Neural network training

Vagaries of optimization

- Non-convex
  - Local optima
  - Sensitivity to initialization
- Vanishing / exploding gradients

$$\frac{\partial z}{\partial z_i} = \frac{\partial z}{\partial z_{n-1}} \frac{\partial z_{n-1}}{\partial z_{n-2}} \cdots \frac{\partial z_{i+1}}{\partial z_i}$$

  - If each term is (much) greater than 1 → *explosion of gradients*
  - If each term is (much) less than 1 → *vanishing gradients*

Vanishing and exploding gradients

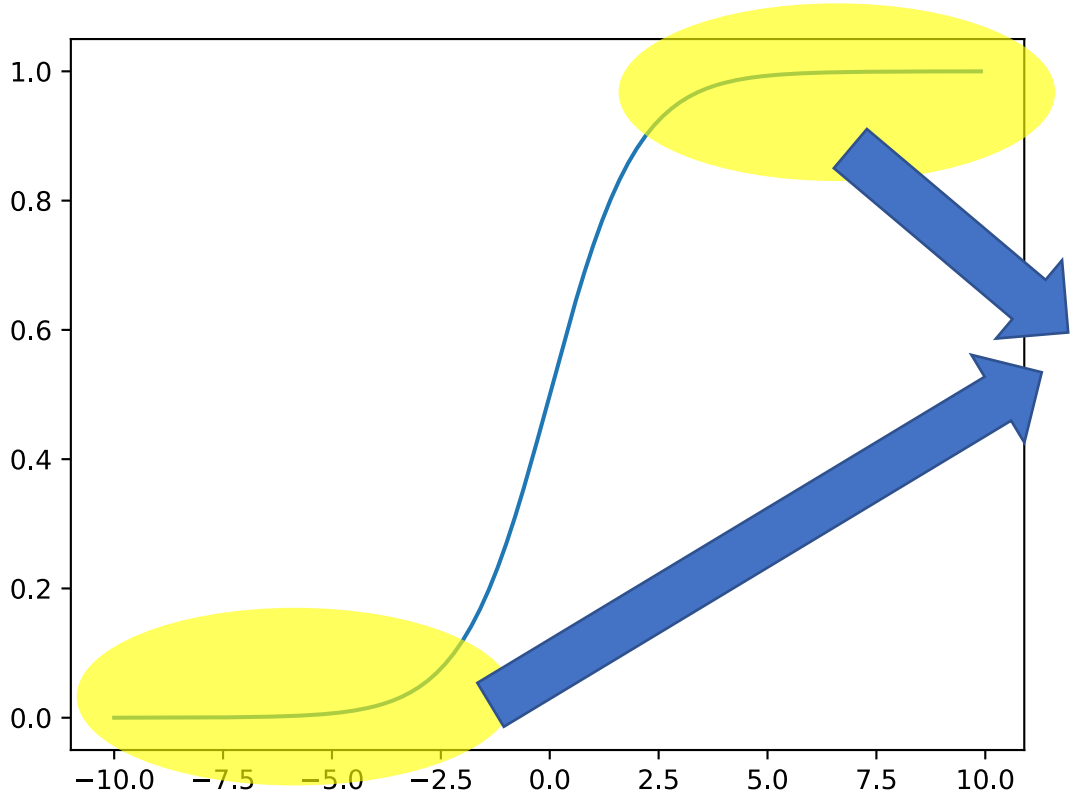
$$\frac{\partial \mathbf{z}}{\partial \mathbf{z}_i} = \frac{\partial \mathbf{z}}{\partial \mathbf{z}_{n-1}} \frac{\partial \mathbf{z}_{n-1}}{\partial \mathbf{z}_{n-2}} \cdots \frac{\partial \mathbf{z}_{i+1}}{\partial \mathbf{z}_i}$$

$$\frac{\partial L}{\partial \mathbf{z}_i} = \frac{\partial L}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{z}_i}$$

$$\lambda_{min}(\frac{\partial \mathbf{z}}{\partial \mathbf{z}_i})\frac{\partial L}{\partial \mathbf{z}} \leq \frac{\partial L}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{z}_i} \leq \lambda_{max}(\frac{\partial \mathbf{z}}{\partial \mathbf{z}_i})\frac{\partial L}{\partial \mathbf{z}}$$

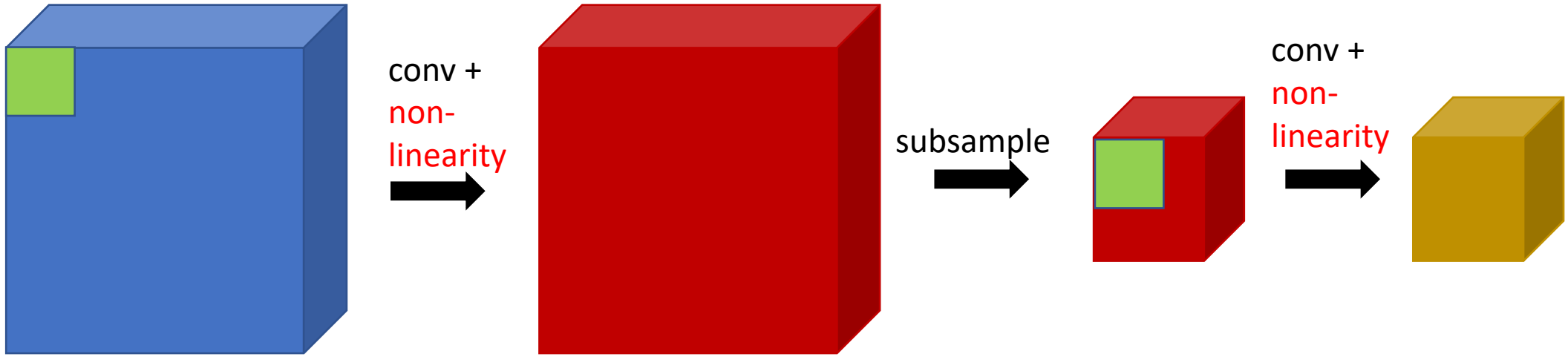
$$\lambda_{max}(UV) \leq \lambda_{max}(U)\lambda_{max}(V)$$
$$\lambda_{min}(UV) \geq \lambda_{min}(U)\lambda_{min}(V)$$
$$\lambda_{max}(A^n) = \lambda_{max}(A)^n$$
$$\lambda_{min}(A^n) = \lambda_{min}(A)^n$$

Sigmoids cause vanishing gradients



Gradient close to 0

Convolution subsampling convolution



Rectified Linear Unit (ReLU)

- max (x,0)
- Also called half-wave rectification (signal processing)

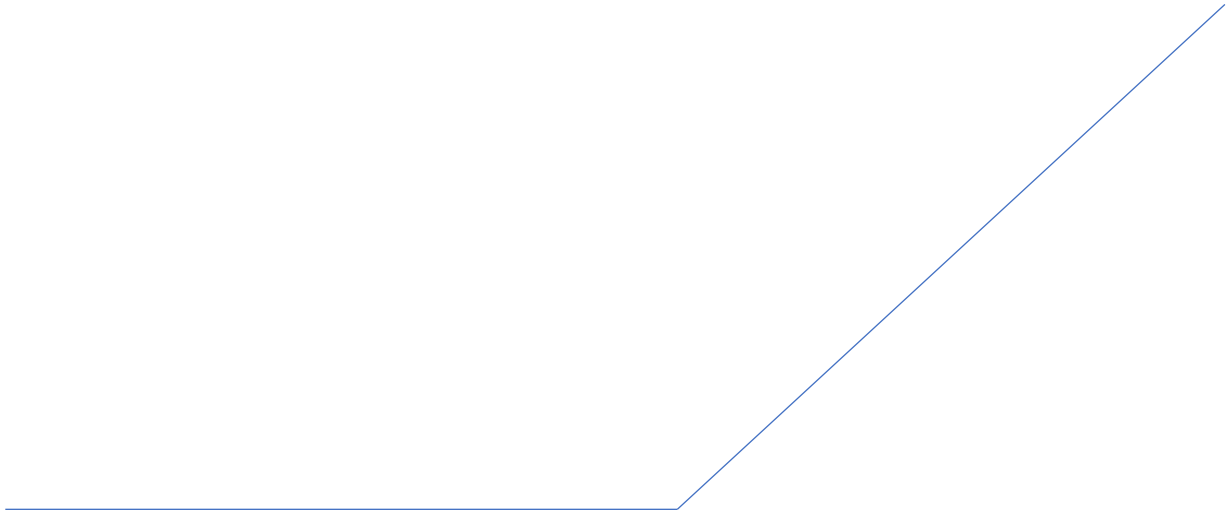


Image Classification

How to do machine learning

- Create training / validation sets
- Identify loss functions
- Choose hypothesis class
- Find best hypothesis by minimizing training loss

How to do machine learning

- Create training / validation sets
- Identify loss functions
- Choose hypothesis class
- Find best hypothesis by minimizing training loss

Multiclass classification !!

$$h(x) = \mathbf{s} \qquad \hat{p}(y = k|x) \propto e^{s_k} \qquad \hat{p}(y = k|x) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

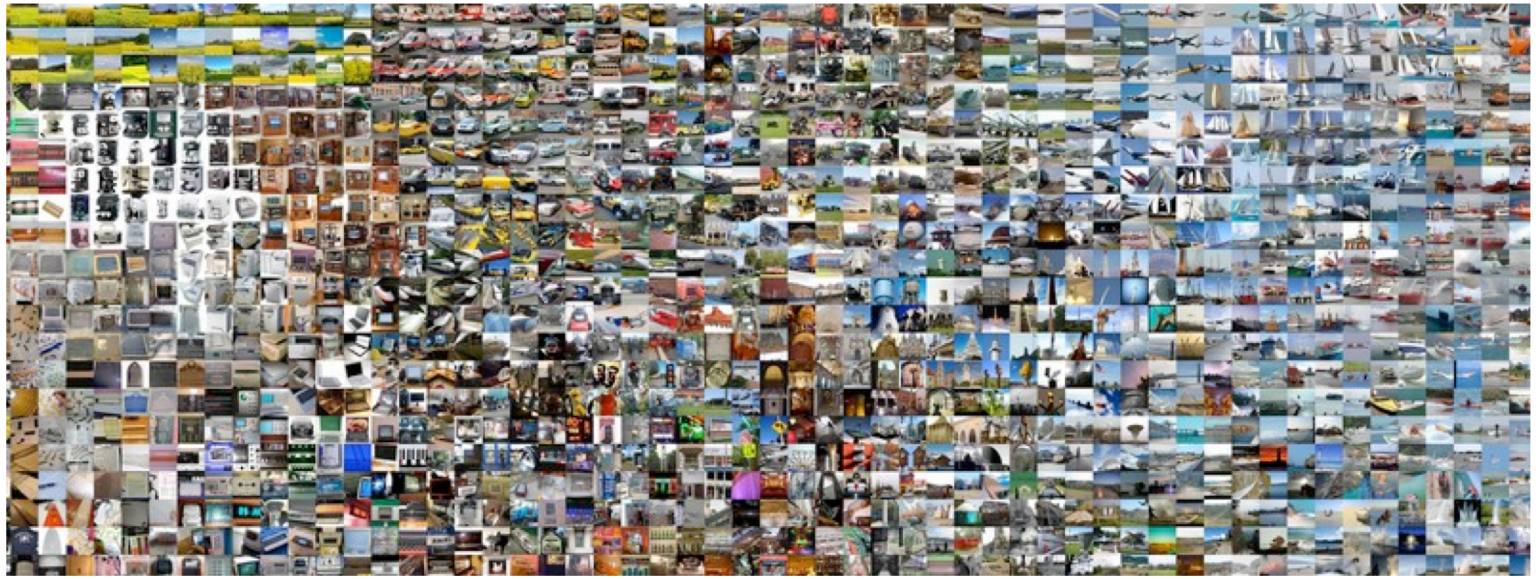
$$L(h(x), y) = -\log \hat{p}(y|x)$$

MNIST Classification

Method	Error rate (%)
Linear classifier over pixels	12

ImageNet

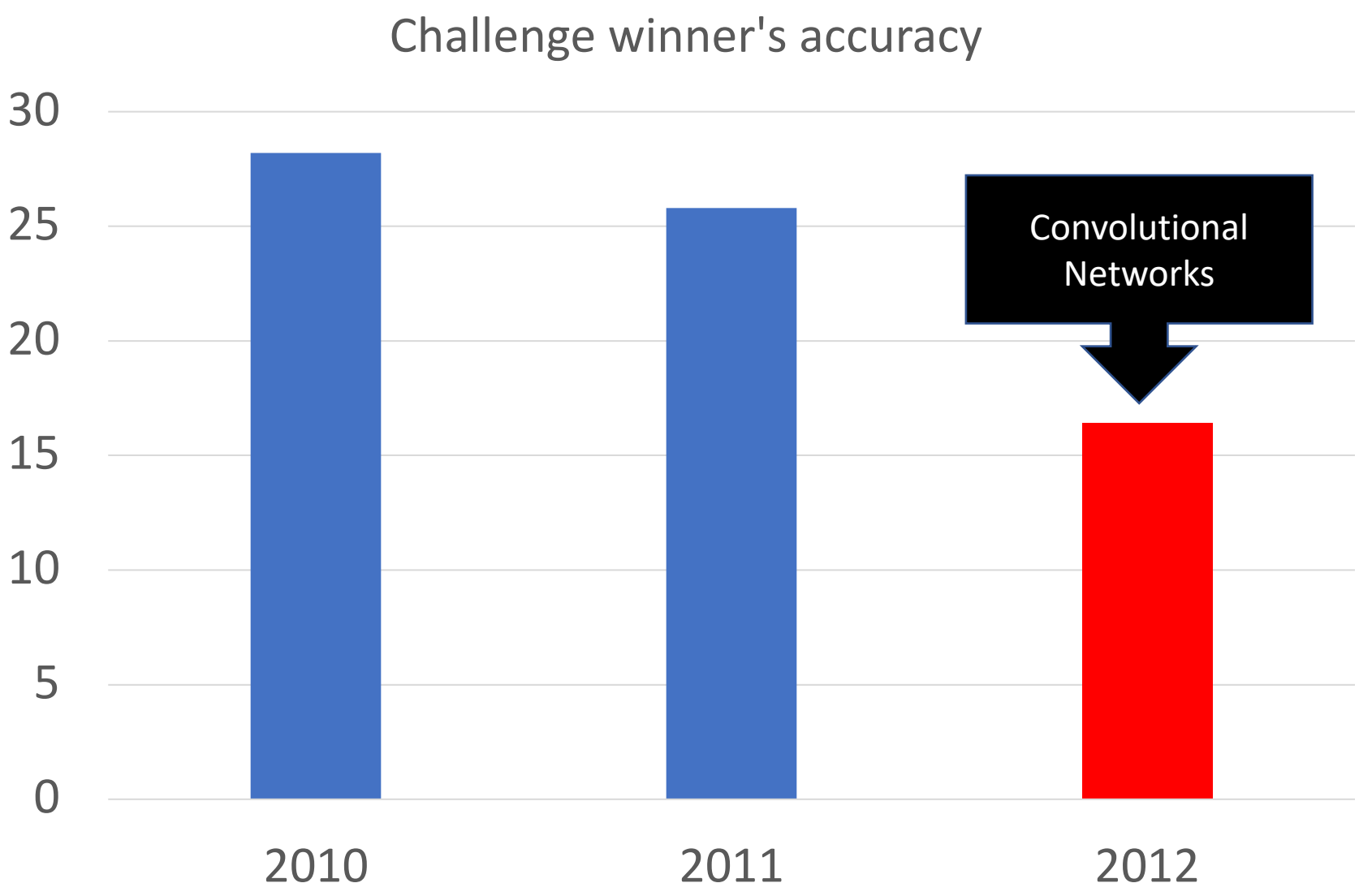
- 1000 categories
- ~1000 instances per category



Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) **ImageNet Large Scale Visual Recognition Challenge**. *International Journal of Computer Vision*, 2015.

ImageNet

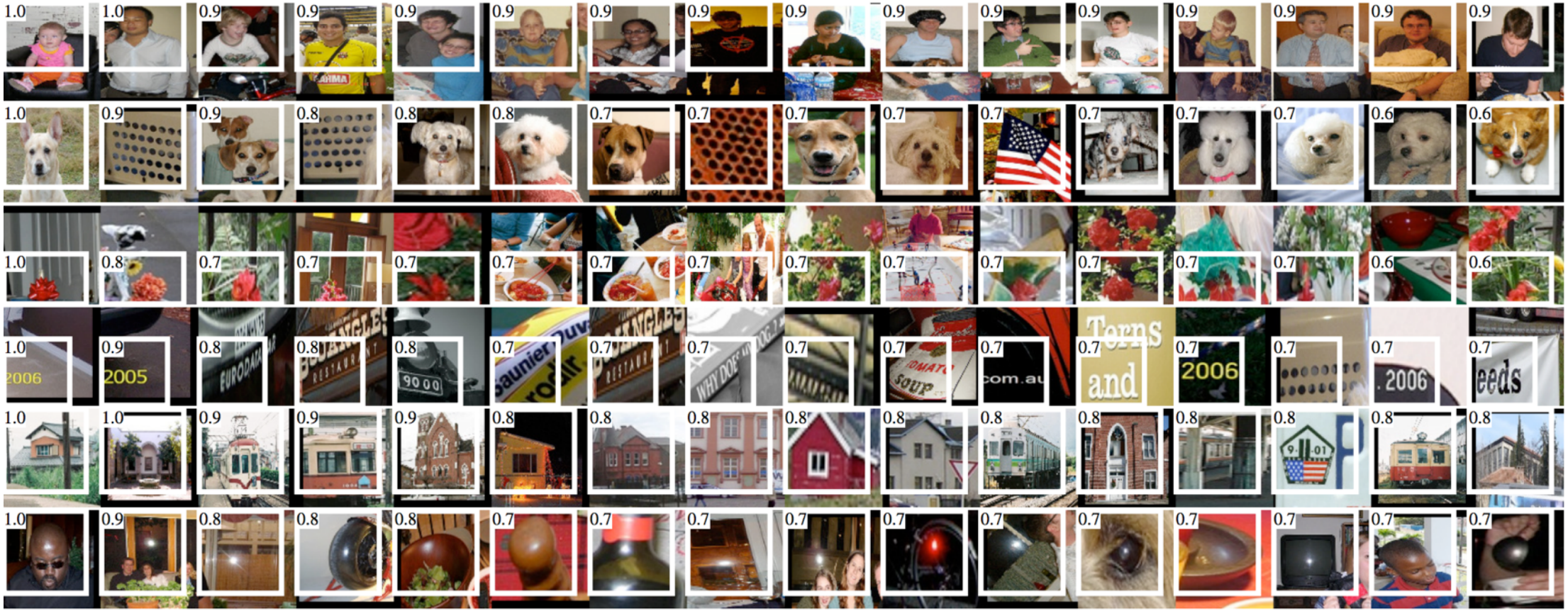
- Top-5 error: algorithm makes 5 predictions, true label must be in top 5
- Useful for incomplete labelings


Challenge winner's accuracy
30
25
20
15
10
5
0
2010
2011
2012
Convolutional Networks

Why do convnets work?

- Claim: ConvNets have way more parameters than traditional models
  - Wrong: contemporary models had same or more parameters
- Claim: Deep models are more expressive than shallow models
  - Wrong: 3 layer neural networks are *universal function approximators*
- What does depth provide?
  - More non-linearities: many ways of expressing non-linear functions
  - More module reuse: really long switch-case vs functions
  - More parameter sharing: most computation is shared amongst categories

Visualizing convolutional networks I

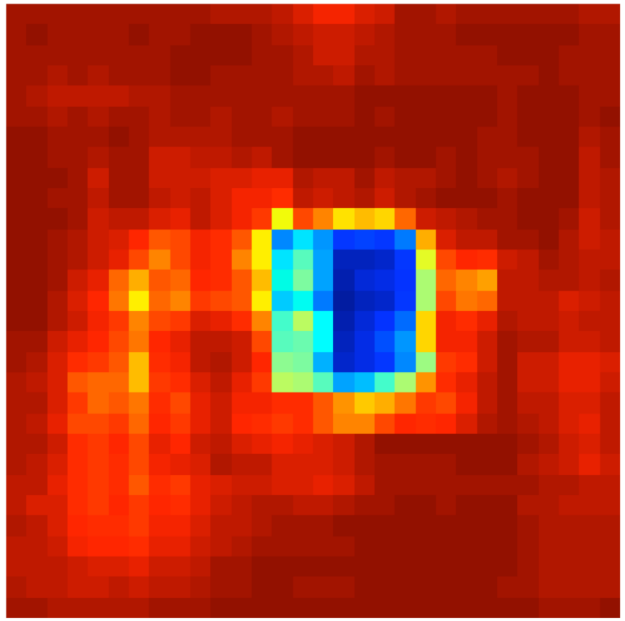


Rich feature hierarchies for accurate object detection and semantic segmentation. R. Girshick, J. Donahue, T. Darrell, J. Malik. In *CVPR,* 2014.

Visualizing convolutional networks II

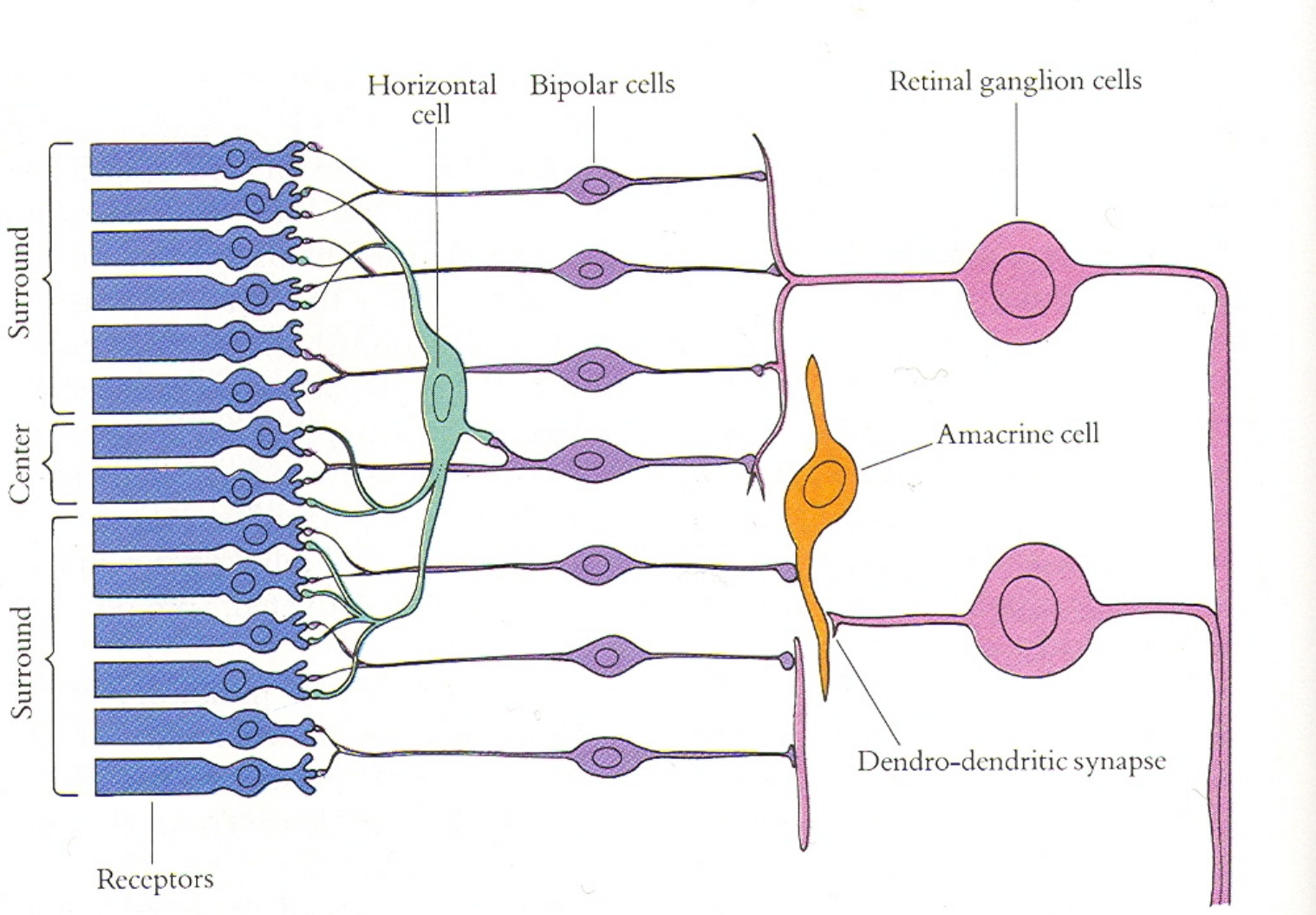
- Image pixels important for classification = pixels when blocked cause misclassification

(d) Classifier, probability of correct class



Visualizing and Understanding Convolutional Networks. M. Zeiler and R. Fergus. In *ECCV 2014.*

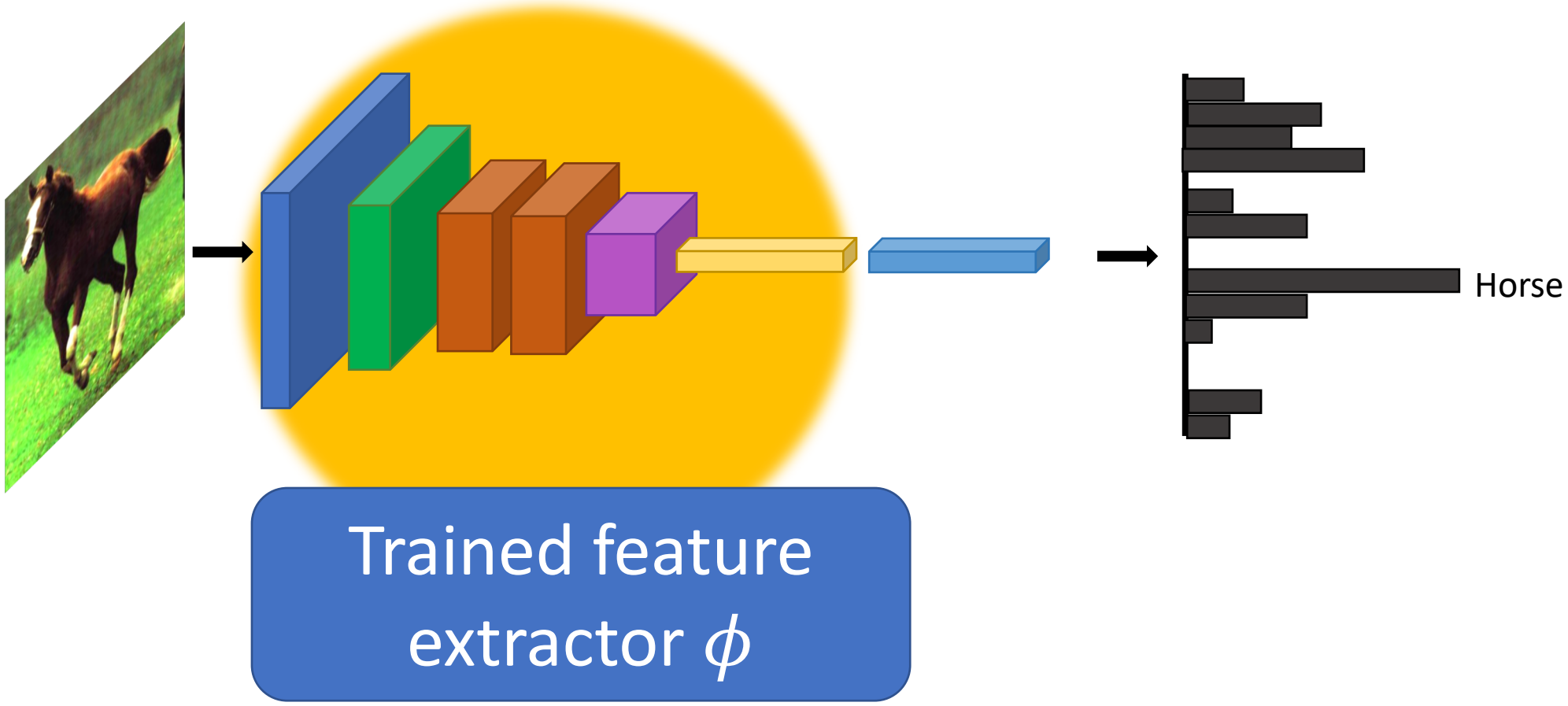
Convolutional Networks and the Brain



Slide credit: Jitendra Malik

Transfer learning

Transfer learning with convolutional networks



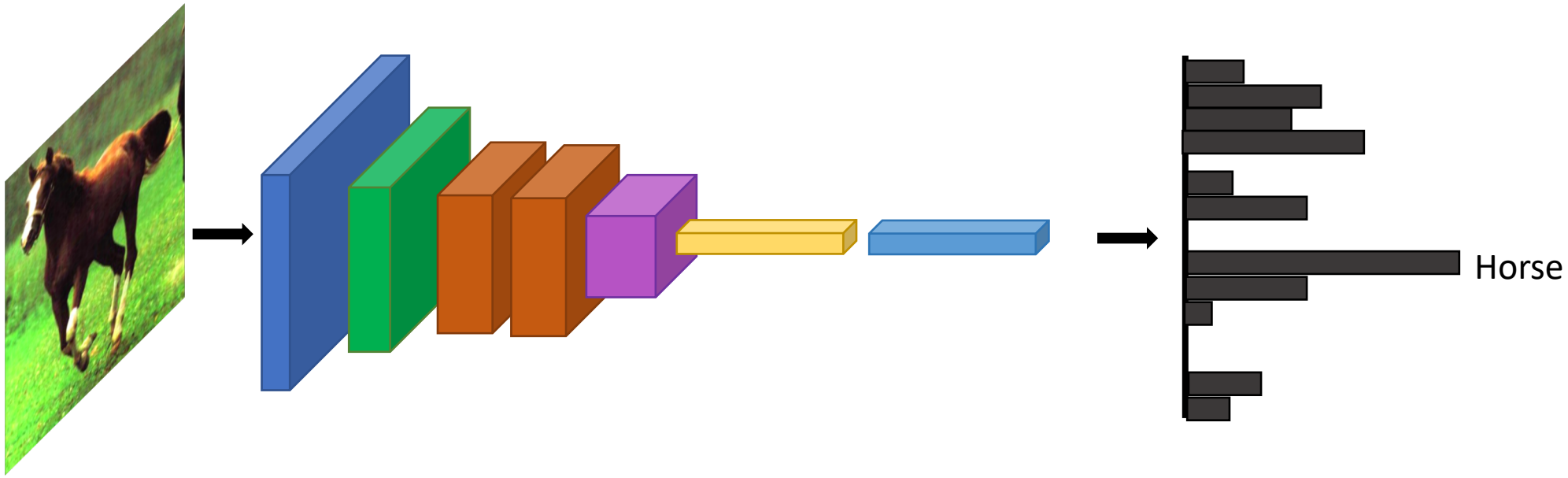
Transfer learning with convolutional networks

Dataset	Non-Convnet Method	Non-Convnet perf	Pretrained convnet + classifier	Improvement
Caltech 101	MKL	84.3	87.7	+3.4
VOC 2007	SIFT+FK	61.7	79.7	+18
CUB 200	SIFT+FK	18.8	61.0	+42.2
Aircraft	SIFT+FK	61.0	45.0	-16
Cars	SIFT+FK	59.2	36.5	-22.7

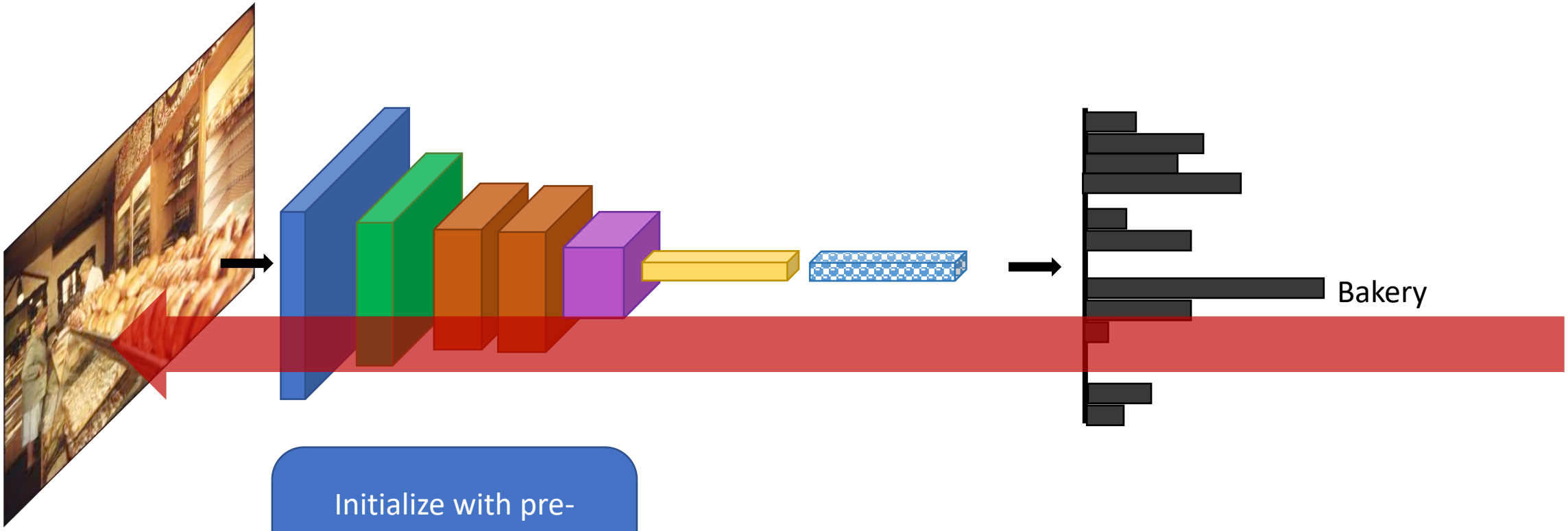
Why transfer learning?

- Availability of training data
- Computational cost
- Ability to pre-compute feature vectors and use for multiple tasks
- *Con: NO end-to-end learning*

Finetuning



Finetuning



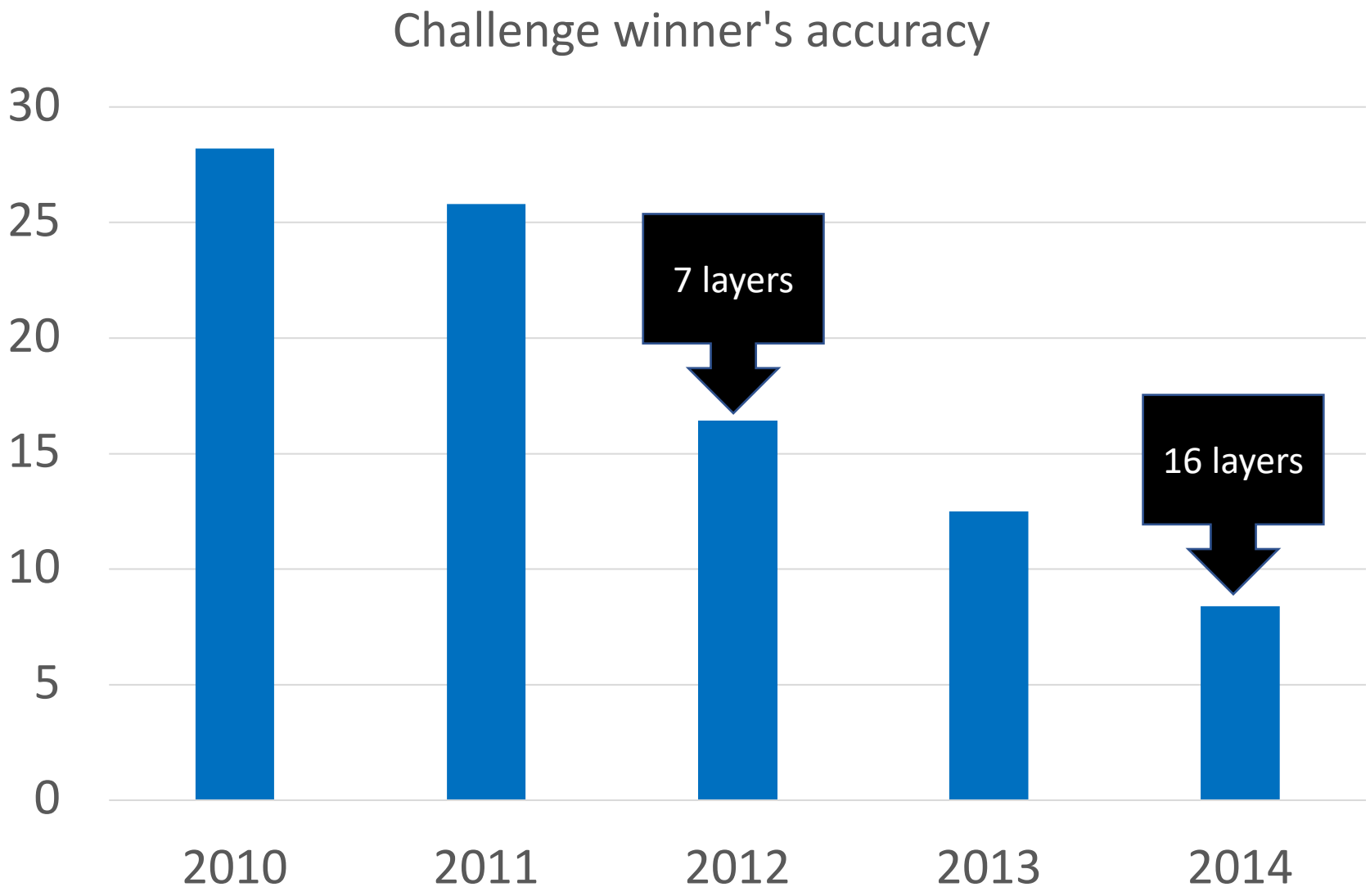
Initialize with pre-trained, then train with low learning rate

Finetuning

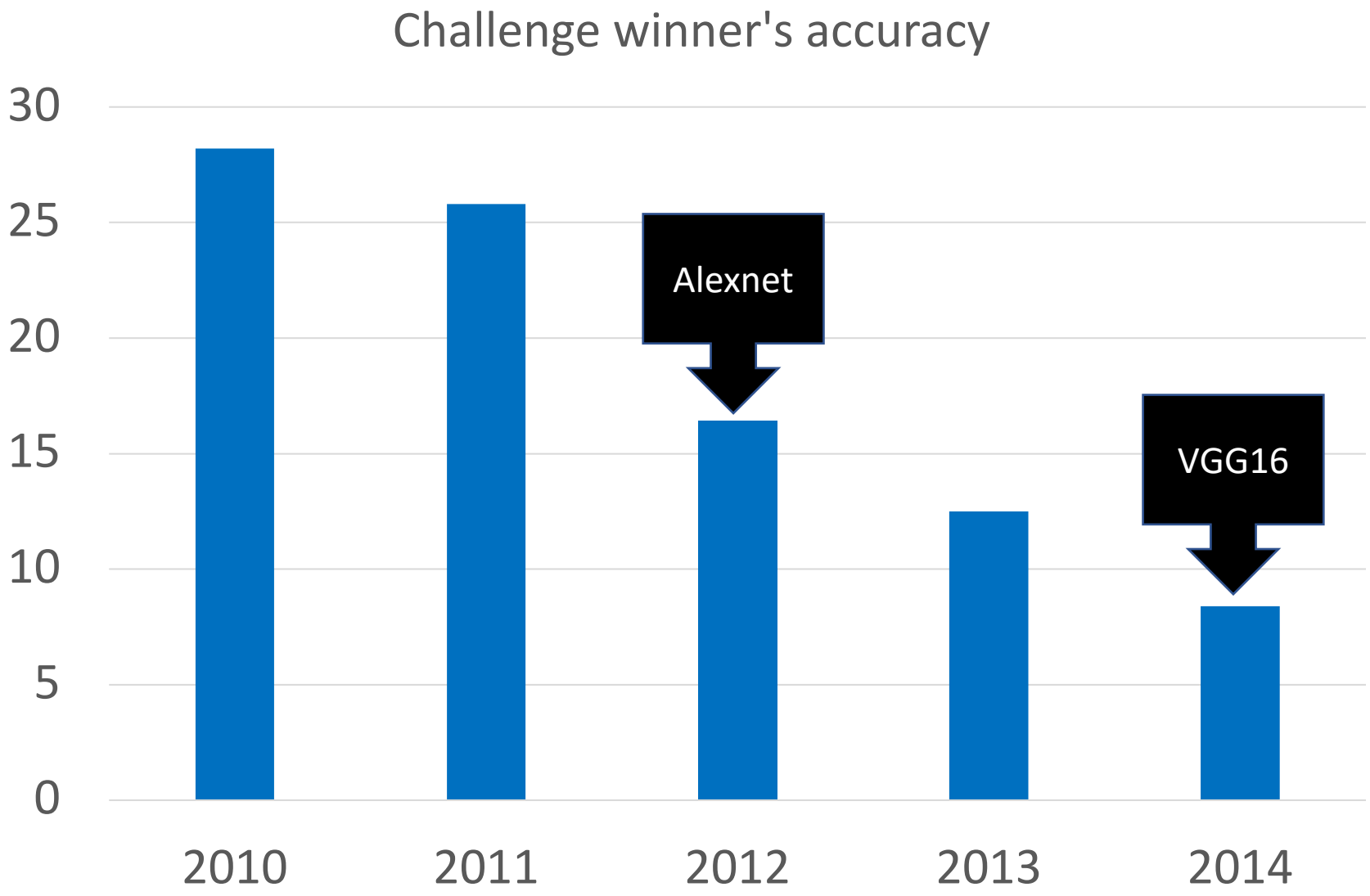
Dataset	Non-Convnet Method	Non-Convnet perf	Pretrained convnet + classifier	Finetuned convnet	Improvement
Caltech 101	MKL	84.3	87.7	88.4	+4.1
VOC 2007	SIFT+FK	61.7	79.7	82.4	+20.7
CUB 200	SIFT+FK	18.8	61.0	70.4	+51.6
Aircraft	SIFT+FK	61.0	45.0	74.1	+13.1
Cars	SIFT+FK	59.2	36.5	79.8	+20.6

Exploring convnet architectures

Deeper is better



Deeper is better



The VGG pattern

- Every convolution is 3x3, padded by 1
- Every convolution followed by ReLU
- ConvNet is divided into “stages”
  - Layers within a stage: no subsampling
  - Subsampling by 2 at the end of each stage
- Layers within stage have same number of channels
- Every subsampling → double the number of channels

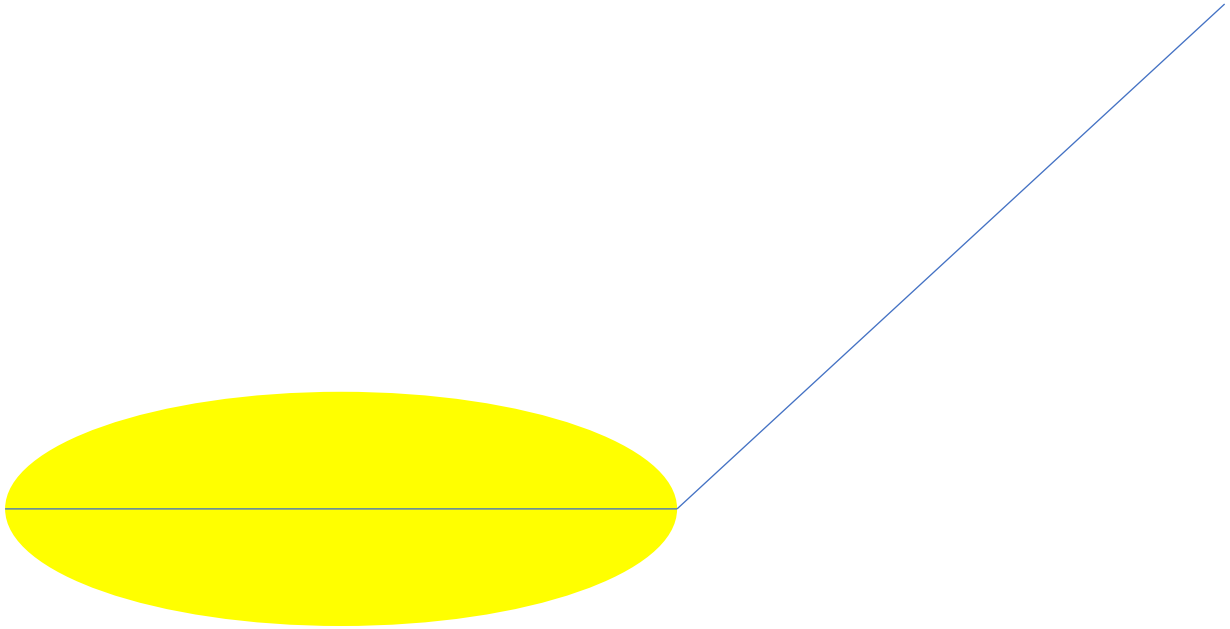
Challenges in training: exploding / vanishing gradients

- Vanishing / exploding gradients

$$\frac{\partial z}{\partial z_i} = \frac{\partial z}{\partial z_{n-1}} \frac{\partial z_{n-1}}{\partial z_{n-2}} \cdots \frac{\partial z_{i+1}}{\partial z_i}$$

  - If each term is (much) greater than 1 → *explosion of gradients*
  - If each term is (much) less than 1 → *vanishing gradients*

Challenges in training: dependence on init



Solutions

- Careful init
- Batch normalization
- Residual connections

Careful initialization

- Key idea: want variance to remain approx. constant
  - Variance increases in backward pass => exploding gradient
  - Variance decreases in backward pass => vanishing gradient
- “MSRA initialization”
  - weights = Gaussian with 0 mean and variance = 2/(k*k*d)

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. K. He, X. Zhang, S. Ren, J. Sun

Batch normalization

- Key idea: normalize so that each layer output has zero mean and unit variance
  - Compute mean and variance for each channel
  - Aggregate over batch
  - Subtract mean, divide by std
- Need to reconcile train and test
  - No ”batches” during test
  - After training, compute means and variances on train set and store

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. S. Ioffe, C. Szegedy. In *ICML,* 2015.

Residual connections

- In general, gradients tend to vanish
- Key idea: allow gradients to flow unimpeded

$$z_{i+1} = f_{i+1}(z_i, w_{i+1})$$

$$\frac{\partial z_{i+1}}{\partial z_i} = \frac{\partial f_{i+1}(z_i, w_{i+1})}{\partial z_i}$$

$$\frac{\partial z}{\partial z_i} = \frac{\partial z}{\partial z_{n-1}} \frac{\partial z_{n-1}}{\partial z_{n-2}} \cdots \frac{\partial z_{i+1}}{\partial z_i}$$

Residual connections

- In general, gradients tend to vanish
- Key idea: allow gradients to flow unimpeded

$$z_{i+1} = g_{i+1}(z_i, w_{i+1}) + z_i \qquad \frac{\partial z_{i+1}}{\partial z_i} = \frac{\partial g_{i+1}(z_i, w_{i+1})}{\partial z_i} + I$$

$$\frac{\partial z}{\partial z_i} = \frac{\partial z}{\partial z_{n-1}} \frac{\partial z_{n-1}}{\partial z_{n-2}} \cdots \frac{\partial z_{i+1}}{\partial z_i}$$

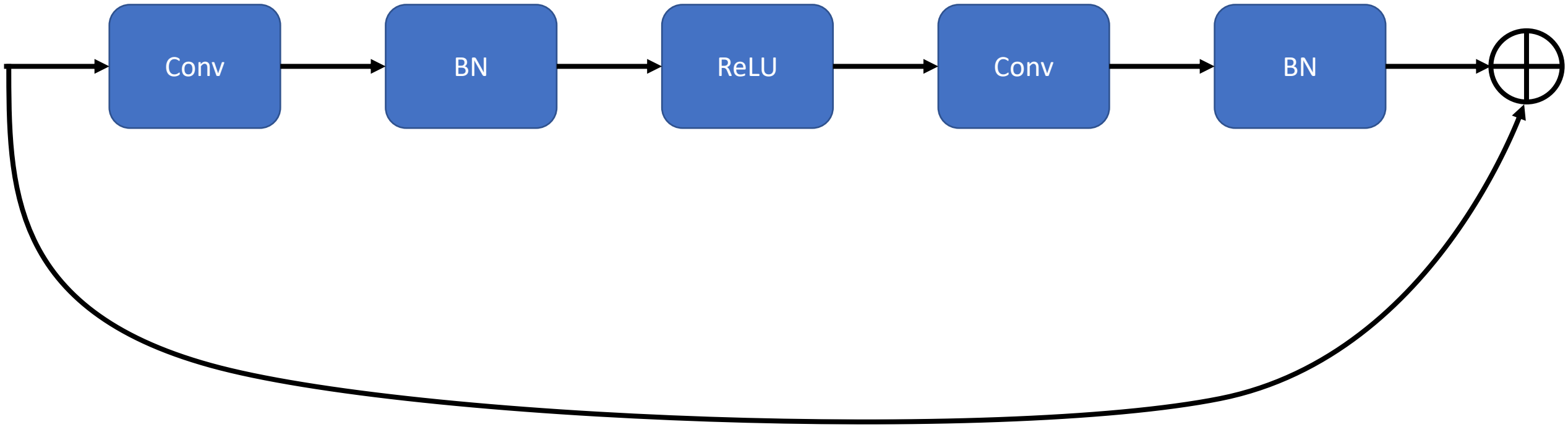
Residual connections

- Assumes all z_i have the same size
- True within a stage
- Across stages?
  - Doubling of feature channels
  - Subsampling
- Increase channels by 1x1 convolution
- Decrease spatial resolution by subsampling

$$z_{i+1} = g_{i+1}(z_i, w_{i+1}) + \mathrm{subsample}(W z_i)$$

A residual block

- Instead of single layers, have residual connections over block



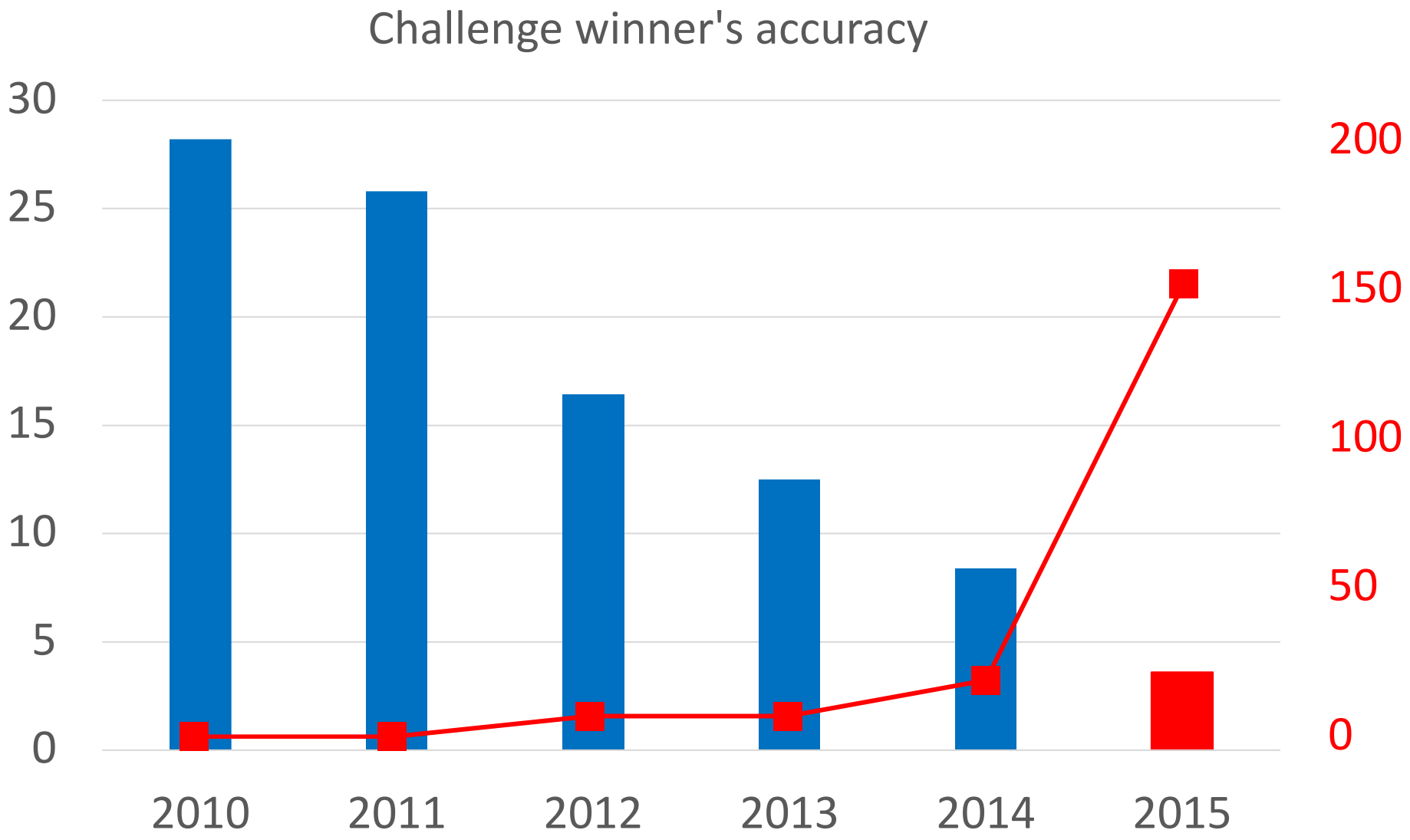
Bottleneck blocks

- Problem: When channels increases, 3x3 convolutions introduce many parameters
  - $3x3xc^2$
- Key idea: use 1x1 to project to lower dimensionality, do convolution, then come back
  - $cxd + 3x3xd^2 + dxc$

The ResNet pattern

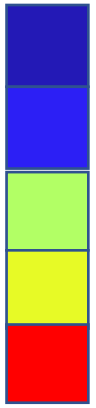
- Decrease resolution substantially in first layer
  - Reduces memory consumption due to intermediate outputs
- Divide into stages
  - maintain resolution, channels in each stage
  - halve resolution, double channels between stages
- Divide each stage into residual blocks
- At the end, compute average value of each channel to feed linear classifier

Putting it all together - Residual networks



Semantic Segmentation

The Task



person

grass

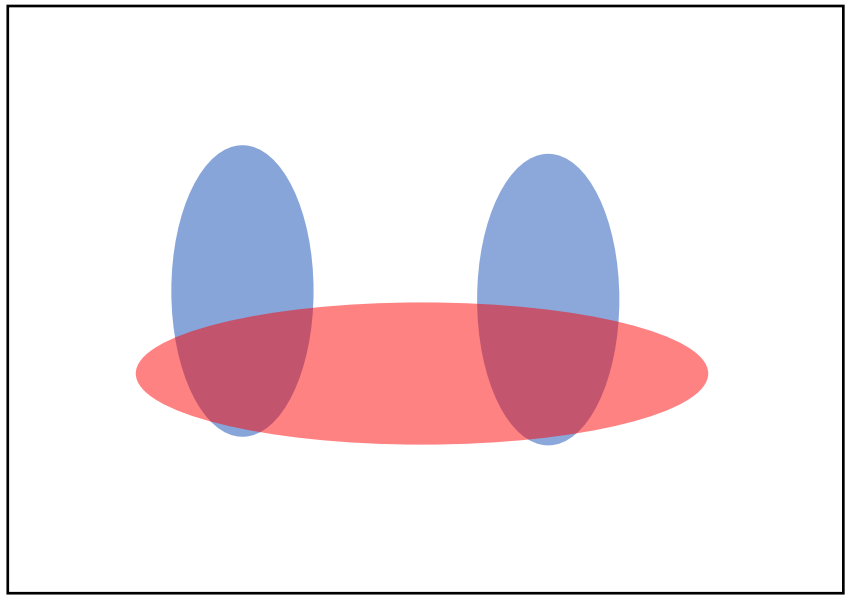
trees

motorbike

road

Evaluation metric

- Pixel classification!
- Accuracy?
  - Heavily unbalanced
- *Intersection over Union*
  - Average across classes and images
- Per-class accuracy
  - Average across classes and images



Things vs Stuff

THINGS

- Person, cat, horse, etc
- Constrained shape
- Individual instances with separate identity
- May need to look at objects

STUFF

- Road, grass, sky etc
- Amorphous, no shape
- No notion of instances
- Can be done at pixel level
- “texture”

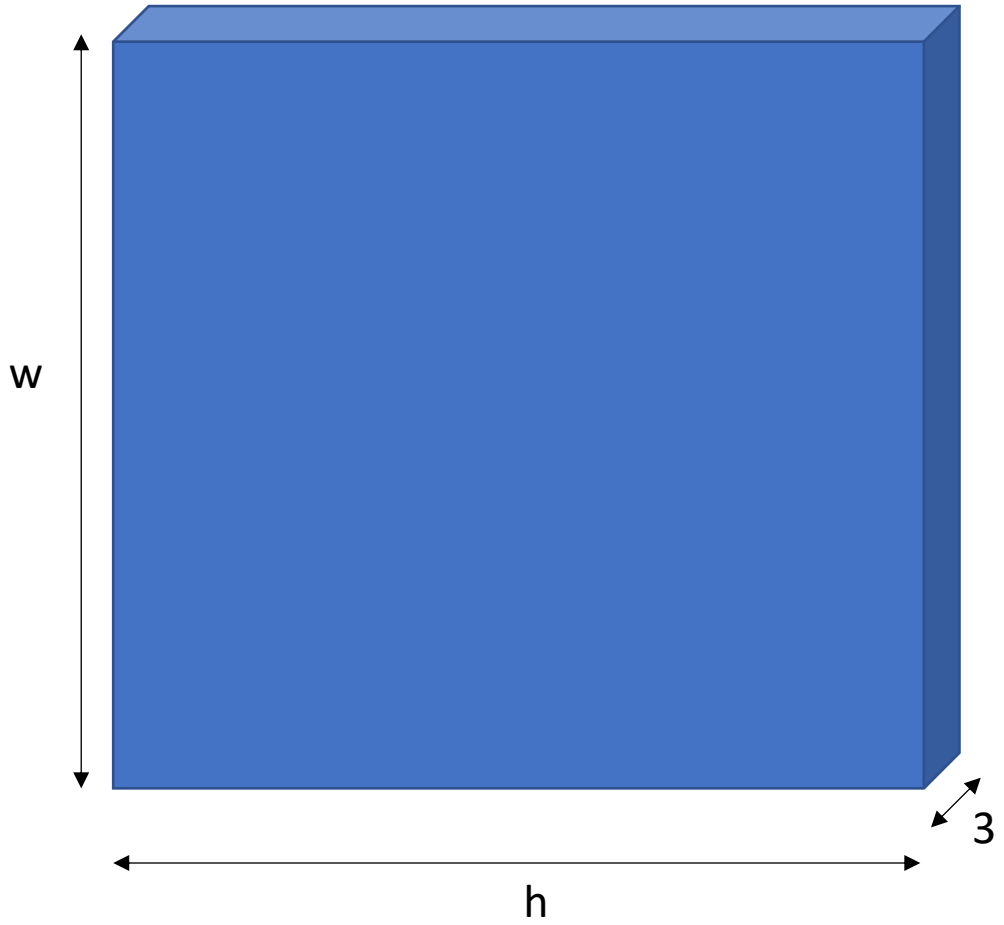
Challenges in data collection

- Precise localization is hard to annotate
- Annotating every pixel leads to heavy tails
- Common solution: annotate few classes (often things), mark rest as “Other”
- Common datasets: PASCAL VOC 2012 (~1500 images, 20 categories), COCO (~100k images, 20 categories)

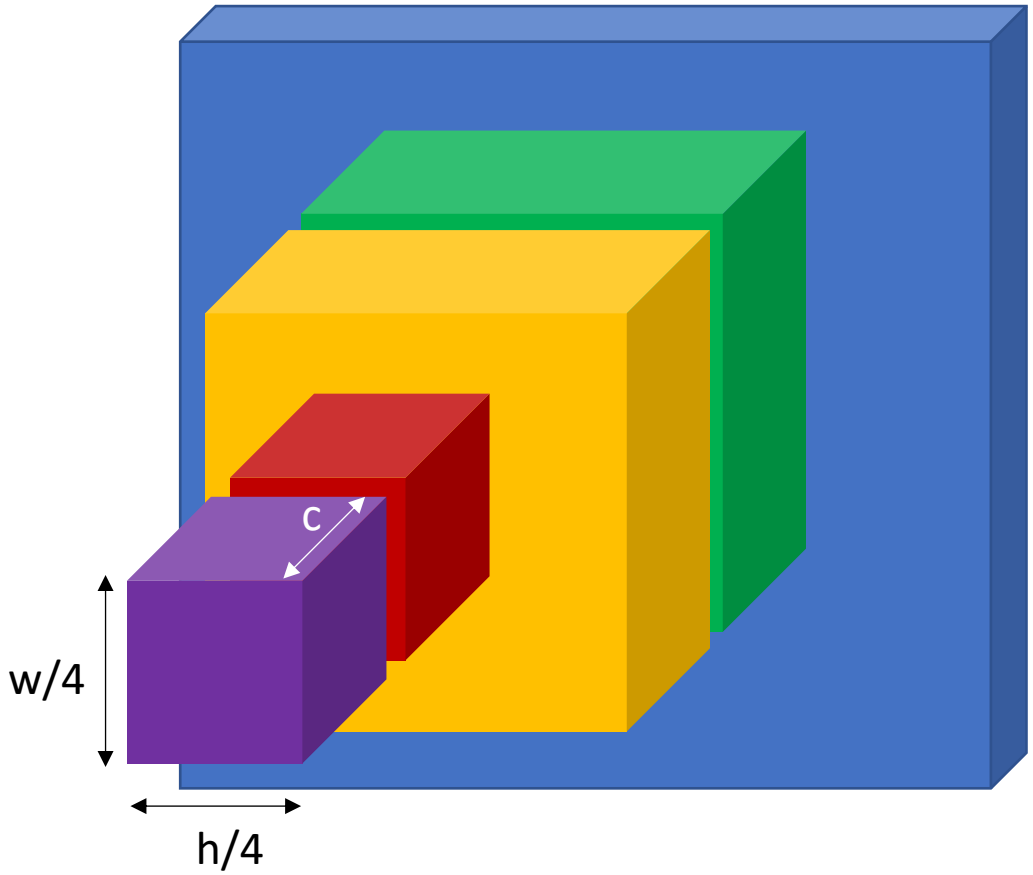
Pre-convnet semantic segmentation

- Things
  - Do object detection, then segment out detected objects
- Stuff
  - ”Texture classification”
  - Compute histograms of filter responses
  - Classify local image patches

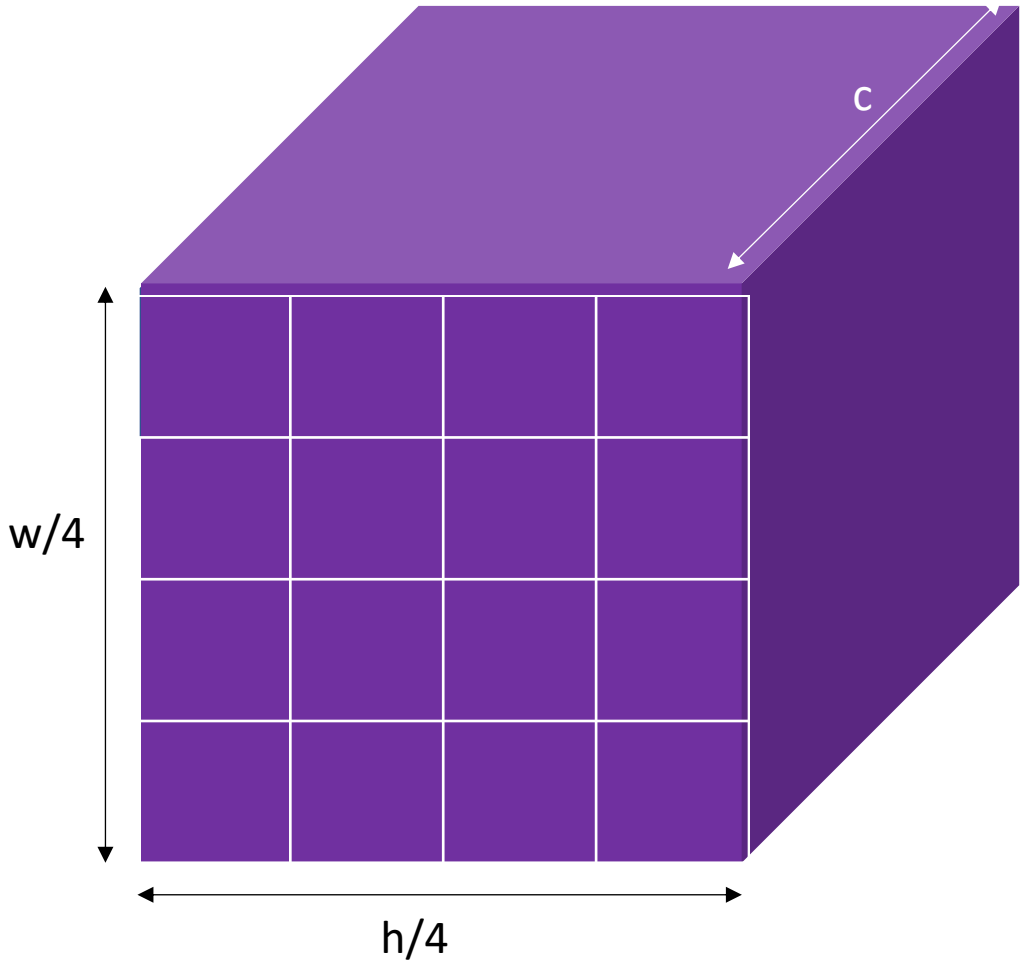
Semantic segmentation using convolutional networks



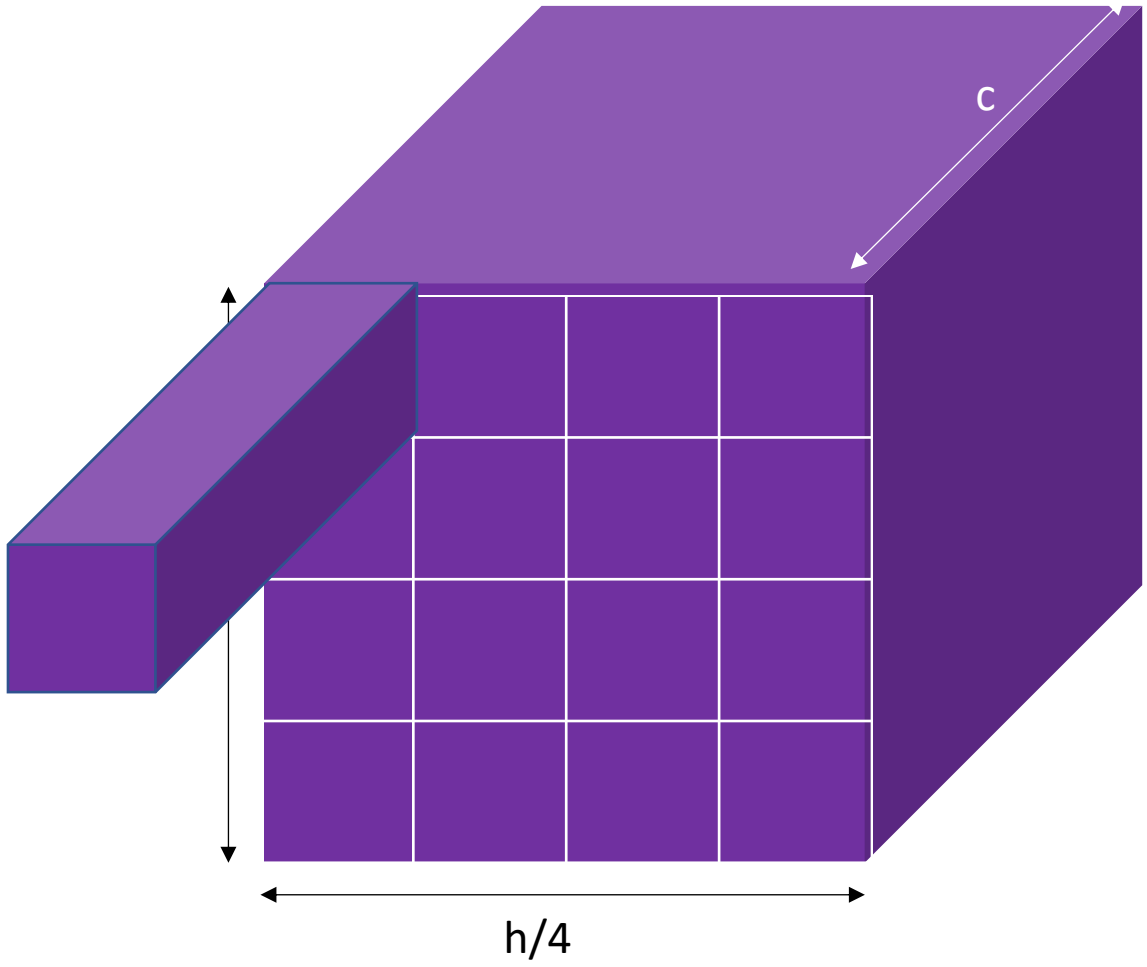
Semantic segmentation using convolutional networks



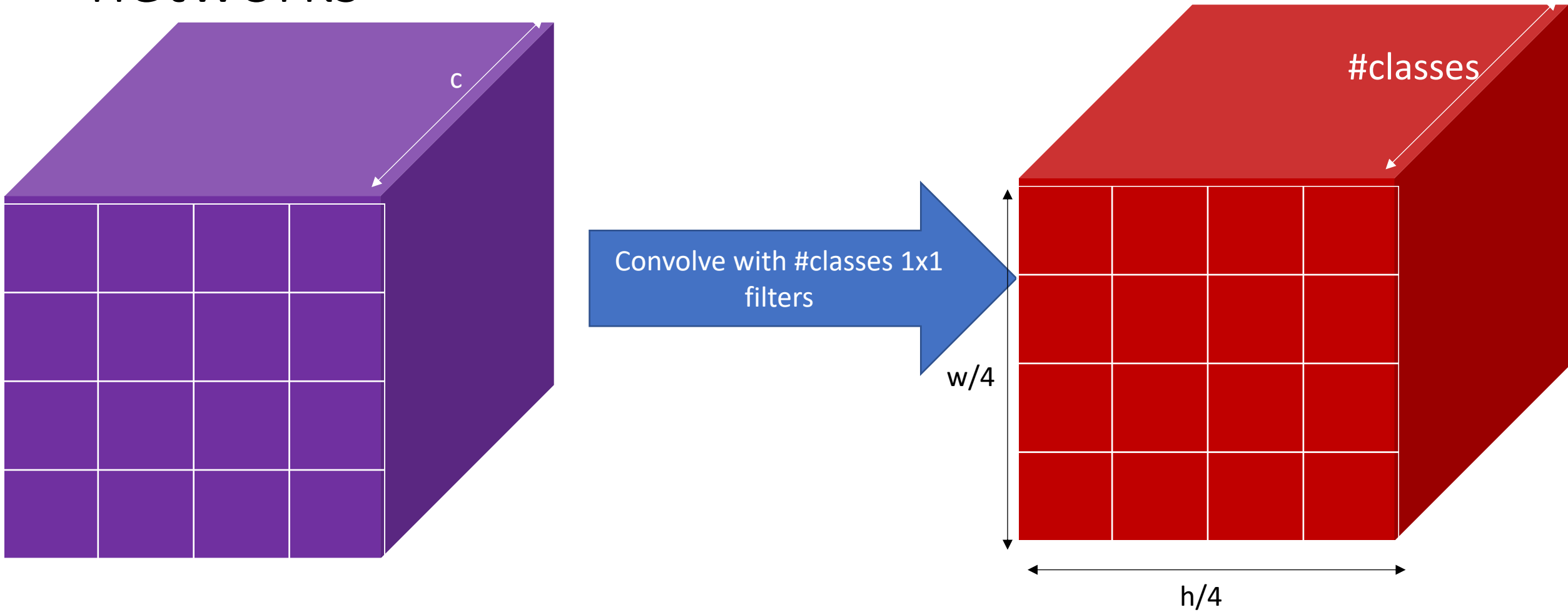
Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks

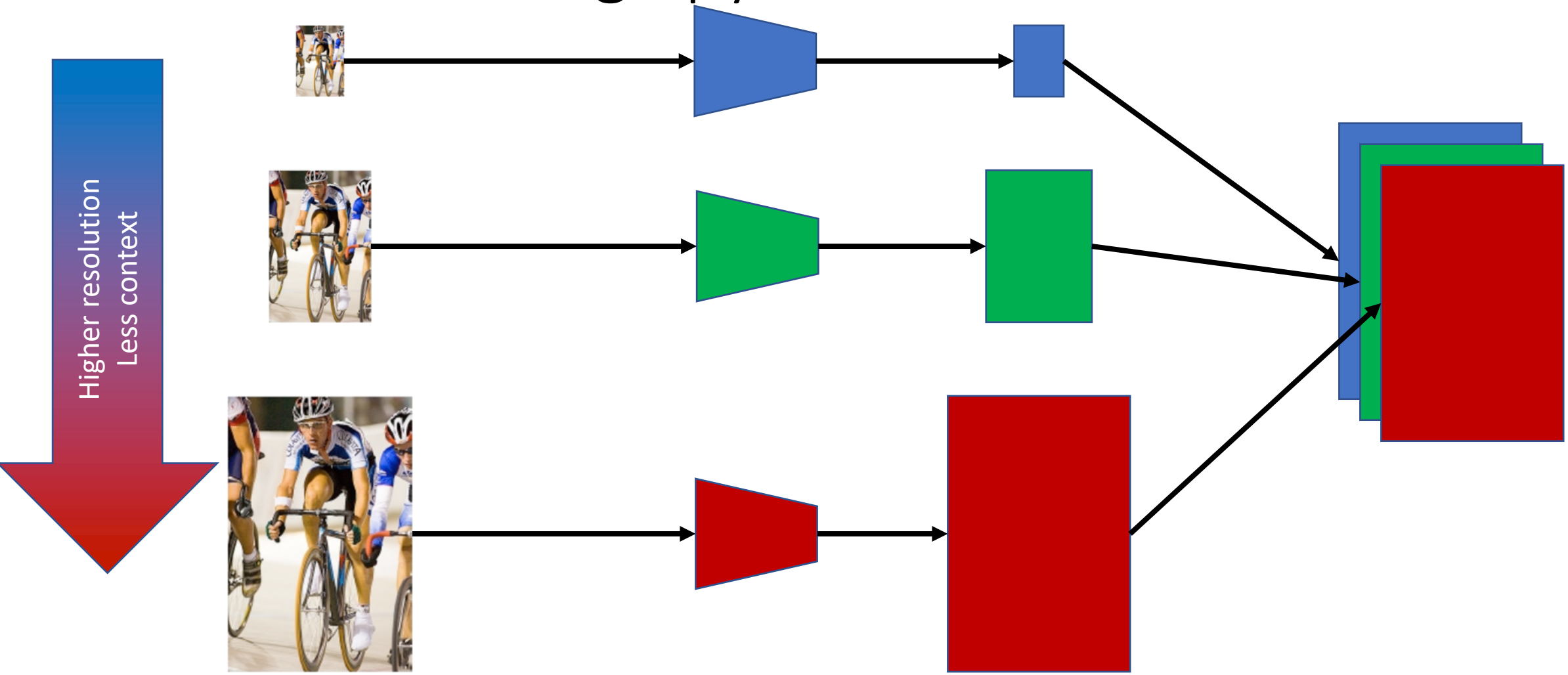
- Pass image through convolution and subsampling layers
- Final convolution with #classes outputs
- Get scores for *subsampled* image
- Upsample back to original size

Semantic segmentation using convolutional networks

The resolution issue

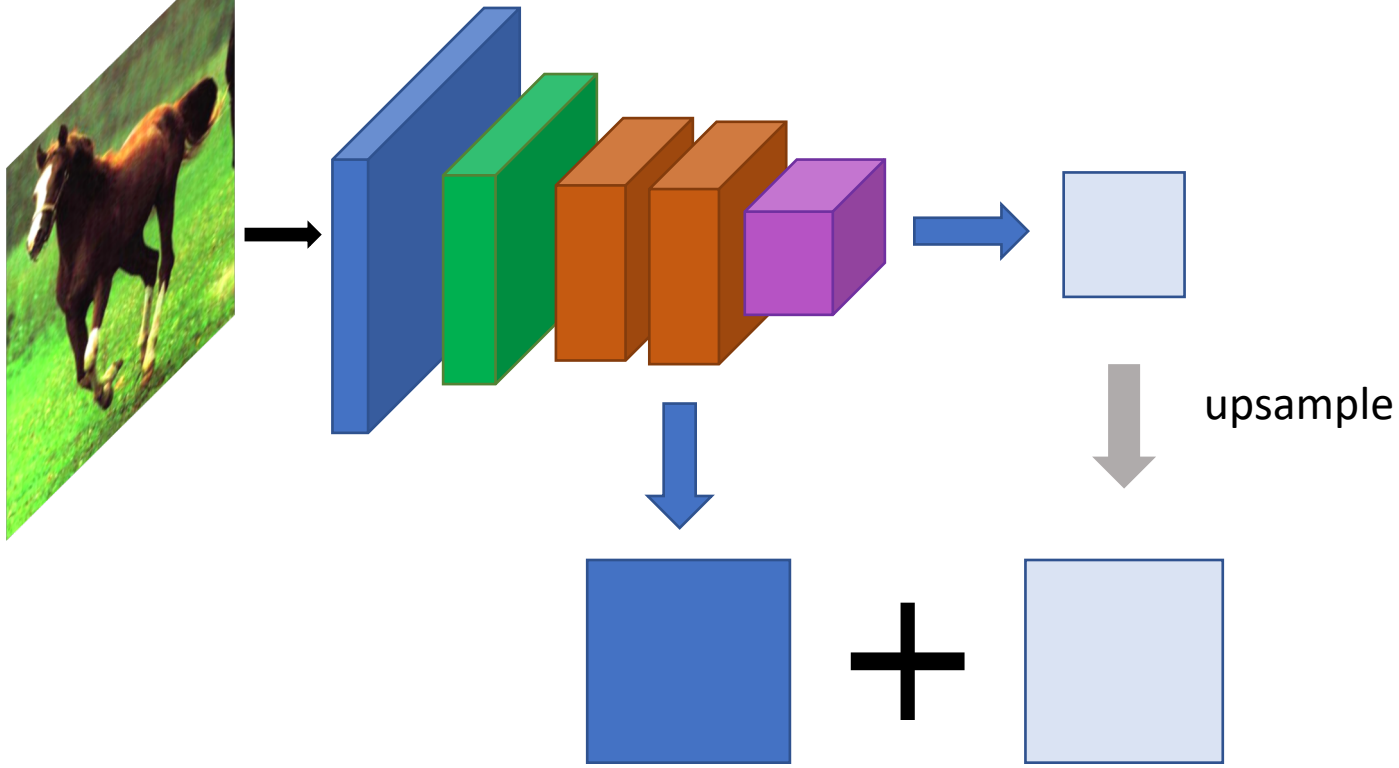
- Problem: Need fine details!
- Shallower network / earlier layers?
  - Not very semantic!
- Remove subsampling?
  - Looks at only a small window!

Solution 1: Image pyramids

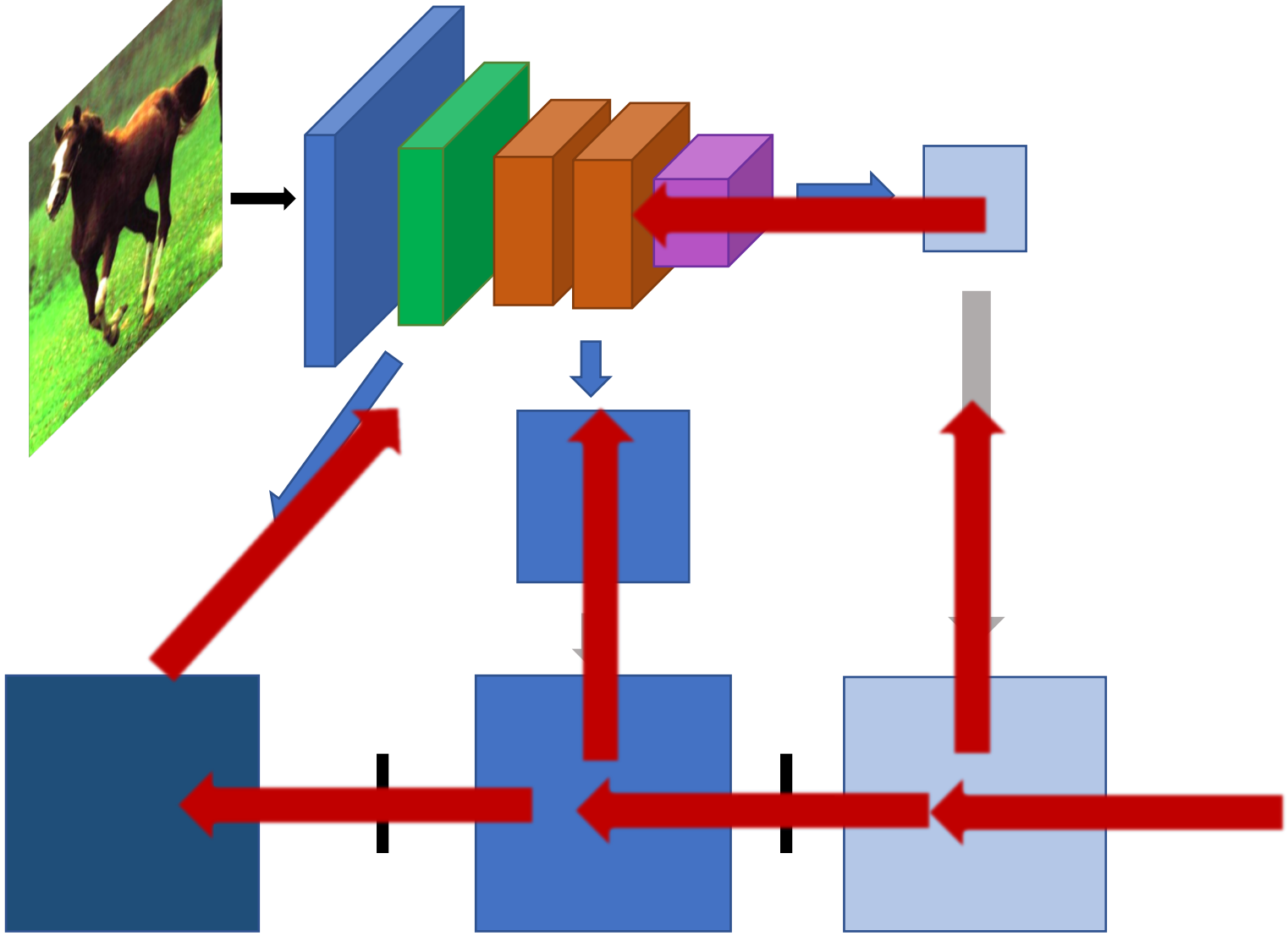


Learning Hierarchical Features for Scene Labeling. Clement Farabet, Camille Couprie, Laurent Najman, Yann LeCun. In *TPAMI,* 2013.

Solution 2: Skip connections



Solution 2: Skip connections

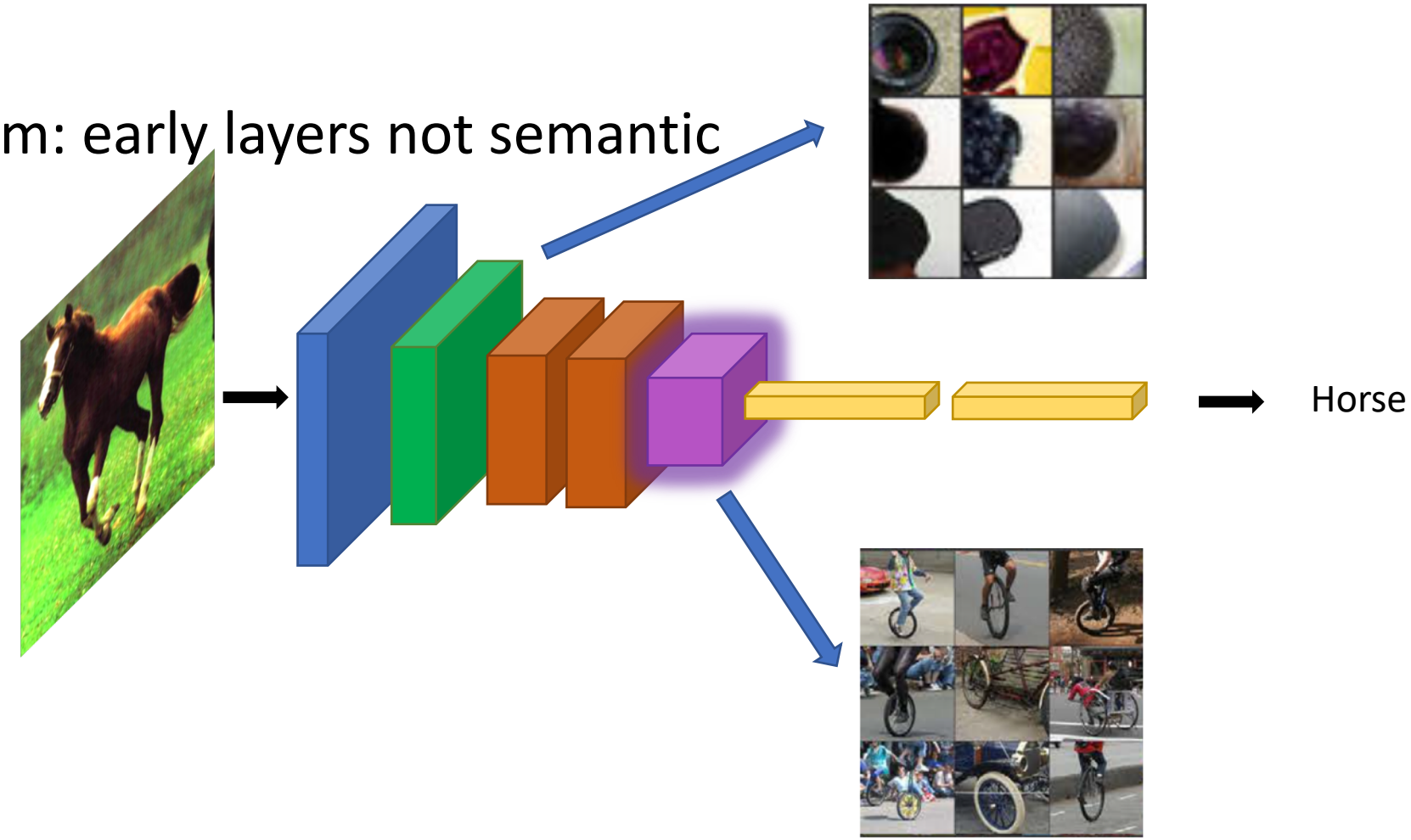


Skip connections

Fully convolutional networks for semantic segmentation. Evan Shelhamer, Jon Long, Trevor Darrell. In *CVPR* 2015

Skip connections

- Problem: early layers not semantic



Visualizations from : M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV* 2014.