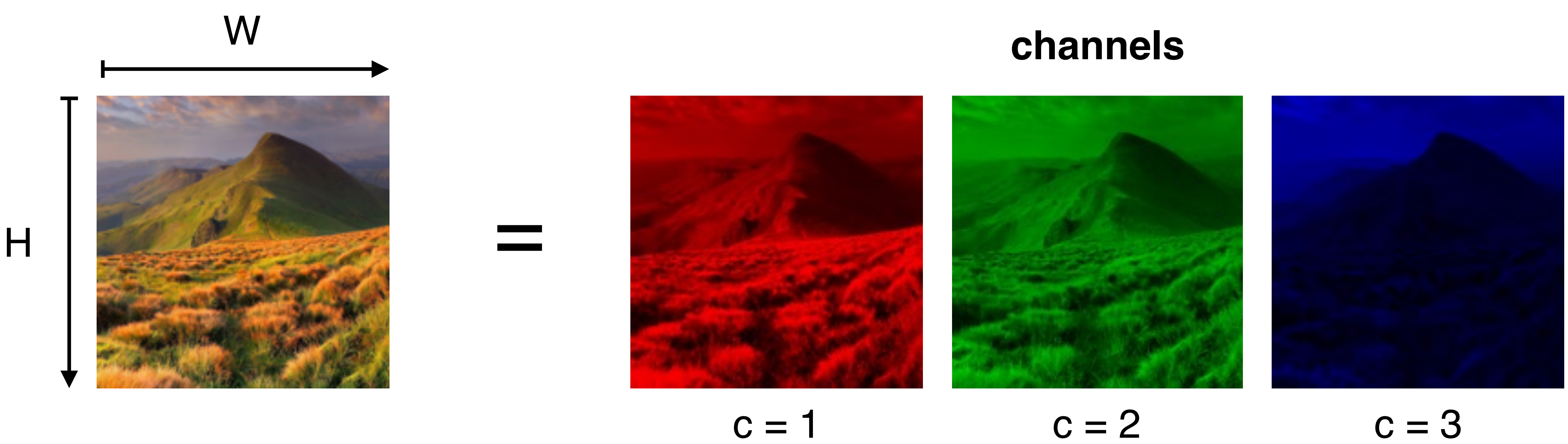


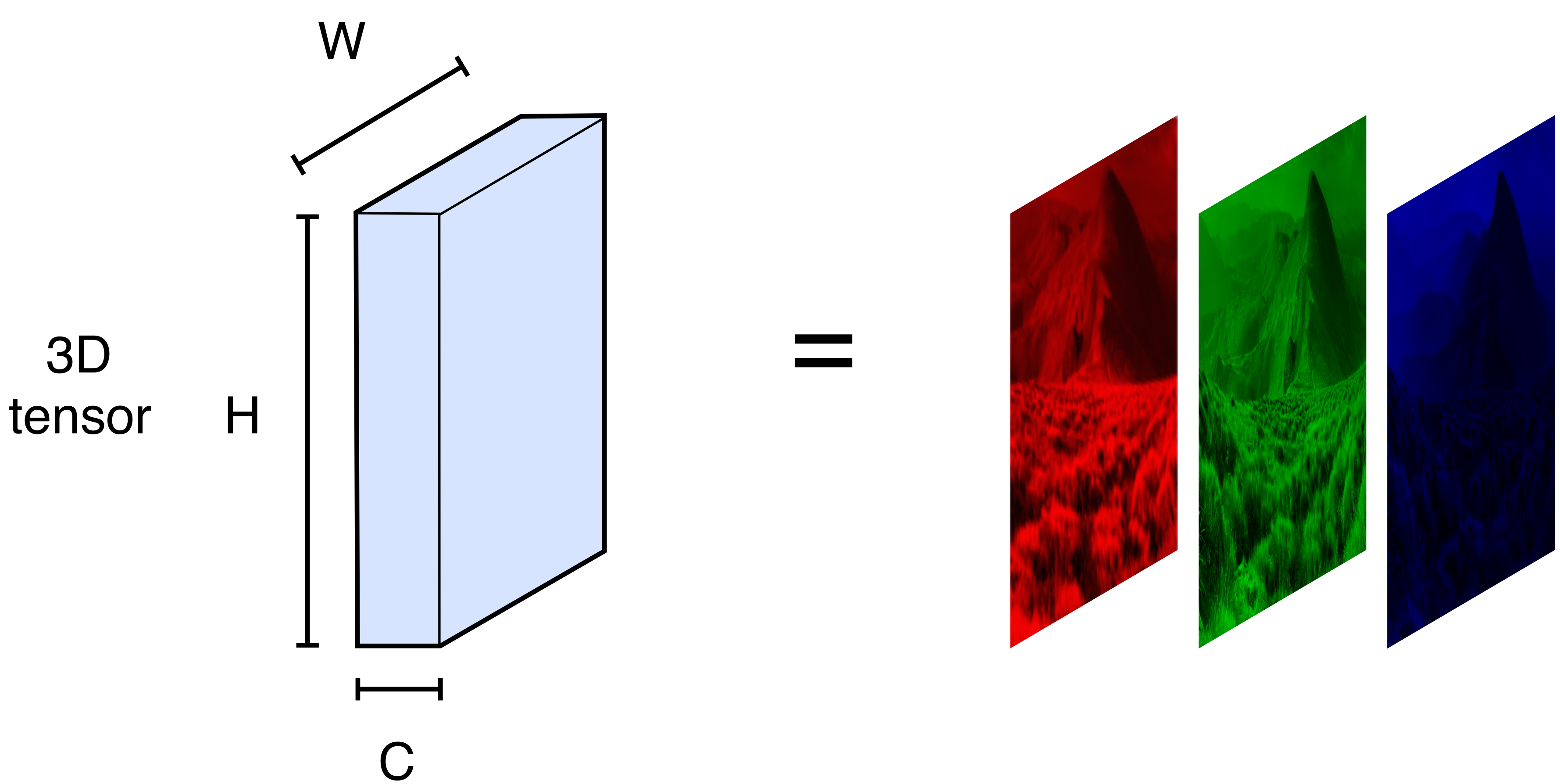


oriented filter

Data = 3D tensors

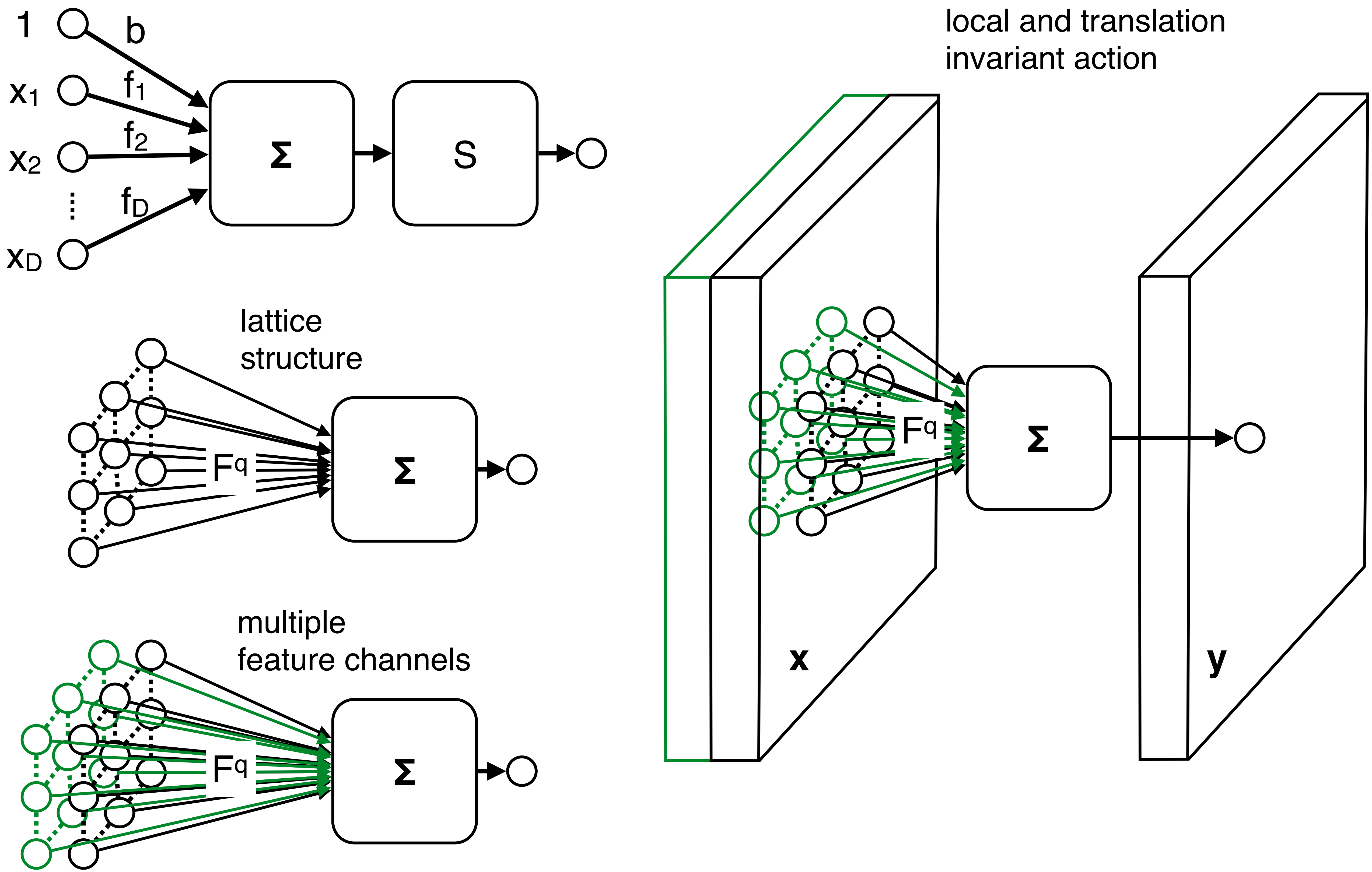
There is a vector of feature channels (e.g. RGB) at each spatial location (pixel).





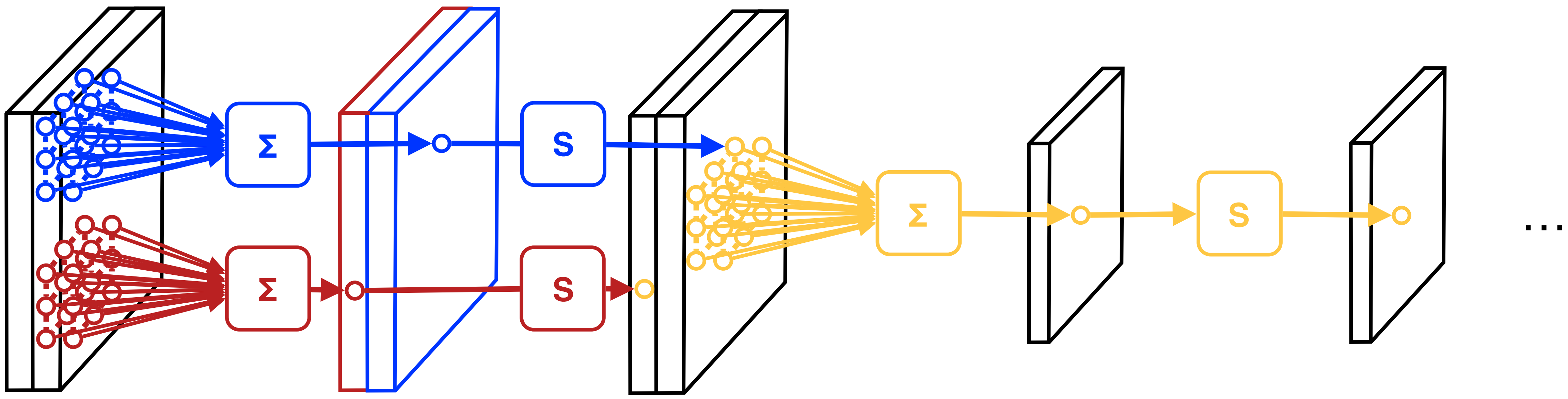
Linear convolution

As a neural network



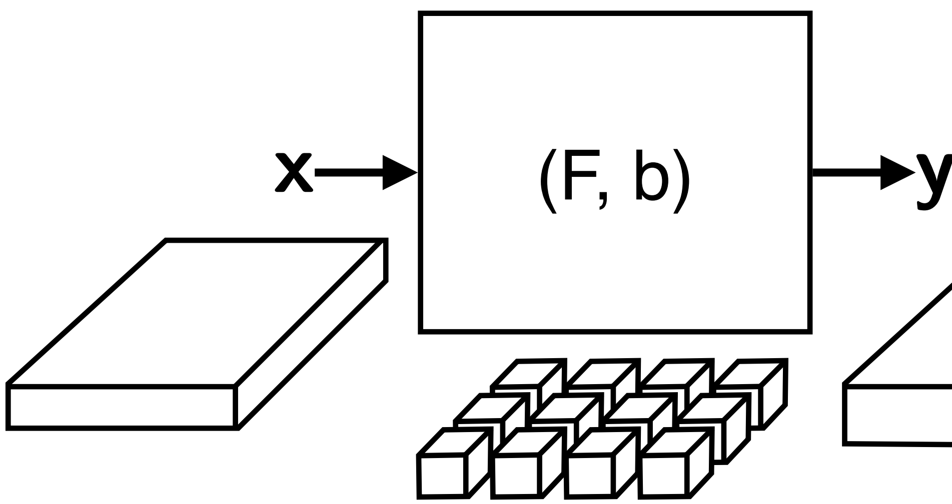
Deep architectures

Repeat linear / non-linear operators



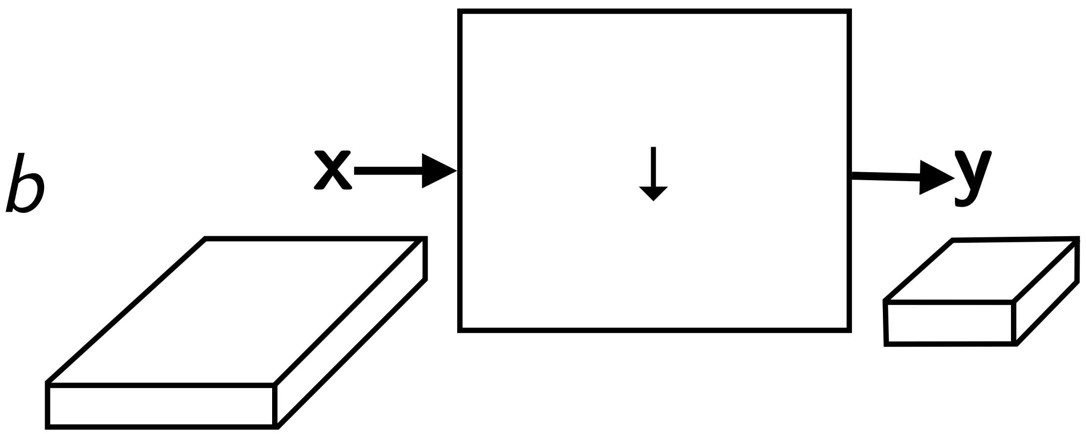
Components of deep architectures

linear 3D filters

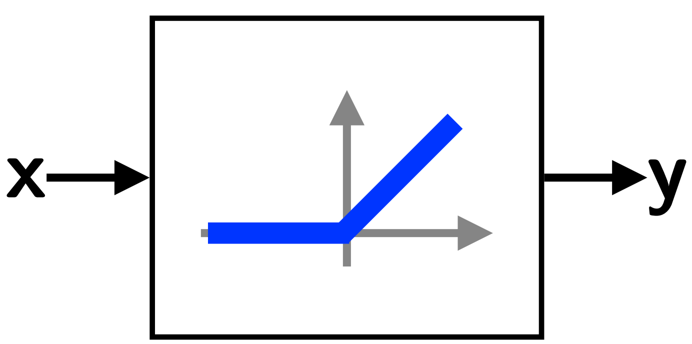


$$\mathbf{y} = F * \mathbf{x} + b$$

downsampling

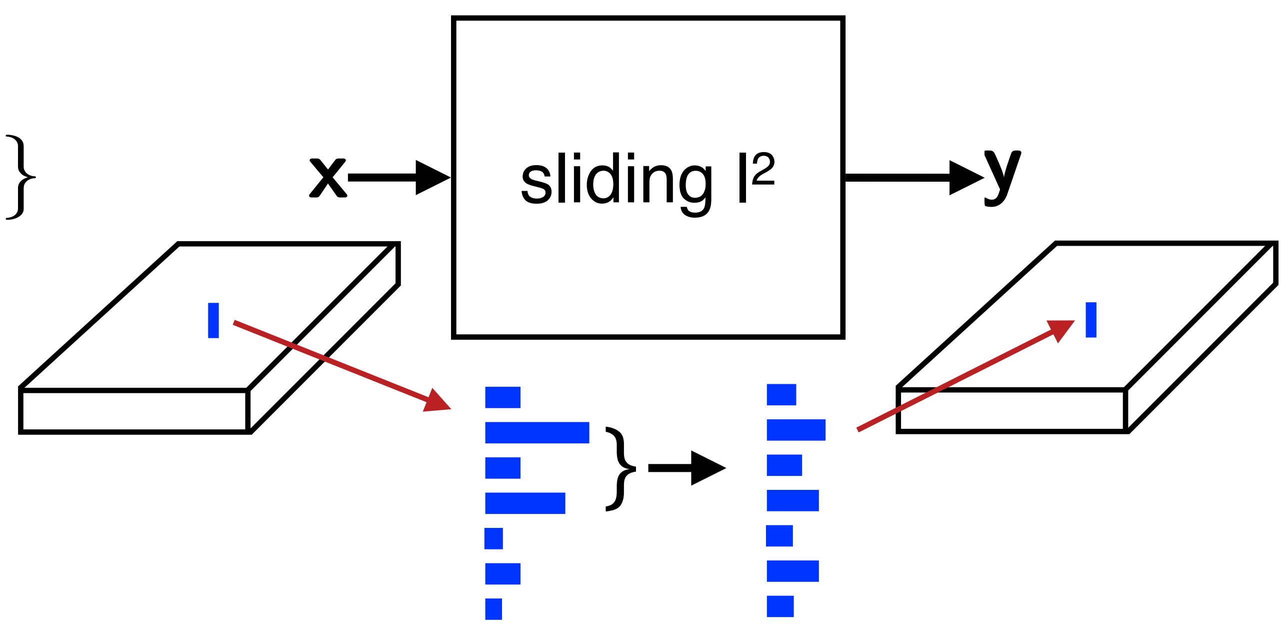


ReLU

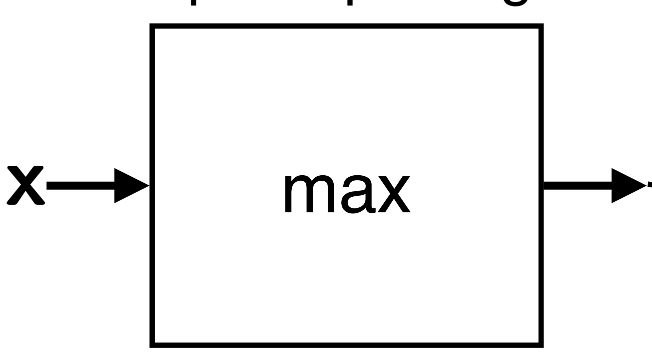


$$\mathbf{y} = \max\{0, \mathbf{x}\}$$

normalization



spatial pooling

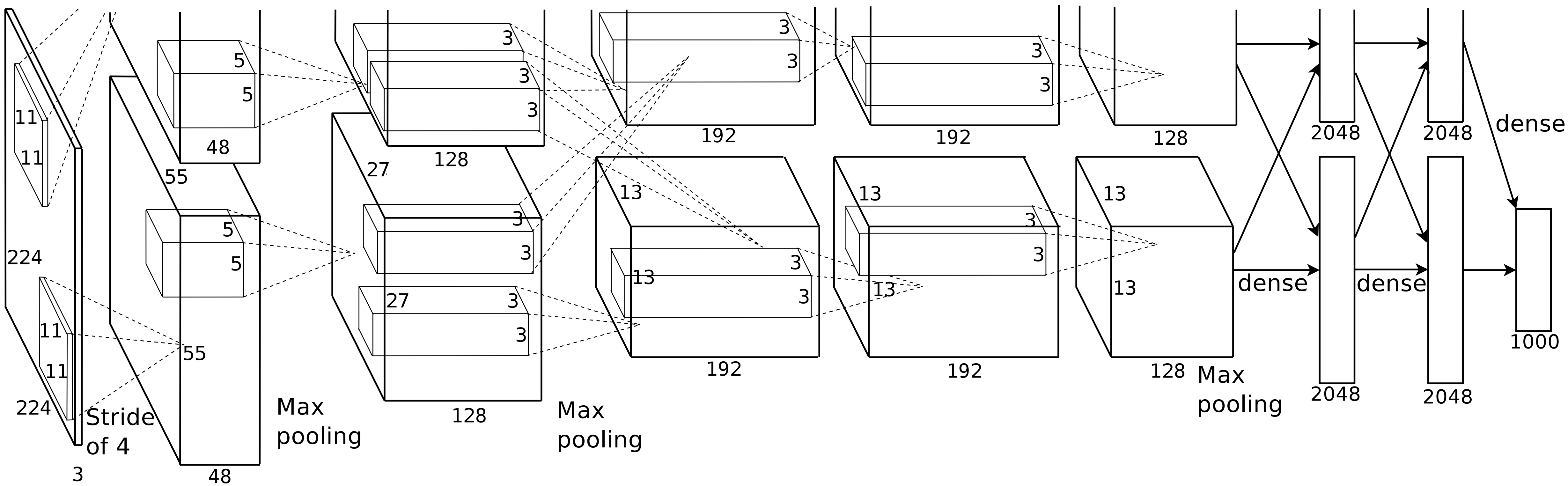


$$y_{ijk} = \max_{pq \in \Omega_{ij}} x_{pqk}$$

Modern convolutional networks

From AlexNet (2012) to ResNet (2015)

Modern convolutional nets



[AlexNet by Krizhevsky et al. 2012]

Excellent **performance** in image understanding tasks

Learn a sequence of **general-purpose representations**

Millions of parameters learned from data

The “**meaning**” of the representation is **unclear**

How deep is deep enough?

AlexNet (2012)

5 convolutional layers

3 fully-connected layers

How deep is deep enough?

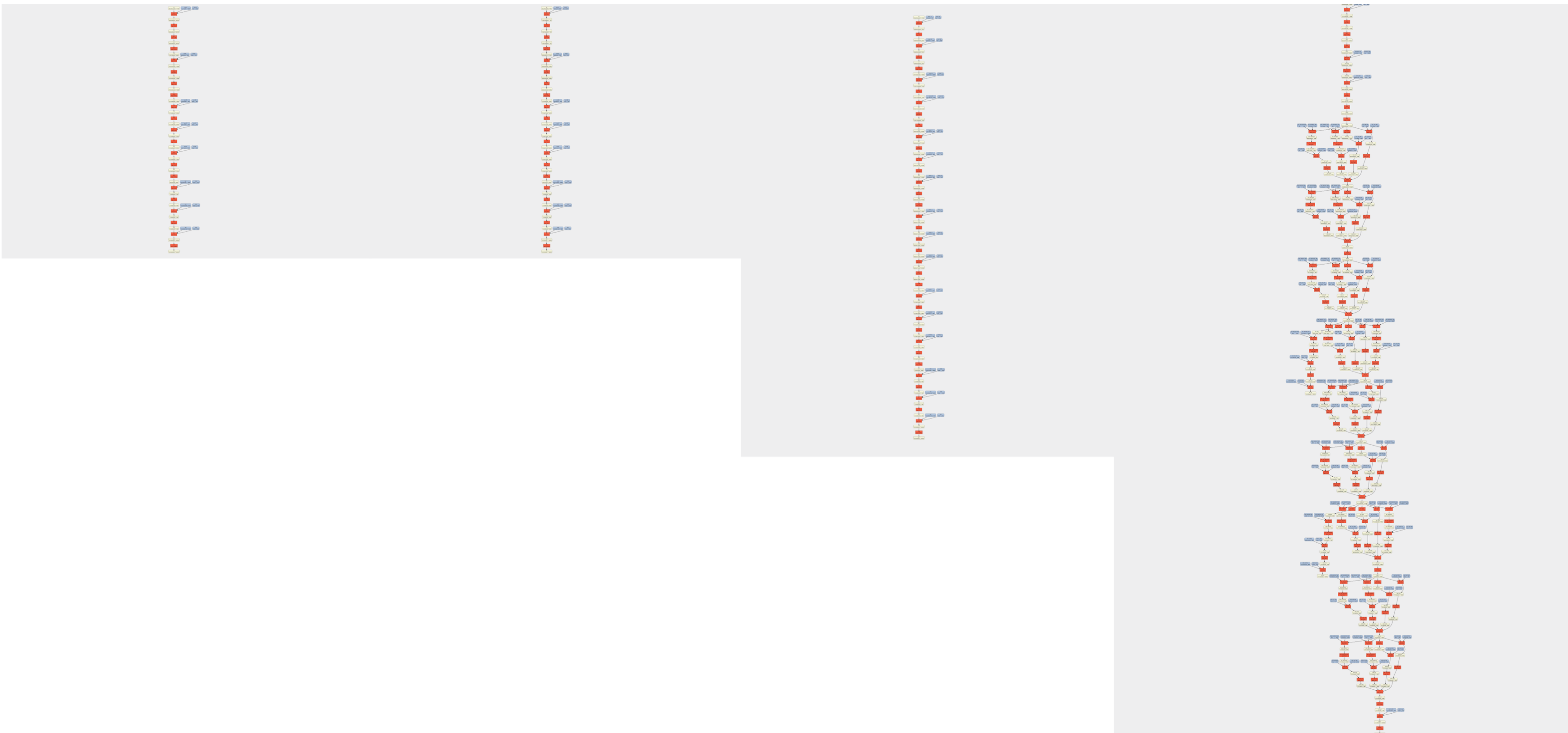
AlexNet (2012)

VGG-M (2013)

VGG-VD-16 (2014)

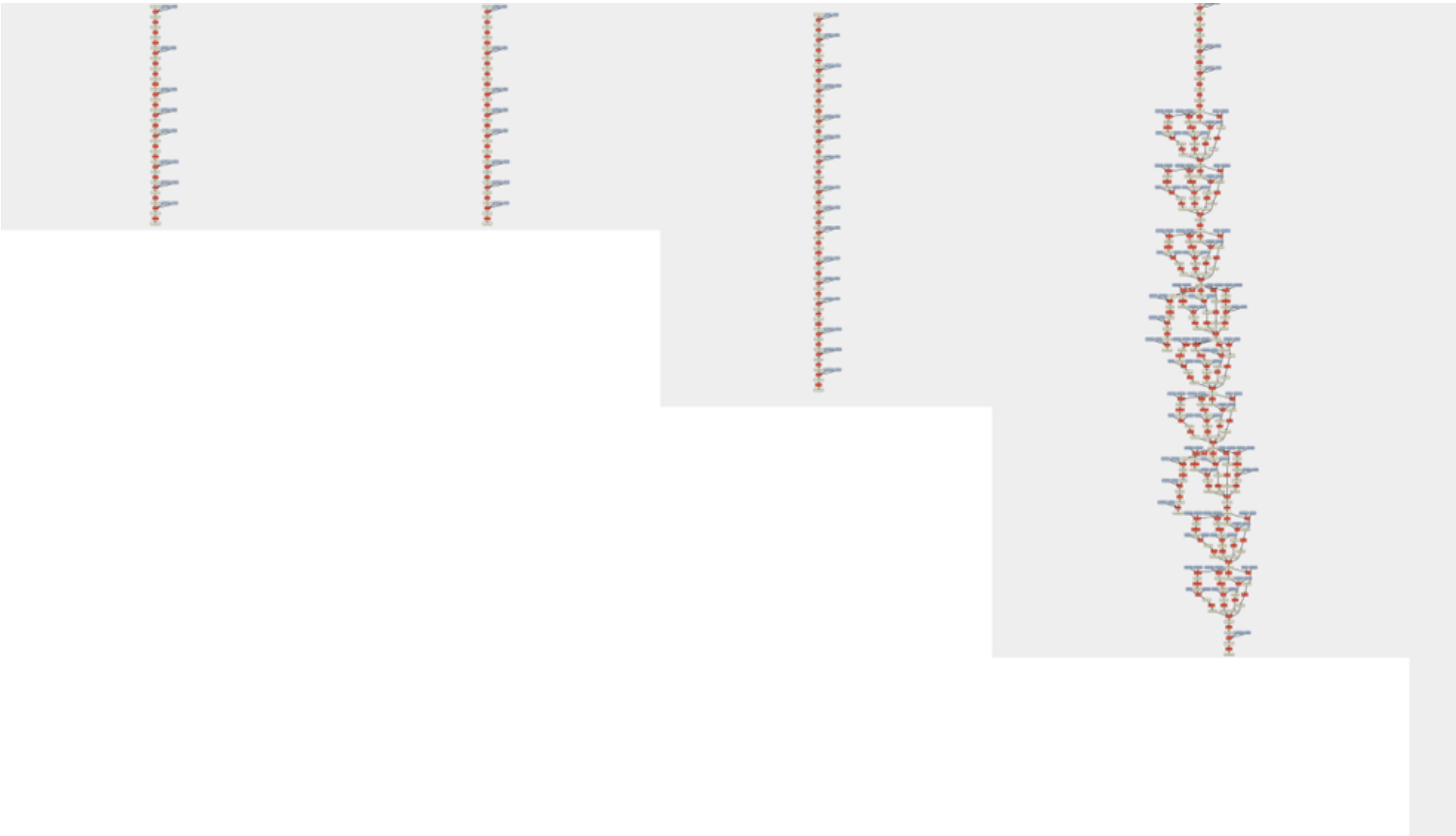
How deep is deep enough?

AlexNet (2012) VGG-M (2013) VGG-VD-16 (2014) GoogLeNet (2014)

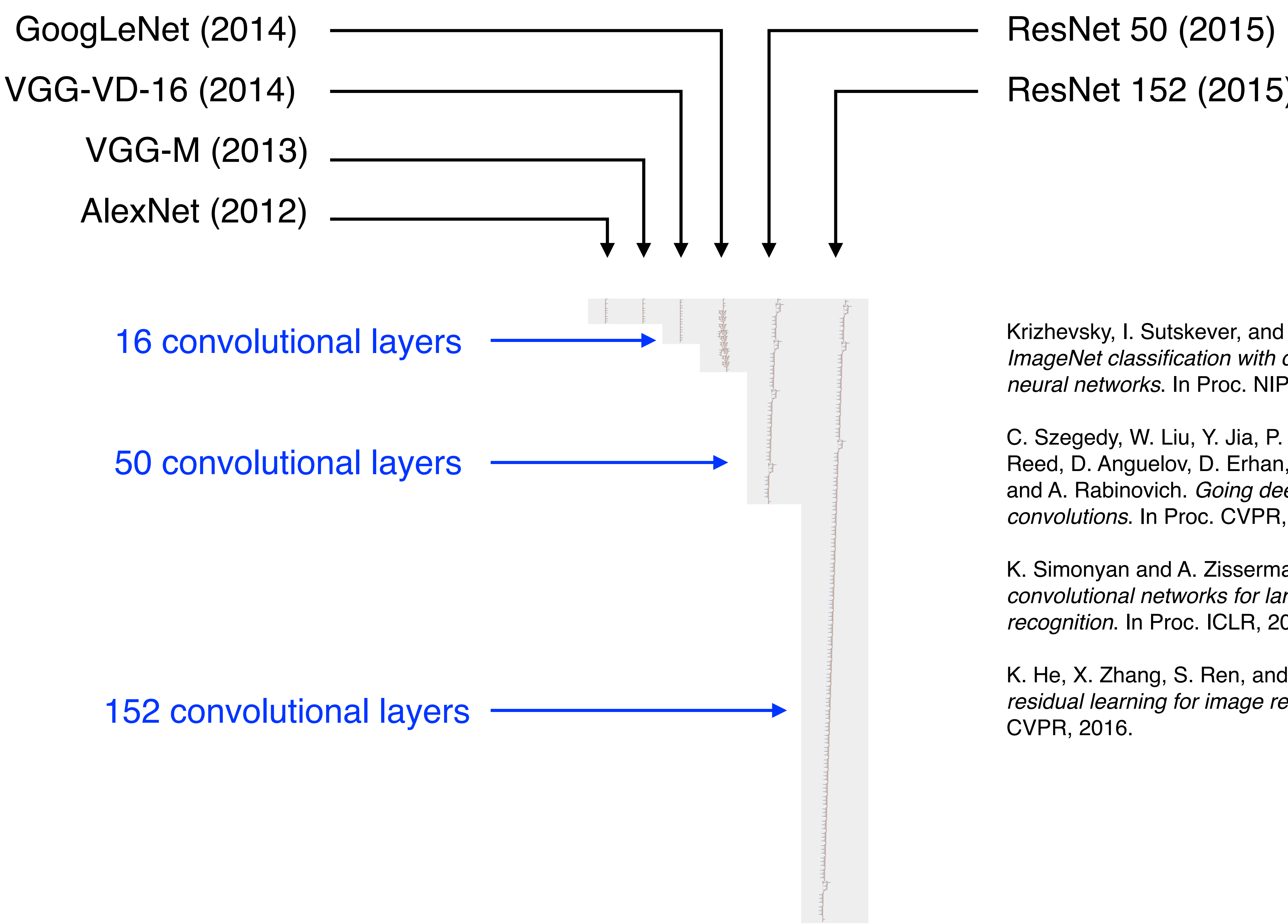


How deep is deep enough?

AlexNet (2012) VGG-M (2013) VGG-VD-16 (2014) GoogLeNet (2014)



How deep is deep enough?



Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet classification with deep convolutional neural networks*. In Proc. NIPS, 2012.

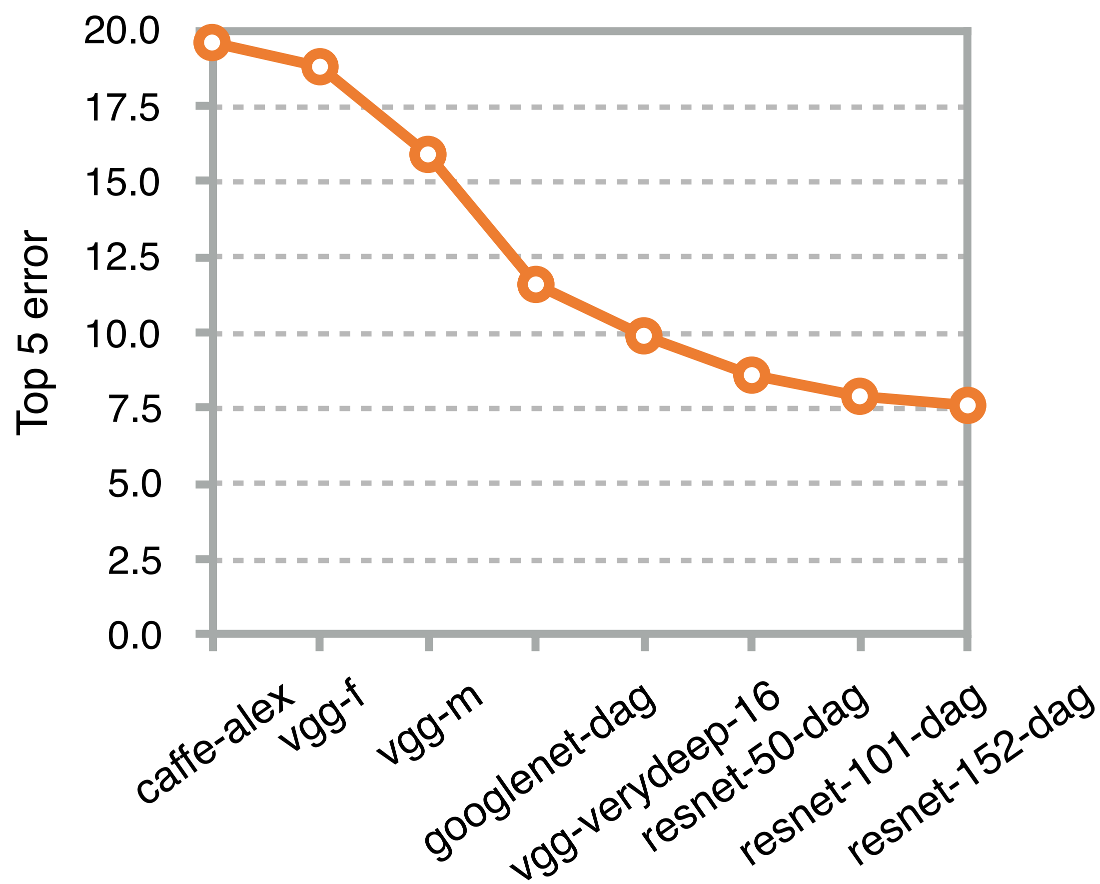
C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going deeper with convolutions*. In Proc. CVPR, 2015.

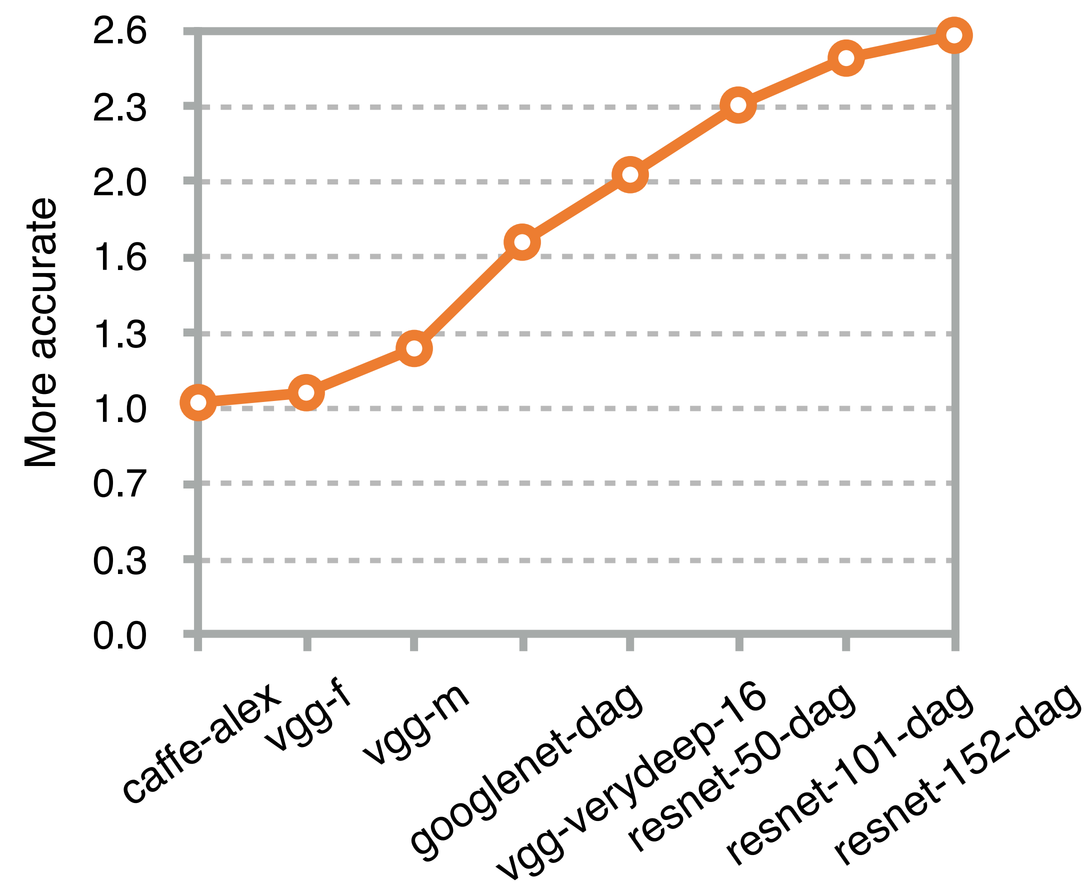
K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. In Proc. ICLR, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. *Deep residual learning for image recognition*. In Proc. CVPR, 2016.

Accuracy

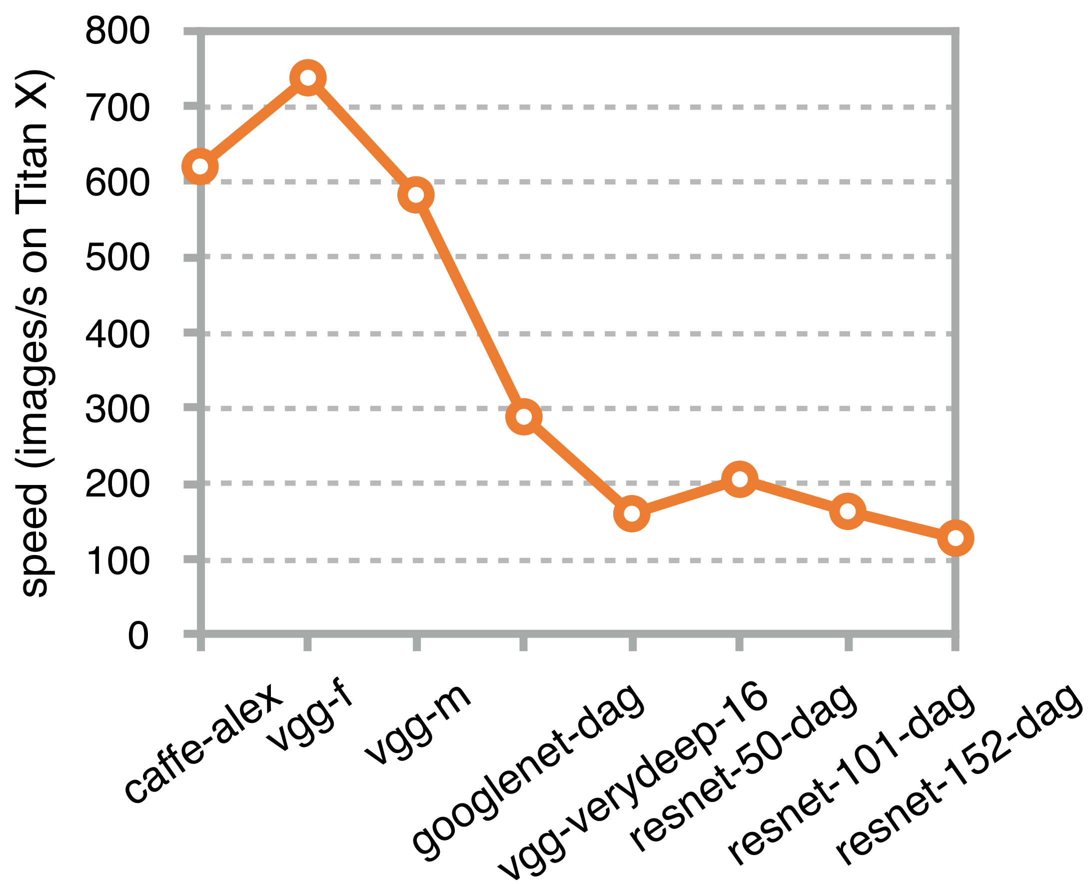
3 × more accurate in 3 years

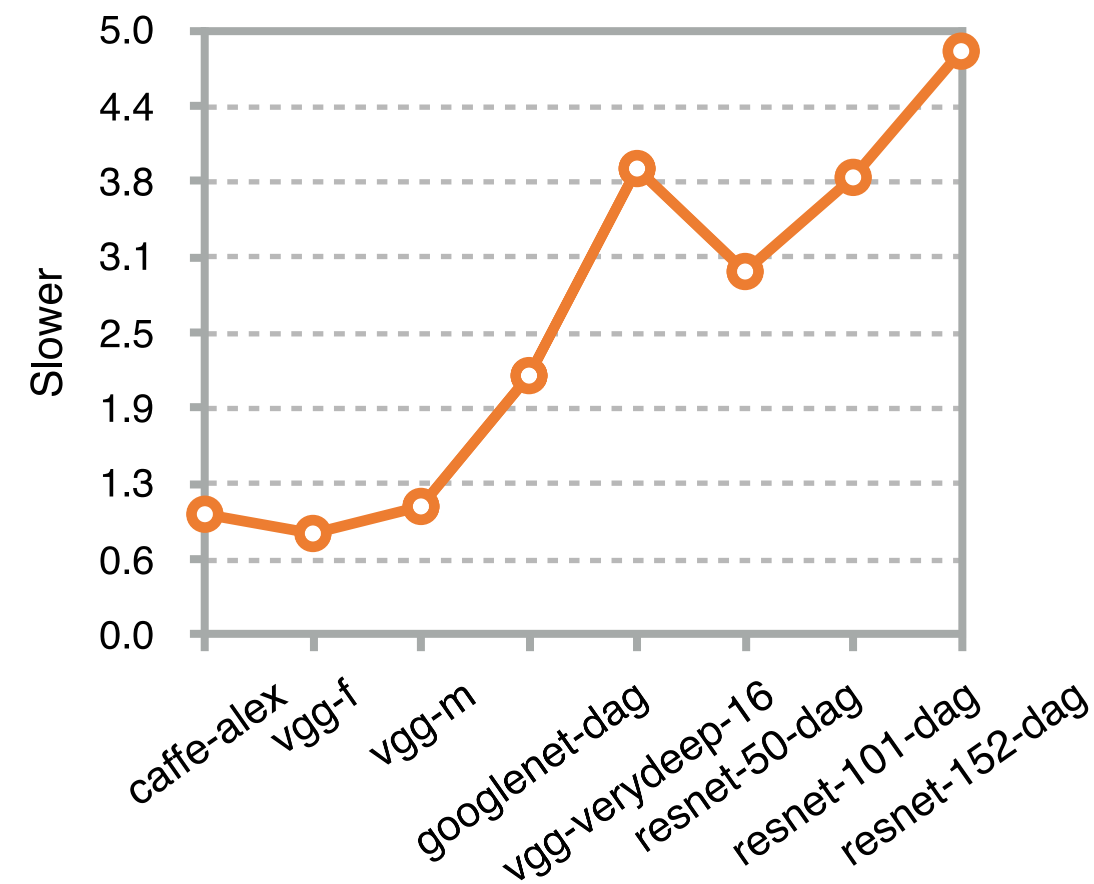




Speed

5 × slower





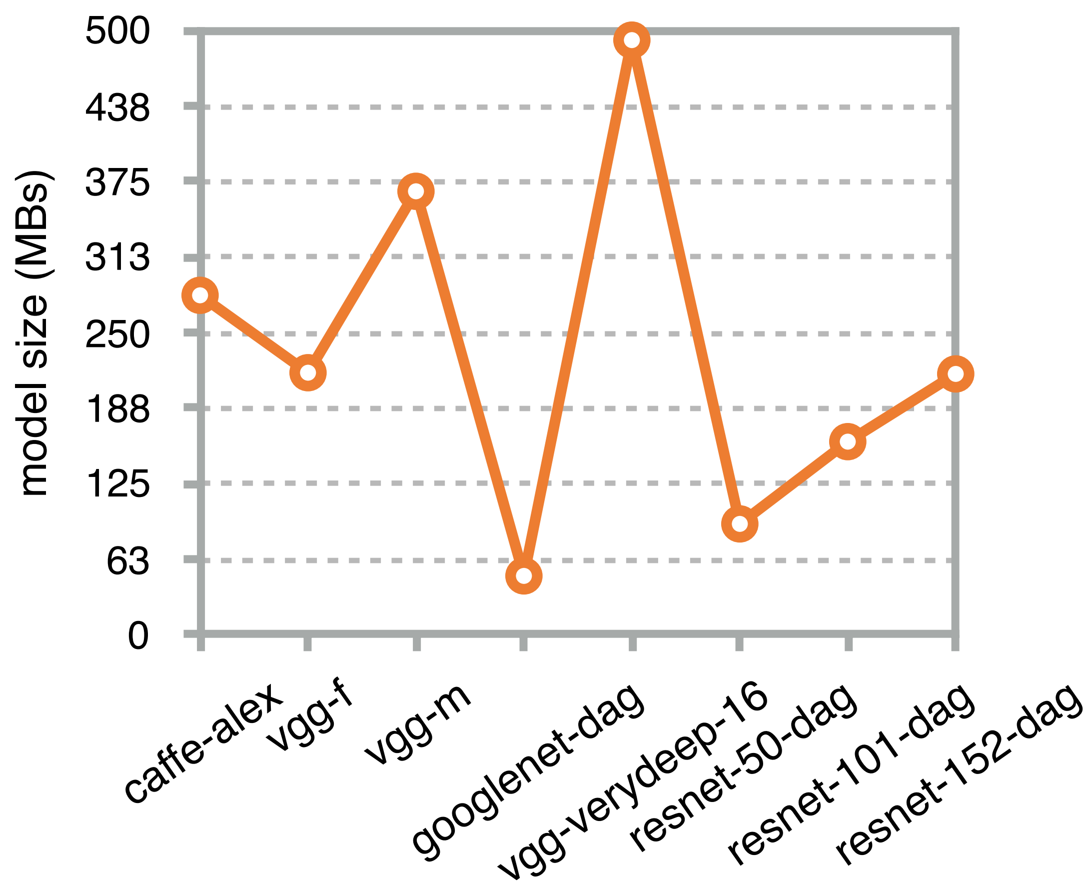
Remark: *101 ResNet* layers same size/speed as *16 VGG-VD* layers

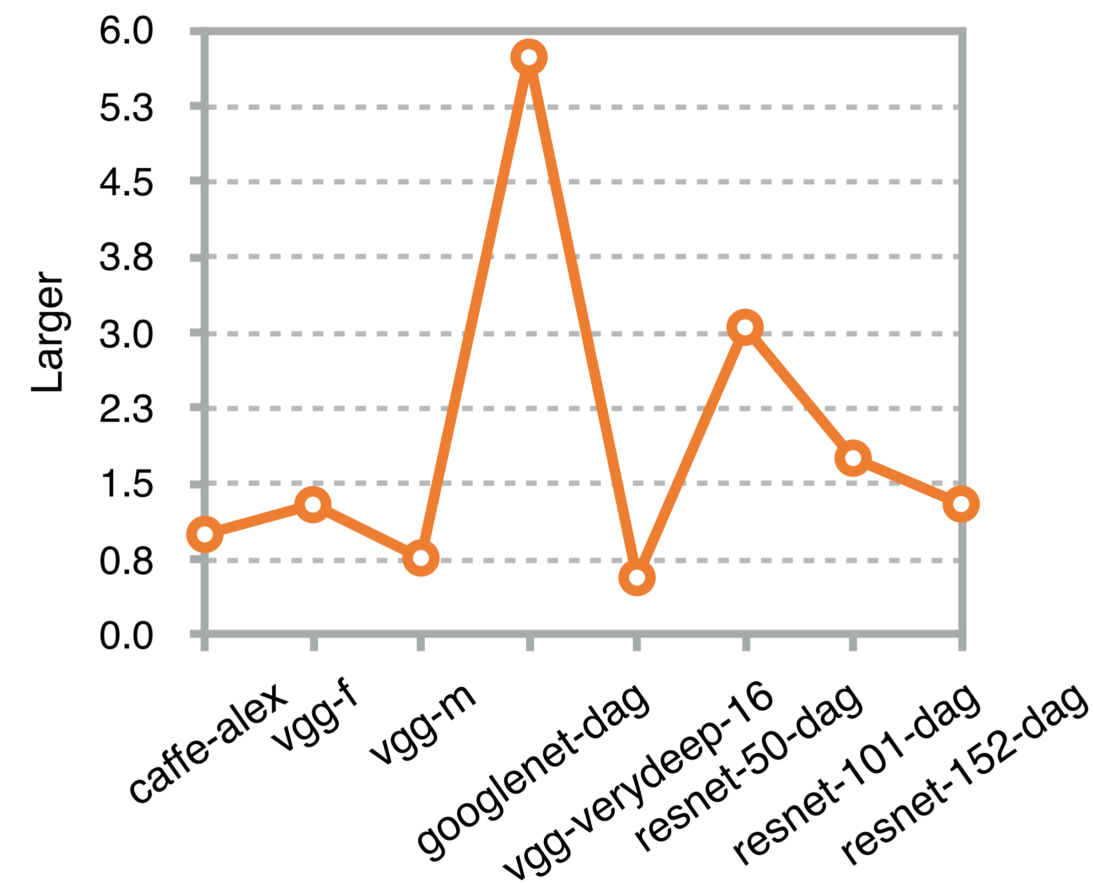
Reason: far fewer feature channels (quadratic speed/space gain)

Moral: optimize your architecture

Model size

Num. of parameters is about the same





Remark: *101 ResNet* layers same size/speed as *16 VGG-VD* layers

Reason: far fewer feature channels (quadratic speed/space gain)

Moral: optimize your architecture

Recent advances

Design guidelines

Batch normalization

Residual learning

Recent advances

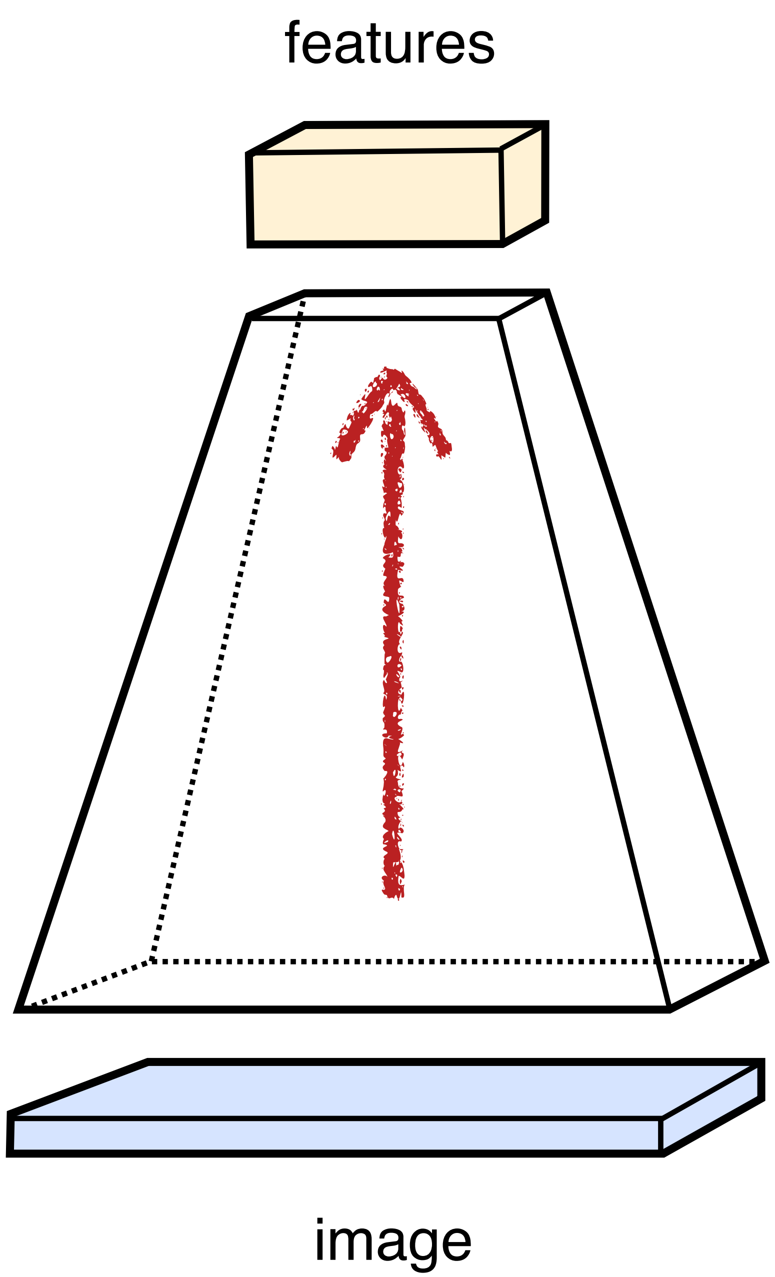
Design guidelines

Batch normalization

Residual learning

Design guidelines

Guideline 1: *Avoid tight bottlenecks*



From bottom to top

- The *spatial resolution* $H \times W$ decreases
- The *number of channels* C increases

Guideline

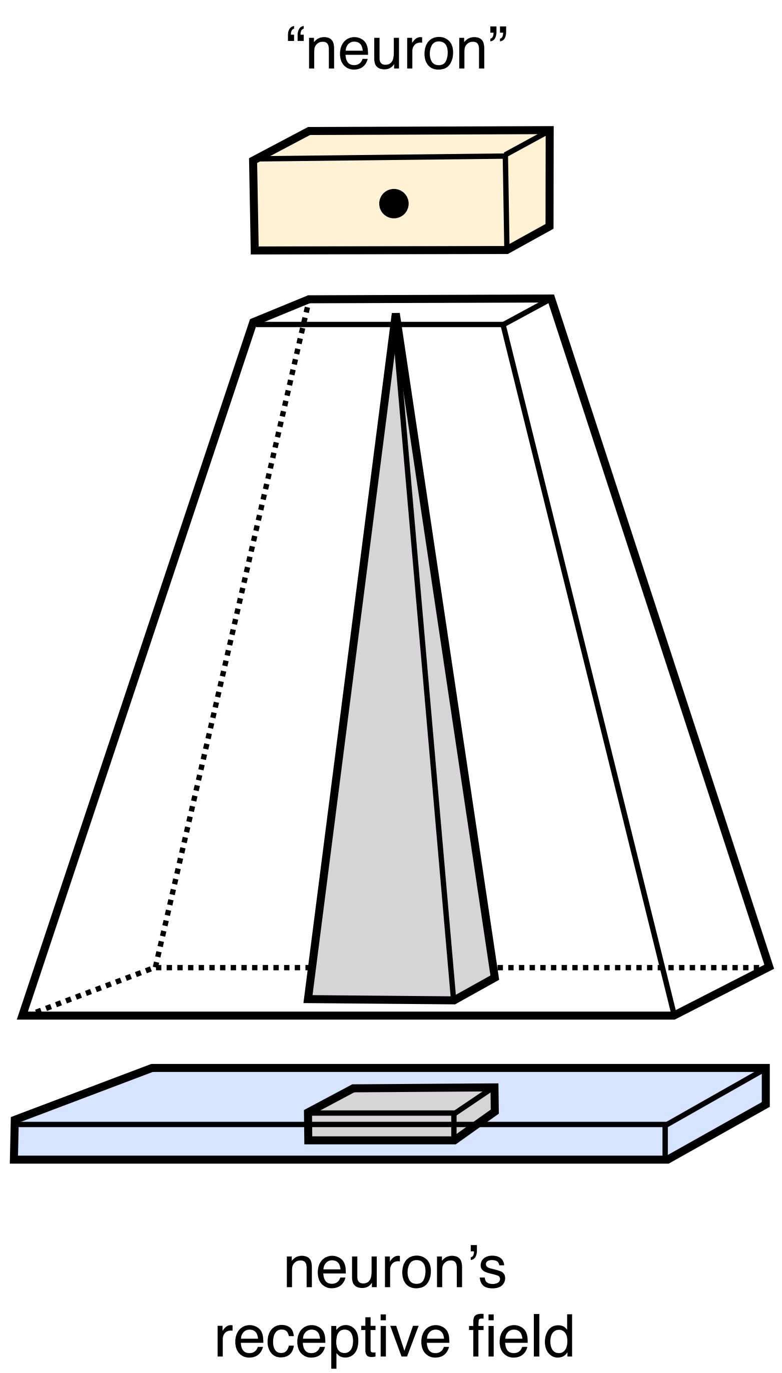
- Avoid tight information bottleneck
- Decrease the *data volume* $H \times W \times C$ slowly

K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. In Proc. ICLR, 2015.

C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens. *Rethinking the inception architecture for computer vision*. In Proc. CVPR, 2016.

Receptive field

Eventually, it must be large enough



Receptive field of a neuron

- The image region influencing a neuron
- Anything happening outside is invisible to the neuron

Importance

- Large image structures cannot be detected by neurons with small receptive fields

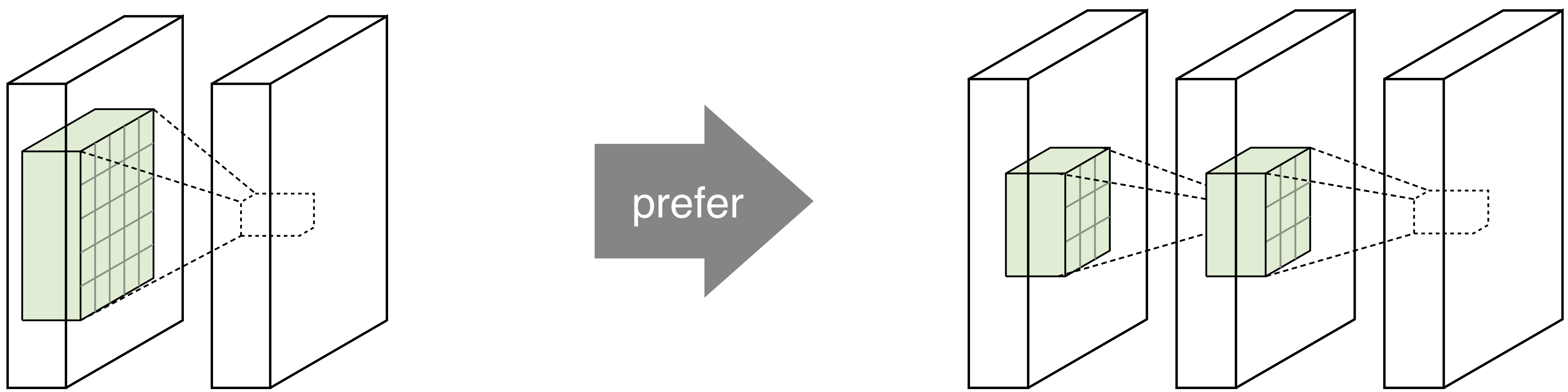
Obtaining large receptive fields

- Large filters
- Chains of small filters

Design guidelines

Guideline 2: *Prefer small filter chains*

One big filter bank | **Two smaller filter banks**



5 × 5 filters + ReLU

3 × 3 filters + ReLU | 3 × 3 filters + ReLU

Benefit 1: less parameters, possibly faster

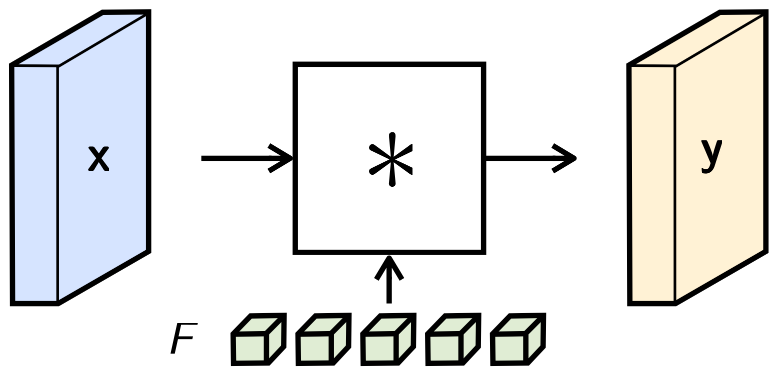
Benefit 2: same receptive field of a bigger filter

Benefit 3: packs two non-linearities (ReLUs)

Design guidelines

Guideline 3: ***Keep the number of channels at bay***

$H \times W \times C$



$H_f \times W_f \times C \times K$

C = num. input channels

K = num. output channels

Num. of operations

$$\frac{H \times H_f}{\text{stride}} \times \frac{W \times W_f}{\text{stride}} \times C \times K$$

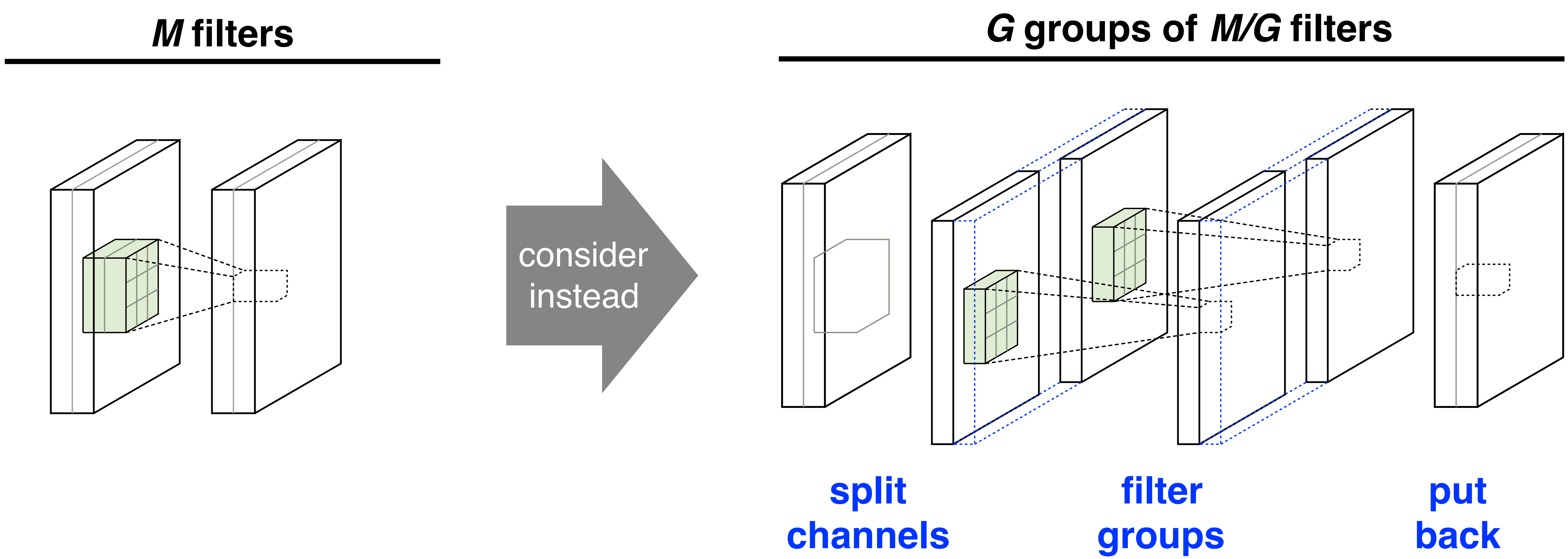
Num. of parameters

$$H_f \times W_f \times C \times K$$

$$\text{complexity} \propto C \times K$$

Design guidelines

Guideline 4: *Less computations with filter groups*

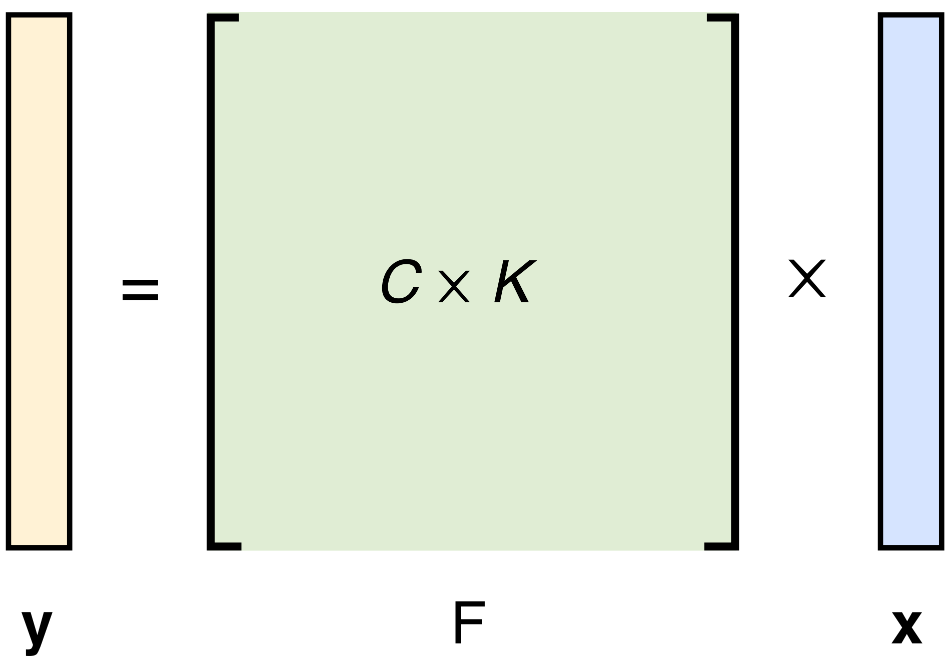


complexity $\propto (C \times K) / G$

Design guidelines

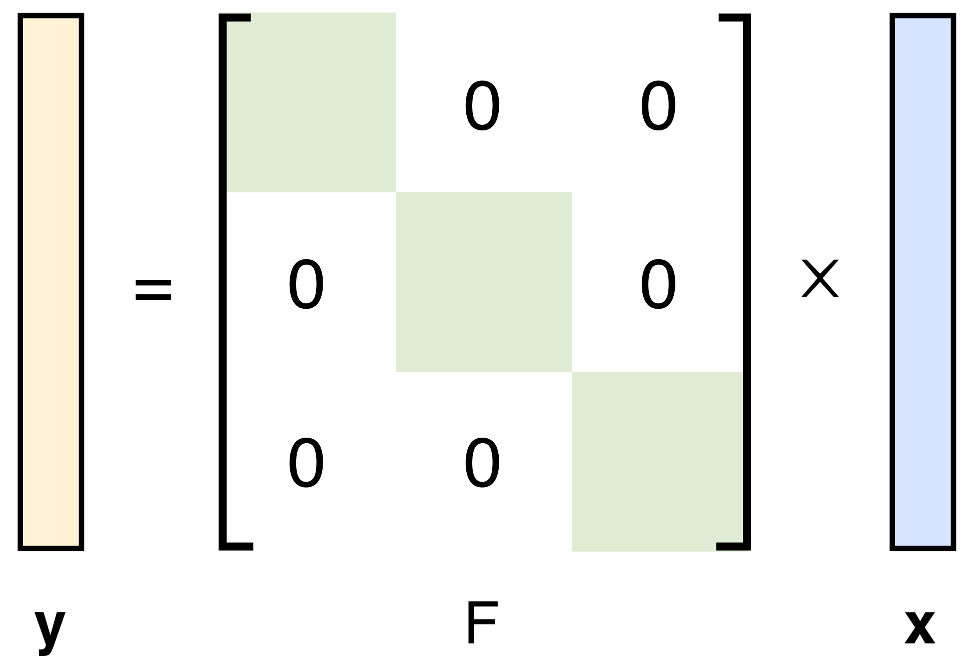
Guideline 4: *Less computations with filter groups*

Full filters



complexity: $C \times K$

Group-sparse filters

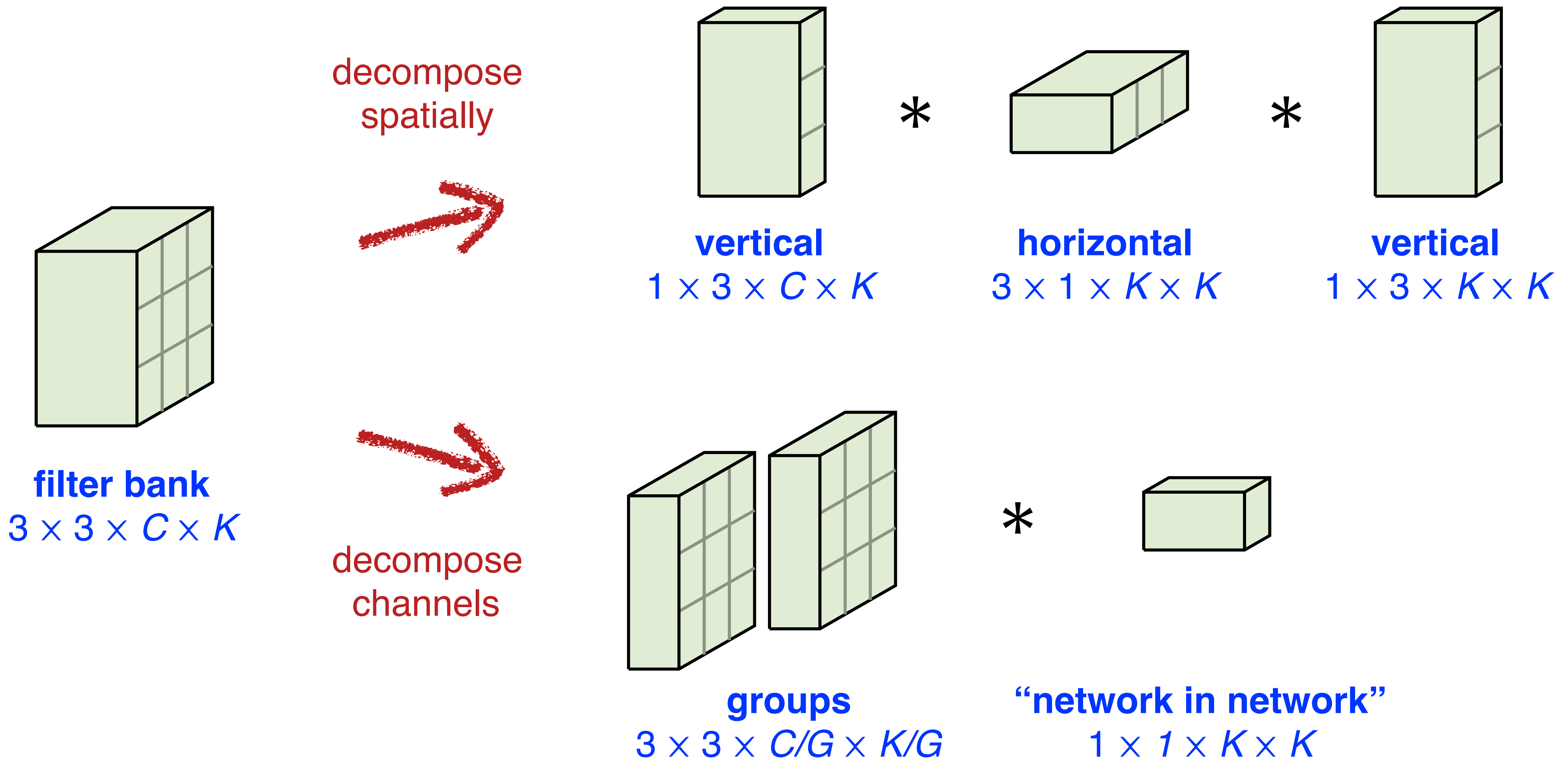


complexity: $C \times K / G$

Groups = filters, seen as a matrix, have a “block” structure

Design guidelines

Guideline 5: *Low-rank decompositions*



Make sure to *mix the information*

Recent advances

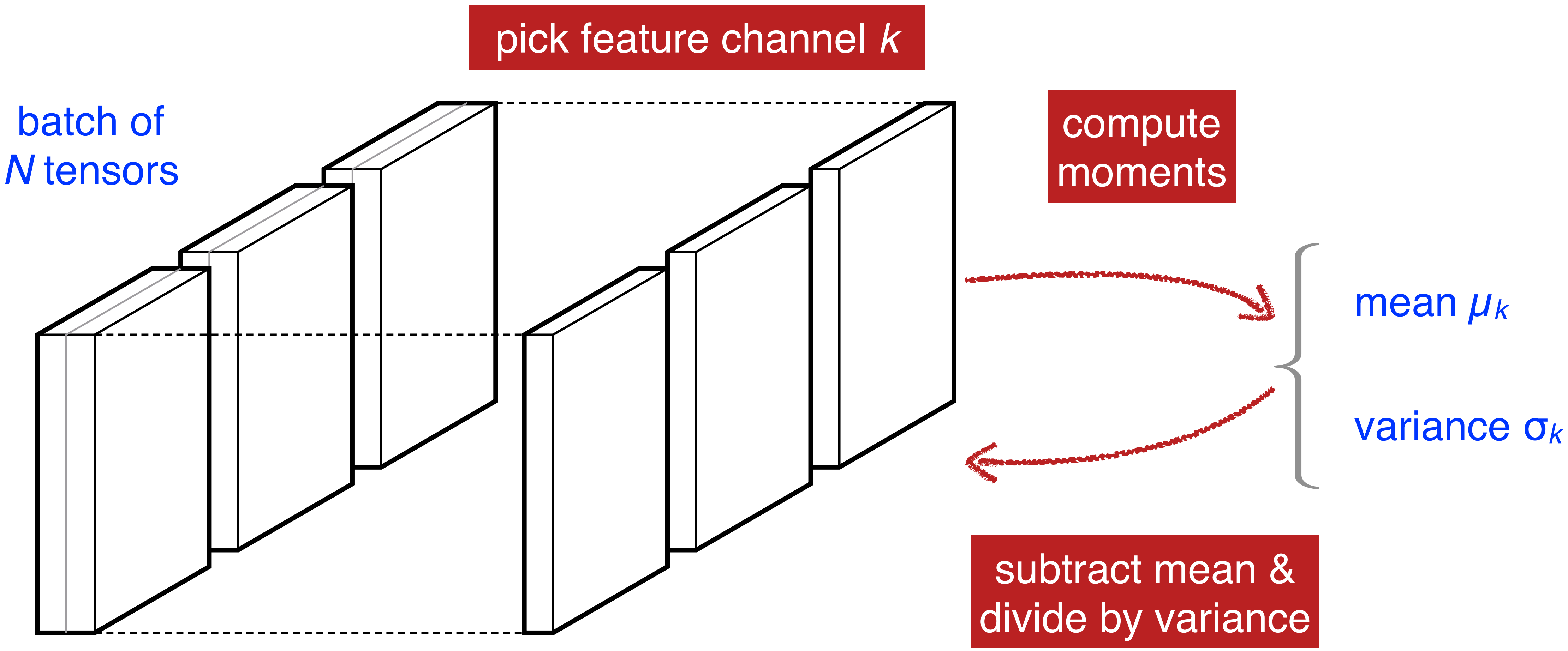
Design guidelines

Batch normalization

Residual learning

Batch normalization

Better condition features



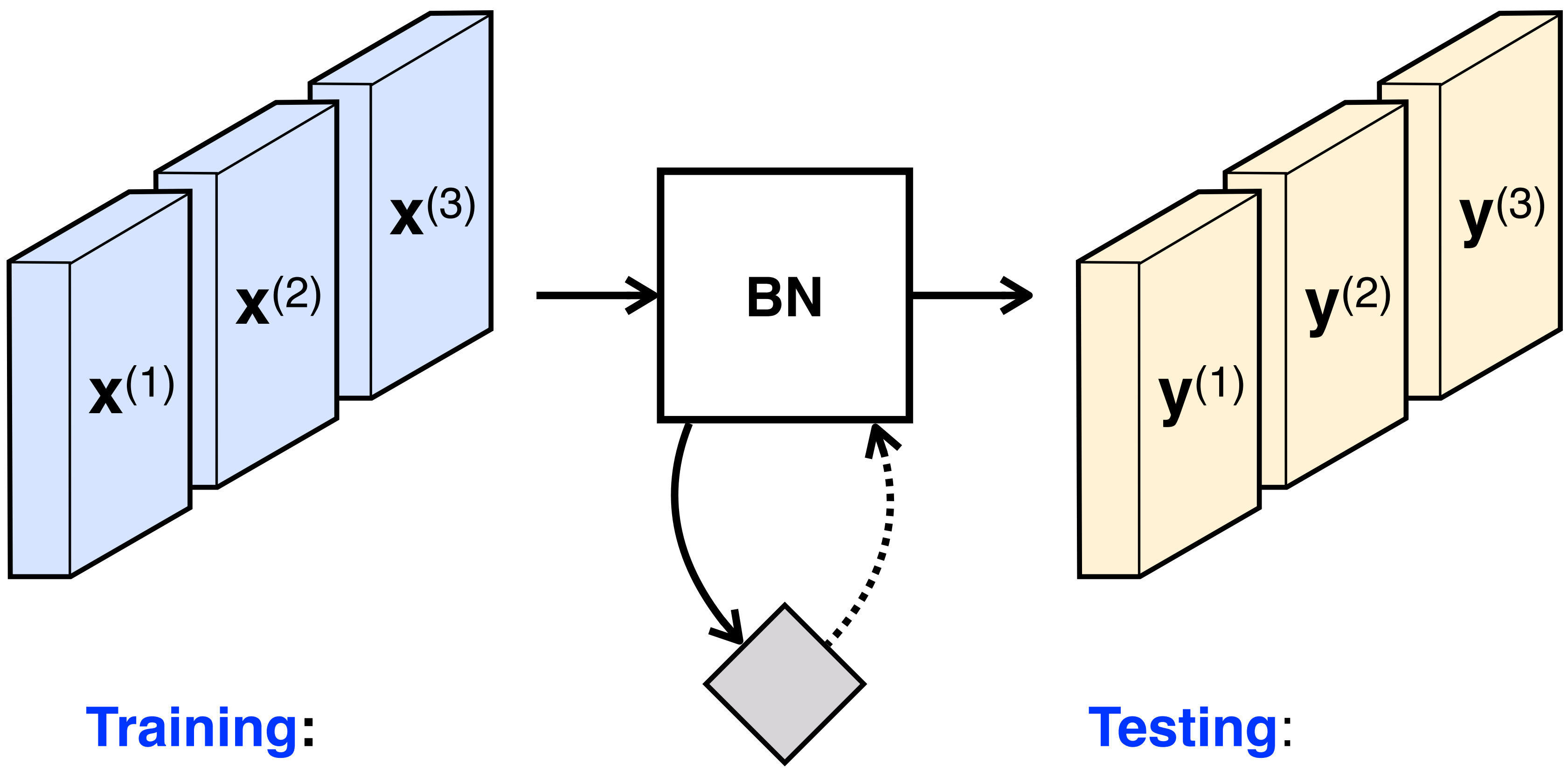
Standardize the response of each feature channel *within the batch*

- Average over spatial locations
- Also, average over multiple images in the batch (e.g. 16-256)

S. Ioffe and C. Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. CoRR, 2015

Batch normalization

Training vs testing modes



Training:
batch-specific
moment averages
are collected

moments
μ, σ

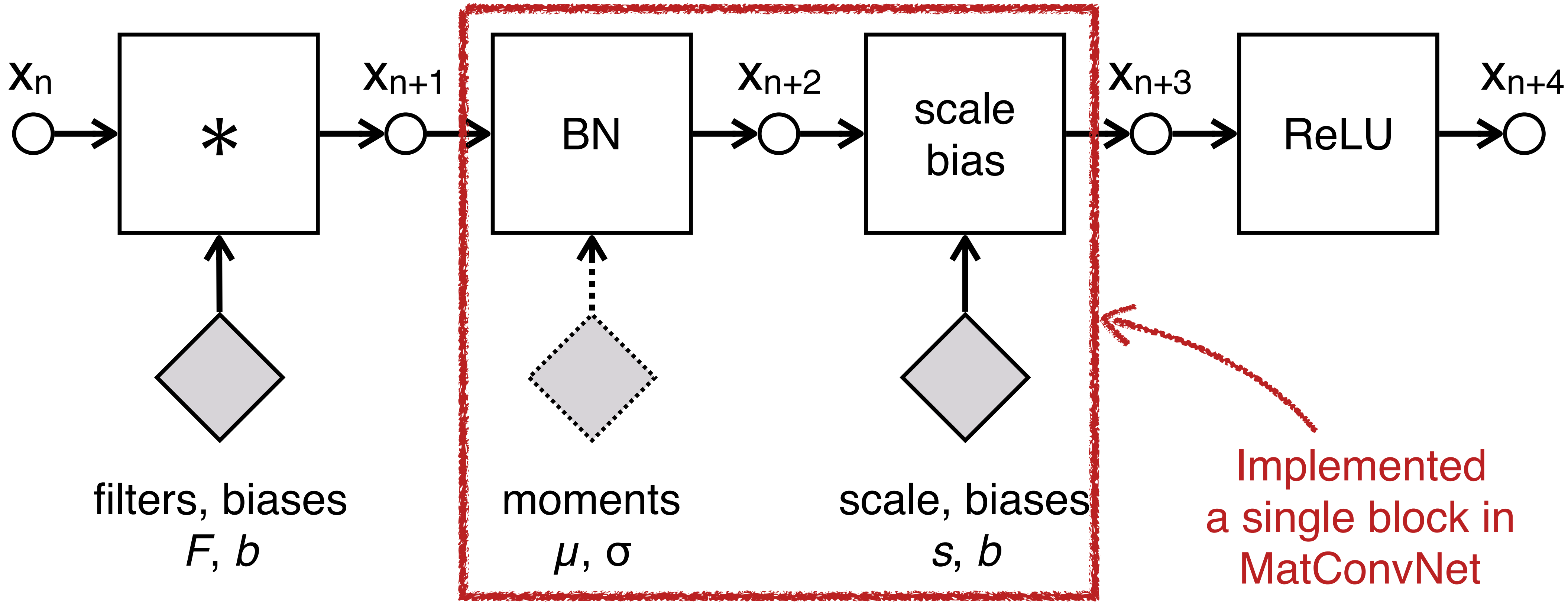
Testing:
moment averages are
used instead
of batch-specific
moments

Moments (mean & variance)

- **Training**: compute anew for each batch
- **Testing**: fixed to their average values

Batch normalization

Utilization



Batch normalization is used after filtering, before ReLU

It is always followed by channel-specific scaling factor *s* and bias *b*

Noisy bias/variance estimation **replaces dropout regularization**

Recent advances

Design guidelines

Batch normalization

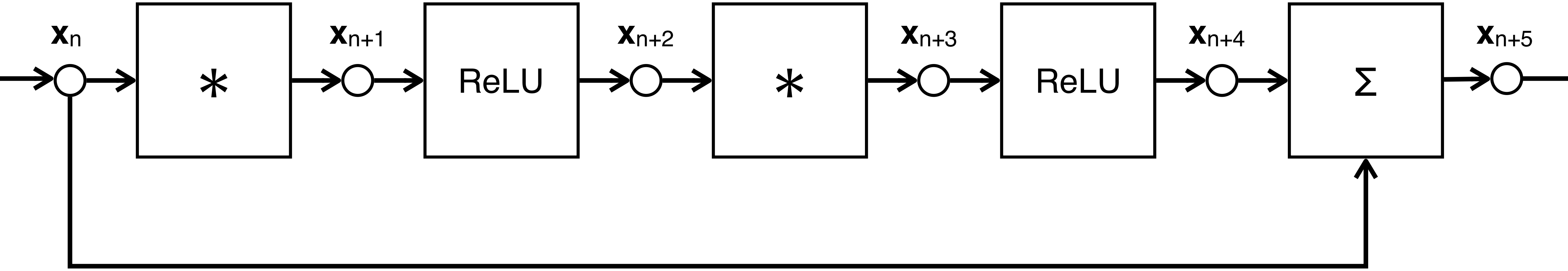
Residual learning

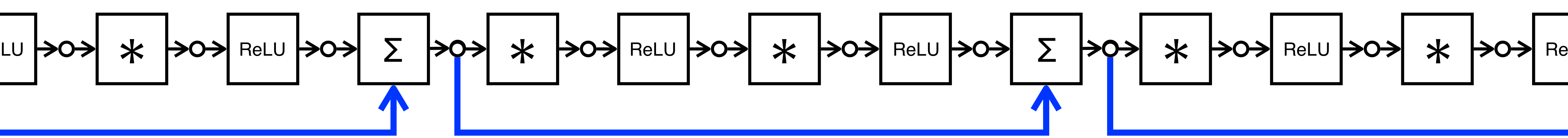
Residual learning

Hardwired identity in parallel with a learned residual transformation

$$\mathbf{x}_{n+5} = \underbrace{\mathbf{x}_n}_{\text{identity}} + \underbrace{(\phi_{\mathrm{ReLU}} \circ \phi_* \circ \phi_{\mathrm{ReLU}} \circ \phi_*)(\mathbf{x}_n)}_{\text{residual}}$$

K. He, X. Zhang, S. Ren, and J. Sun. *Deep residual learning for image recognition*. In Proc. CVPR, 2016.





Deep nets in vision: 2012-2015

Impact of deep learning in vision

- **2012** amazing results by AlexNet in the ImageNet challenge
- **2013-15** massive 3x improvement
- **2016-19** more improvements?

What have we learned

- Several **incremental tweaks** over the base AlexNet
- There is still space for improvements to the base model

Things that work

- Deeper architectures
- Smarter architectures (groups, low rank decompositions, …)
- Batch normalization
- Residual connections

Understanding visual representations

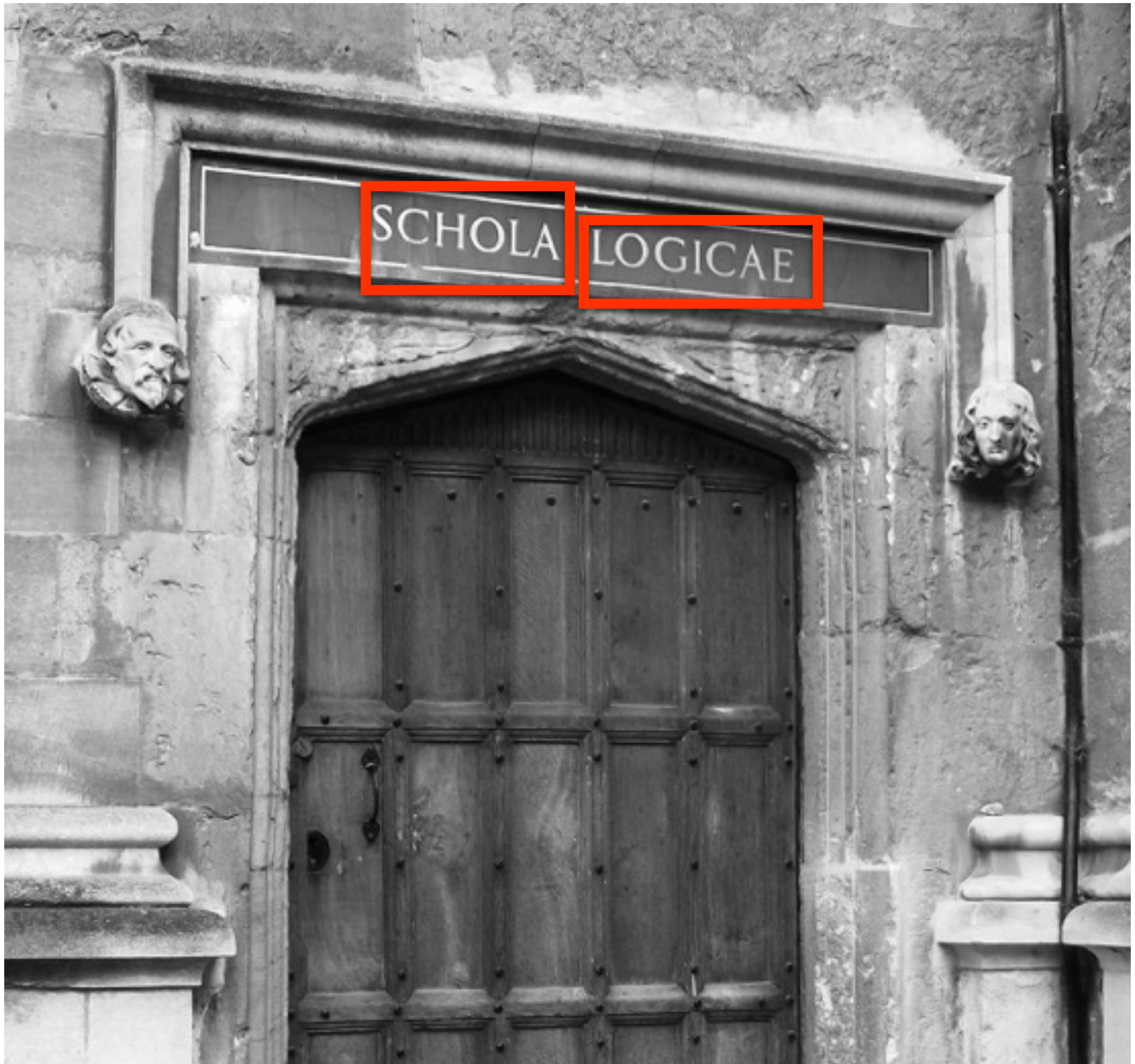
Modern convolutional neural networks

Applications

Segmentation: “fully convolutional” networks

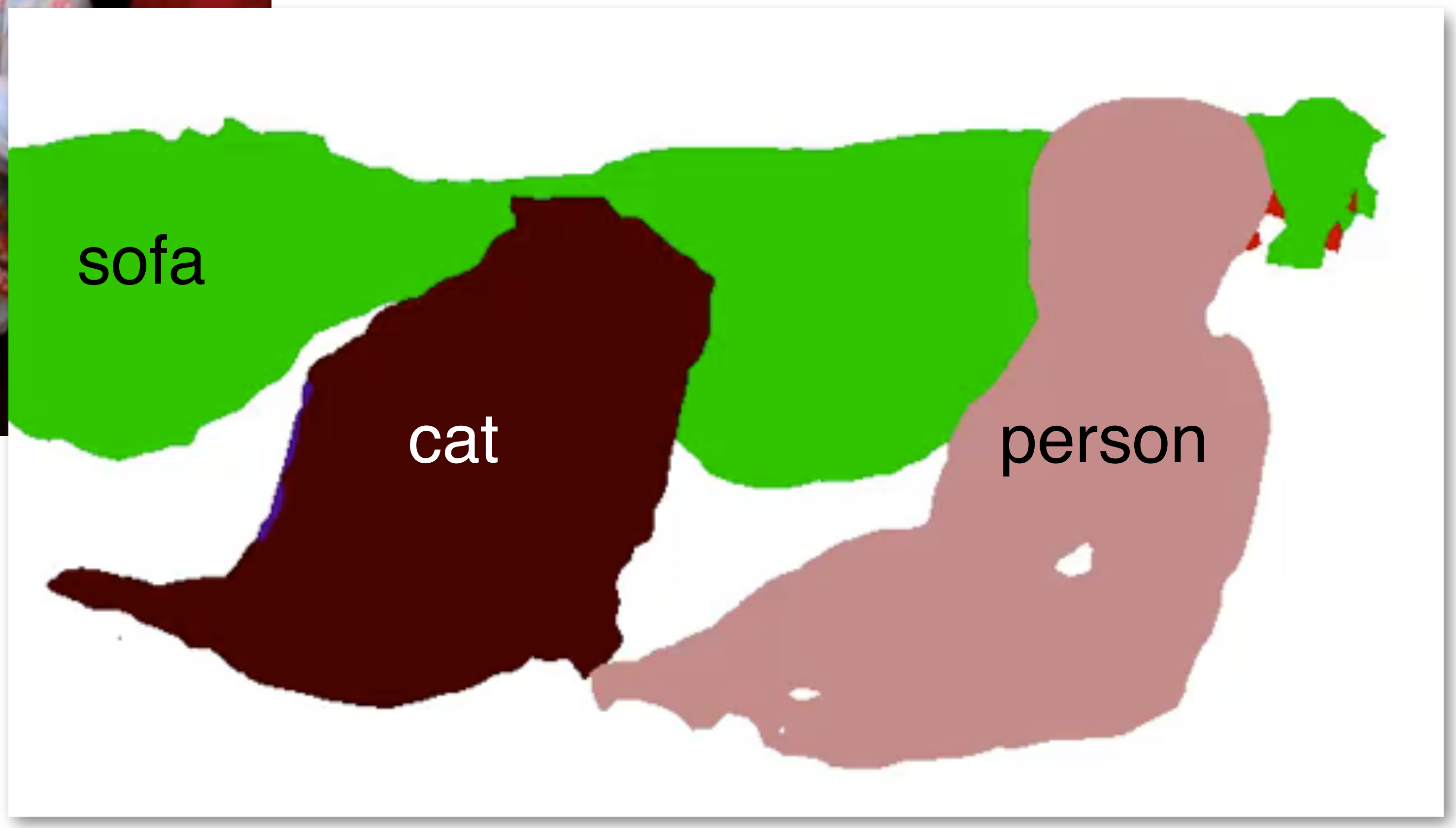
Object detection: R-CNN and weak supervision

Applications



Semantic image segmentation

Label individual pixels



Face analysis

Detection, verification, recognition, emotion, 3D fitting

E.g. VGG-Face

Text spotting

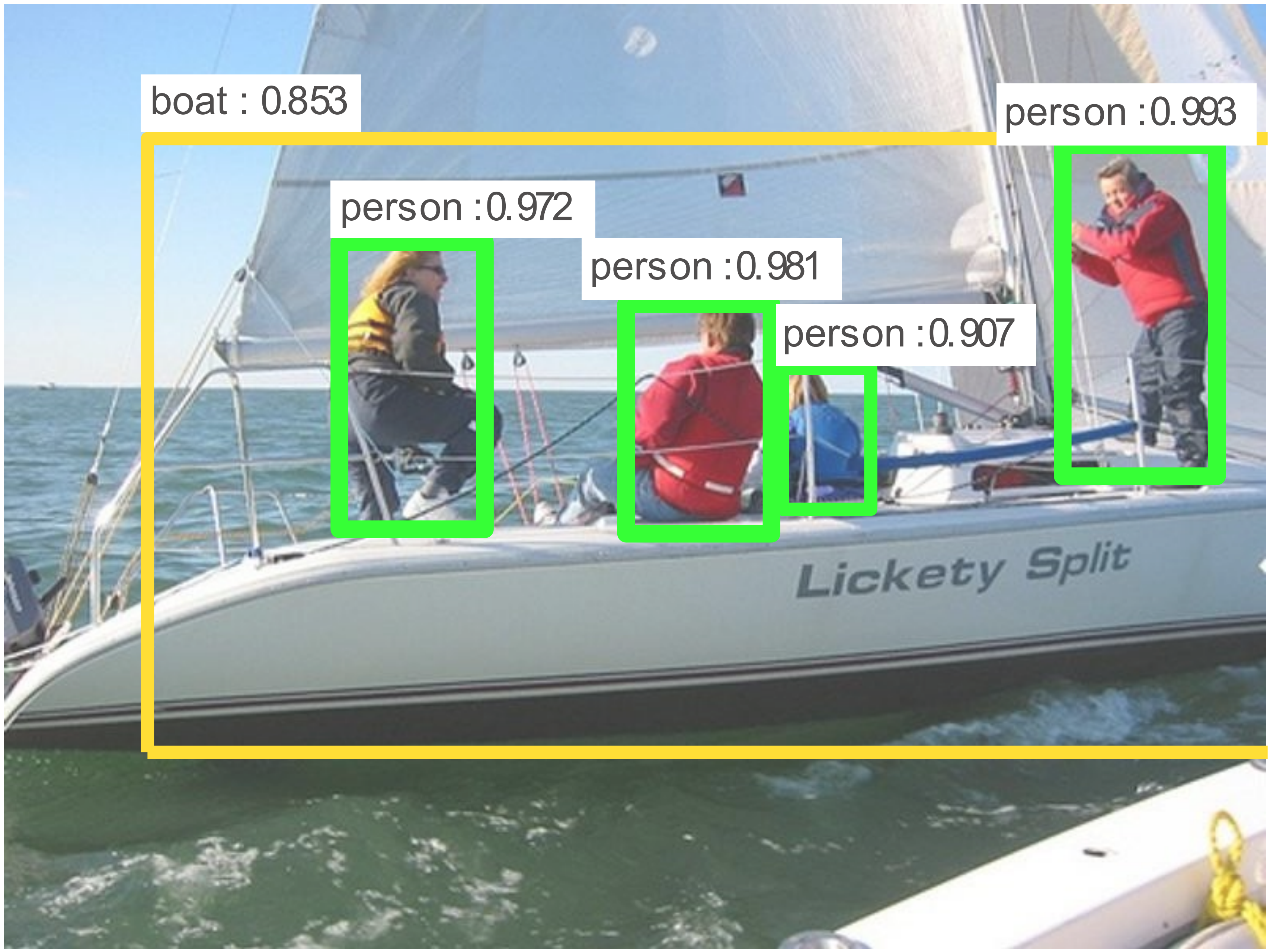
Detection, word recognition, character recognition

E.g. SynthText and VGG-Text

http://zeus.robots.ox.ac.uk/textsearch/#/search/

Object detection

Extract individual object instances



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

Understanding visual representations

Modern convolutional neural networks

Applications

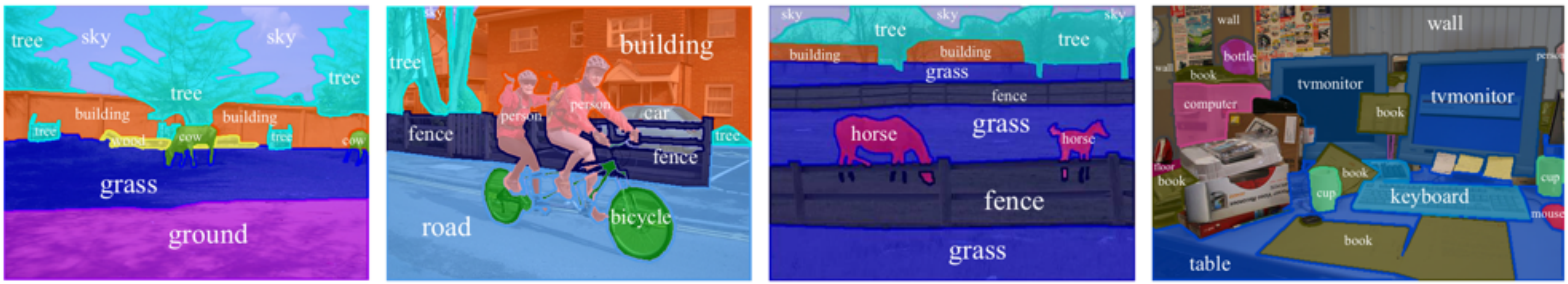
Segmentation: “fully convolutional” networks

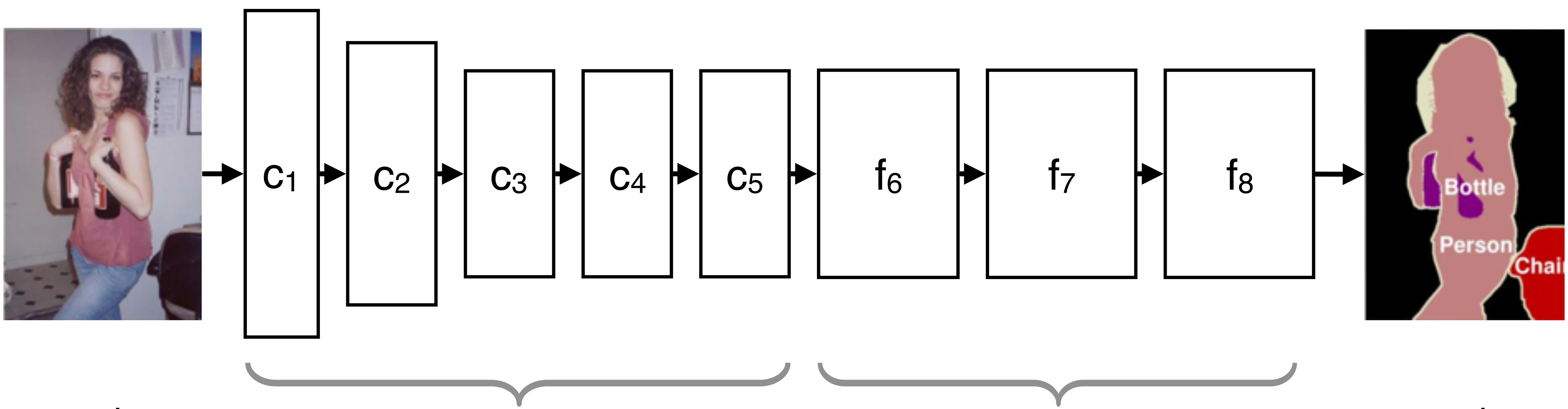
Object detection: R-CNN and weak supervision

Semantic segmentation

Semantic image segmentation

Label individual pixels





input = image

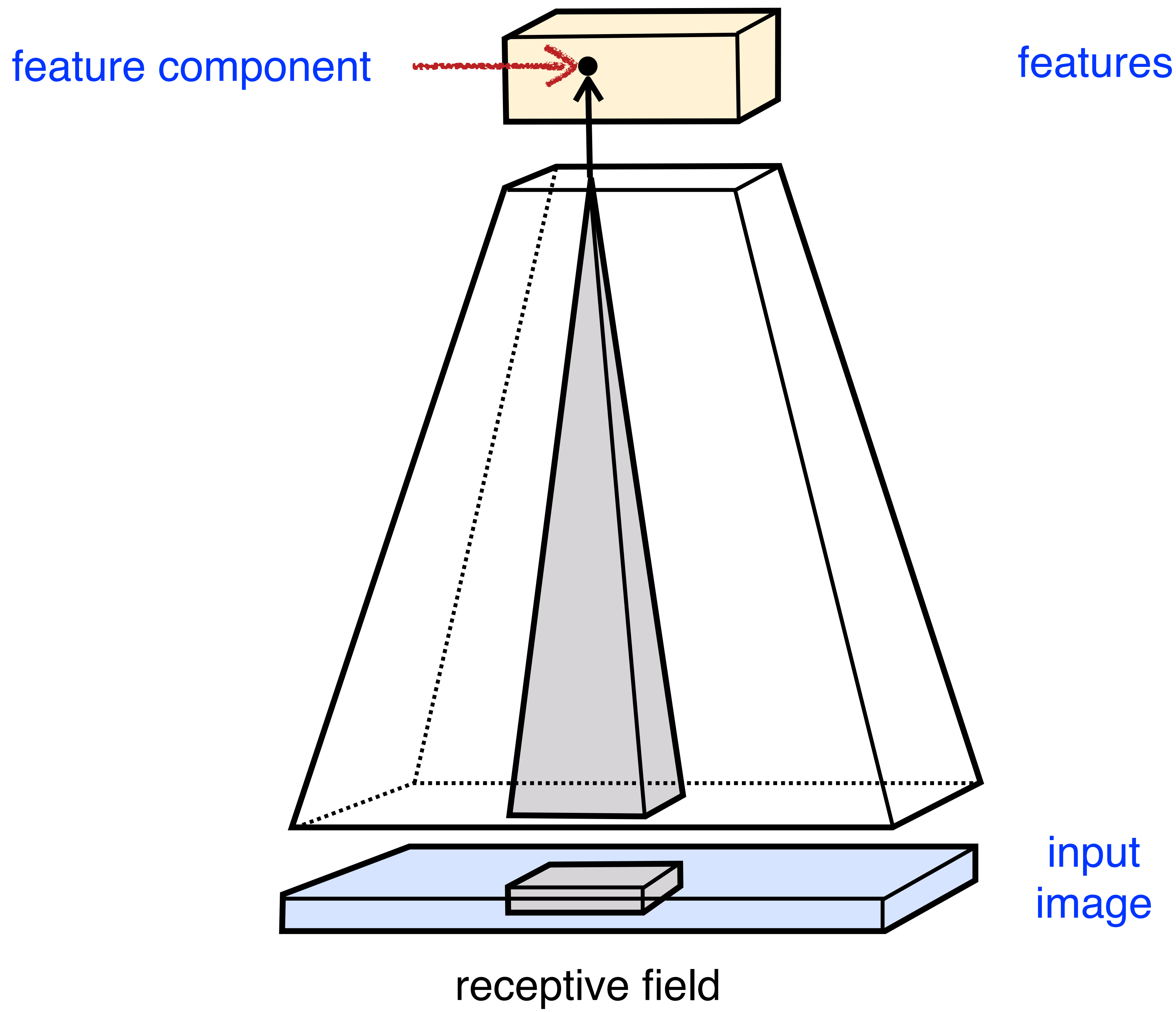
convolutional

fully-connected

output = image

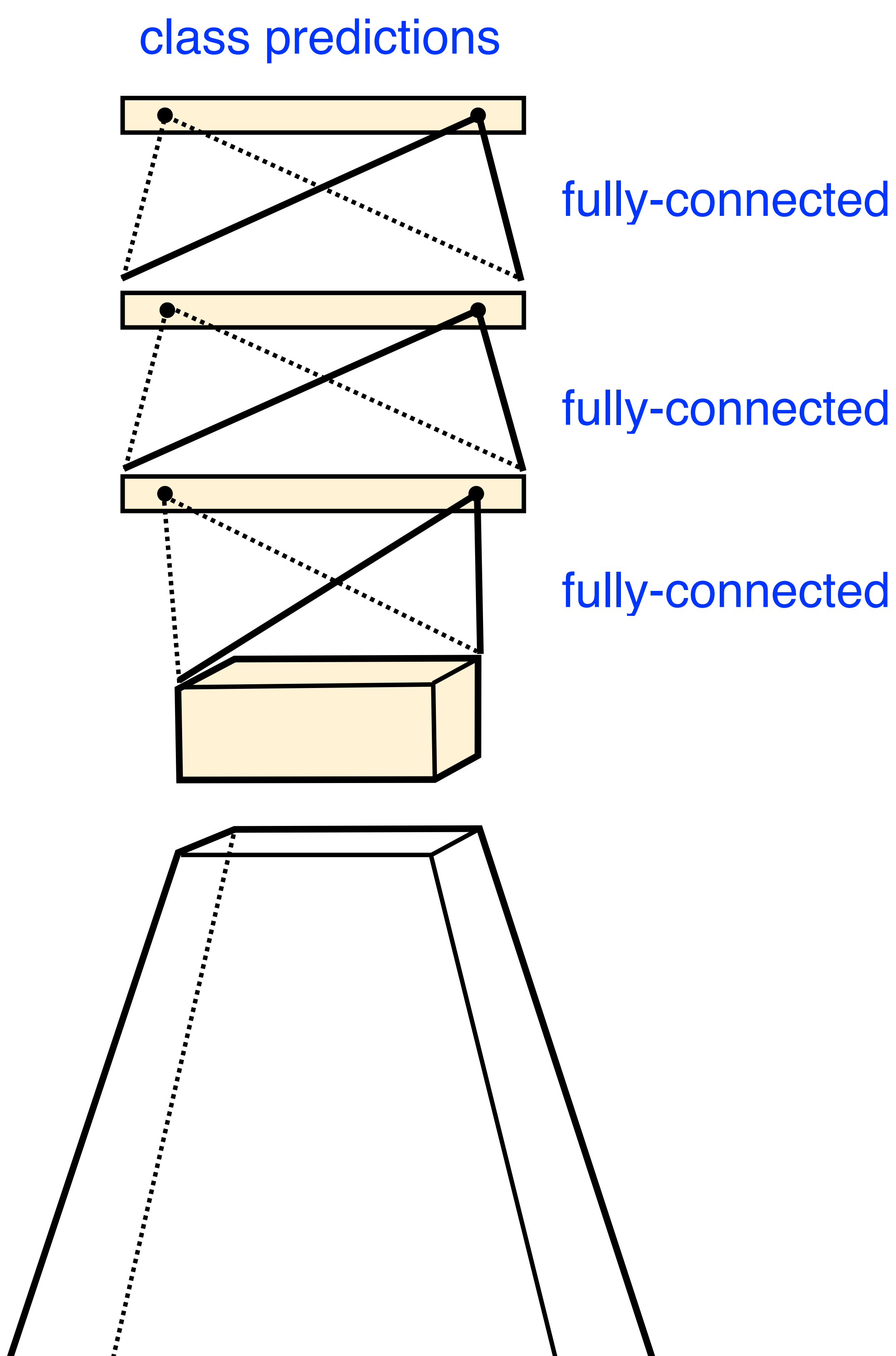
Convolutional layers

Local receptive field



Fully connected layers

Global receptive field



Convolutional vs fully connected

Comparing the receptive fields

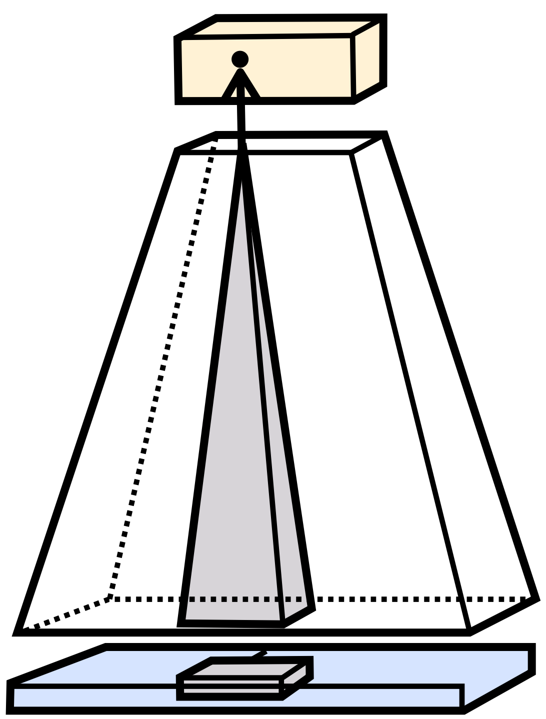
Downsampling filters

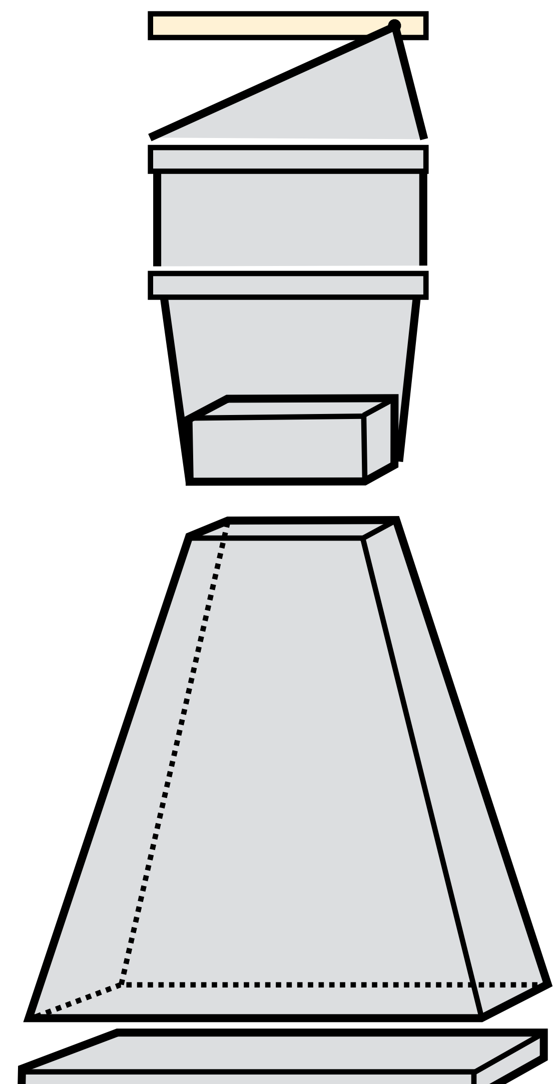
Responses are spatially selective, can be used to localize things.

Upsampling filters

Responses are global, do not characterize well position.

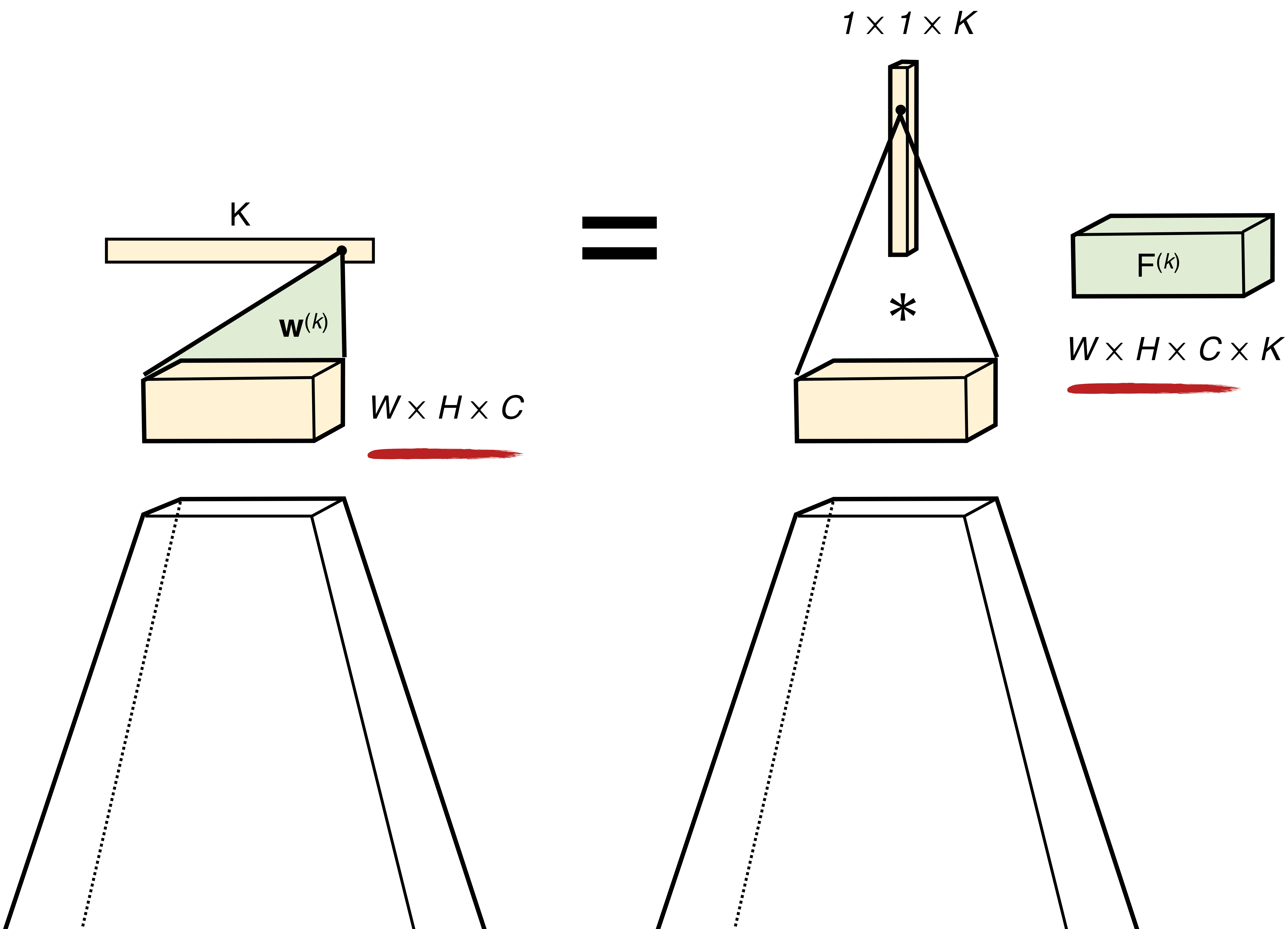
Which one is more useful for pixel level labelling?



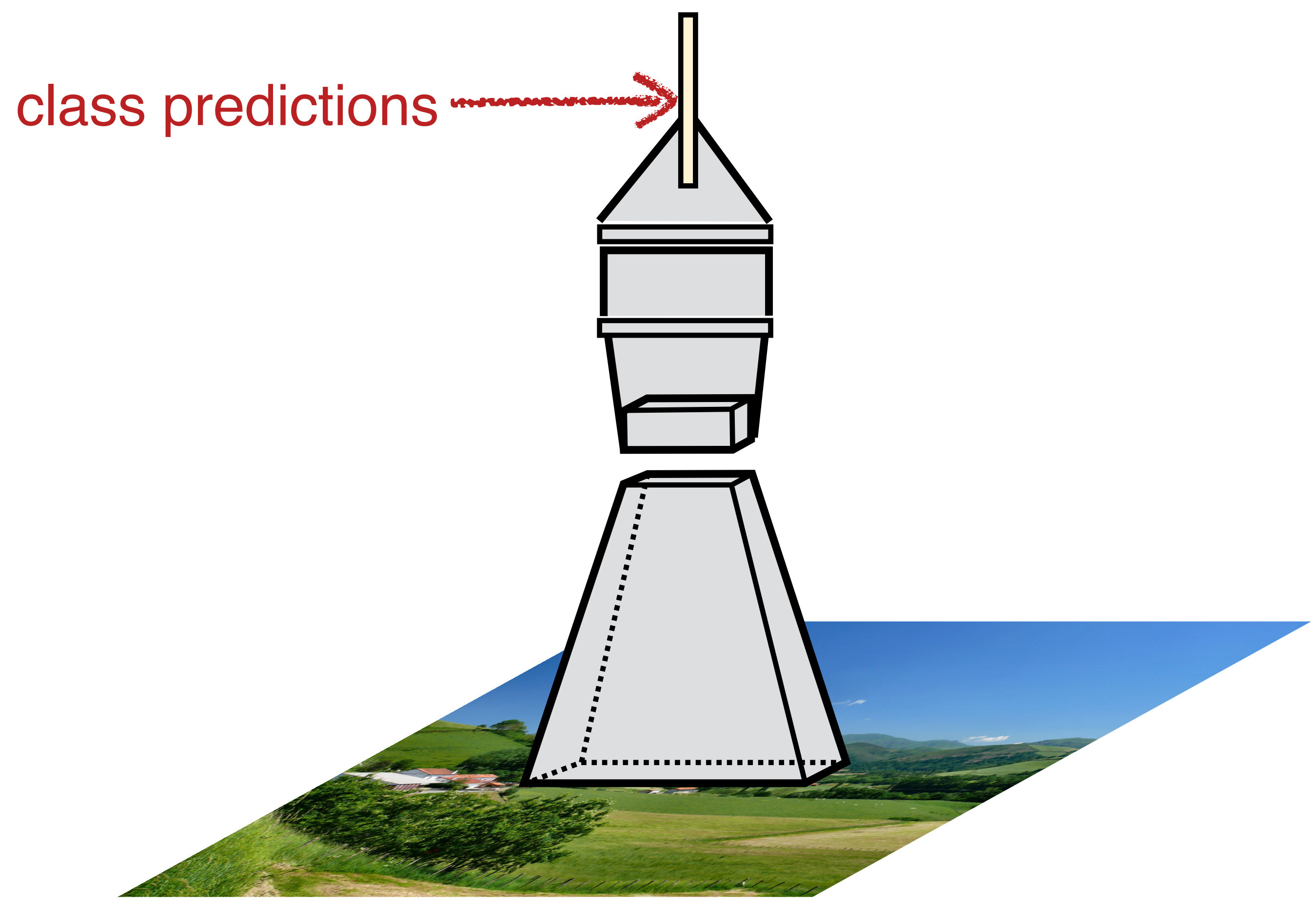


Fully-connected = very large filter

"FC" is just a name for a particular filter configuration

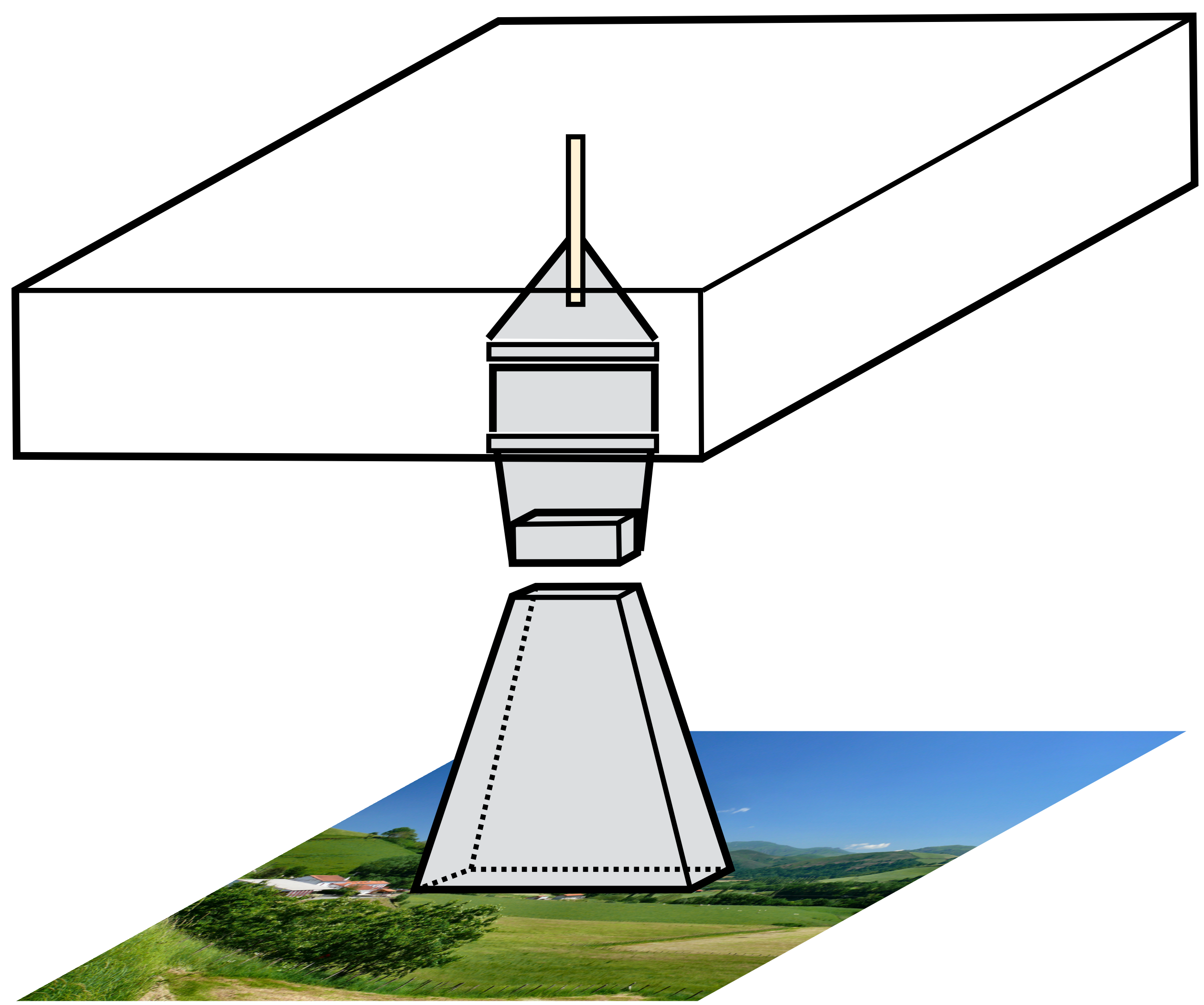


Fully-convolutional neural networks



J. Long, E. Shelhamer, and T. Darrell. *Fully convolutional models for semantic segmentation.* In Proc. CVPR, 2015

Fully-convolutional neural networks



Dense evaluation

- Apply the whole network convolutional
- Estimates a vector of class probabilities at each pixel

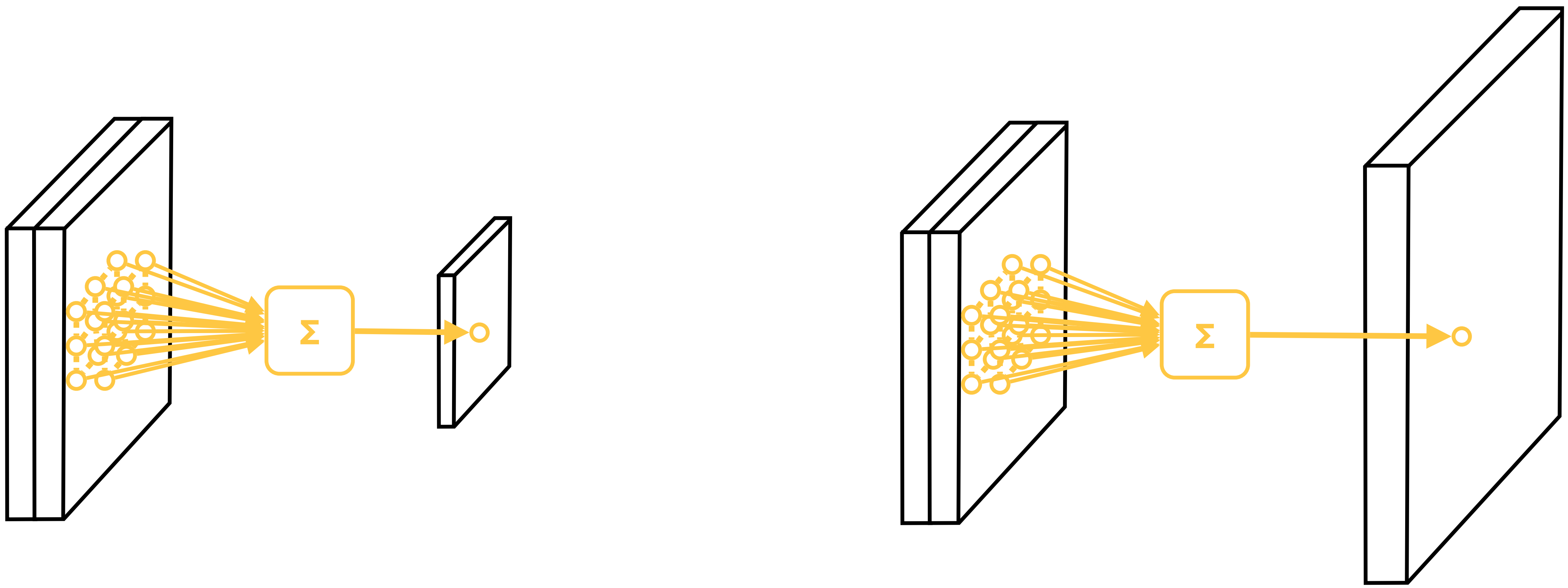
Downsampling

- In practice most network downsample the data fast
- The output is very low resolution (e.g. 1/32 of original)

Upsampling the result

Interpolating filter

Downsampling filters | **Upsampling filters**



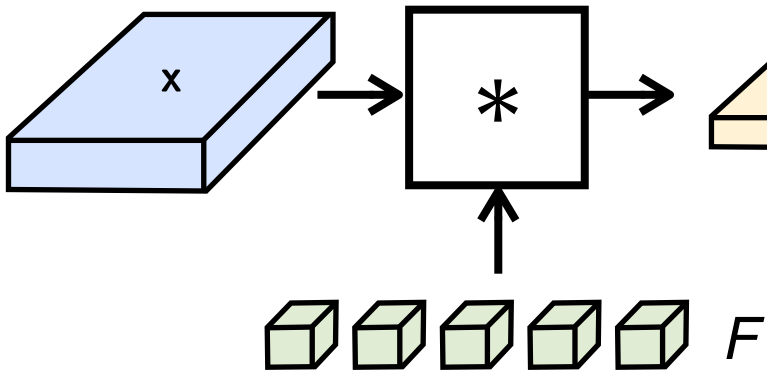
Upsampling filters allow to increase the resolution of the output

Very useful to get full-resolution segmentation results

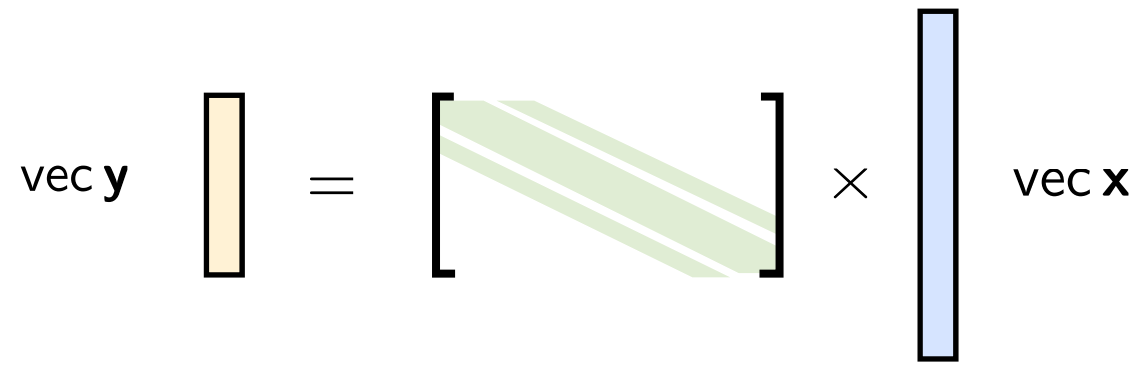
Deconvolution layer

Or convolution transpose

Convolution

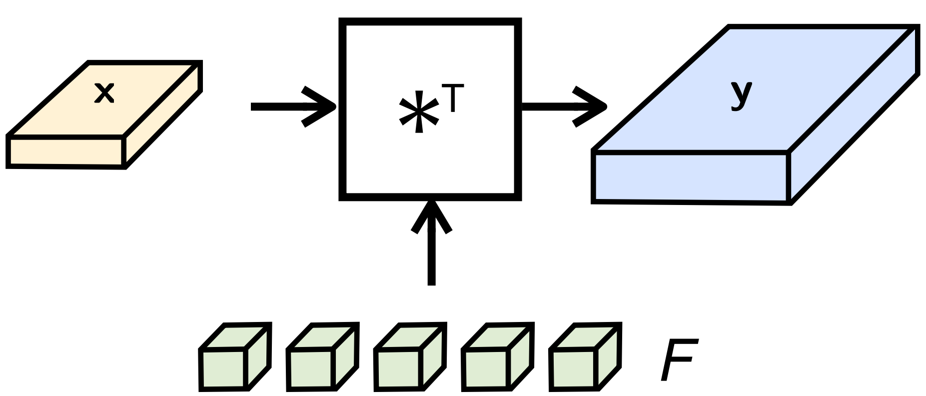


As matrix multiplication



Banded matrix equivalent to *F*

Convolution transpose

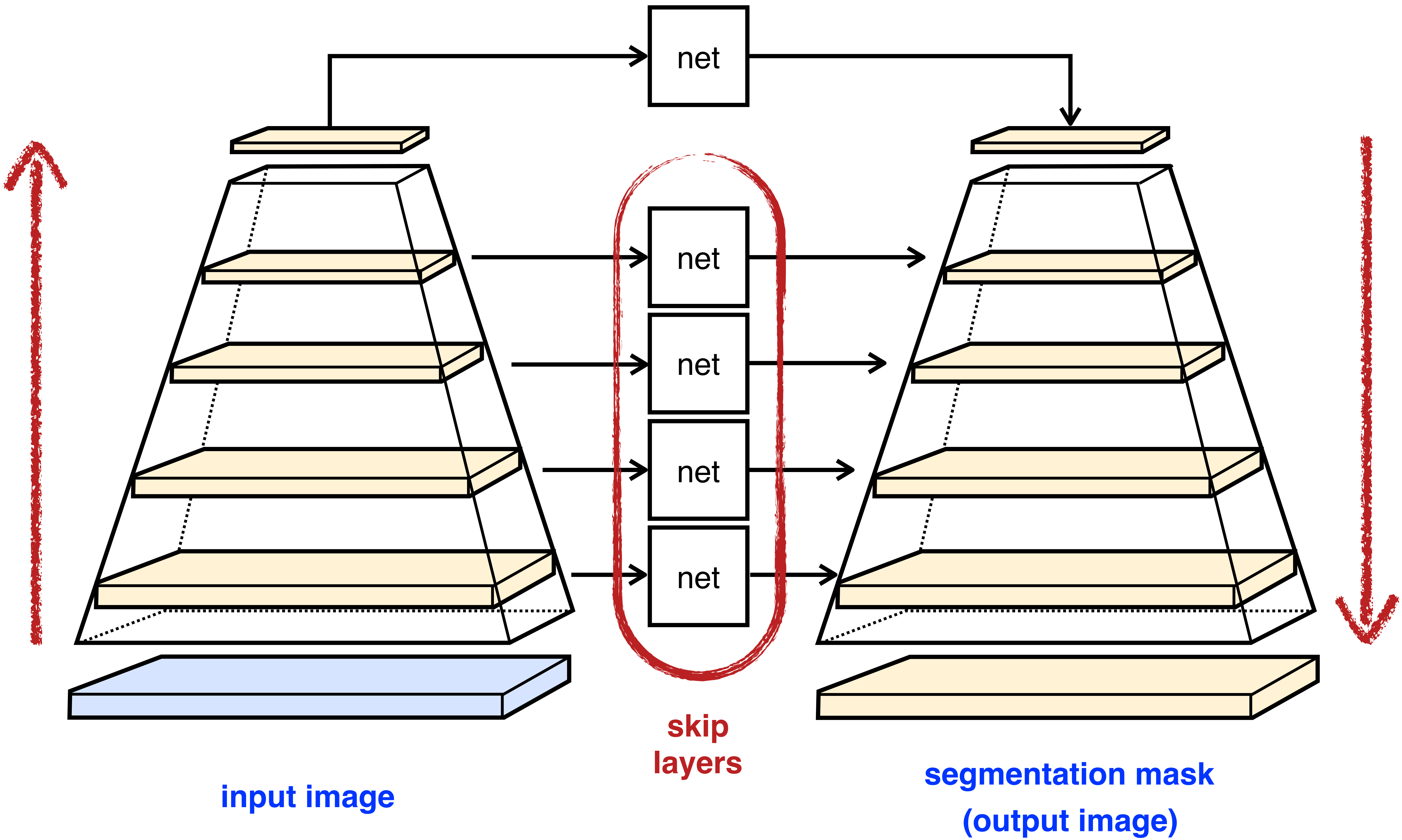


Transposed

Transposed matrix

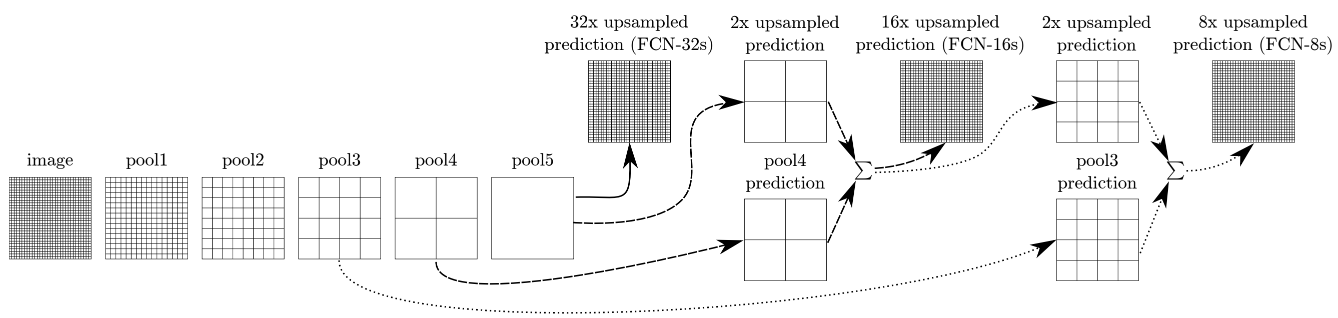
U-architectures

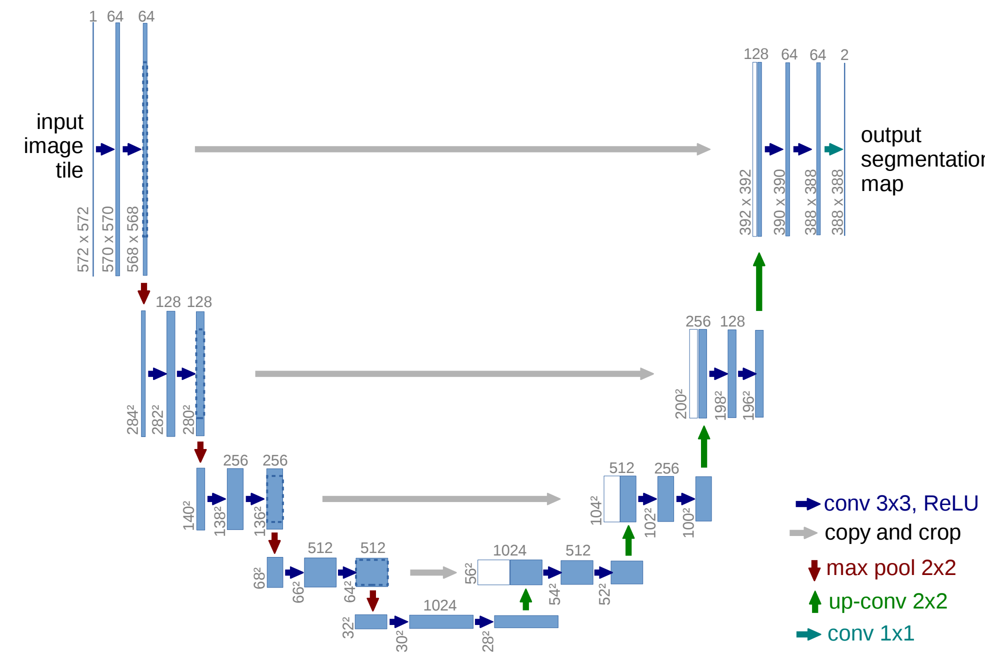
From image to image

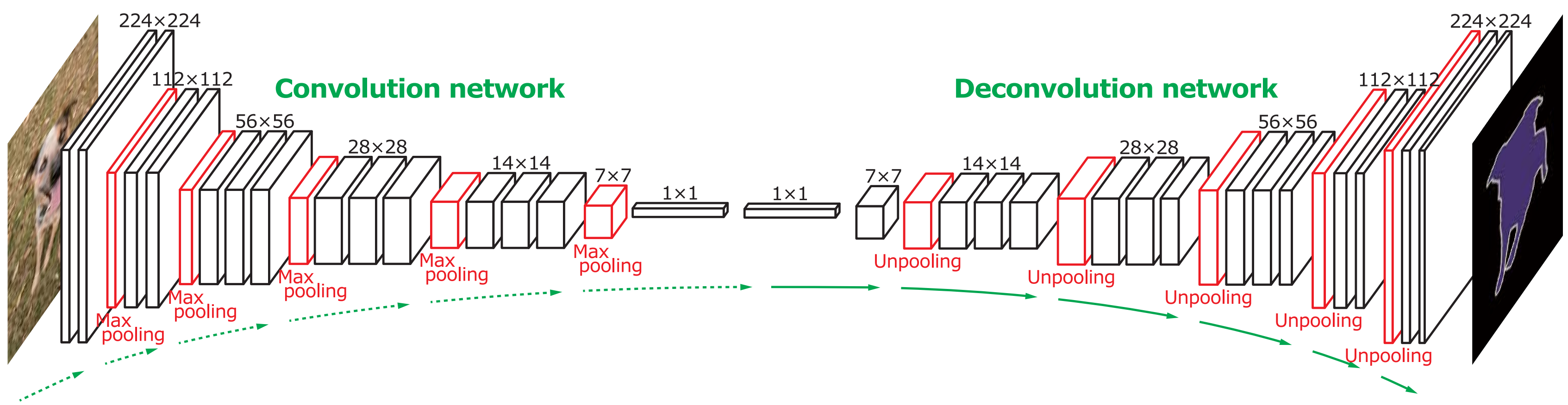


U-architectures

Several variants: FCN, U-arch, deconvolution, …







J. Long, E. Shelhamer, and T. Darrell. *Fully convolutional models for semantic segmentation.* In Proc. CVPR, 2015
H. Noh, S. Hong, and B. Han. *Learning deconvolution network for semantic segmentation.* In Proc. ICCV, 2015
O. Ronneberger, P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation.* In Proc. MICCAI, 2015

Understanding visual representations

Modern convolutional neural networks

Applications

Segmentation: “fully convolutional” networks

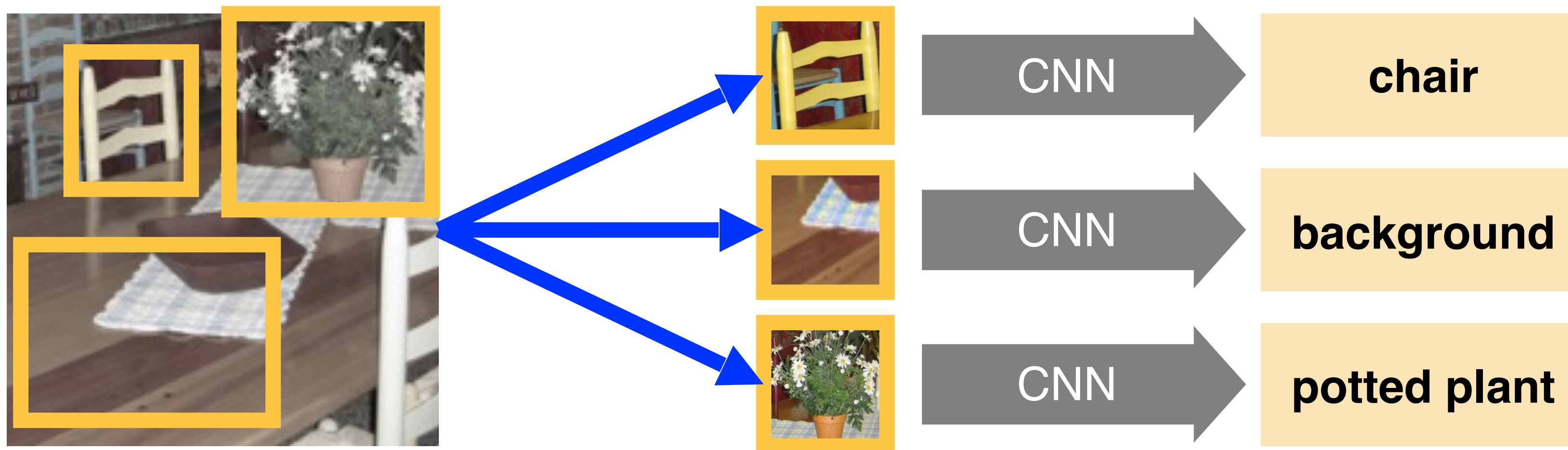
Object detection: R-CNN and weak supervision

Object detection

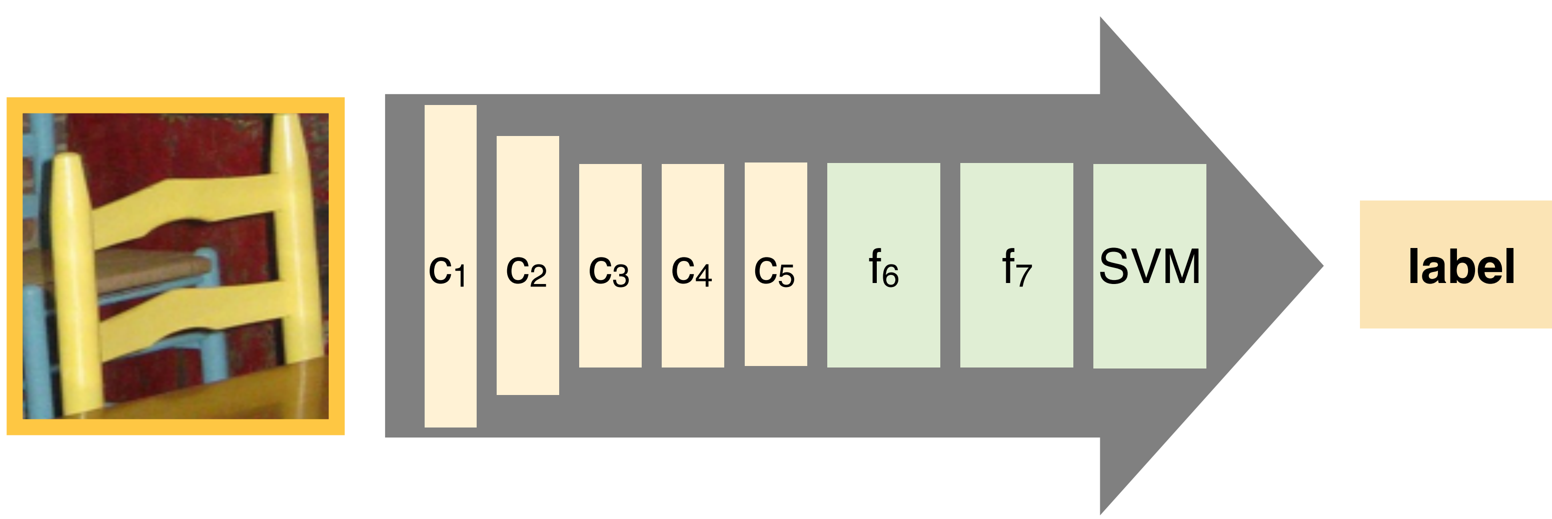
Detections with conv nets

Region-based Convolutional Neural Network (R-CNN)

Pros: simple and effective



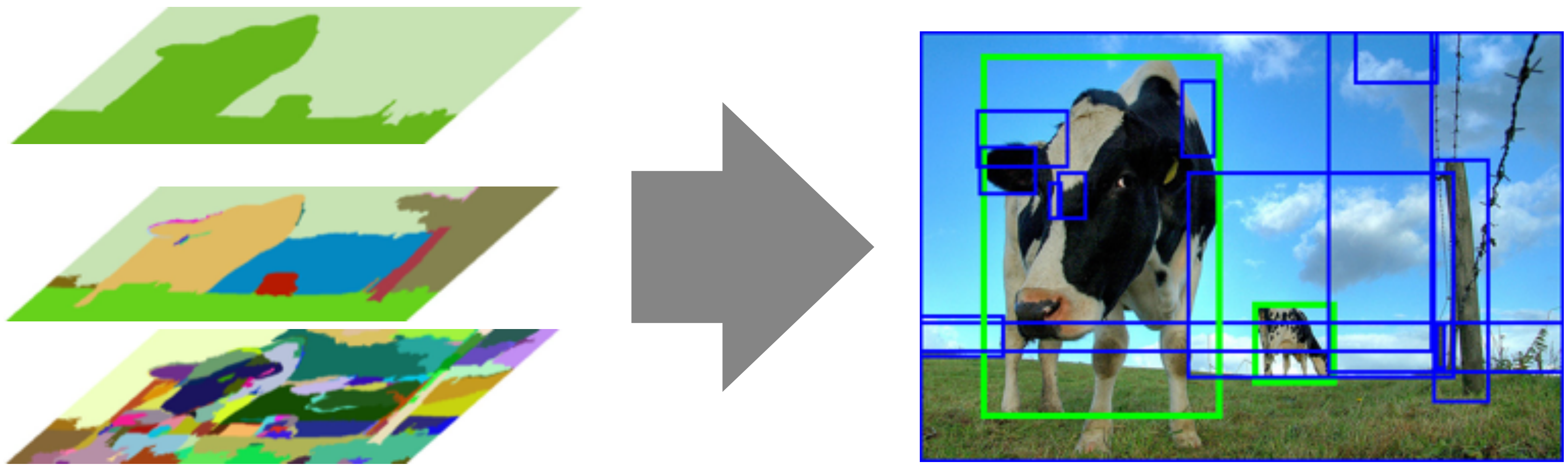
Cons: slow as the CNN is re-evaluated for each tested region



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

Region proposals

Cut down the number of candidates



Proposal-method: Selective Search [van de Sande, Uijlings et al.]

- hierarchical segmentation
- each region generates a ROI
- ~ 2000 regions / image

From proposals to CNN features

Dilate, crop, reshape

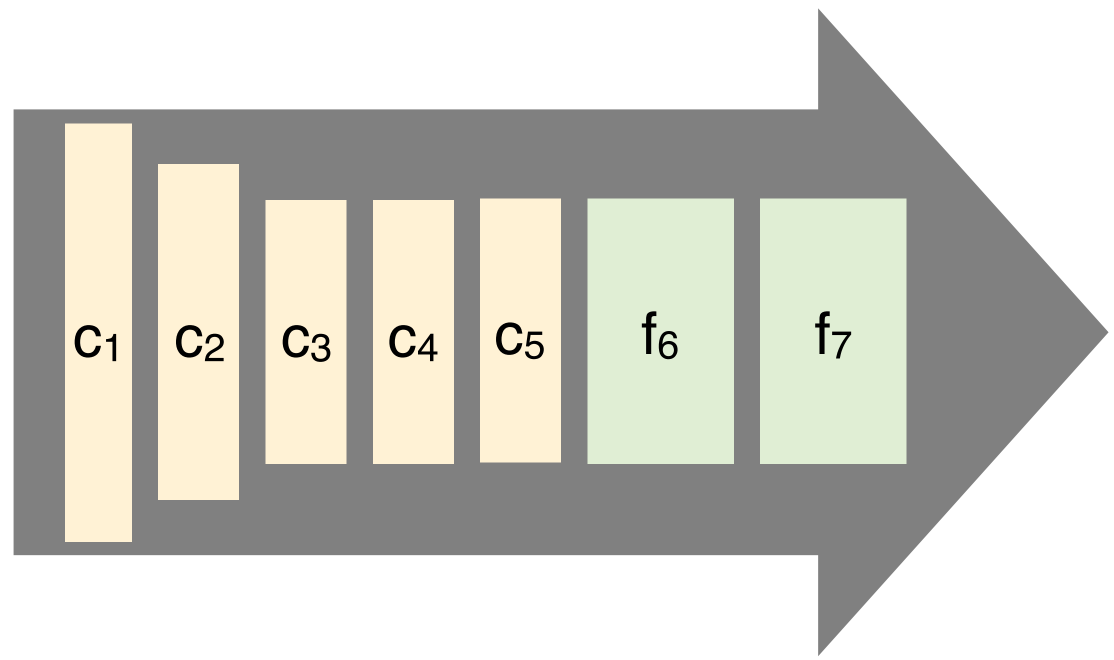
Propose

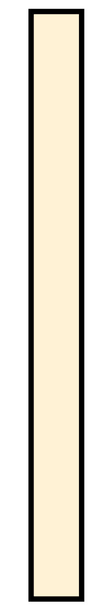
Dilate

Crop & scale
Anisotropic
227 x 227

From proposals to CNN features

Evaluate CNN





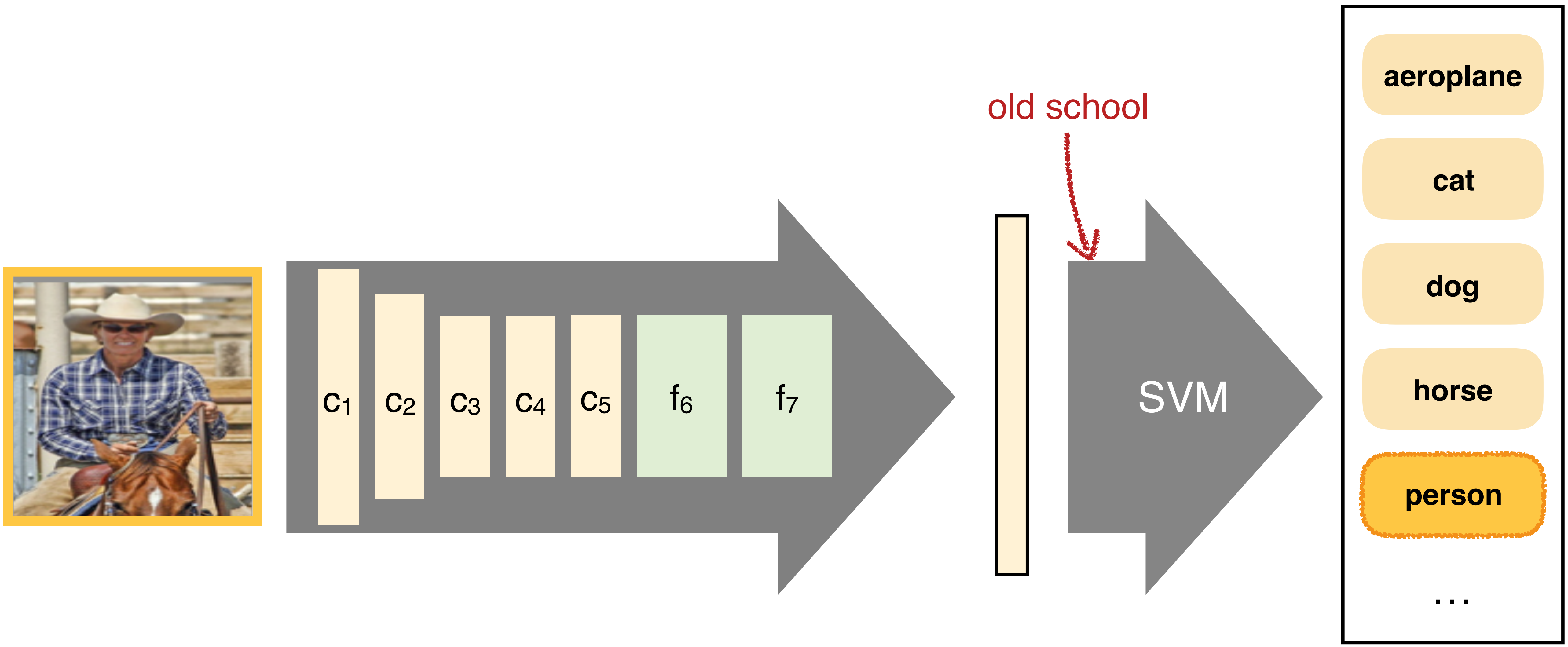
Scale
Anisotropic
227 x 227

CNN features
Up to FC-7
AlexNet

Feature vector
4096 D

Classification of a region

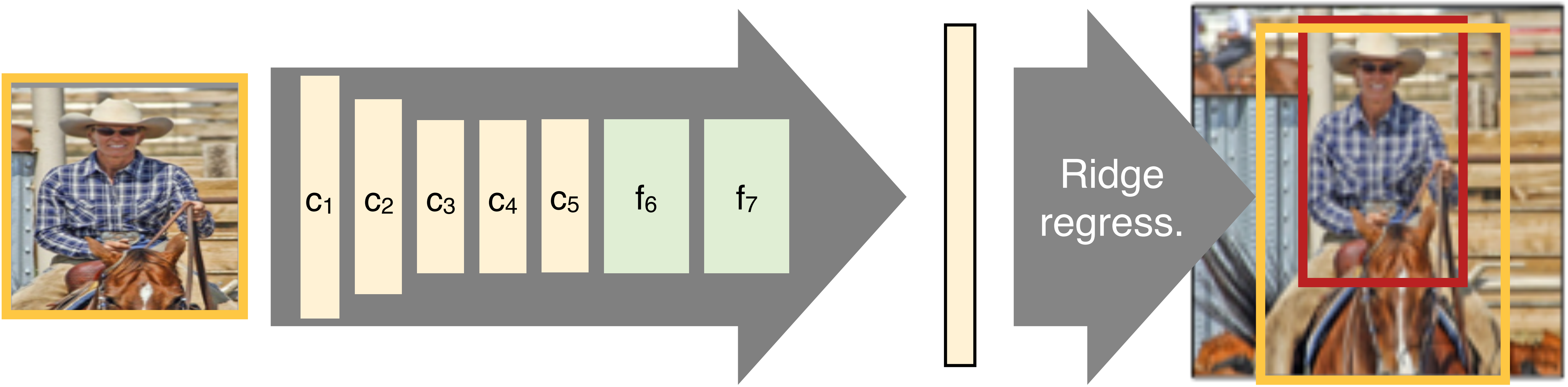
Run an SVM or similar on top



Scale	**CNN features**	**Feature vector**	**Label**
Anisotropic 227 x 227	Up to FC-7 AlexNet	4096 D	One out of *N*

Region adjustment

Bounding-box regression



Scale
Anisotropic
227 x 227

CNN features
Up to FC-7
AlexNet

Feature vector
4096 D

Box adjustment
dx1, dx2, dy1, dy2

Training: what is a positive or negative box?

Based on overlap with ground truth

treat as **positive**

overlap > 70%

treat as **negative**

overlap < 30%

Ren, He, Girshick, & Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

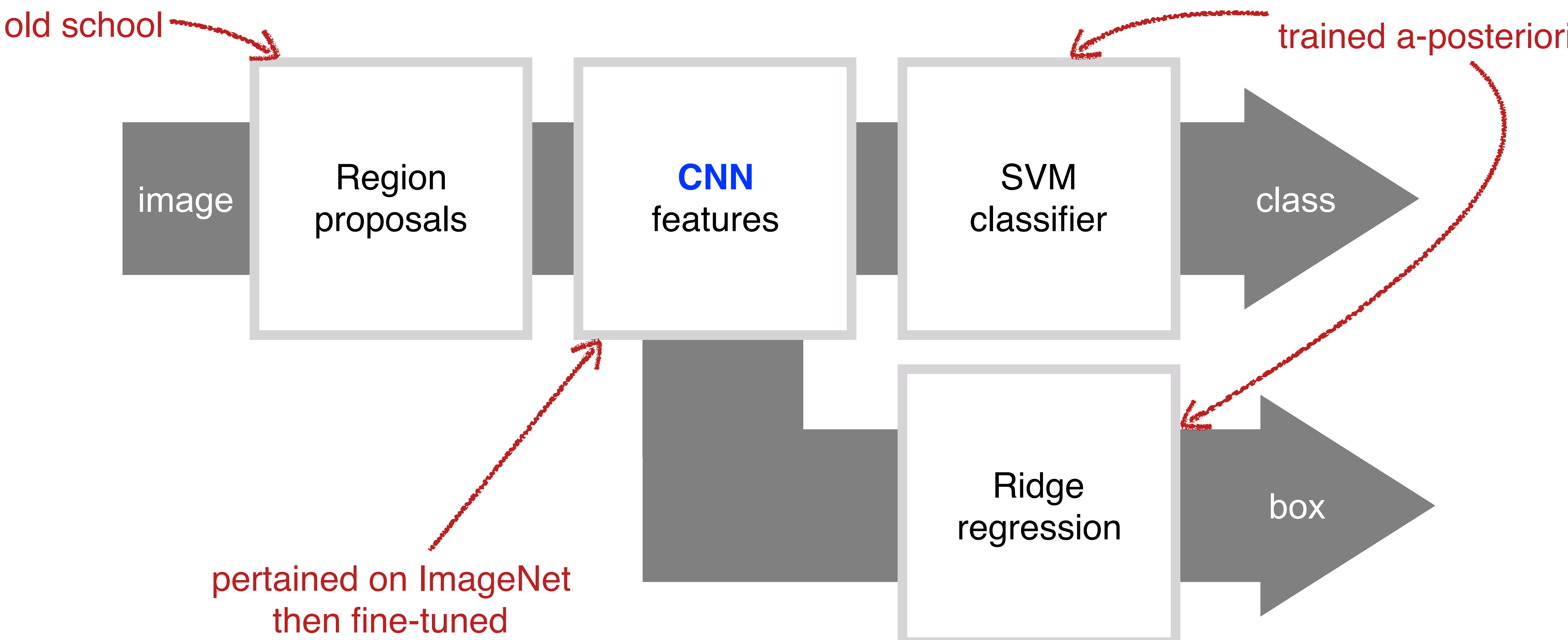
R-CNN results on PASCAL VOC

At the time of introduction (2013)

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2013)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%
SegDPM (Fidler et al. 2013)		40.4%
R-CNN (TorontoNet)	54.2%	50.2%
R-CNN (TorontoNet) + bbox regression	58.5%	53.7%
R-CNN (VGG-VD)	62.1%	
R-CNN (ONet) + bbox regression	66.0%	62.9%

R-CNN summary

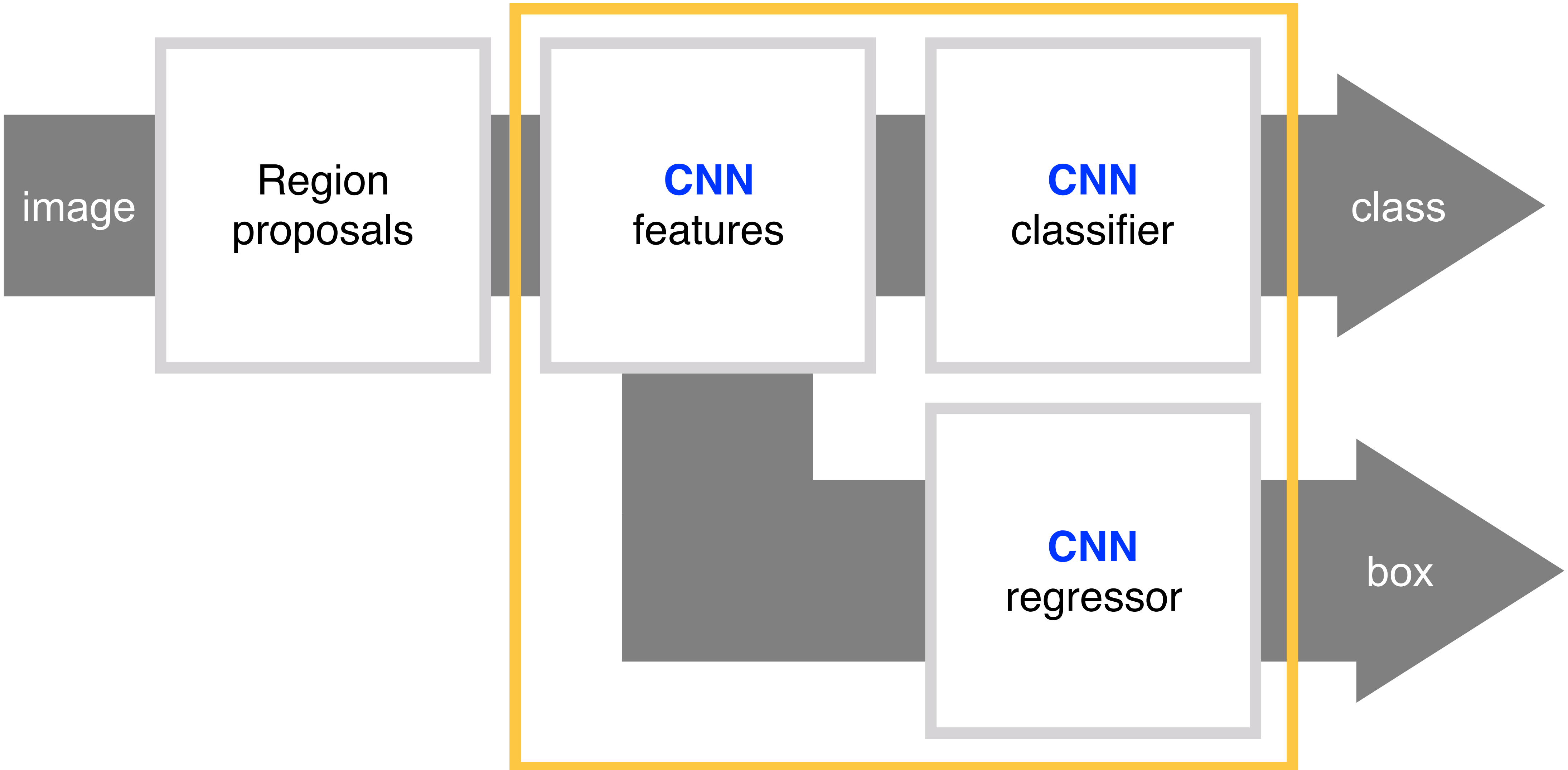
Region-based Convolutional Neural Network



Can we achieve end-to-end training?

Towards better R-CNNs

Region-based Convolutional Neural Network

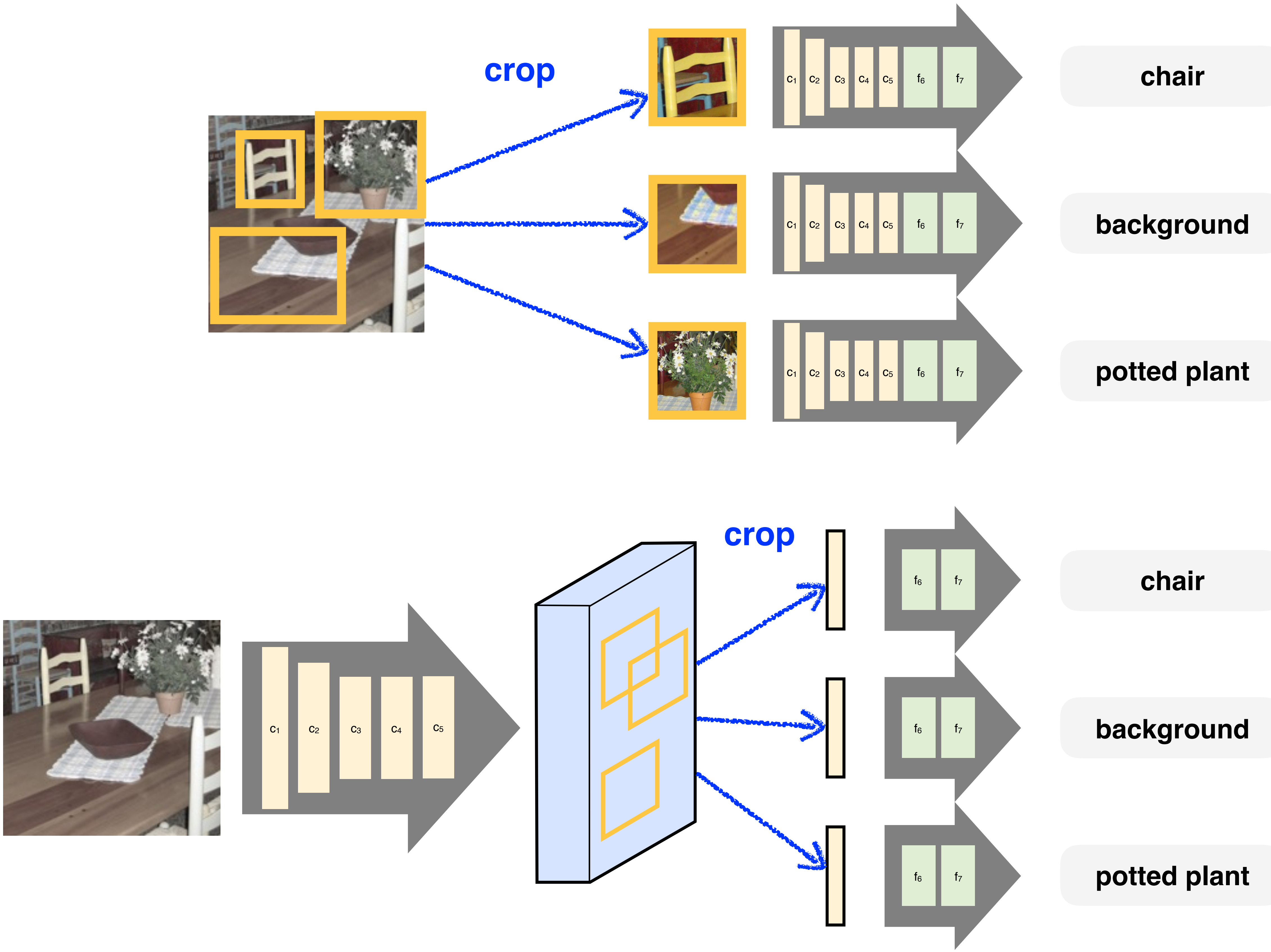


End-to-end training

Except for region proposals

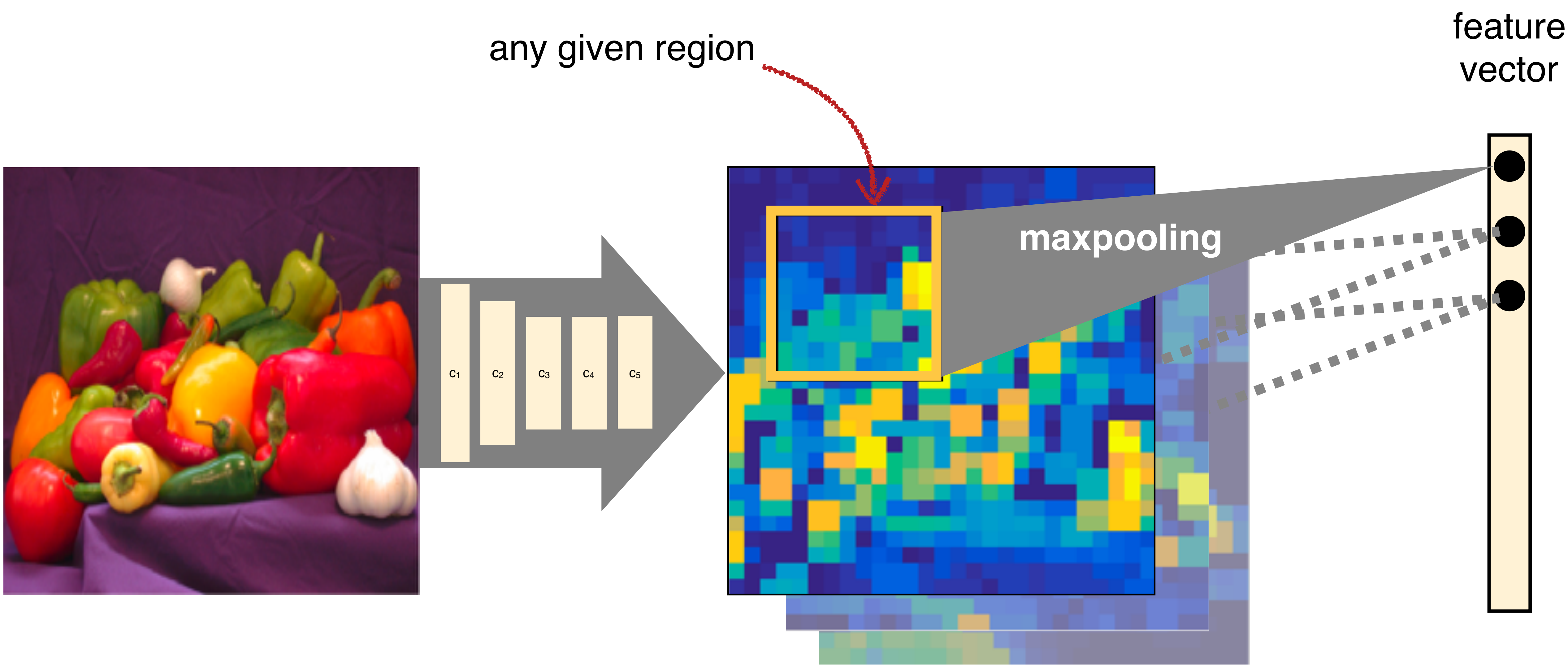
Problem: this is still pretty slow!

Accelerating R-CNN



The Spatial (Pyramid) Pooling layer

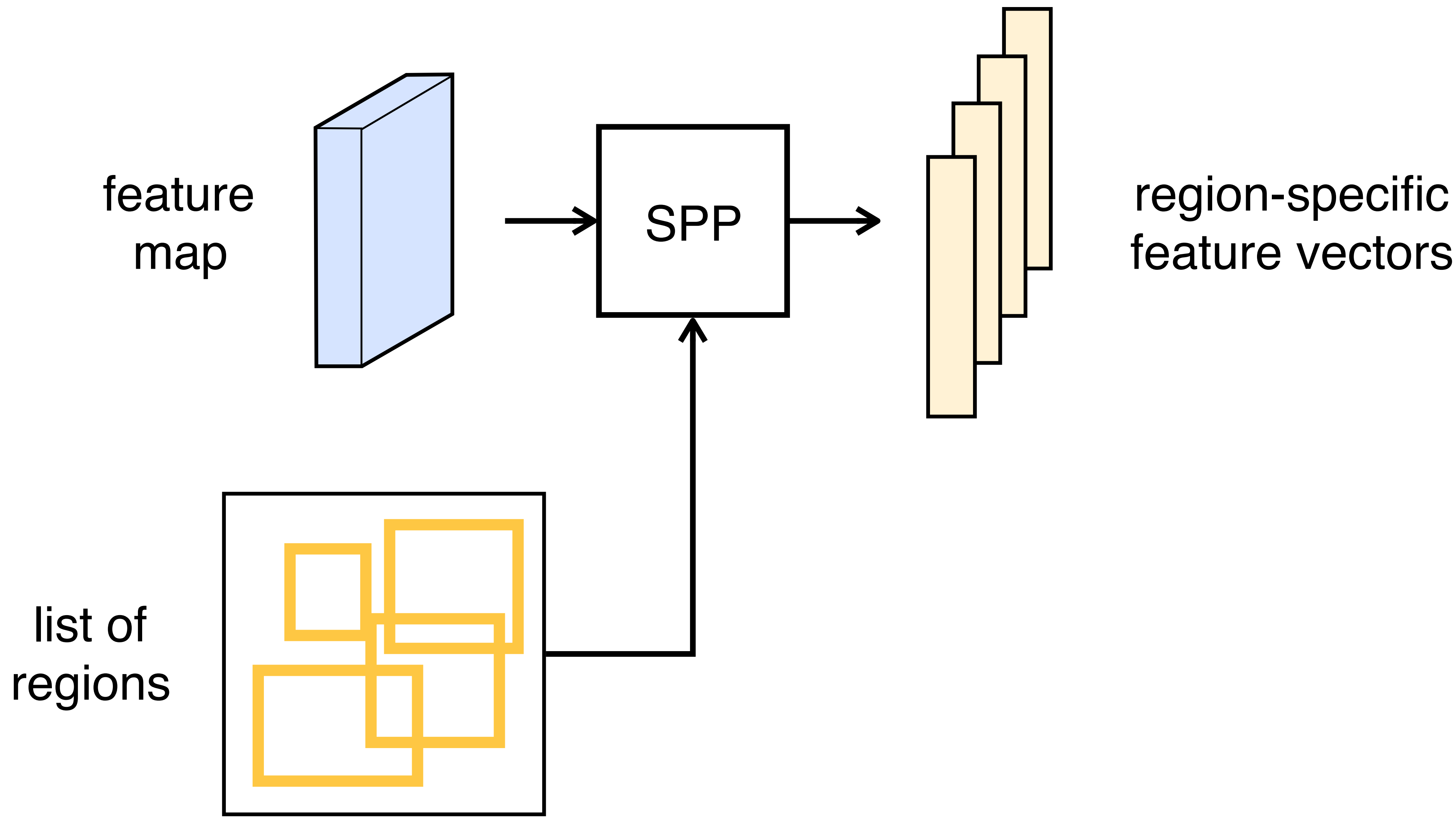
Max pooling in arbitrary regions



He, Zhang,Ren & Sun, “Spatial Pyramid Pooling (SPP) in Deep Convolutional Networks for Visual Recognition”, ECCV 2014

The Spatial (Pyramid) Pooling layer

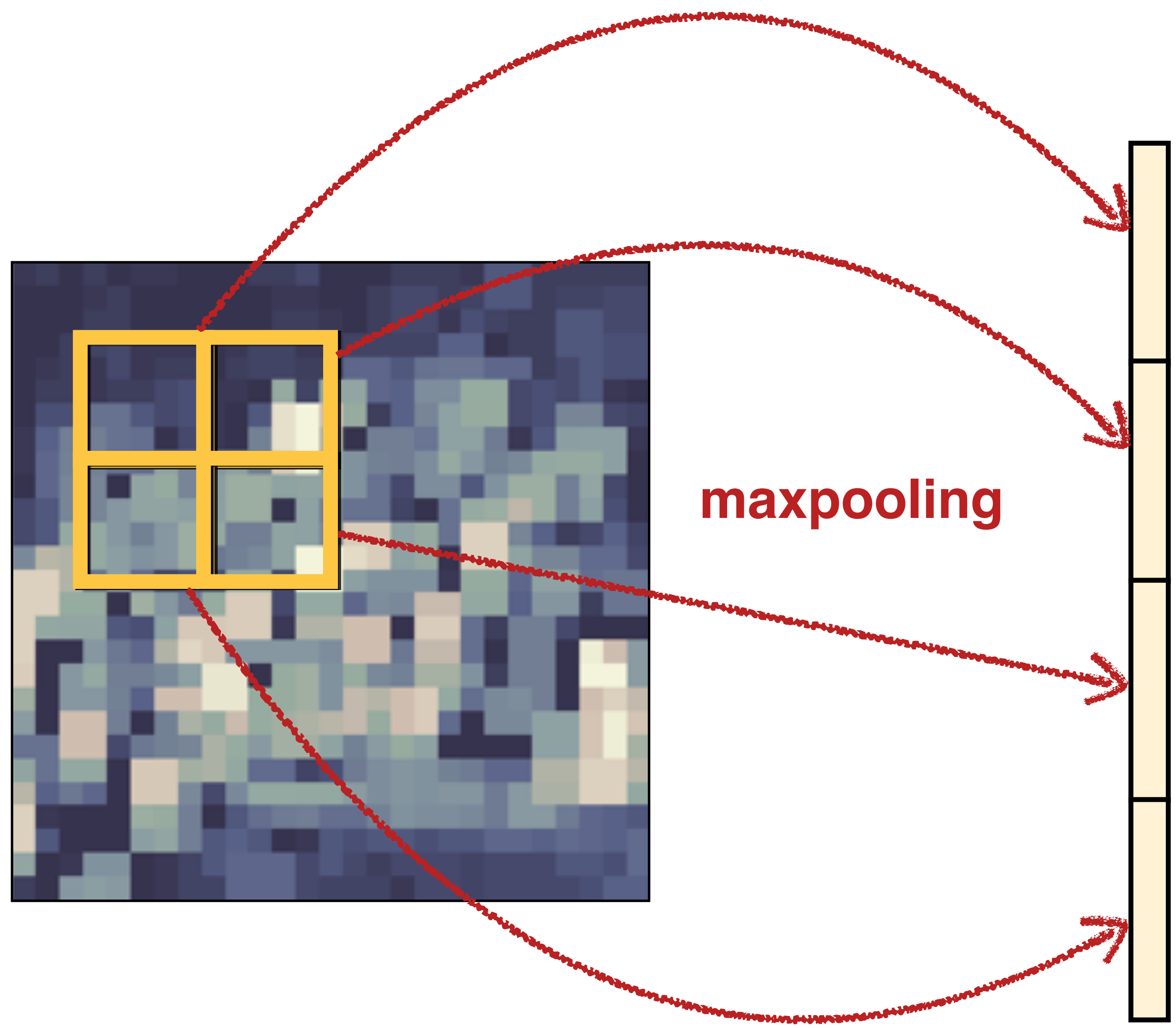
As a building block



He, Zhang,Ren & Sun, “Spatial Pyramid Pooling (SPP) in Deep Convolutional Networks for Visual Recognition”, ECCV 2014

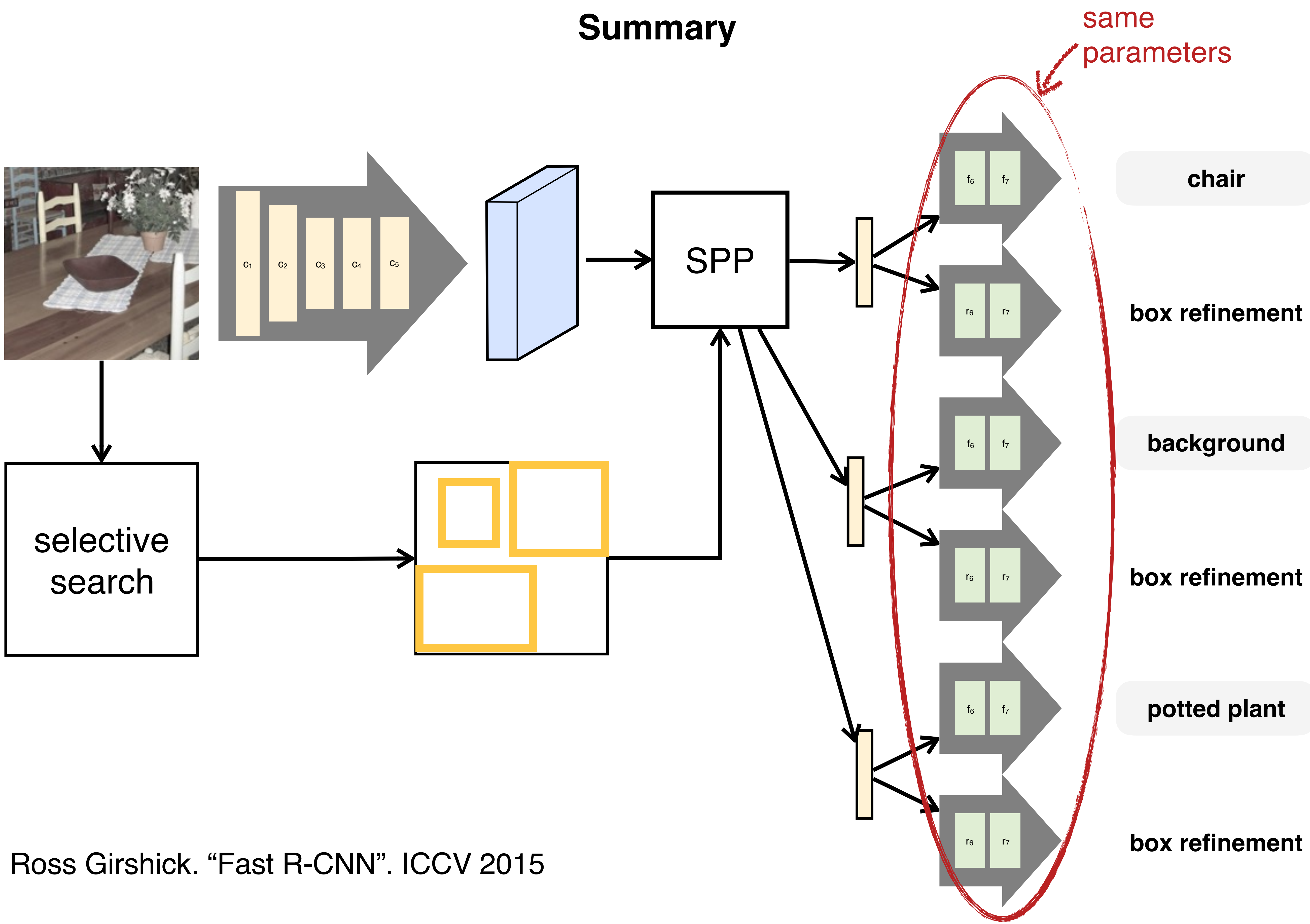
The Spatially **Pyramid** Pooling Layer

Same as above, but for multiple subdivisions



Fast R-CNN

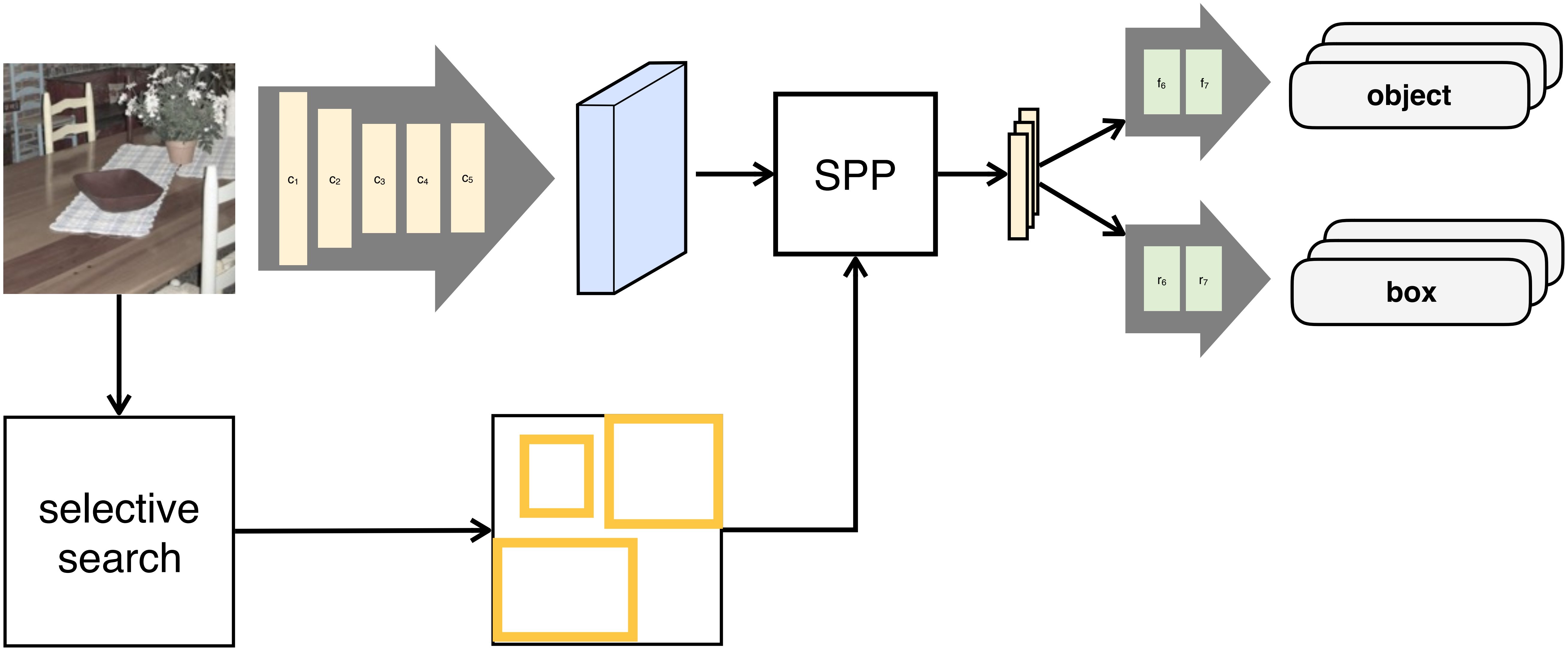
Summary



Ross Girshick. "Fast R-CNN". ICCV 2015

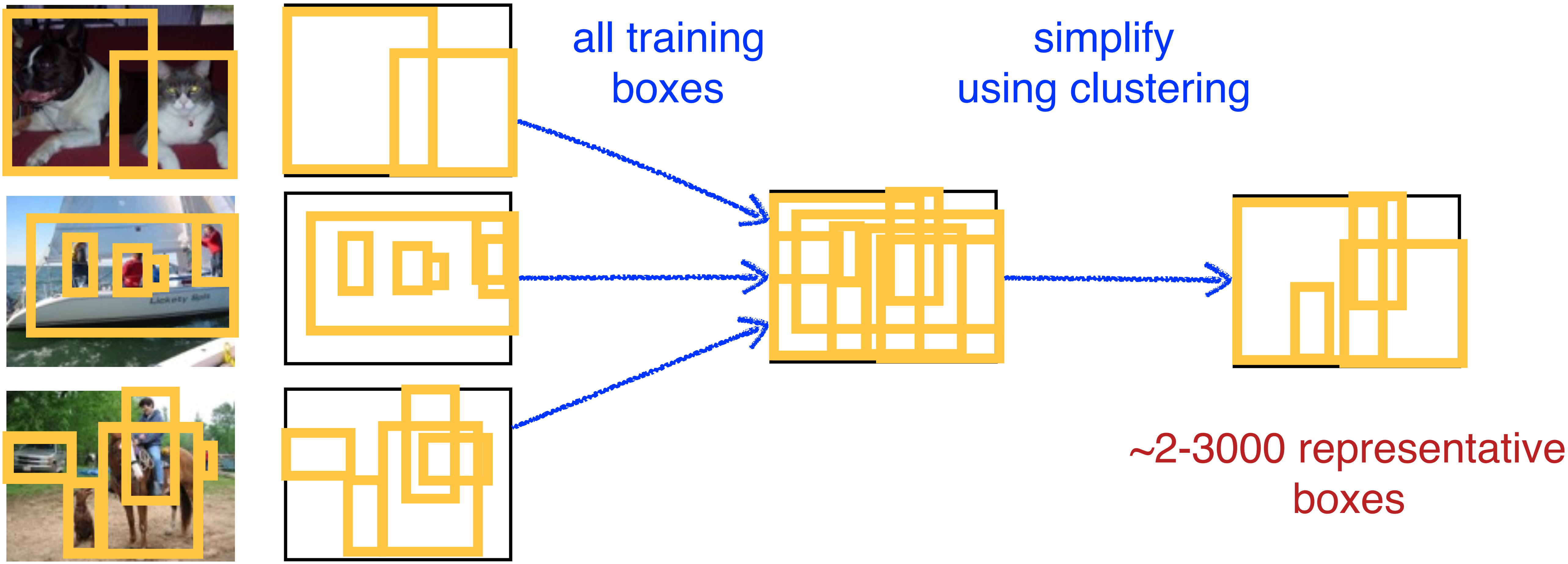
Fast R-CNN

Summary



R-CNN minus R

Fixed image-independent proposal set



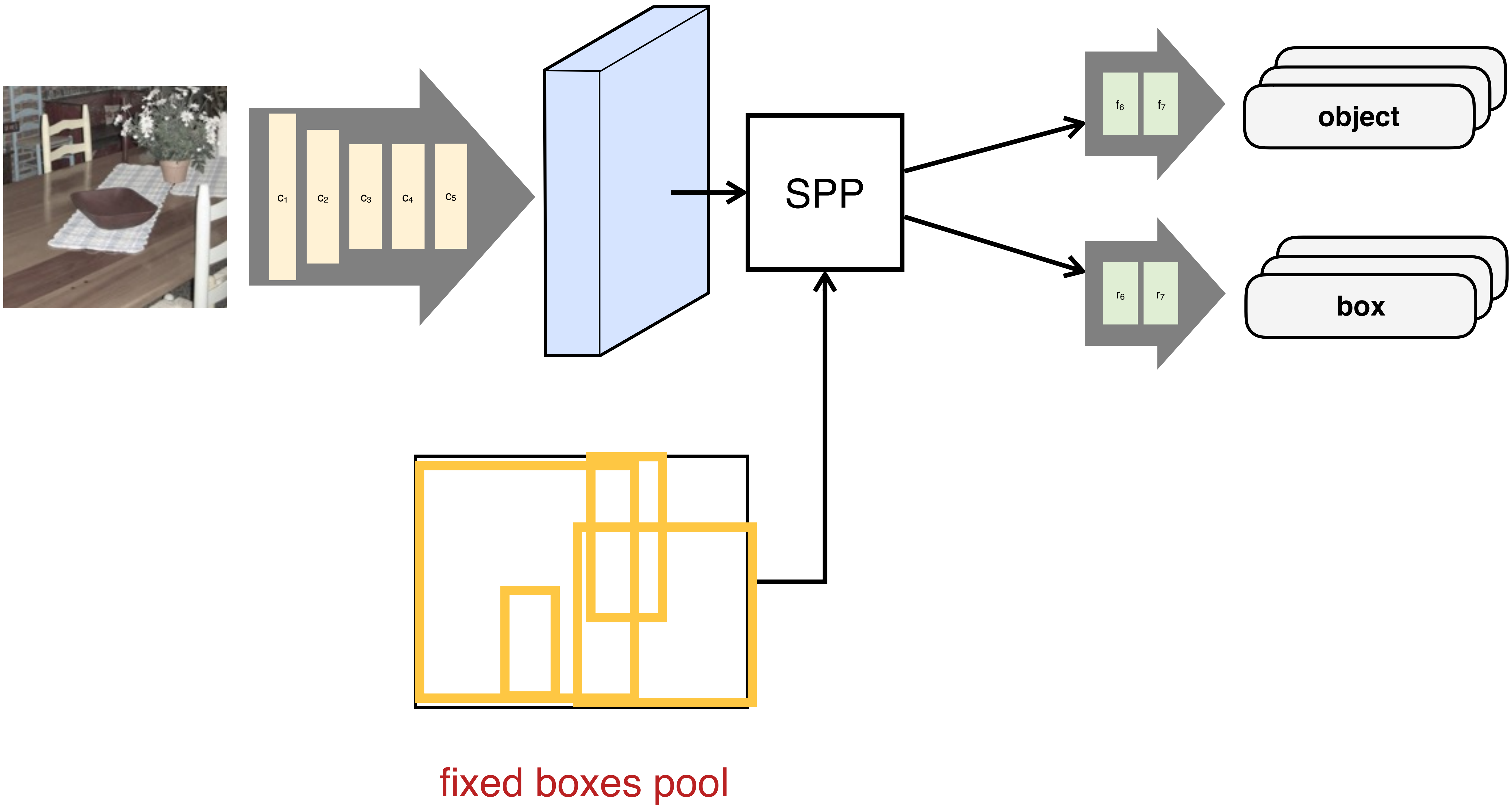
Fixed proposal generation

- Take all bounding boxes in the training set
- Run K-means clustering to distill a few thousands

[Lenc Vedaldi BMVC 2015]

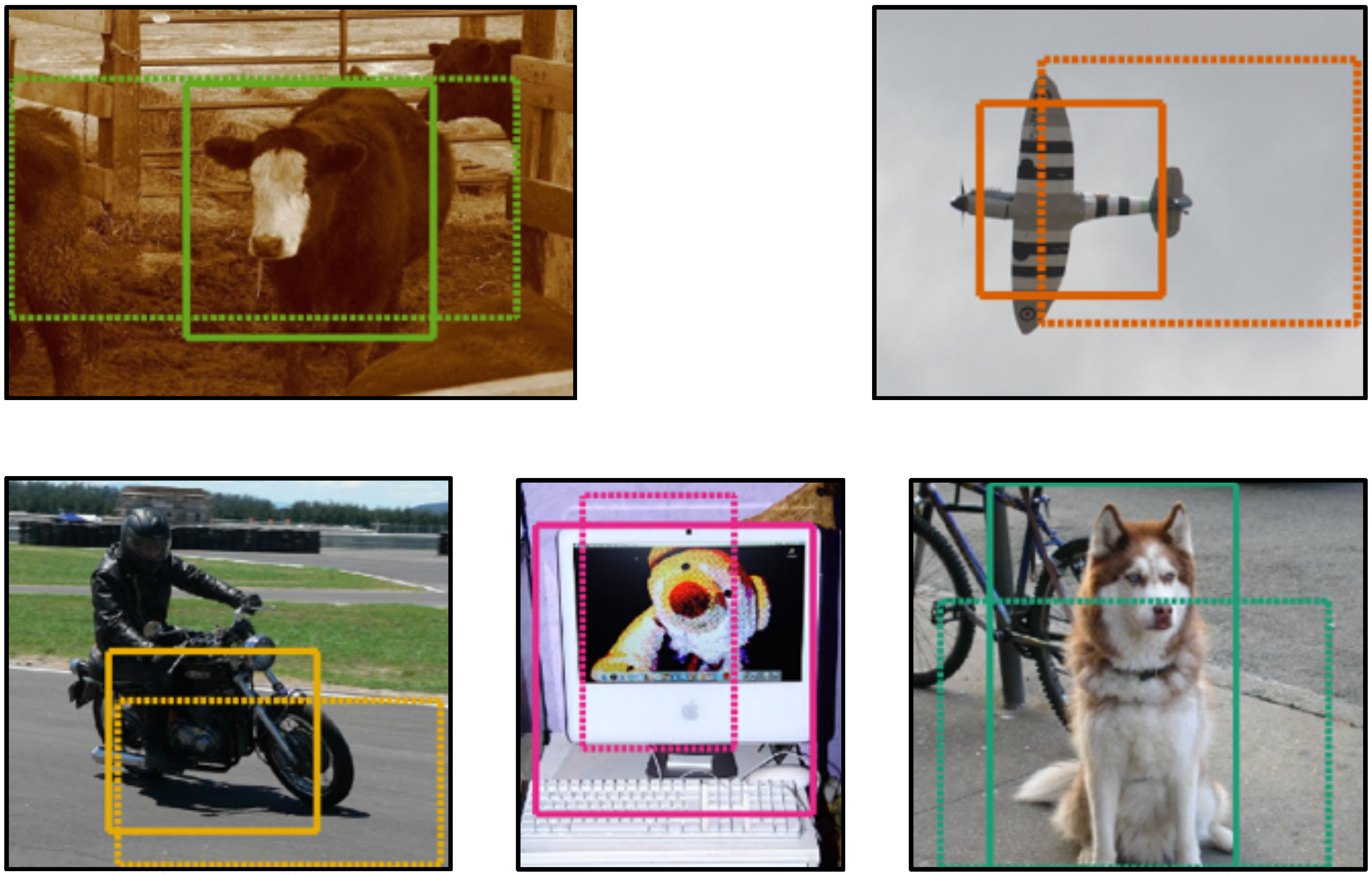
R-CNN minus R

Replace image-specific boxes with a fixed pool



Why does it work?

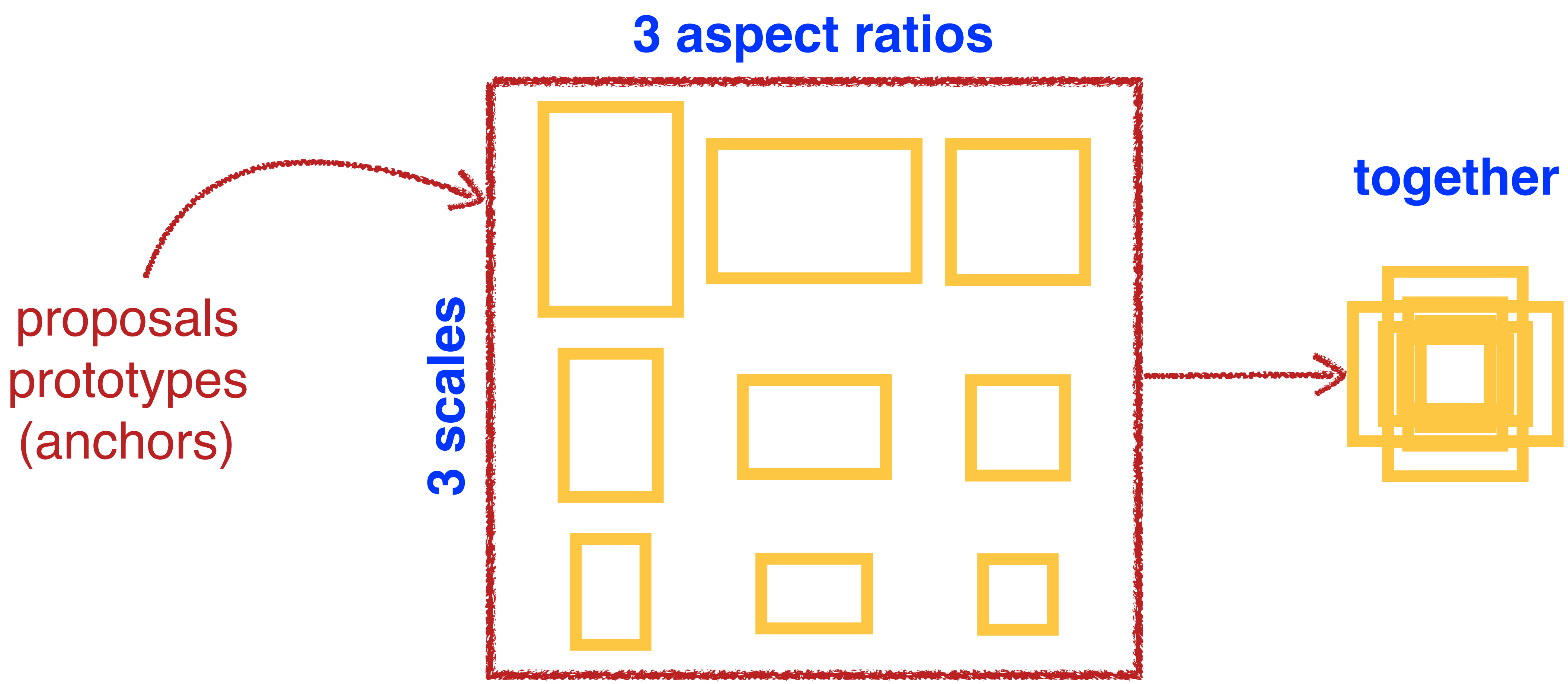
Answer: regression is quite powerful



Dashed line: initial **Solid line**: corrected by the CNN

Faster R-CNN

Better fixed proposals



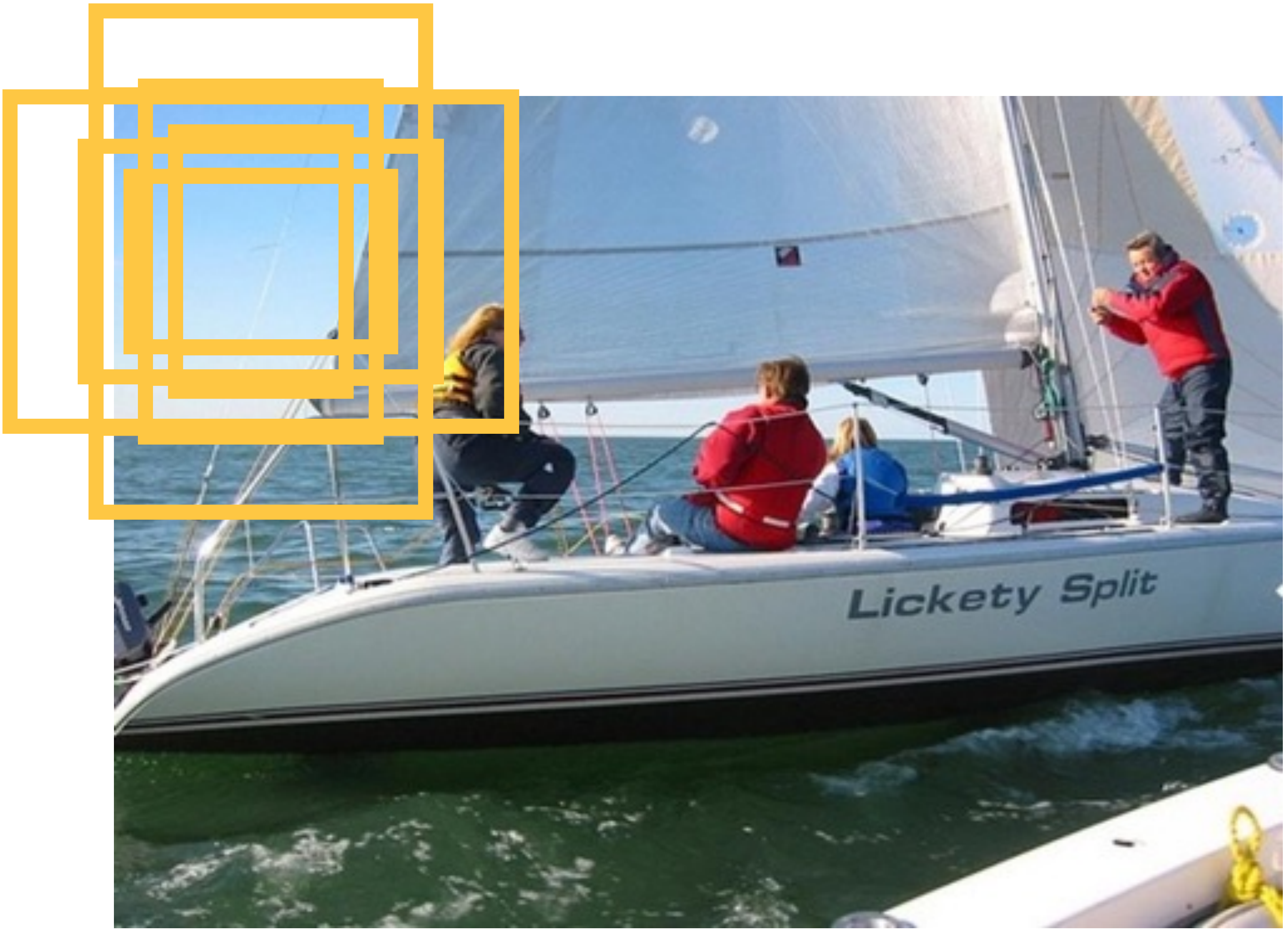
Ideas:

- Better fixed region proposal sampling
- Iterated classification: propose box, refine box, classify box
- Proposal shape specific classifier / regressors

Ren, He, Girshick, & Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.

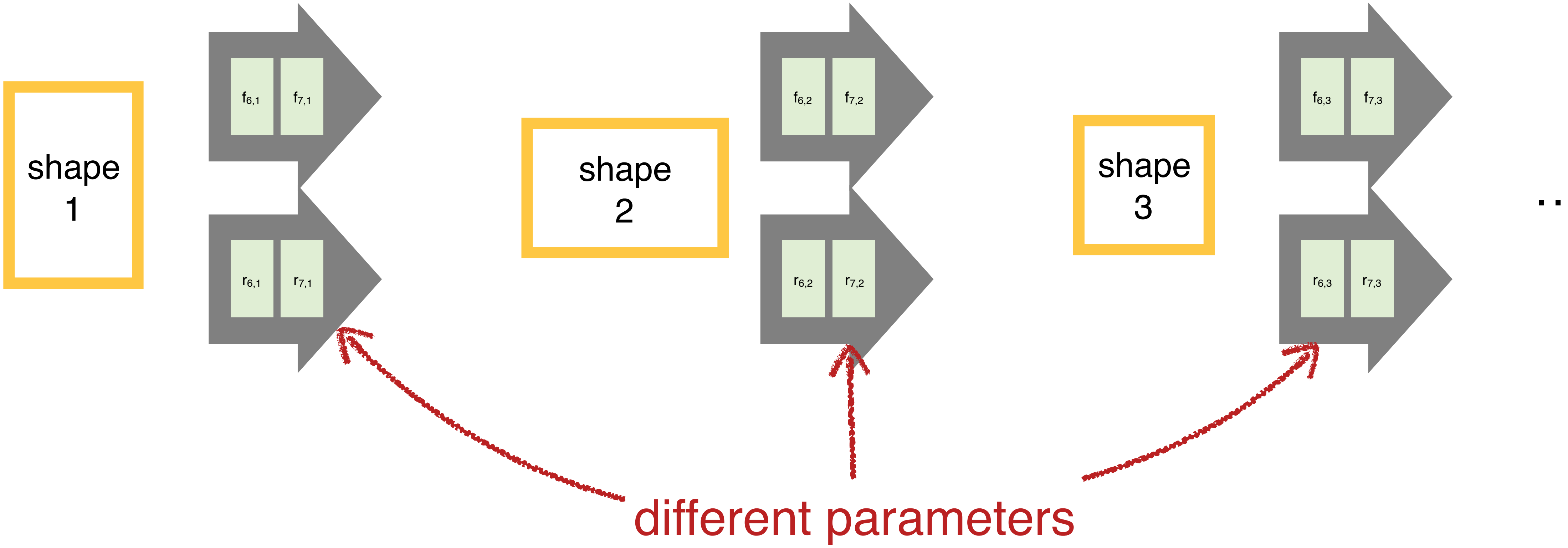
Faster R-CNN

Better fixed proposals



Ren, He, Girshick, & Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

Shape-specific classifiers / regressors



Model parameters: translation invariant but shape/scale specific

Object aspects are learned by brute force

Ren, He, Girshick, & Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

Multi-scale representations

Three strategies

scale image **scale feature filters** **fixed scale features**

model parameters shared for all scales

recompute features for each scale

cannot “see” fine details even when visible

“brute force” modeling of scale

compute features at a single scale

can model fine details

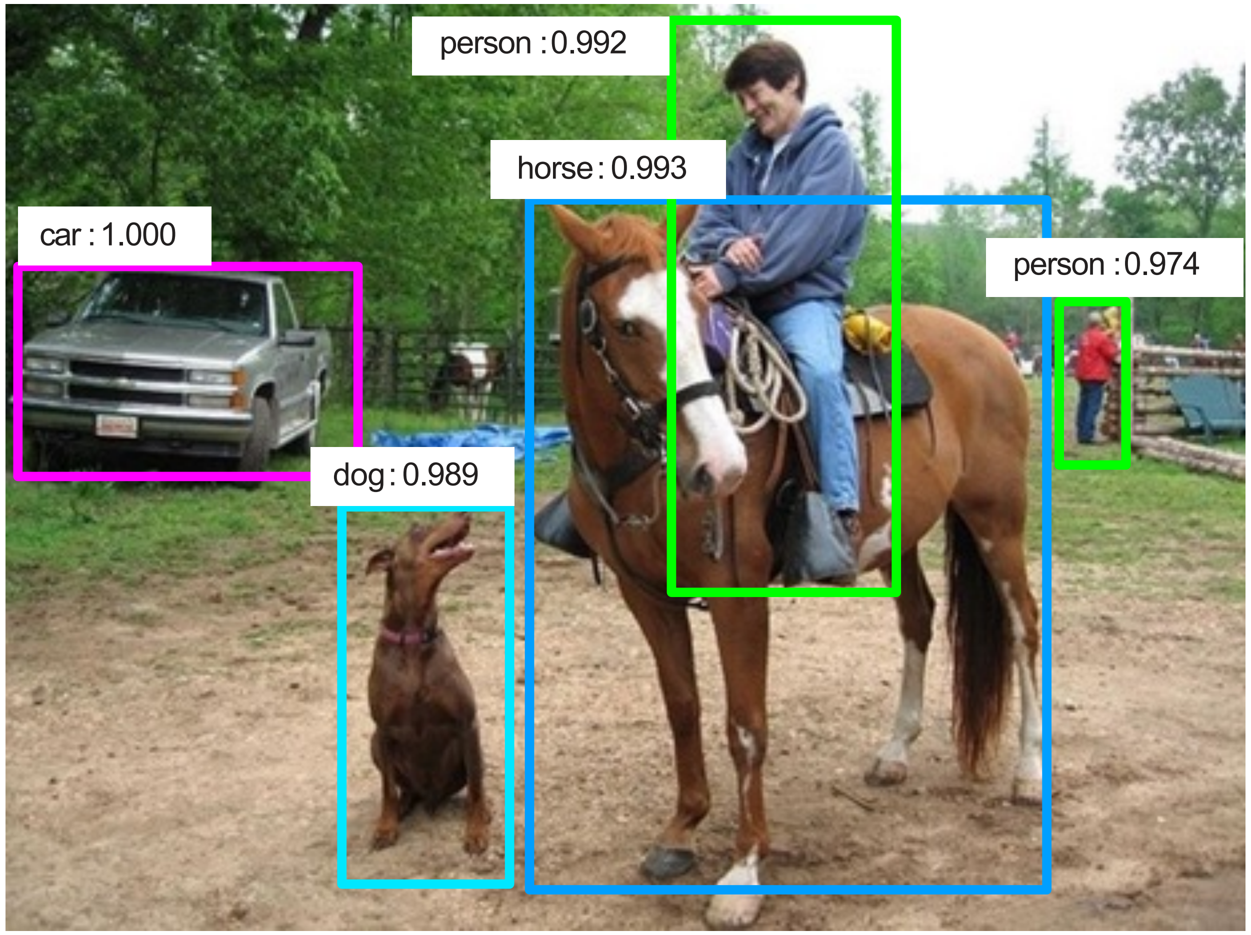
Fast and Faster R-CNN performance

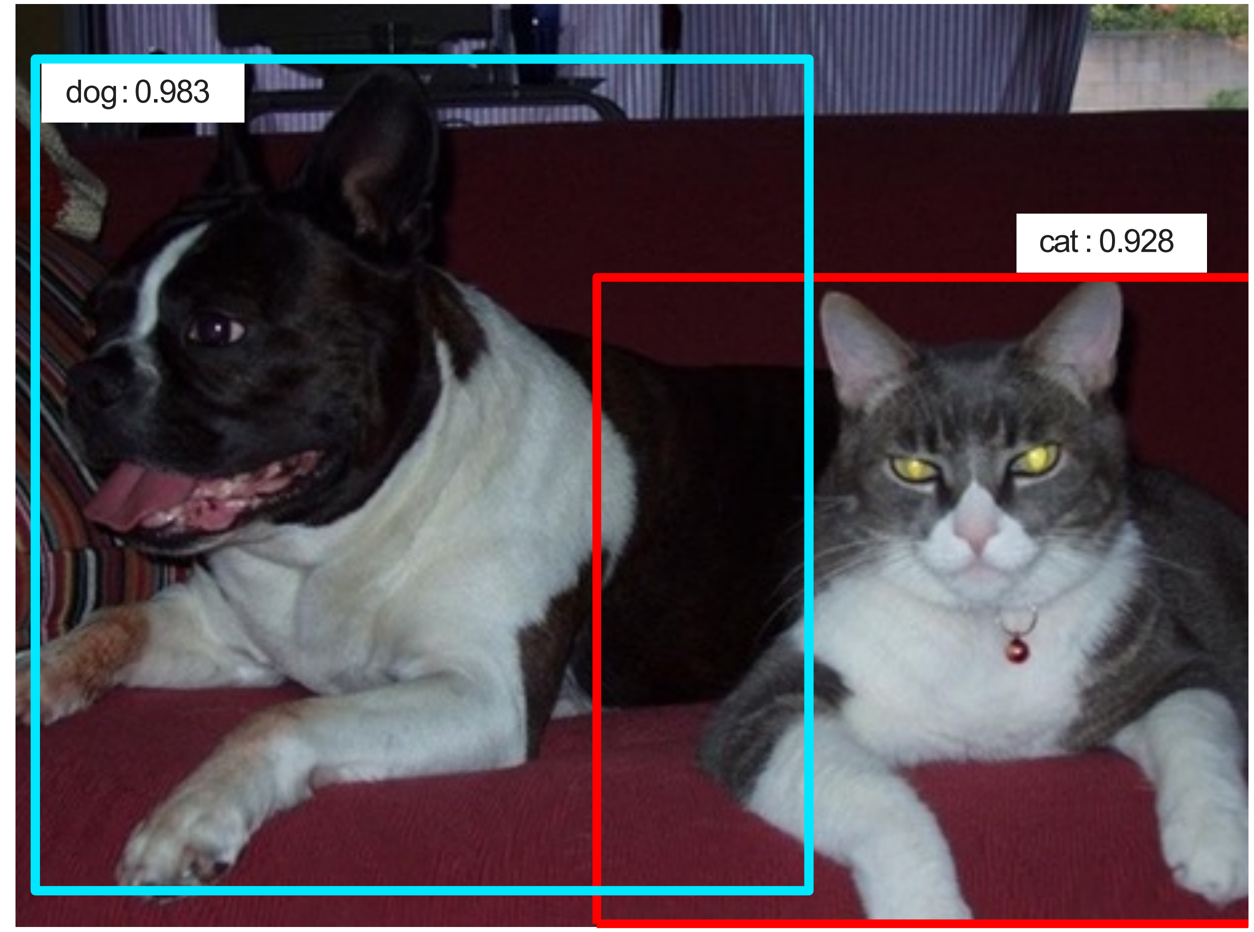
Better, faster!

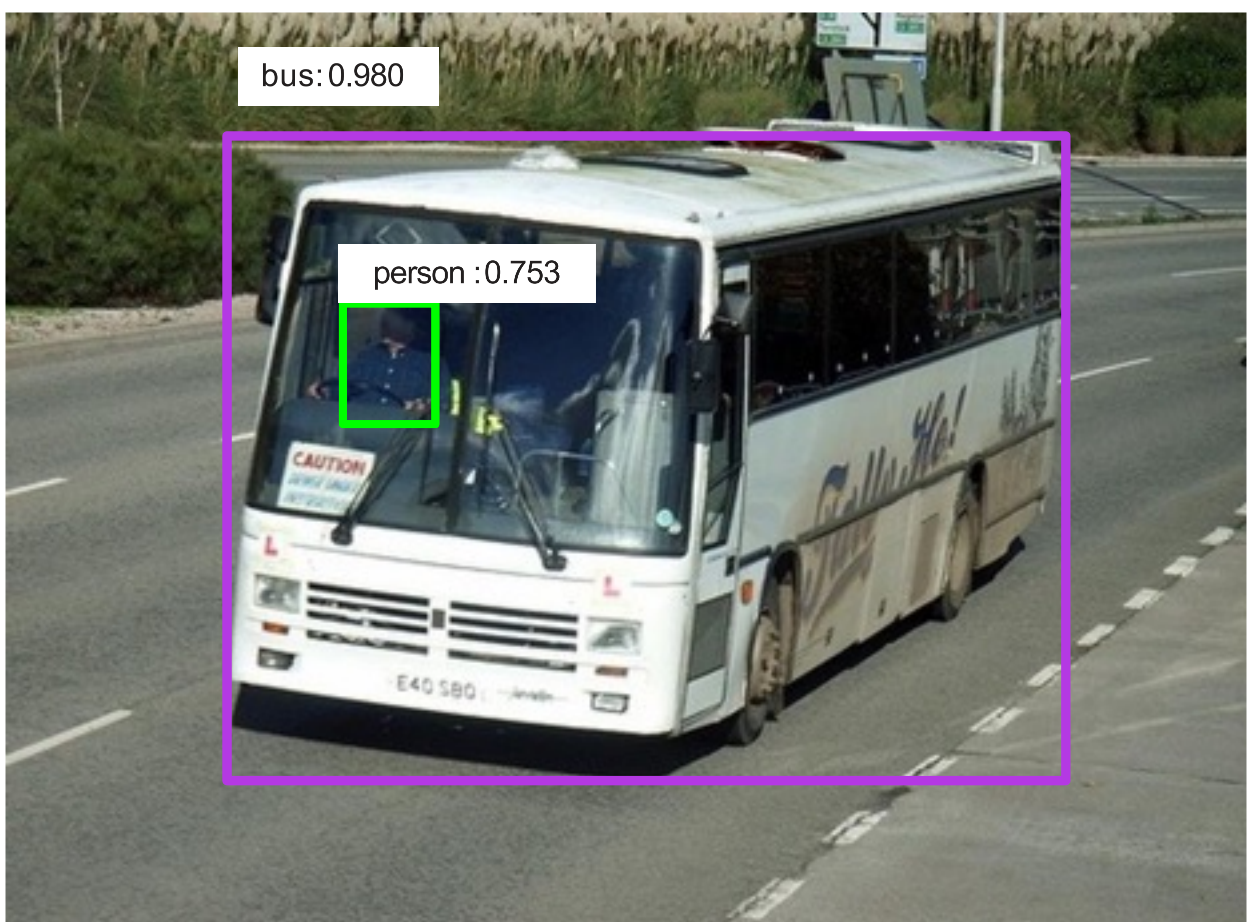
Method	Time / image	mAP (%)
R-CNN	~50s	66.0
Fast R-CNN	~2s	66.9
Faster R-CNN	198ms	69.9

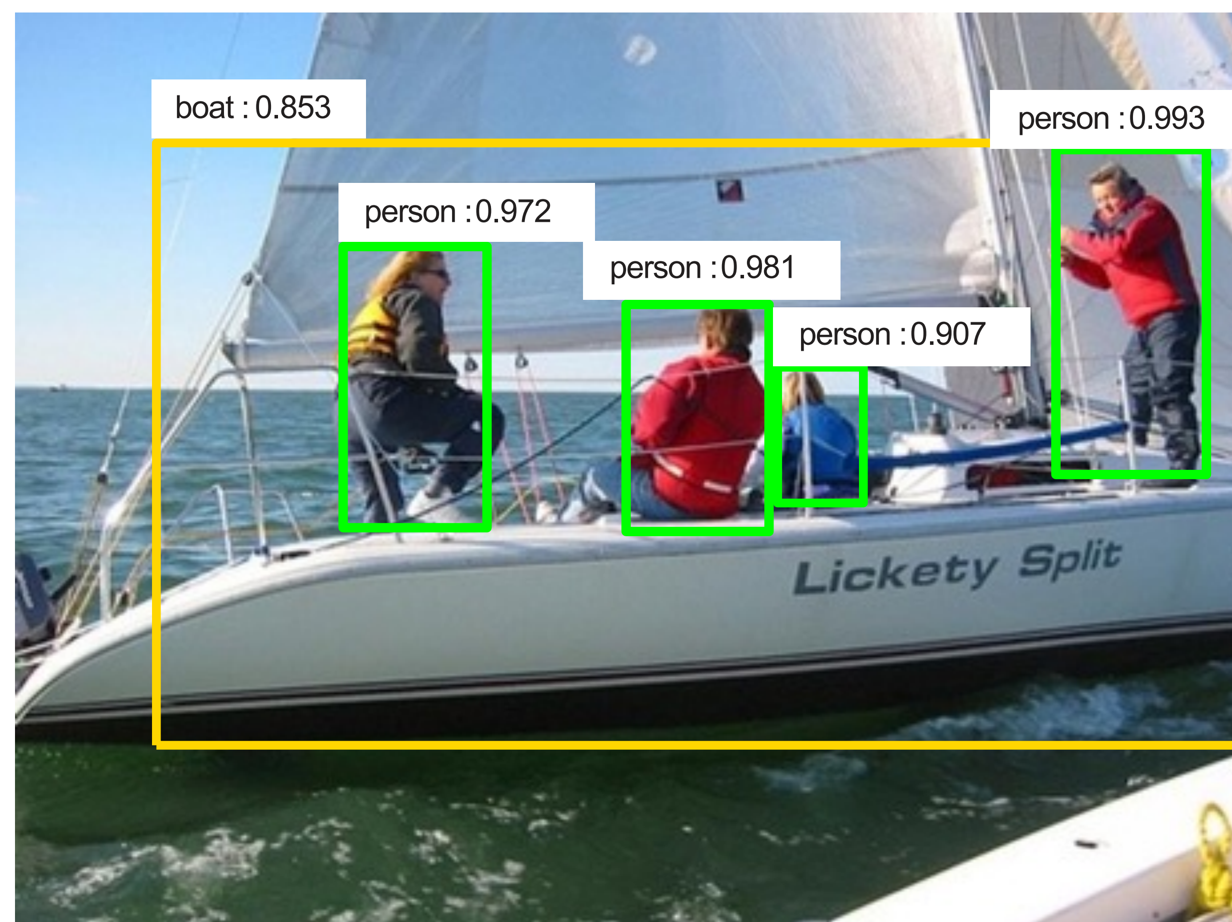
Detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet.

Example detections





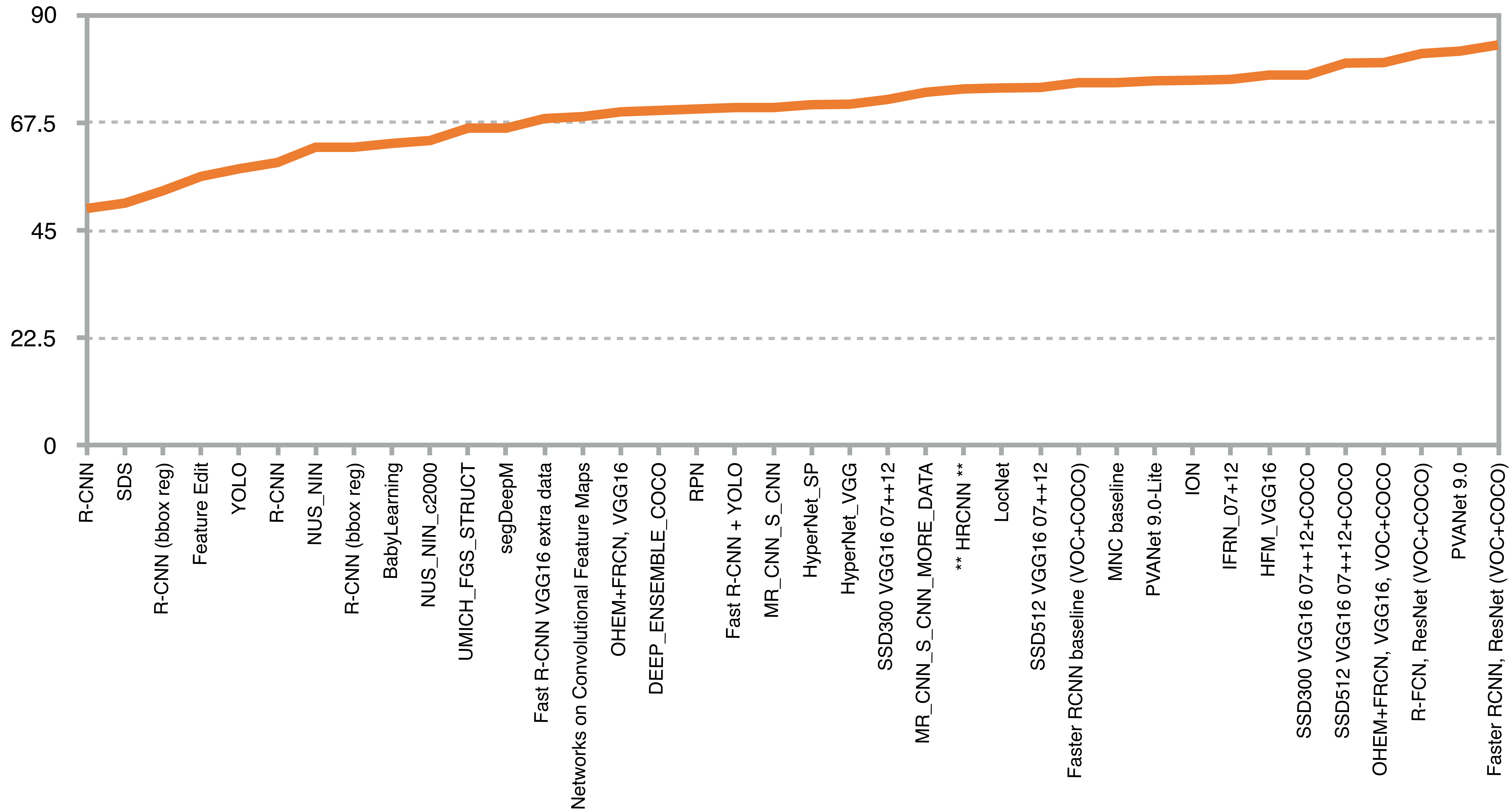




PASCAL VOC Leaderboards

Detection challenge (comp4: train on own data)

http://tinyurl.com/h7uzkov



2014 2016

4x better in error

Supervision required

Still a major limitation

Can get around in various ways, for example by using **synthetic data**

A. Gupta et al.. Synthetic data for text localisation in natural images. Proc. CVPR, 2016

Weakly-supervised learning

Use partial labelings

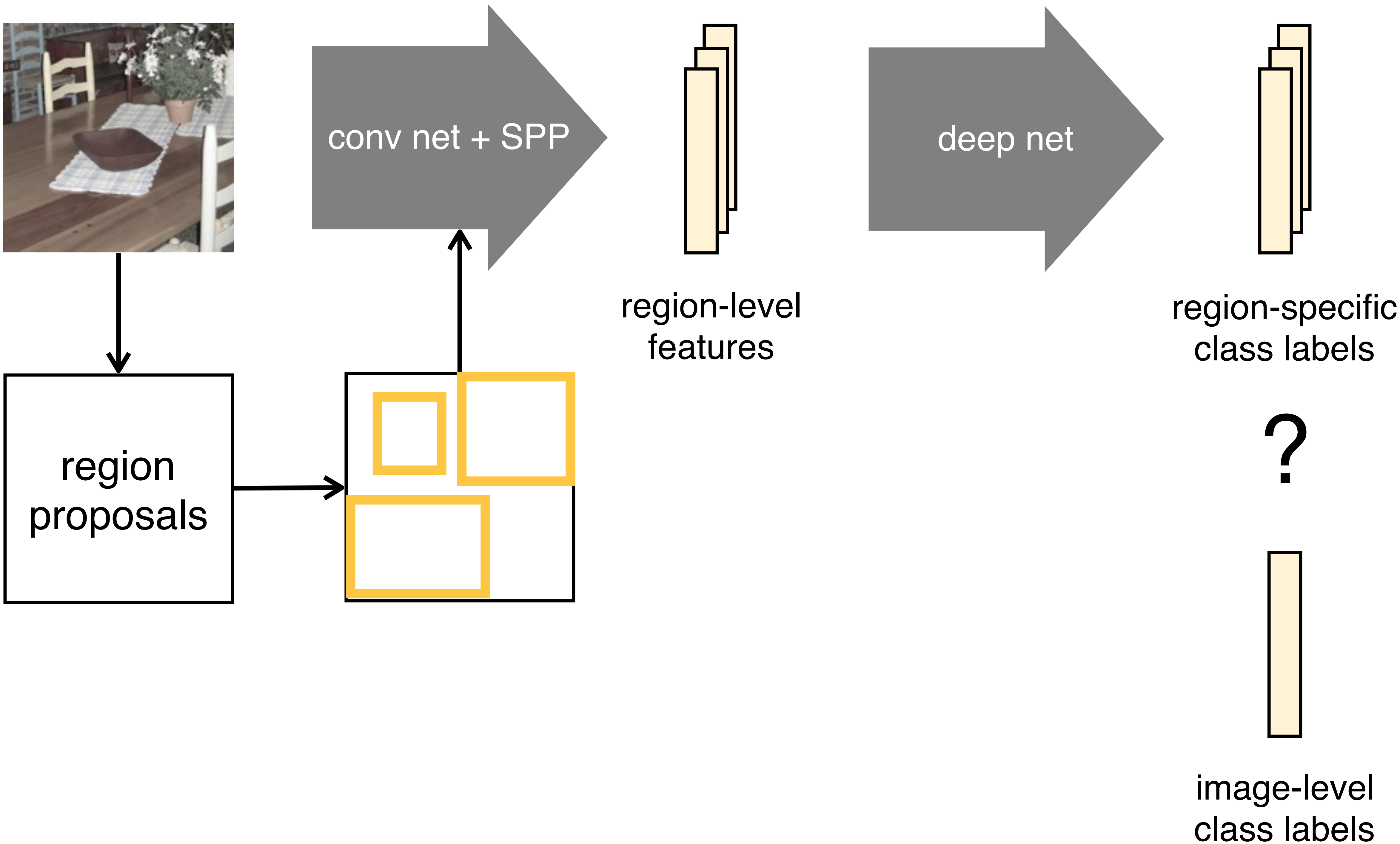
From this

To this

"There are
a cat, a person,
a chair"

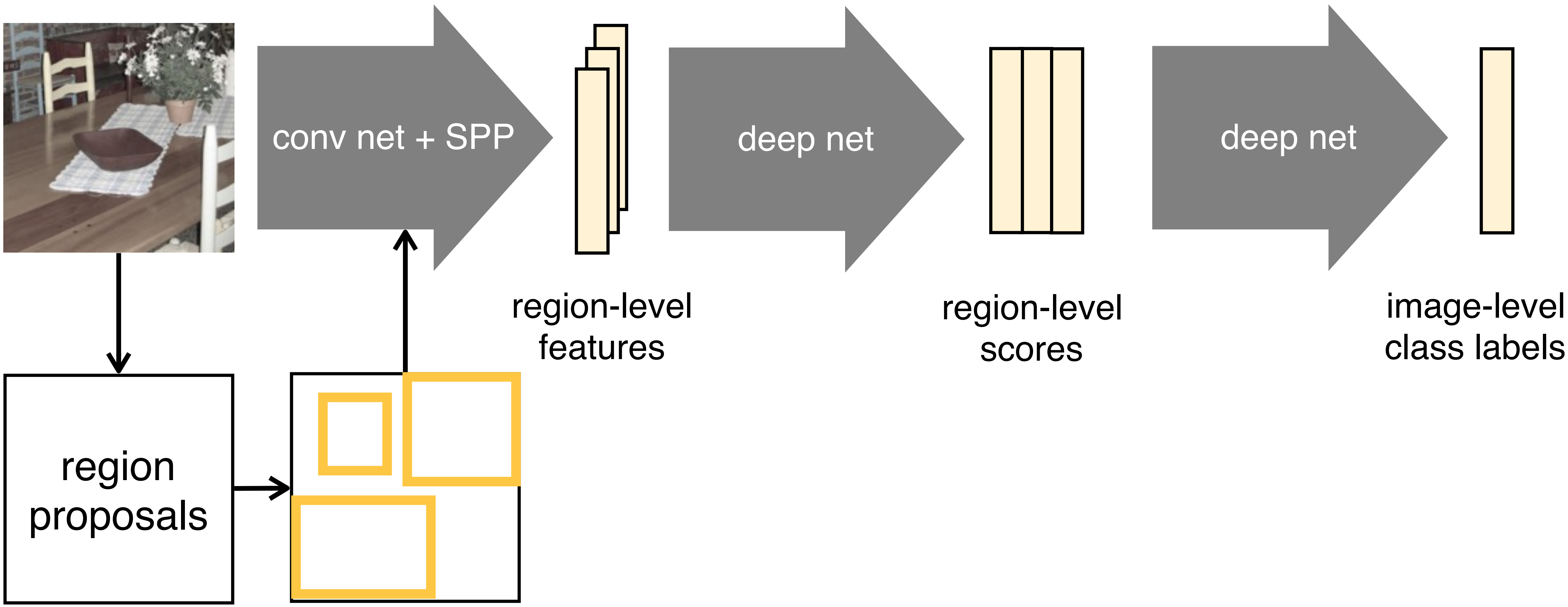
Towards weak supervision

High-level view



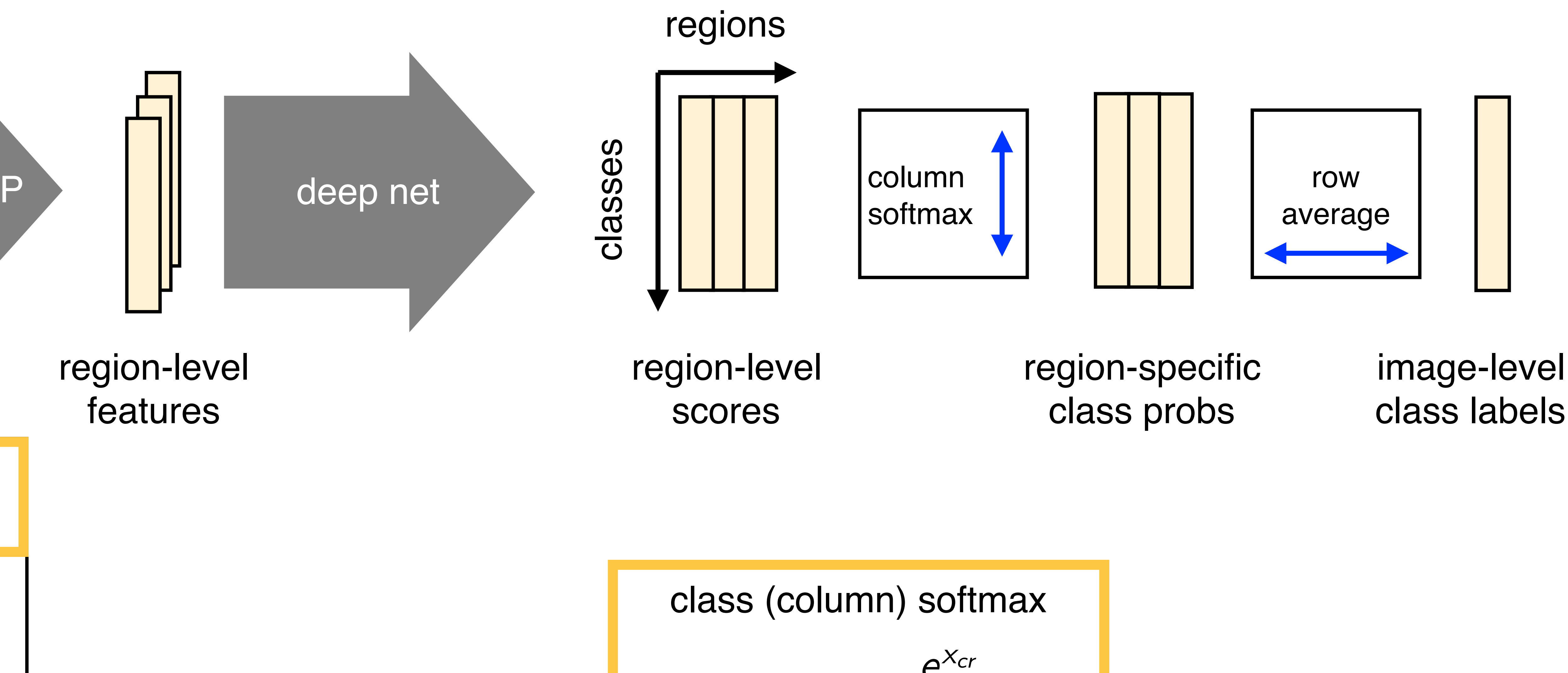
Towards weak supervision

Region scores to class labels



Towards weak supervision

Class labels = average of region labels

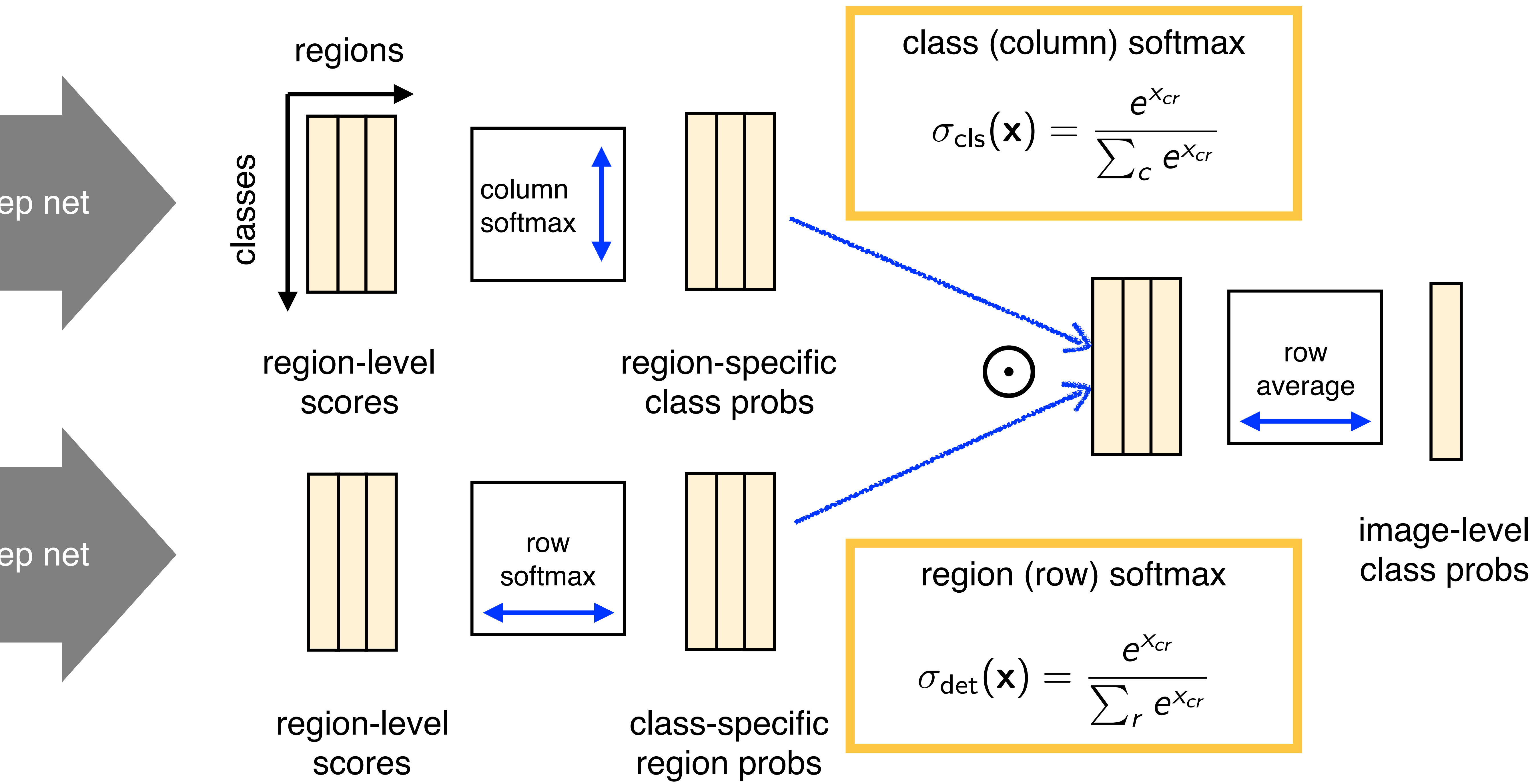


class (column) softmax

$$\sigma_{\text{cls}}(\mathbf{x}) = \frac{e^{x_{cr}}}{\sum_c e^{x_{cr}}}$$

Towards weak supervision

Combine both class and region information



class (column) softmax

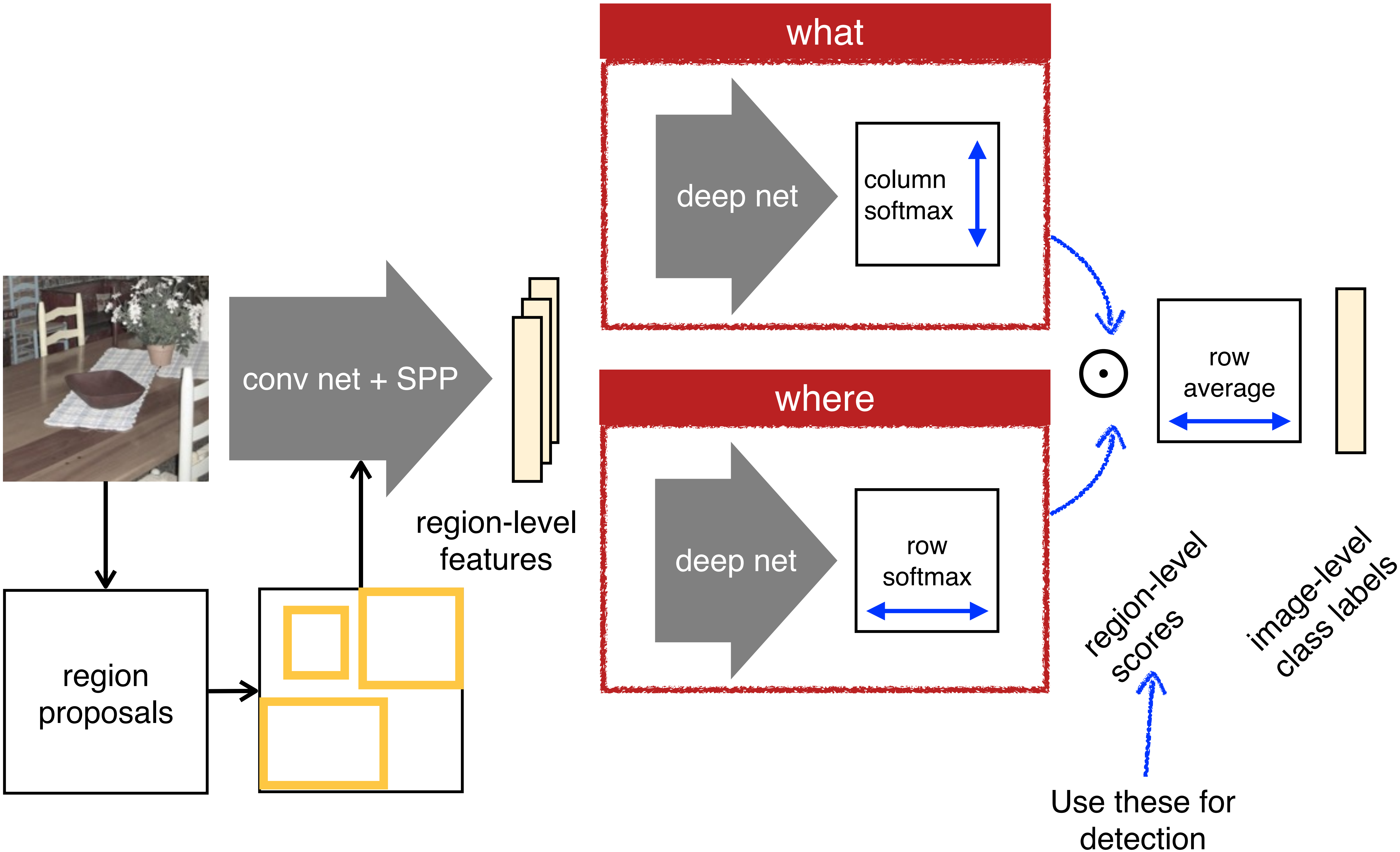
$$\sigma_{\text{cls}}(\mathbf{x}) = \frac{e^{x_{cr}}}{\sum_c e^{x_{cr}}}$$

region (row) softmax

$$\sigma_{\text{det}}(\mathbf{x}) = \frac{e^{x_{cr}}}{\sum_r e^{x_{cr}}}$$

Two-streams weakly-supervised R-CNN

Overview



[Bilen Vedaldi CVPR 2016]

Results

Two streams vs state-of-the-art

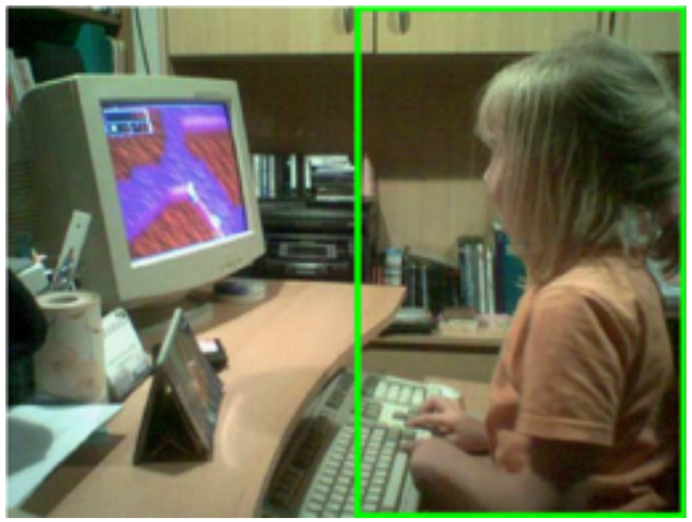
PASCAL VOC Detection (mAP)				
	Wang@ ECCV14	Bilen@ CVPR15	Cinbis@ PAMI16	Two Streams ++
PASCAL VOC07	30.2	27.7	31.6	**39.3**
PASCAL VOC10	27.4	-	-	**36.2**

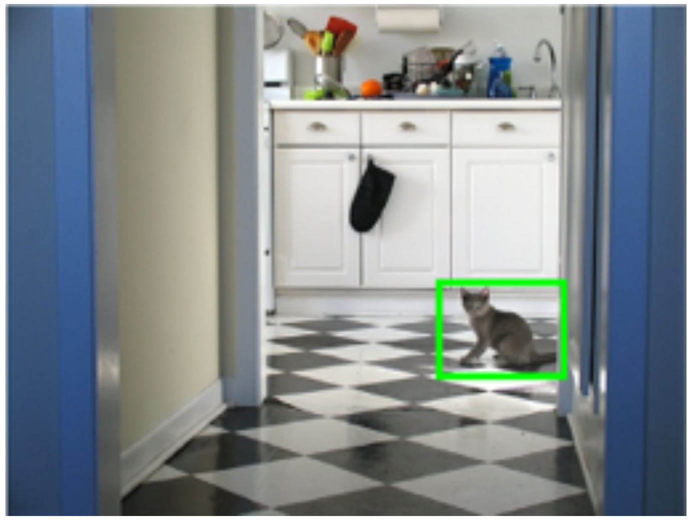
Two streams vs single stream

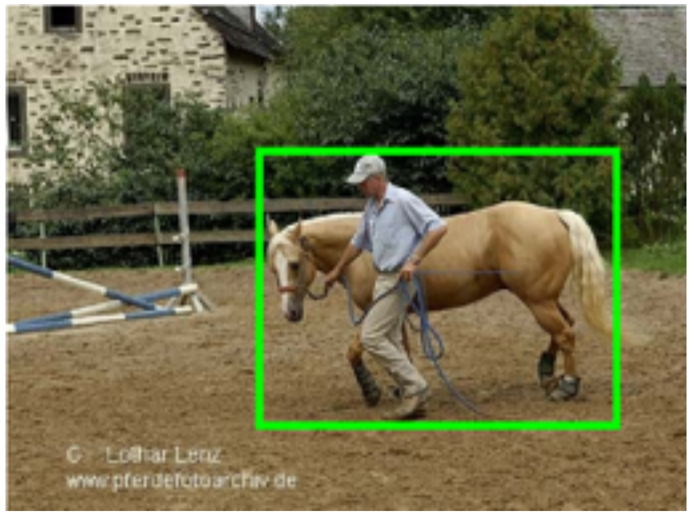
PASCAL VOC Detection (mAP)			
	Single Stream	Two Streams	Two Streams ++
PASCAL VOC07	21.6	30.9	39.3

Results

Good results







Failure modes

Summary

Modern CNNs are still “new” technology

- Expect bigger and meaner CNNs to further improve performance
- CNNs are more than big balls of parameters
- We do not really understand what they do

CNNs can address directly many interesting applications

- Classification
- Segmentation
- Detection
- Regression

Addressing the supervision problem

- Transfer learning
- Synthetic data
- Pseudo-tasks

Still a long way to go

Integrated computer vision

- One network to rule them all

Cognition

- Integrate perception and “the rest”
- Planning, memory, background knowledge, …

No labels required

- Unsupervised
- Alternatively supervised (reinforcement, pseudo-tasks)

Spatial reasoning

- From image-centric to object-centric and scene-centric understanding
- Representing deformable 3D shape
- Dynamic environments, physics