Computer Vision

ASAI-ER 2023

SAMUELE SALTI

samuele.salti@unibo.it

What is Computer Vision?

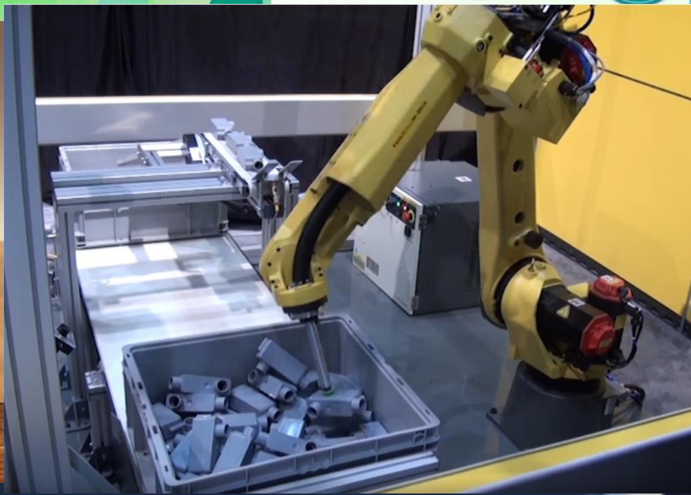
The science (and art) of making computers gain a high-level understanding of images (and videos, and 3D data, and ...)

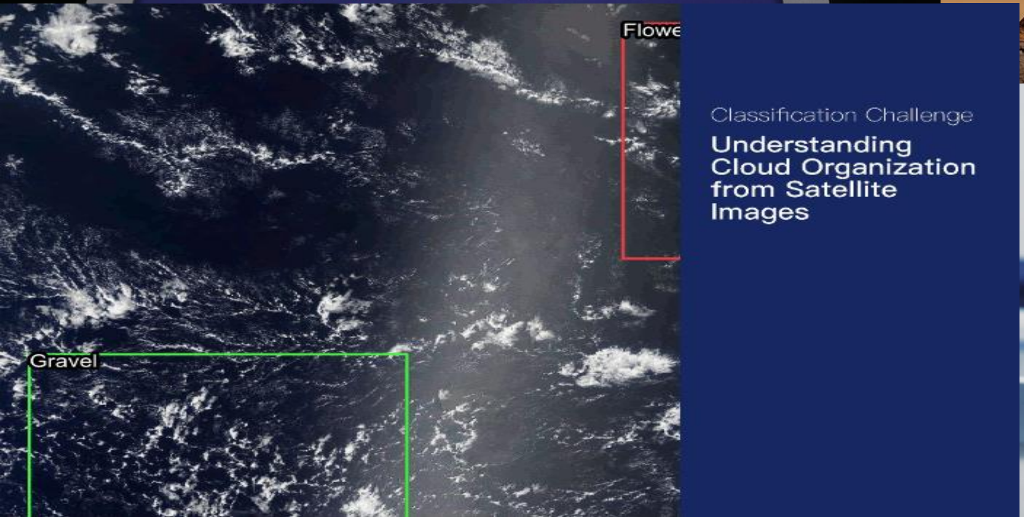
Example credit: Andrej Karpathy
https://karpathy.github.io/2012/10/22/state-of-computer-vision/

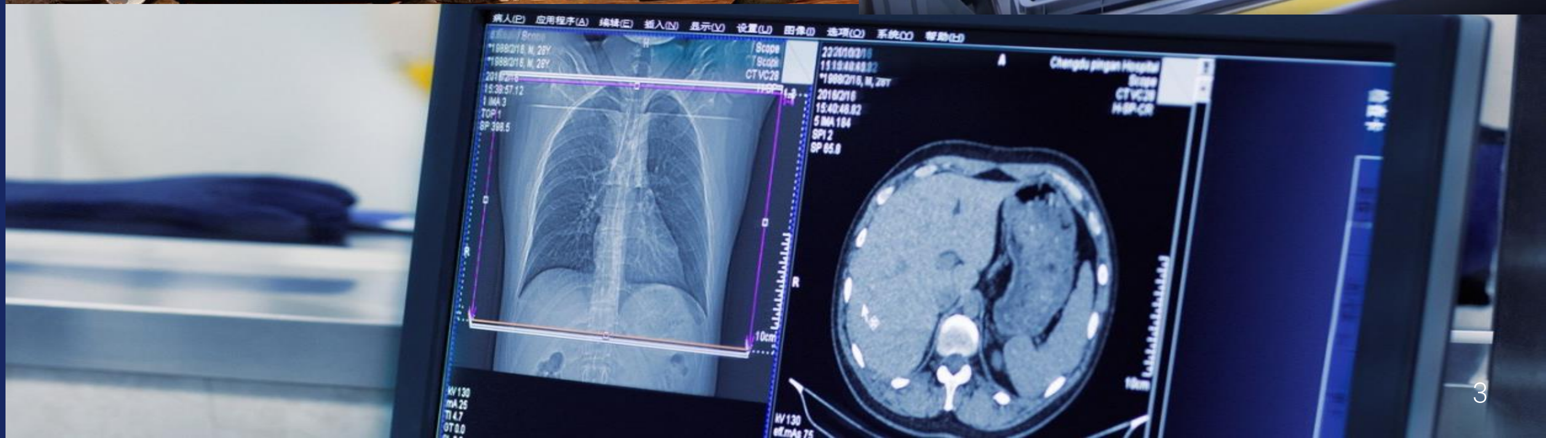

Electabuzz CP 270
AR


POKÉMON
53 / 250
EGGS
9 / 9
CP 485 Hypno
CP 469 Gloom
CP 432 Pidgeot
CP 412 Jynx
CP 408 Golbat
CP 401 Magneton
CP 367 Hitmonlee
CP 361 Electabuzz
CP 327 Jynx
CP 326 Raticate
CP 248
CP 238
CP


AlphaGo
Lee Sedol




Flowe
Gravel
Classification Challenge
Understanding Cloud Organization from Satellite Images

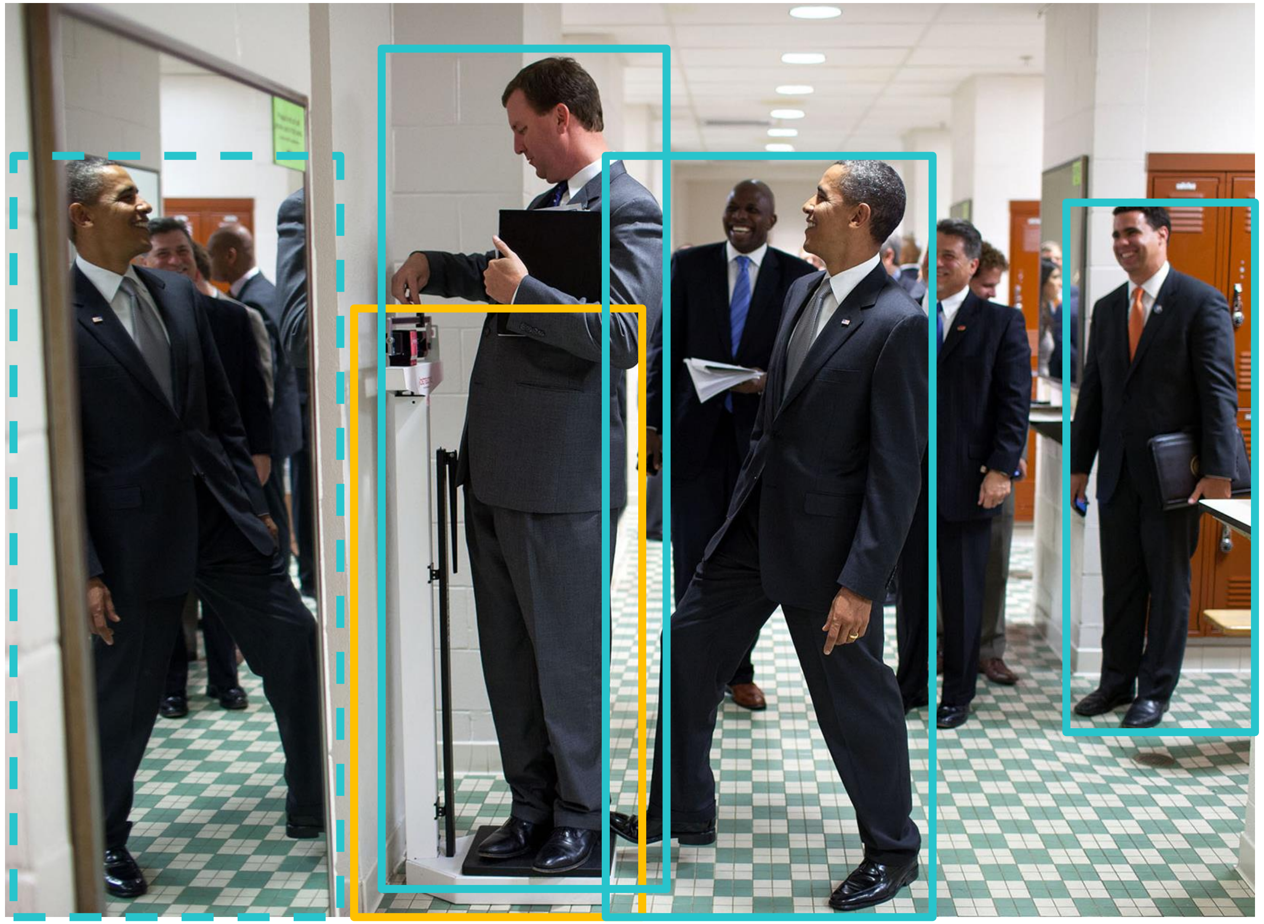


What does an image tell you?

3D structure, place classification

What does an image tell you?

Interesting "objects"



What does an image tell you?

Object boundaries

What does an image tell you?

Relative depth placement

What does an image tell you?

Named entity recognition

What does an image tell you?

A story

The previous slide has not aged well…

This is a picture of Barack Obama. He is a former president of the United States.

How many people are there in this picture?

There are at least 5 people in this picture.

Where was this picture taken?

It was taken in a school.

How many mirrors are there?

There are at least 2 mirrors in this picture.

What is the person standing on?

The person is standing on a rug.

He is standing on a scale.

I think you are right.

What is he doing?

He is looking at the scale.

Where is Obama's foot positioned?

Obama's foot is positioned on the right side of the scale.

What happens as a result?

The scale shows a higher weight.

Is the person on the scale aware of it?

I think he is not aware of it.

Do you think that's why people are laughing?

I think so.

https://twitter.com/inoryy/status/1522621712382234624
https://www.deepmind.com/blog/tackling-multiple-tasks-with-a-single-visual-language-model

…or has it?

This is a picture of Barack Obama. He is a former president of the United States.

How many giraffes are there?

There are two giraffes.

…but….

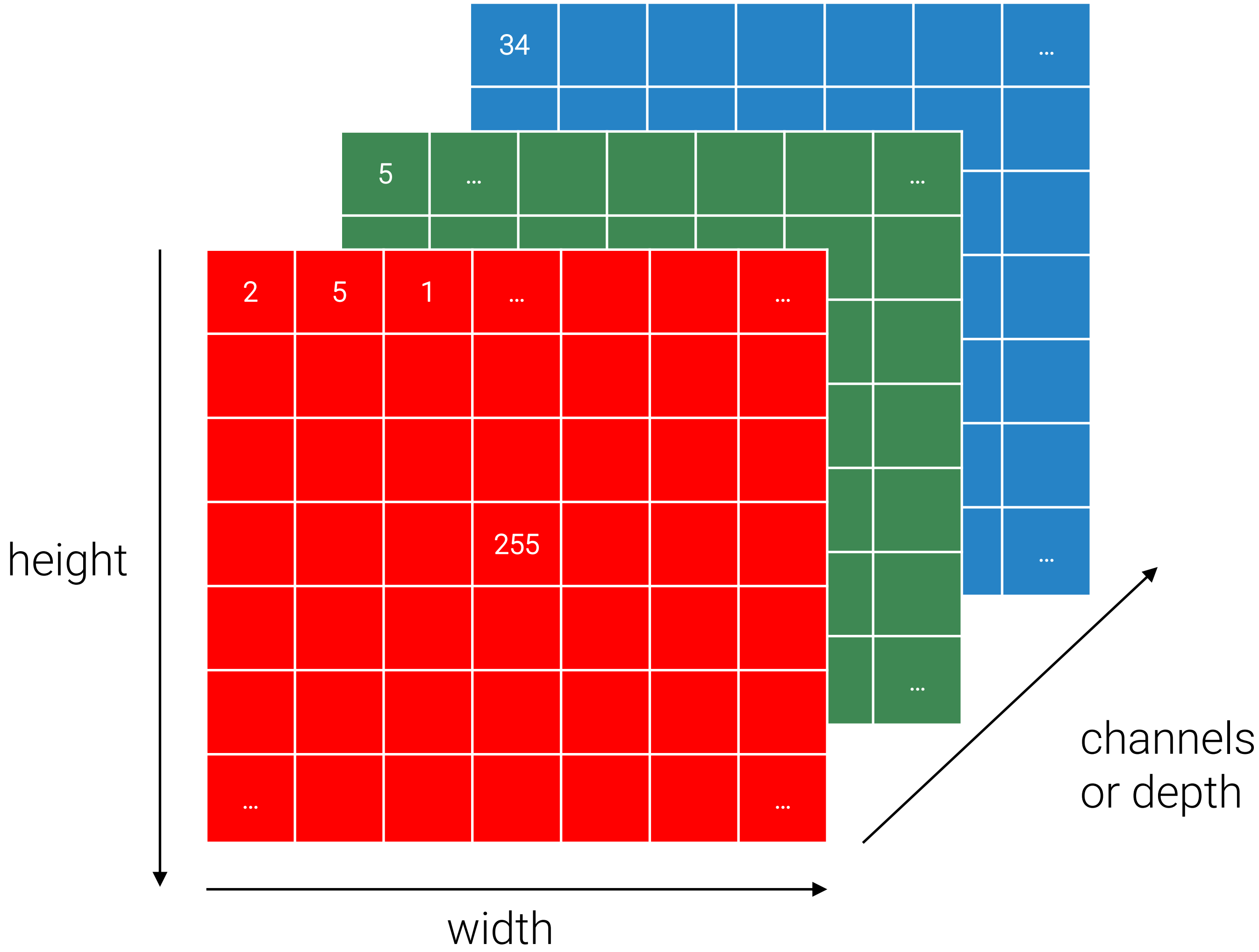
This is a picture of Barack Obama. He is a former president of the United States.

How many giraffes are there in the image?

I can't see any giraffes in the image.

https://twitter.com/geoffreyirving/status/1522669461278539777

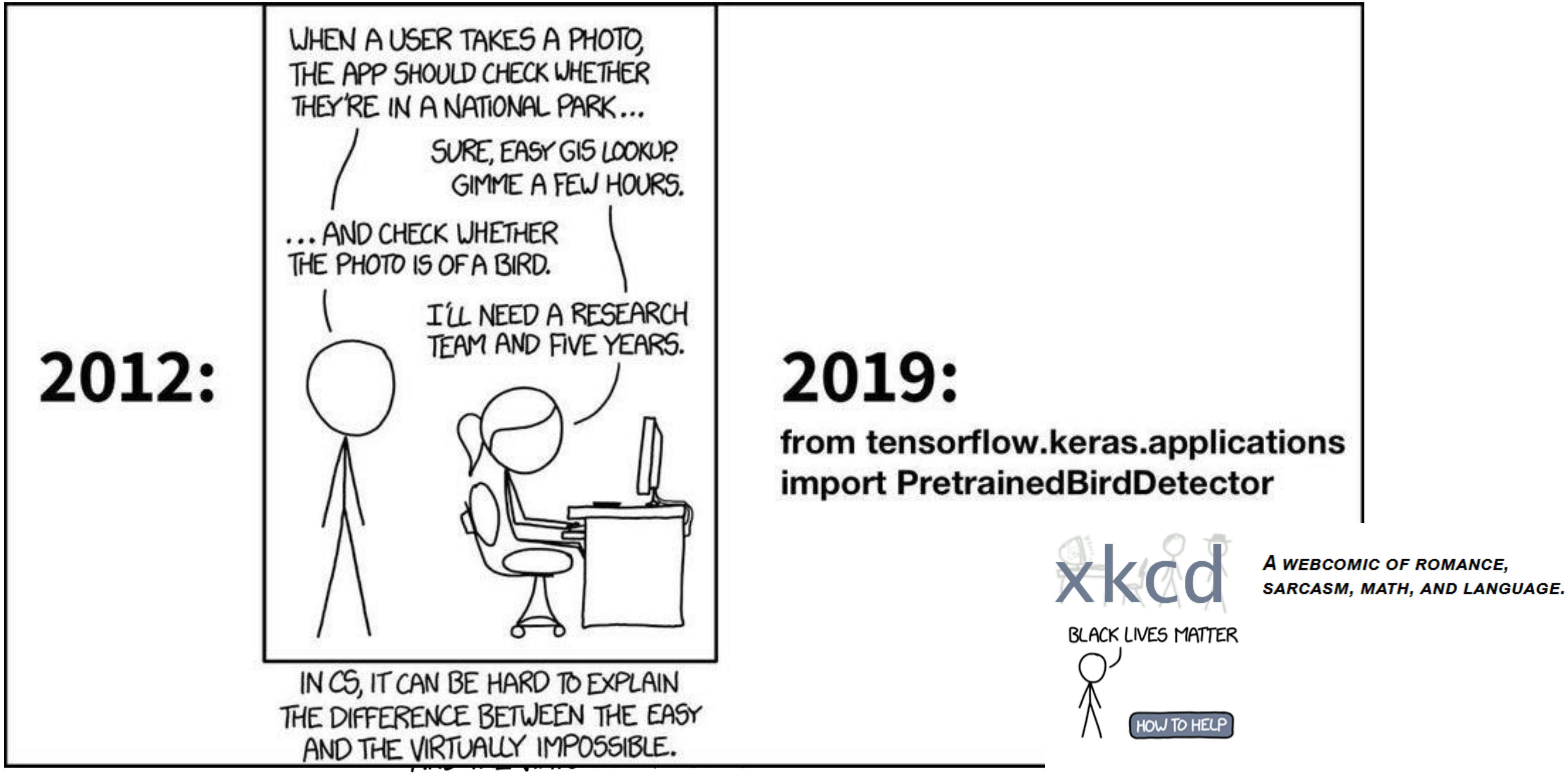
RGB images are tensors in a computer



Moravec’s paradox

Moravec wrote in 1988, "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility"

Computer vision is (was?) hard



https://xkcd.com/1425/

Image Classification

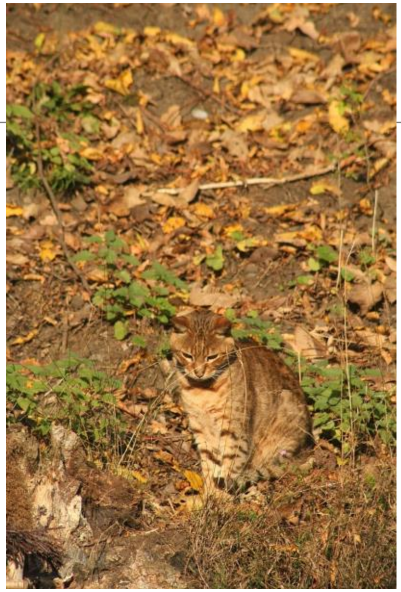
Input

Output
Choose among these categories

Dog
Cat
Bird
Frog
Person

Some challenges

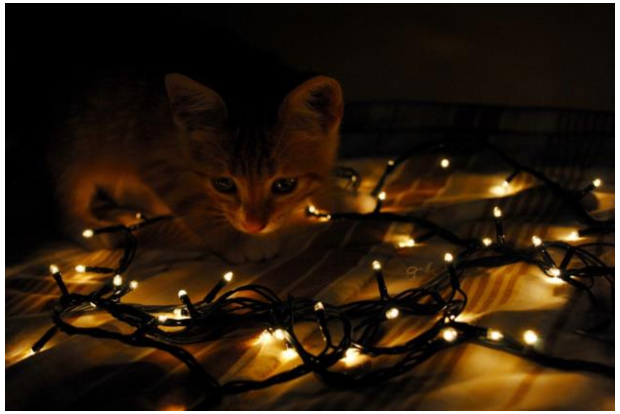
Intraclass variations



Background clutter

Occlusions

Viewpoint variations



Illumination changes

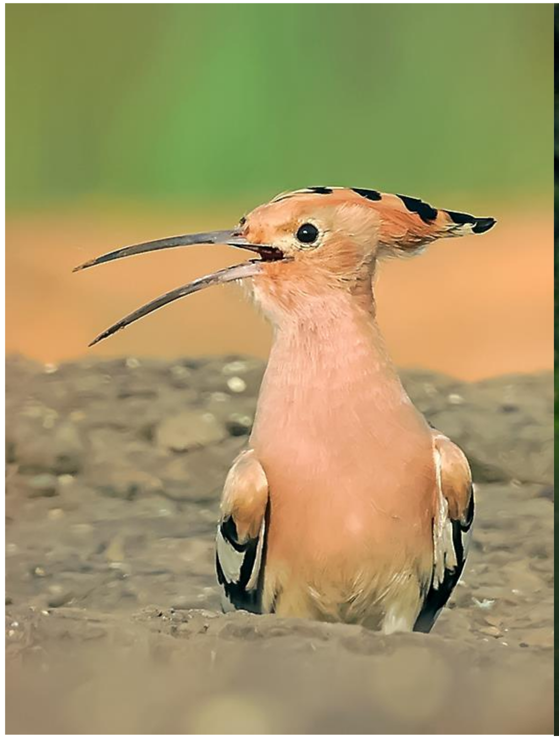
General weirdness of the world…

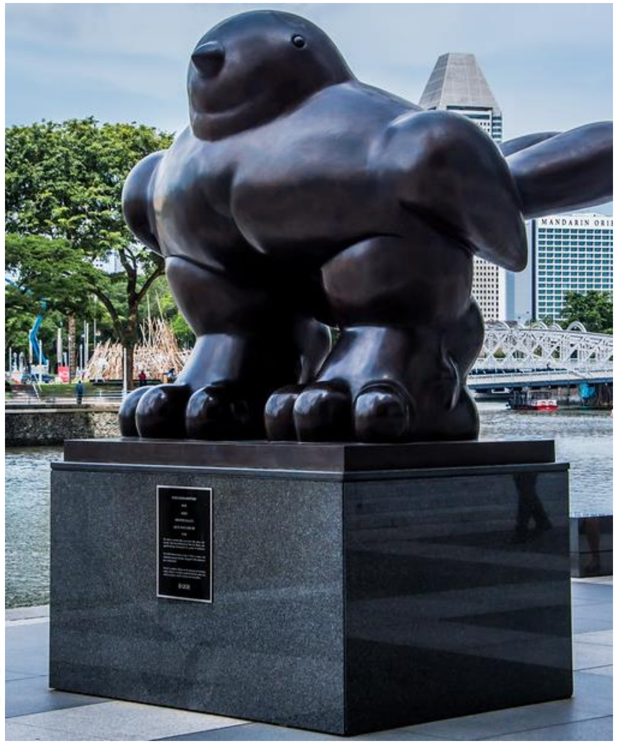
Categories as numbers

$f($  $) = 2$

0 -> Dog
1 -> Cat
2 -> Bird
3 -> Frog
4 -> Person

Birds…

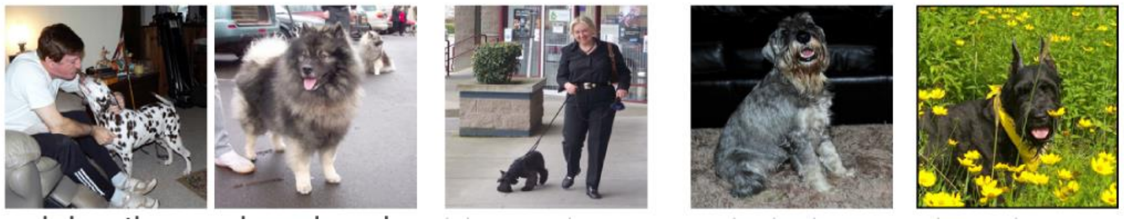




Traditional Computer Vision techniques, e.g. handcrafted rules based on edges, need **a controlled environment**, usually feasible in industrial vision applications, otherwise they are very brittle.

(Supervised) Machine learning to the rescue

Dog

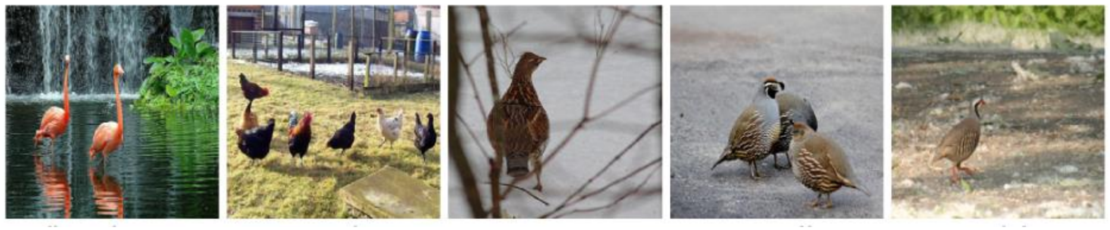


…

Cat

…

Bird

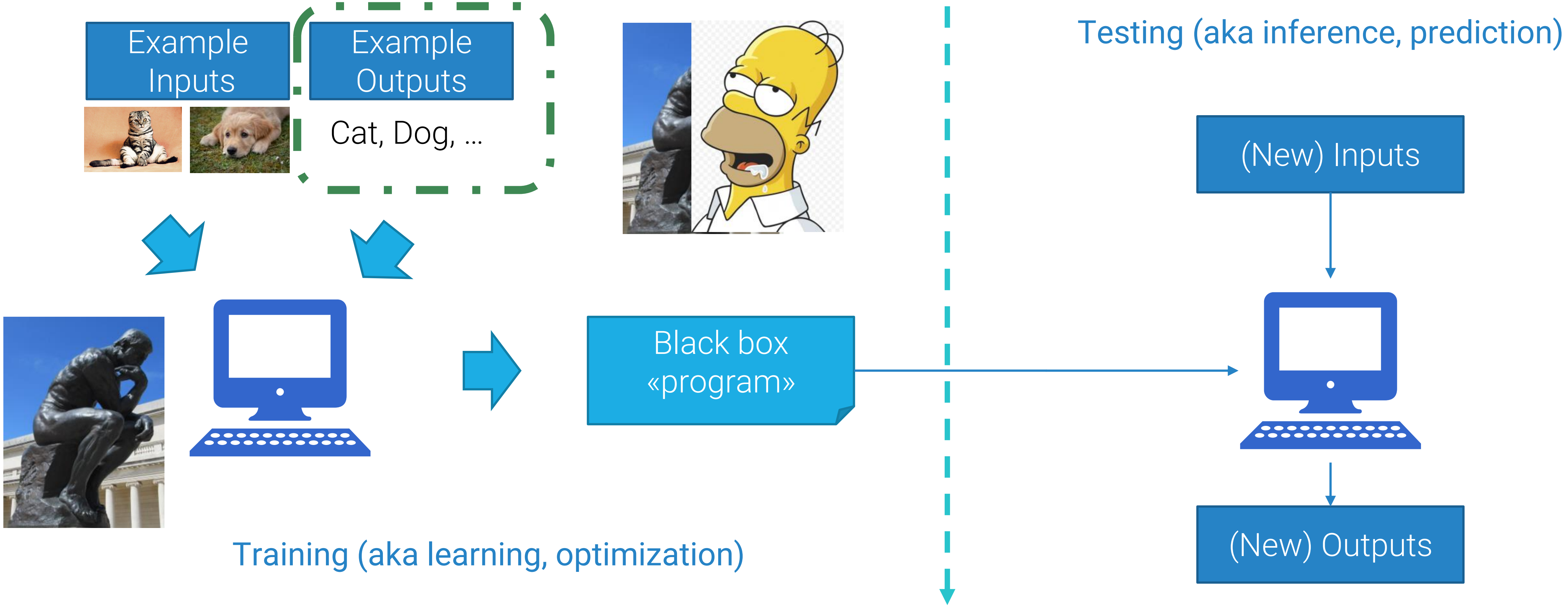


…

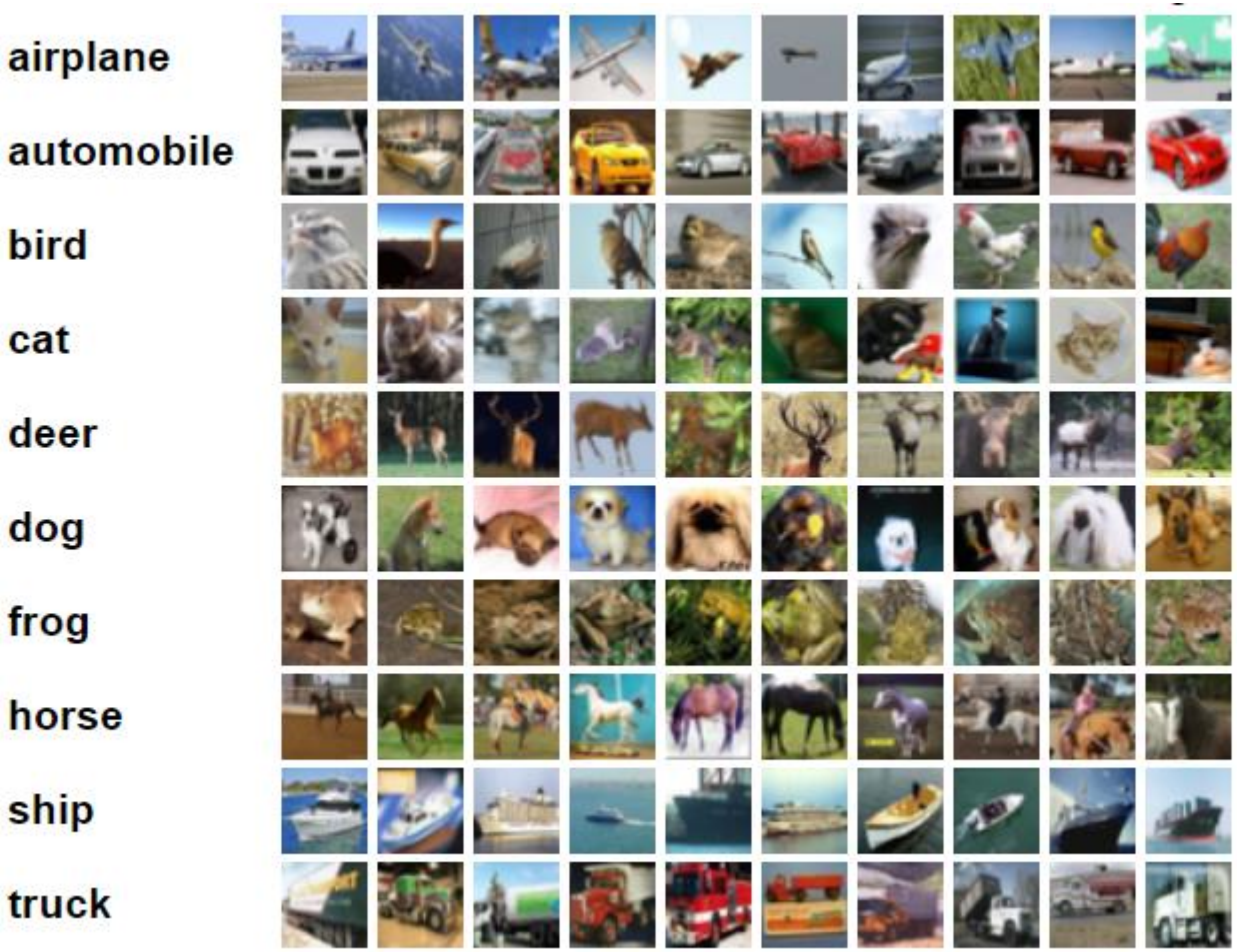
$f($  $) = 2$

Machine learning or data-driven approach

We can think of machine learning as a new way to instruct computers about what we want them to do.



CIFAR 10



Subset of the 80 million Tiny Images dataset

https://www.cs.toronto.edu/~kriz/cifar.html

10 classes

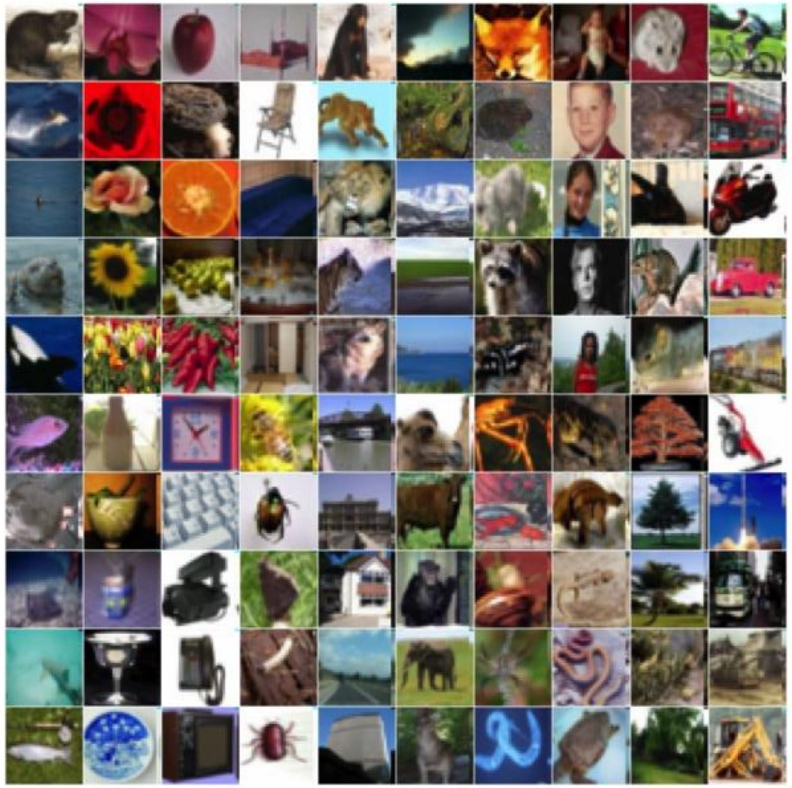
50k training images

10k testing images

32x32 RGB images

Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

CIFAR 100



Another subset of the 80 million Tiny Images dataset

100 classes

50k training images (500 per class)

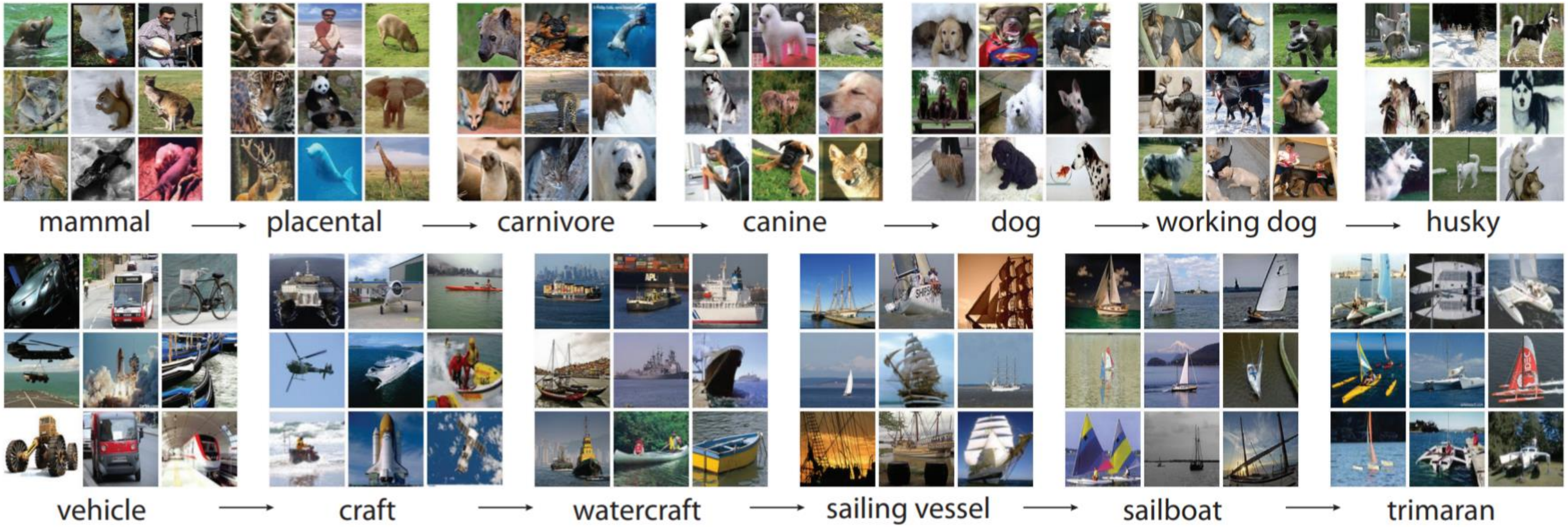
10k testing images (100 per class)

32x32 RGB images

Hierarchical structure: 20 super-classes with 5 sub-classes each

Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

ImageNet / ImageNet 21k



14 millions RGB images at full and variable resolution with average size about 400 × 350.

Hierarchical structure: modelled on about 21k synsets from **WordNet** (out of 50k)

Deng et al, “ImageNet: A Large-Scale Hierarchical Image Database”, CVPR 2009

Linear classifier

$f(\boldsymbol{x} =$  $;\theta) = 2$

0 (plane)	45.4
1 (car)	128.3
2 (bird)	253
...	0.23
	-1.34
	4
	56
	-63
	78
	2

$\xrightarrow{argmax} 2$

$$f(\boldsymbol{x};\boldsymbol{W}) = \boldsymbol{W}\boldsymbol{x} = \boldsymbol{scores}$$

$\boldsymbol{aka\ logits}$

32x32x3=3072x1 CIFAR image

10x3072

10x1

What does a linear model learn?

Accuracy about 38% on CIFAR10

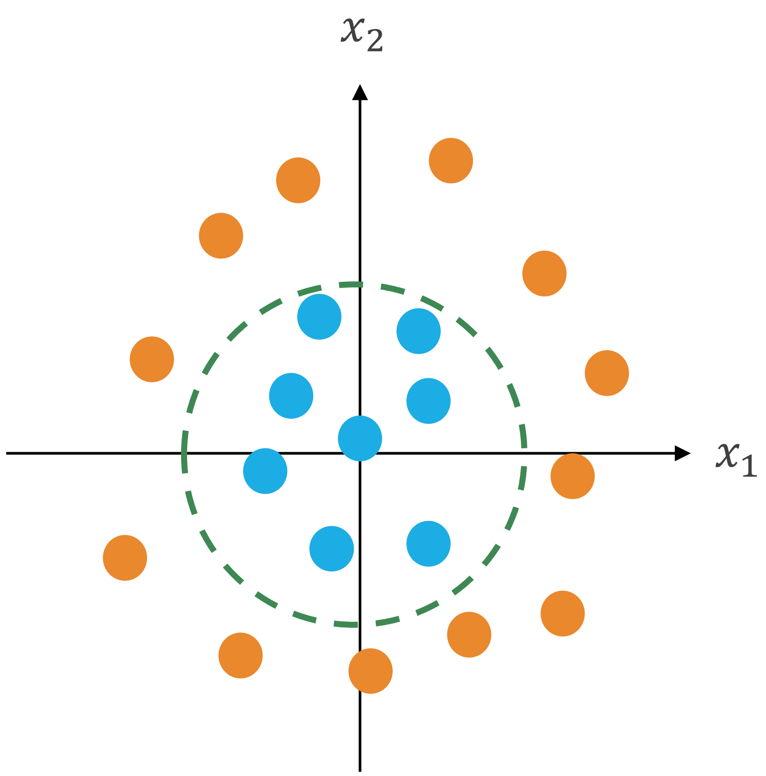
Let's use the template matching interpretation of a linear classifier to understand what the model is learning

It looks like the background color is the predominant feature used by the model

Moreover, one template cannot capture multiple appearances within one class, e.g. rotated cars, trucks, etc..

Distance between templates and images is still a distance in input space, same problem we had with k-NN classifier, and performance is similar

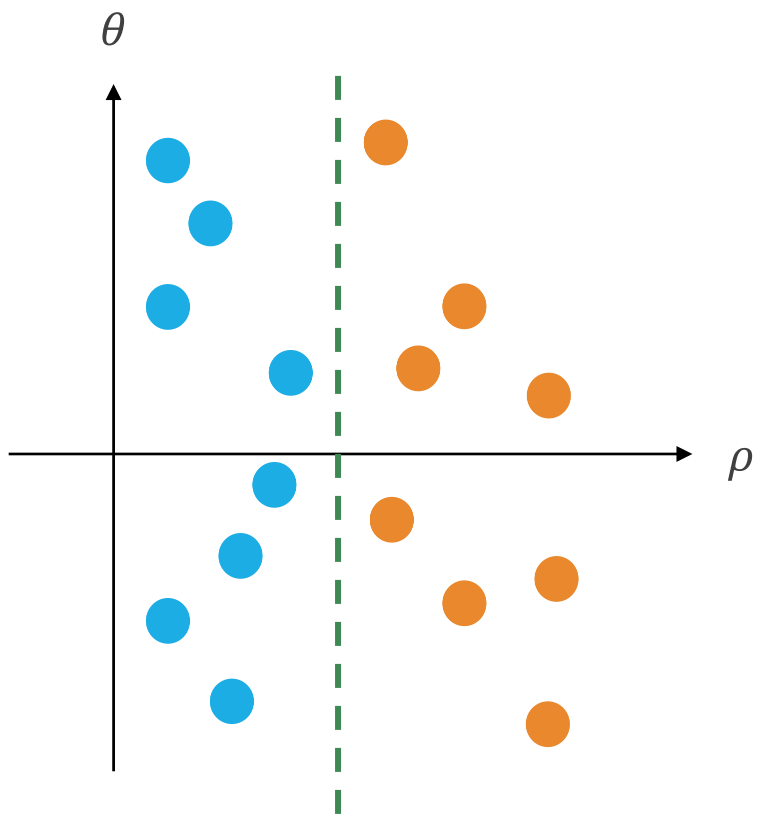
Representation is important



Switch to polar coordinates

$$\rho = \sqrt{x_1^2 + x_2^2}$$

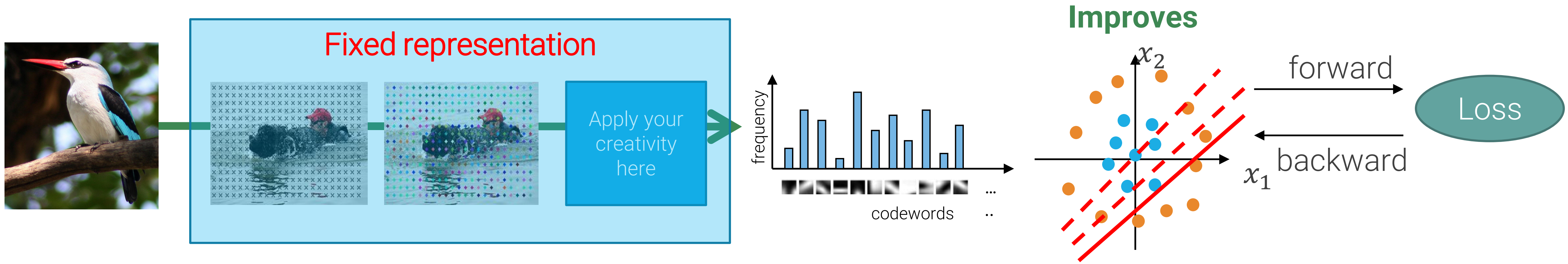
$$\theta = \tan^{-1}\frac{x_2}{x_1}$$

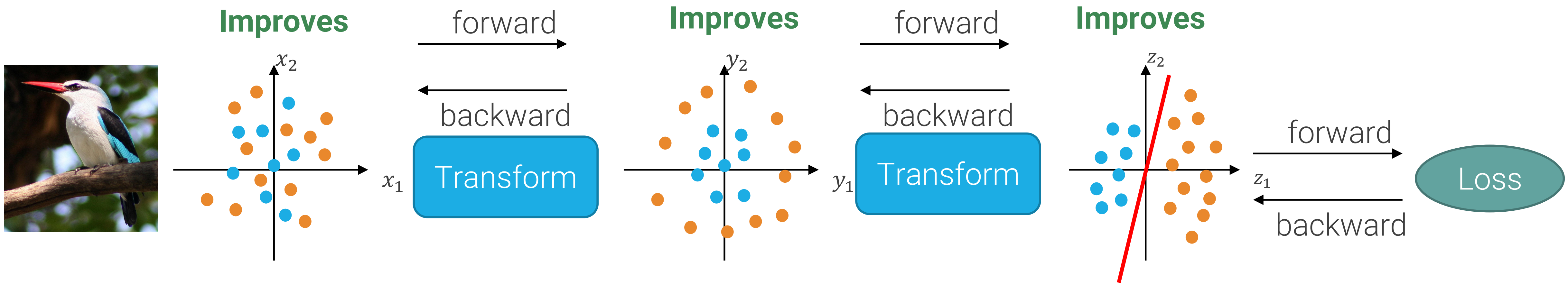


Non-linear decision boundary in input space

Linear decision boundary in feature space

Representation learning

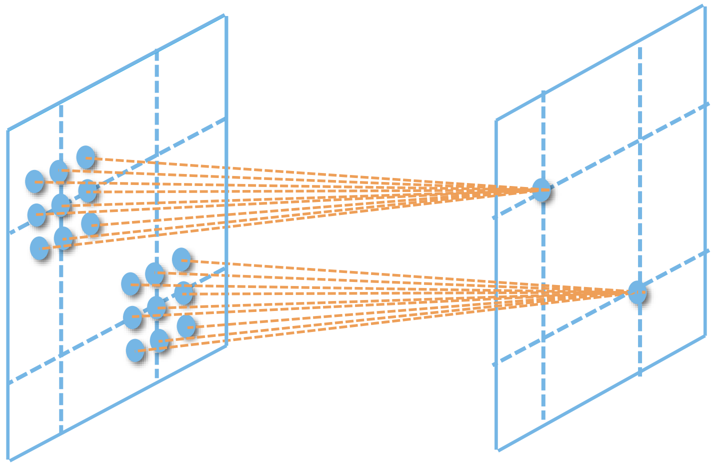




Deep learning ≈ Representation learning

Convolutions

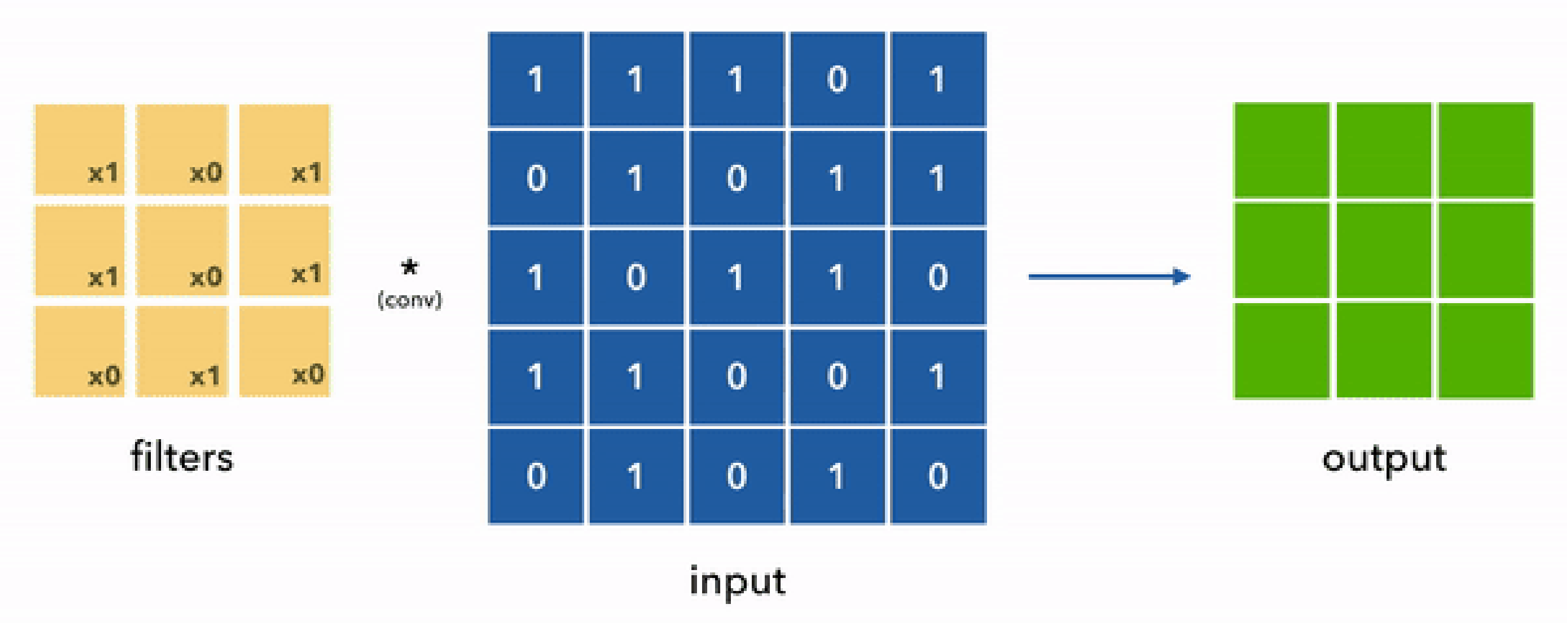
In traditional image processing and computer vision, we usually rely on convolution/correlation with hand-crafted filters (kernels) to process images (e.g. denoise or detect local features).



- Unlike linear layers, in a convolution, the input and output are not flattened, i.e. convolution preserves the spatial structure of images.
- Unlike linear layers, a convolution processes only a – small – set of neighboring pixels at each location. In other words, each output unit is connected only to local input units. This realizes a so called local receptive field.
- Unlike linear layers, the parameters associated with the connections between an output unit and its input neighbors are the same for all output units. Thus, parameters are said to be shared and the convolution seamlessly learns the same detector, regardless of the input position.

Convolutions embody inductive biases dealing with the structure of images: images exhibit informative local patterns that may appear everywhere across an image.

Convolution - animation



https://colab.research.google.com/github/GokuMohandas/Made-With-ML/blob/main/notebooks/11_Convolutional_Neural_Networks.ipynb

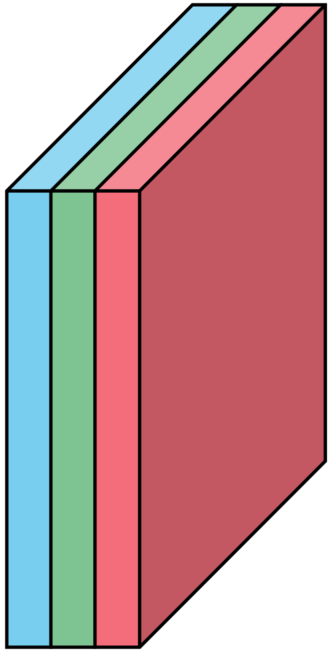
Multiple input channels

Images have 3 channels, so convolution kernels will be a 3-dimensional tensors of size $3 \times H_K \times W_K$ and

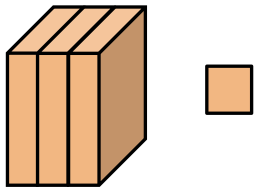
$$[K * I](j,i) = \sum_{n=1}^{3}\sum_{m}\sum_{l} K_n(m,l)\, I_n(j-m, i-l) + \boldsymbol{b}$$

This is still a 2D convolution, but over vector-valued functions, not a 3D convolution (notice we do not slide over channels)

As usual, we actually compute an affine function, so we also have a bias term



Input image, e.g. $\mathbf{3} \times \mathbf{32} \times \mathbf{32}$

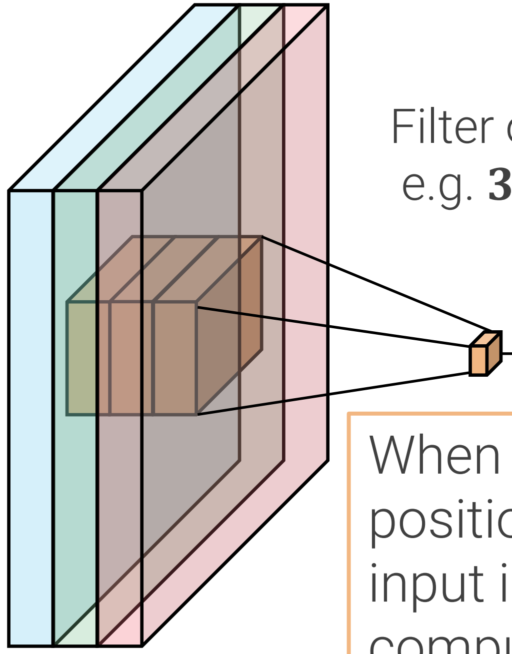


Filter or kernel, e.g. $\mathbf{3} \times \mathbf{5} \times \mathbf{5}$

Filters and input depth always match, and the third dimension of a filter is usually implicit, i.e. we define this as a "5 by 5 convolution", but it has $5 \times 5 \times 3 = 75$ parameters (76 with the bias), not 25

Output activation

$$[K * I](j,i) = \sum_{n=1}^{3} \sum_{m} \sum_{l} K_n(m,l) I_n(j-m, i-l) + b$$



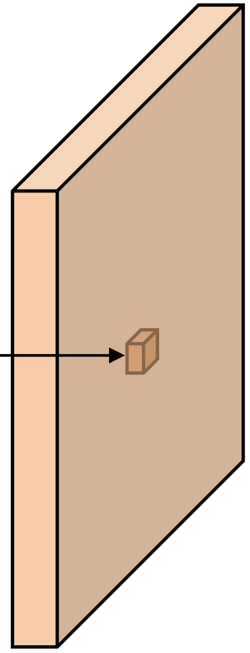
Filter or kernel,
e.g. $\mathbf{3} \times \mathbf{5} \times \mathbf{5}$

When the filter is at position (j,i) on the input image, we compute one number

Input image,
e.g. $\mathbf{3} \times \mathbf{32} \times \mathbf{32}$

By sliding the filter over the image, we get a single-valued output image, which is the output activation

It is also called feature map, activation map, etc…

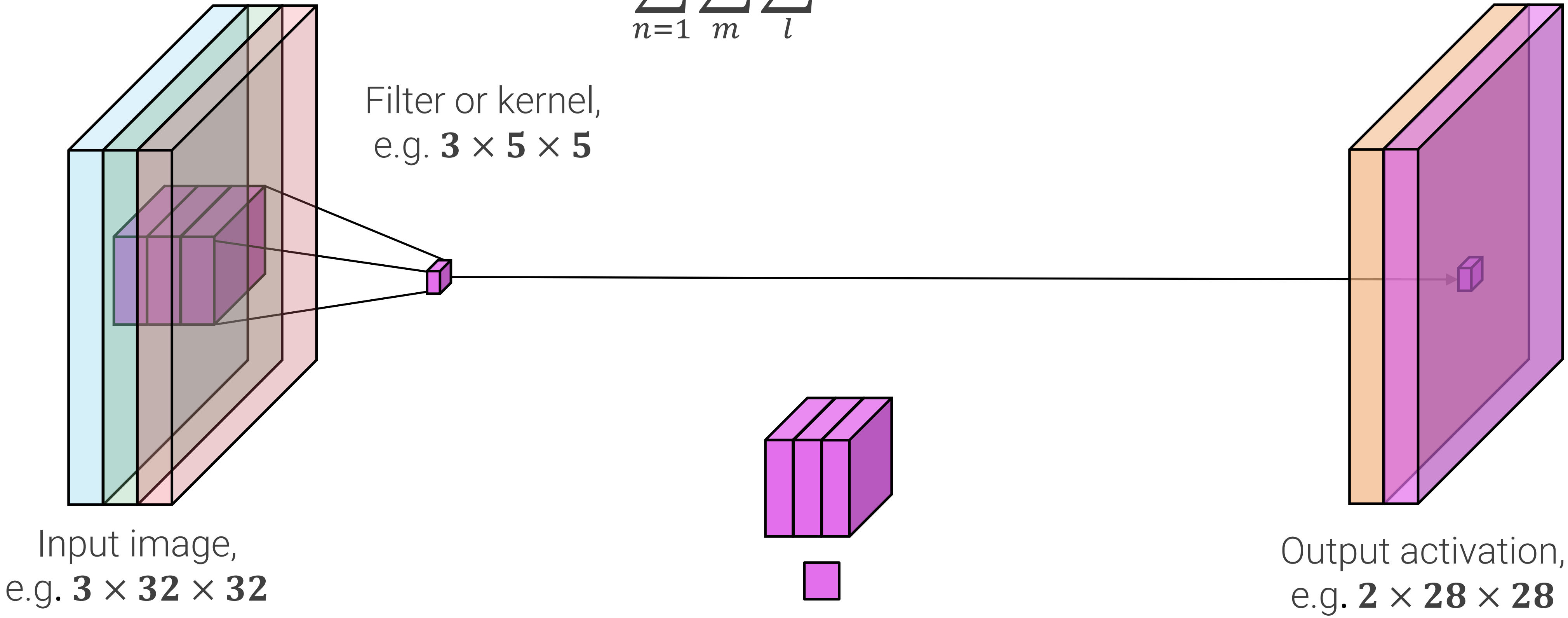


Output activation,
e.g. $\mathbf{1} \times \mathbf{28} \times \mathbf{28}$

Activation

We can repeat it with a second filter, with different weights, e.g. a filter that detects horizontal edges instead of vertical ones

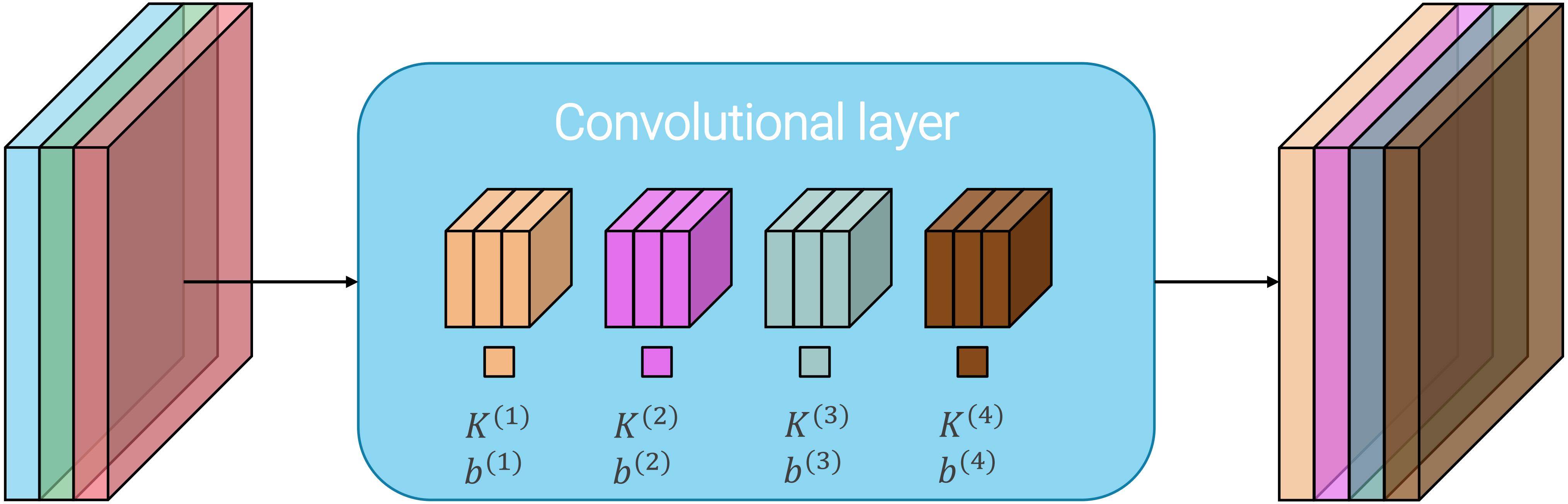
$$\left[\boldsymbol{K}^{(2)} * I\right](j,i) = \sum_{n=1}^{3}\sum_{m}\sum_{l} \boldsymbol{K}_{\boldsymbol{n}}^{(2)}(m,l)I(j-m,i-l) + \boldsymbol{b}^{(2)}$$



Convolutional layer

If we have **4** filters, each of size $\mathbf{3 \times 5 \times 5}$, we can describe the overall operation realized by the layer as

$$[K * I]_k(j,i) = \sum_{n=1}^{3} \sum_m \sum_l \boldsymbol{K_n^{(k)}}(m,l) I(j-m, i-l) + b^{(k)} \qquad k = 1, \dots, 4$$



Input image,
e.g. $\mathbf{3 \times 32 \times 32}$

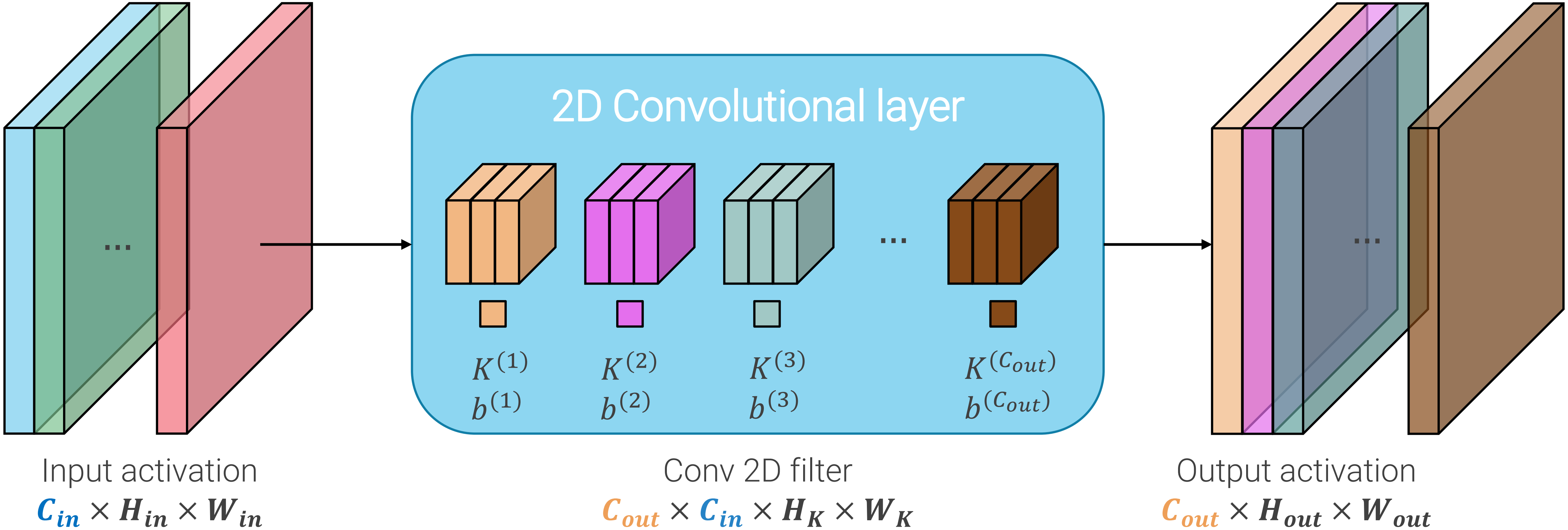
Conv 2D filter
e.g. $\mathbf{4 \times 3 \times 5 \times 5}$

Output activation,
e.g. $\mathbf{4 \times 28 \times 28}$

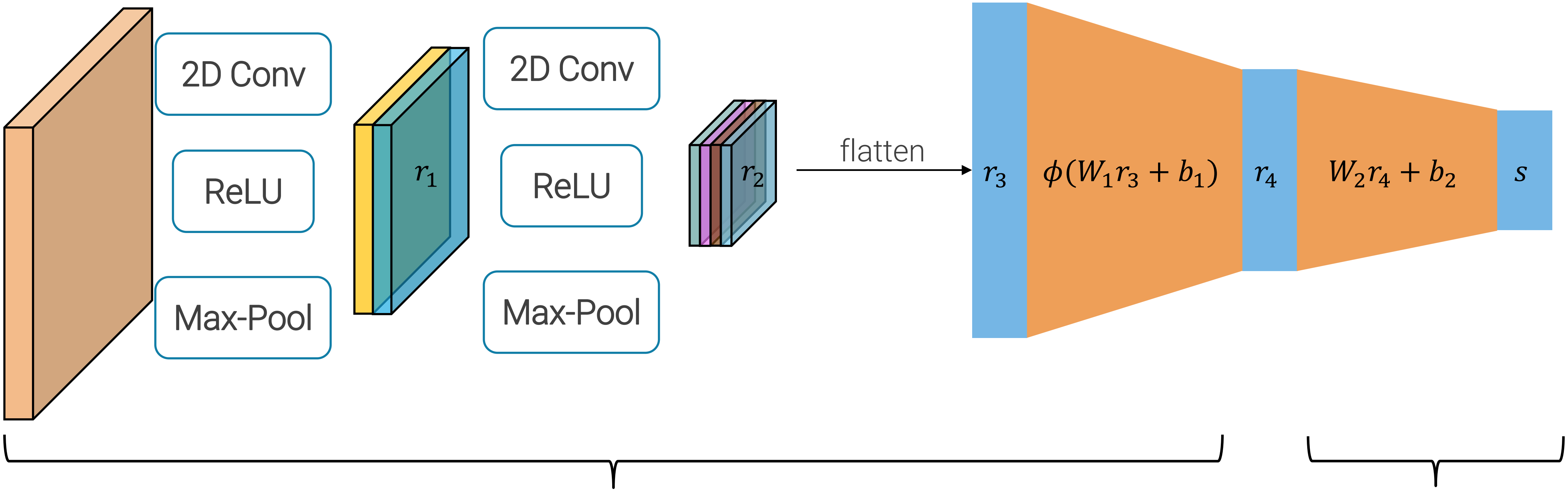
Convolutional layer

In the general case, we compute C_{out} convolutions between vector-valued kernels and input activations

$$[K * I]_k(j,i) = \sum_{n=1}^{C_{in}} \sum_m \sum_l K_n^{(k)}(m,l) I_n(j-m, i-l) + b^{(k)} \qquad k = 1, \dots, C_{out}$$



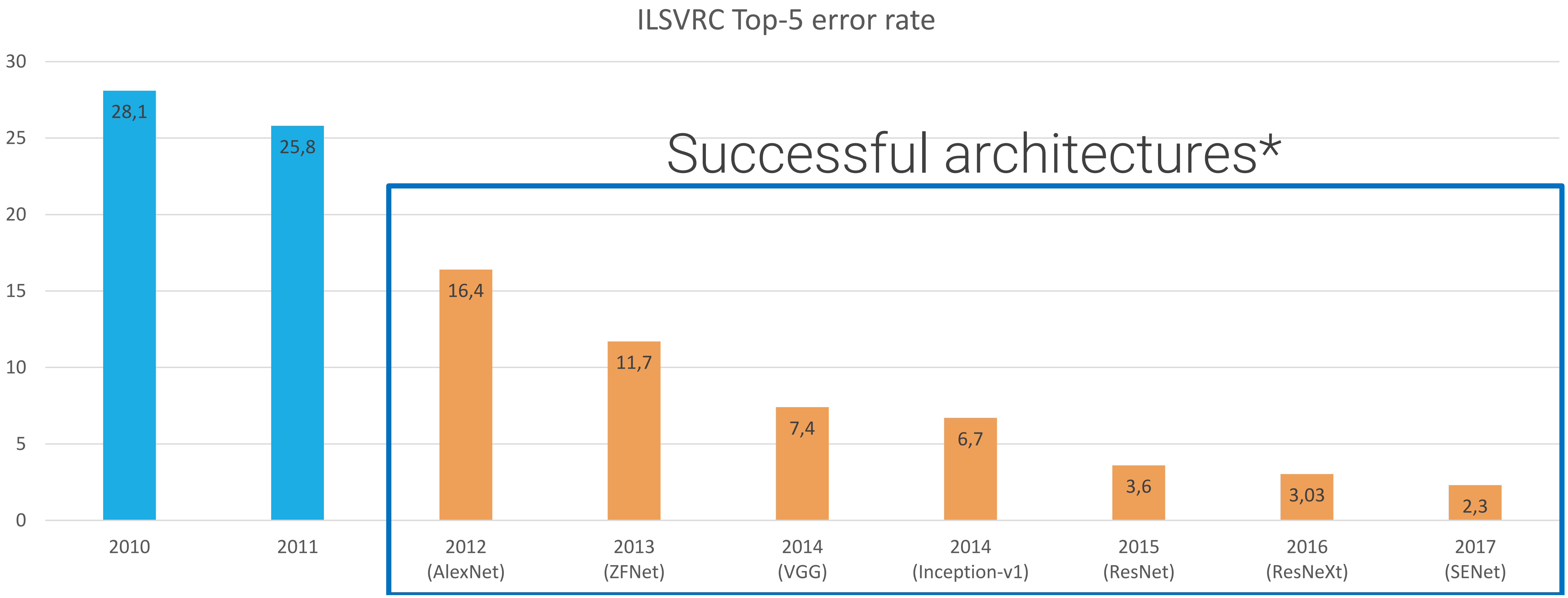
Convolutional Neural Networks



N convolutional+pooling layers followed by M linear layers
This is also called the feature extractor

The final linear layer is also called the classifier

ILSVRC error rate evolution

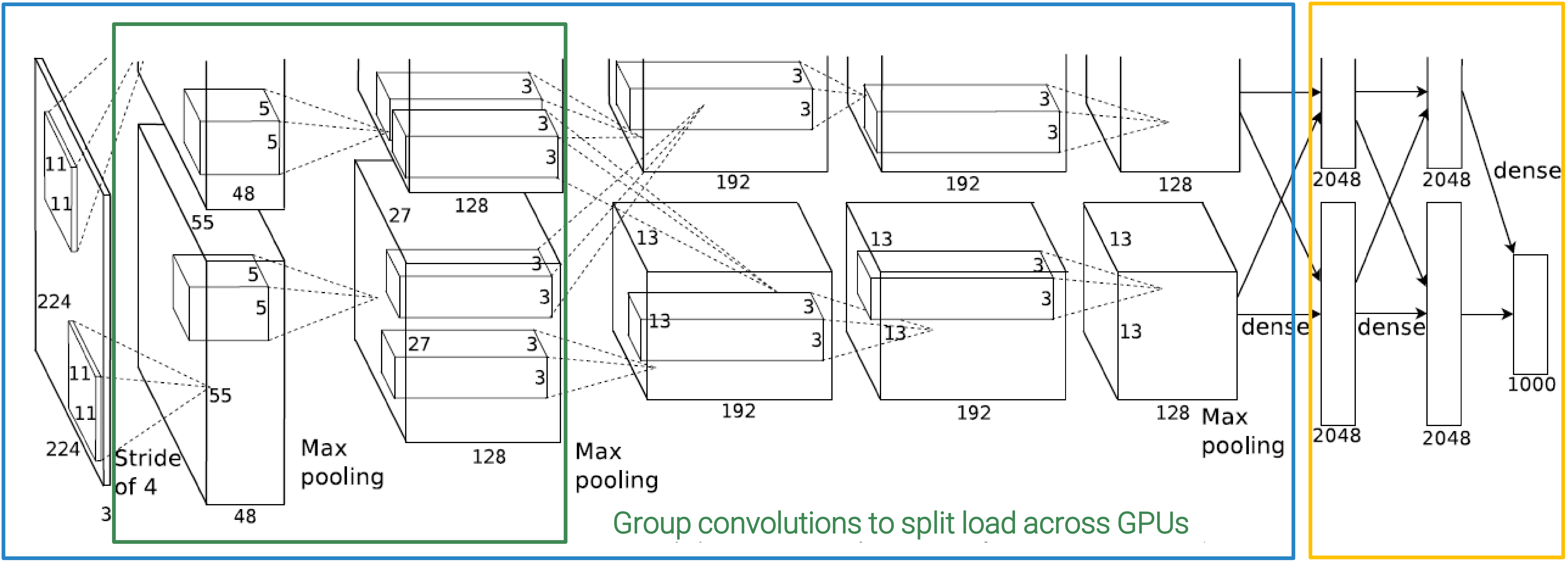


*Results based on ensembles and, sometimes, heavy test-time augmentation

AlexNet

5 convolutional layers (conv+relu) optionally followed by max-pooling

3 fc layers



Group convolutions to split load across GPUs

AlexNet

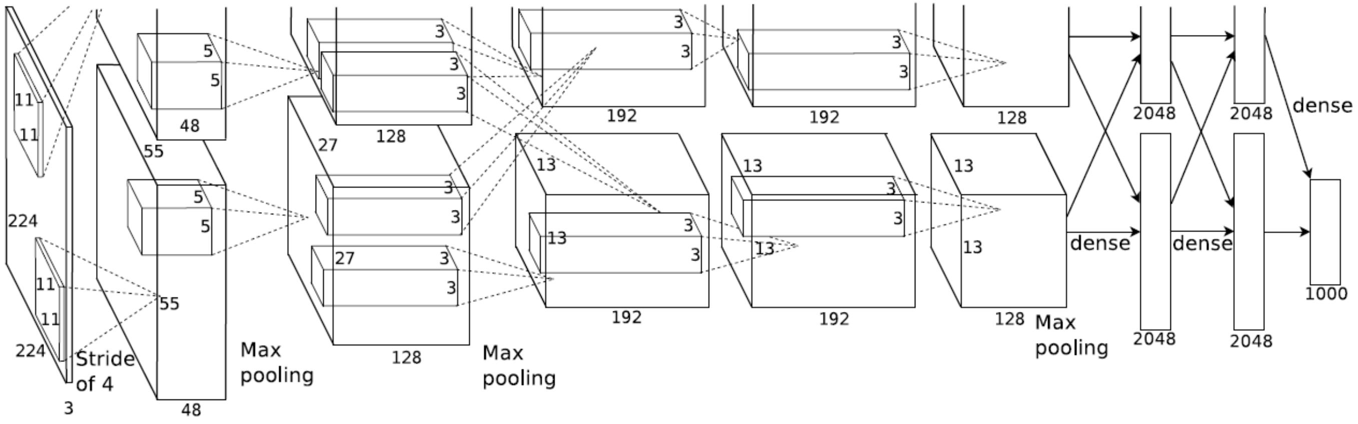
Won ILSVRC 2012.

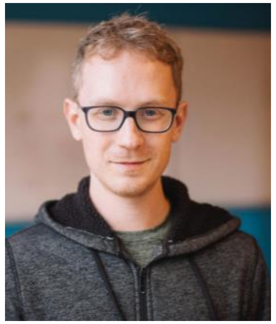
Was trained on two GTX580 GPUs.

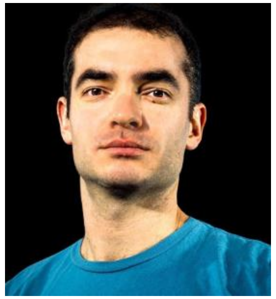
Used local response normalization (LRN) in some layers, not used in subsequent architectures.

Took between five and six days to train

“All our experiments suggest that our results can be improved simply by waiting for faster GPUs and bigger datasets to become available.”







Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NeurIPS 2012

VGG: Deep but regular

Second place in ILSVRC 2014, 7.5% top-5 error

Commit to explore the effectiveness of simple design choices, by allowing only the combination of :

- 3x3 convolutions, S=1, P=1
- 2x2 max-pooling, S=2, P=0
- #channels doubles after each pool

Dropped local response normalization (LRN)

Batch norm not invented yet! Pre-initialization of deeper networks with weights from shallower architectures crucial to let training progress (unless smart initialization strategies are used).

				VGG-16	VGG-19
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					

Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition", ICLR 2015

Stages

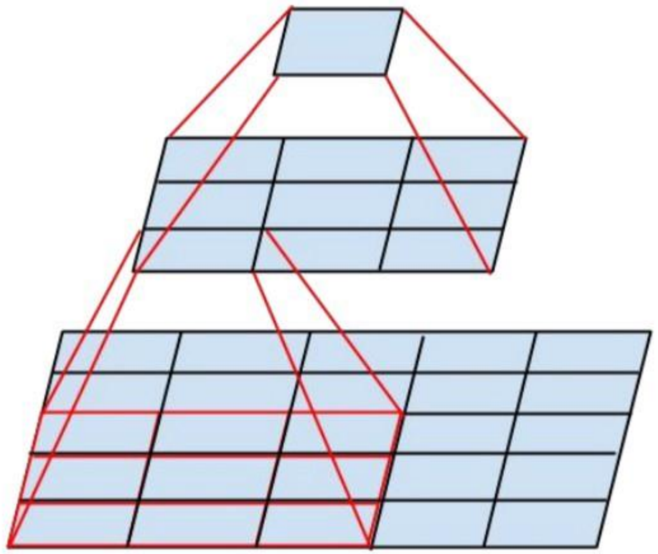
VGG introduces the idea of designing a network as repetitions of **stages**, i.e. a fixed combination of layers that process activations **at the same spatial resolution**.

In VGG, stages are either:

- conv-conv-pool
- conv-conv-conv-pool
- conv-conv-conv-conv-pool

One stage **has same receptive field of larger convolutions** but **requires less params and computation** and introduces more non-linearities.

No free-lunch, though: **memory for activations doubles**



VGG-16	VGG-19
D 16 weight layers	E 19 weight layers
conv3-64 conv3-64	conv3-64 conv3-64
conv3-128 conv3-128	conv3-128 conv3-128
conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool	maxpool
FC-4096	FC-4096
FC-4096	FC-4096
FC-1000	FC-1000
soft-max	soft-max

Conv layer	Params	Flops	ReLUs	#Activations
$C \times C \times 5 \times 5\,, S = 1, P = 2$	$25C^2 + C$	$50C^2 W_{in} H_{in}$	1	$C \times W_{in} \times H_{in}$
2 stacked $C \times C \times 3 \times 3, S = 1, P = 1$	$18C^2 + 2C$	$36C^2 W_{in} H_{in}$	2	$2 \times C \times W_{in} \times H_{in}$

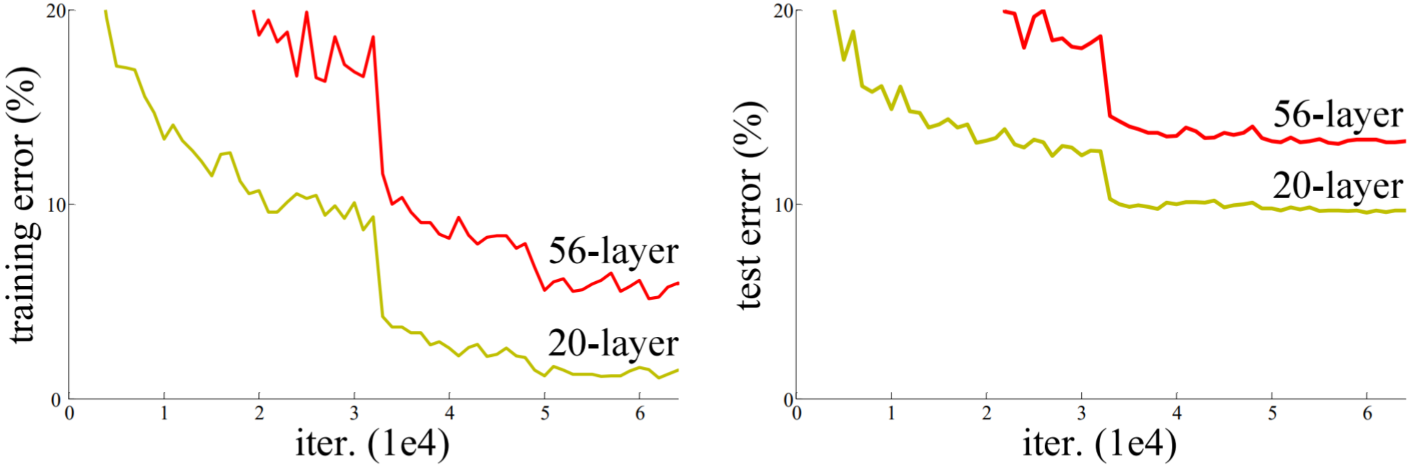
Residual Networks

VGG lesson: growing depth improves performance. Yet, stacking more layers doesn't automatically improve performance.

Too many parameters increase overfitting and hurts generalization? We also observe higher training errors, so overfitting it's not the only reason, there is also a training problem, even when using Batch Norm.

Yet, a solution exists by construction: if a network with 20 layers achieves performance X, then we can stack 36 more identity layers and we should keep performance at X.

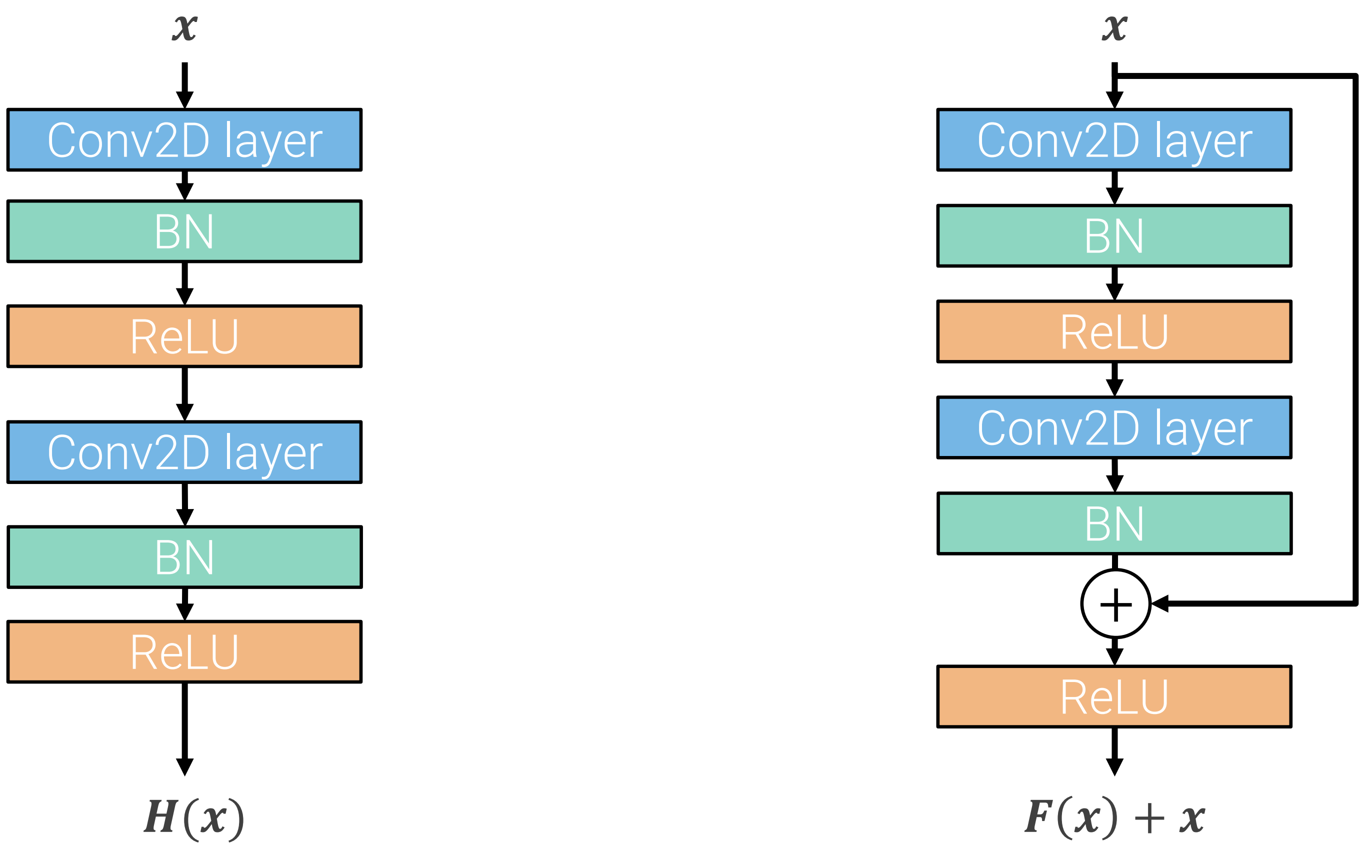
SGD is not able to find this solution with the parameterization we use for layers: optimizing very deep networks is hard.



Kaiming He et al., "Deep Residual learning for image recognition", CVPR 2016

Residual block

The proposed solution is to change the network so that learning identity functions is easy by introducing residual blocks. Implemented by adding skip connections skipping two convolutional layers.



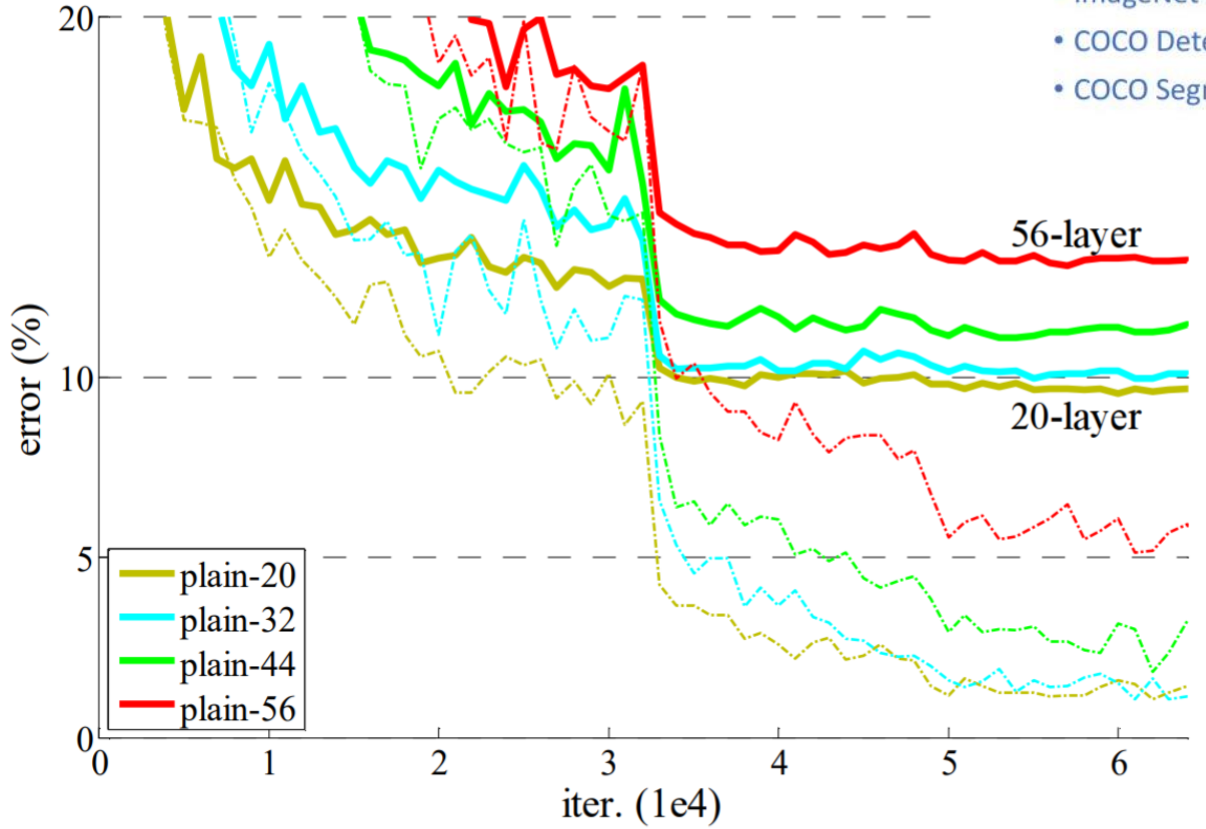
Weights usually initialized to be very small (or 0 for biases). Network starts with the identity function and learns an "optimal" perturbation of it.

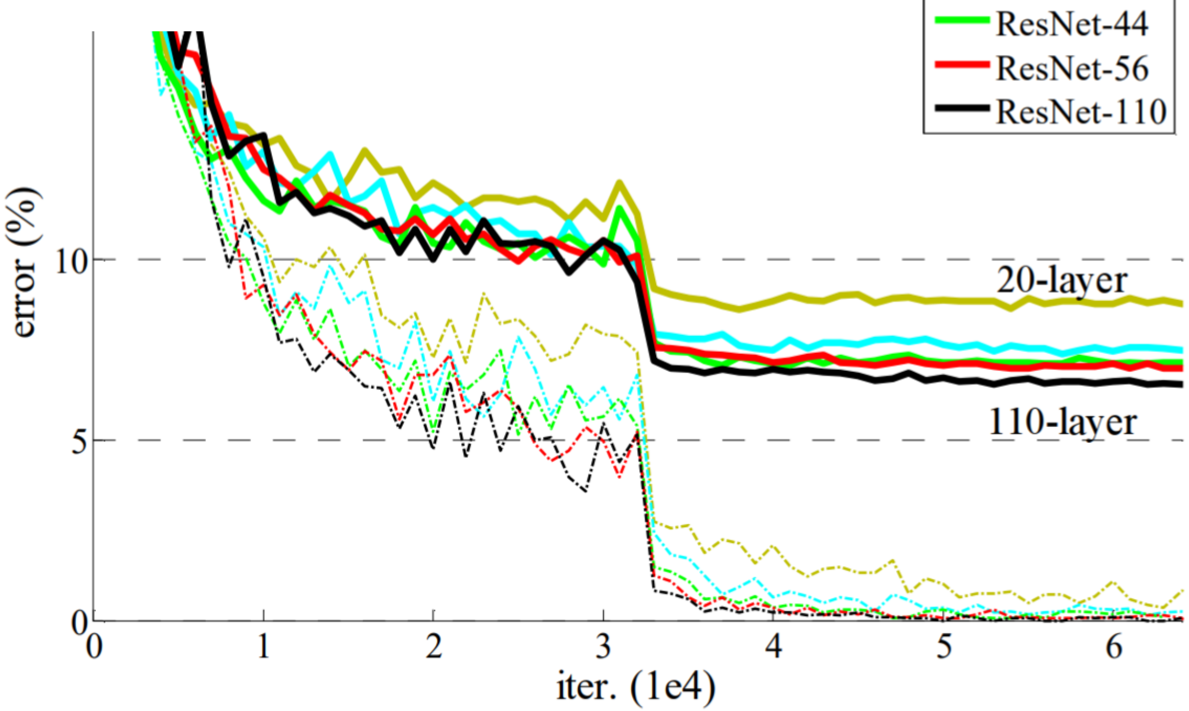
It makes heavy use of batch-norm

Results updated

MSRA @ ILSVRC & COCO 2015 Competitions

- **1st places in all five main tracks**
  - ImageNet Classification: *"Ultra-deep"* (quote Yann) 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd





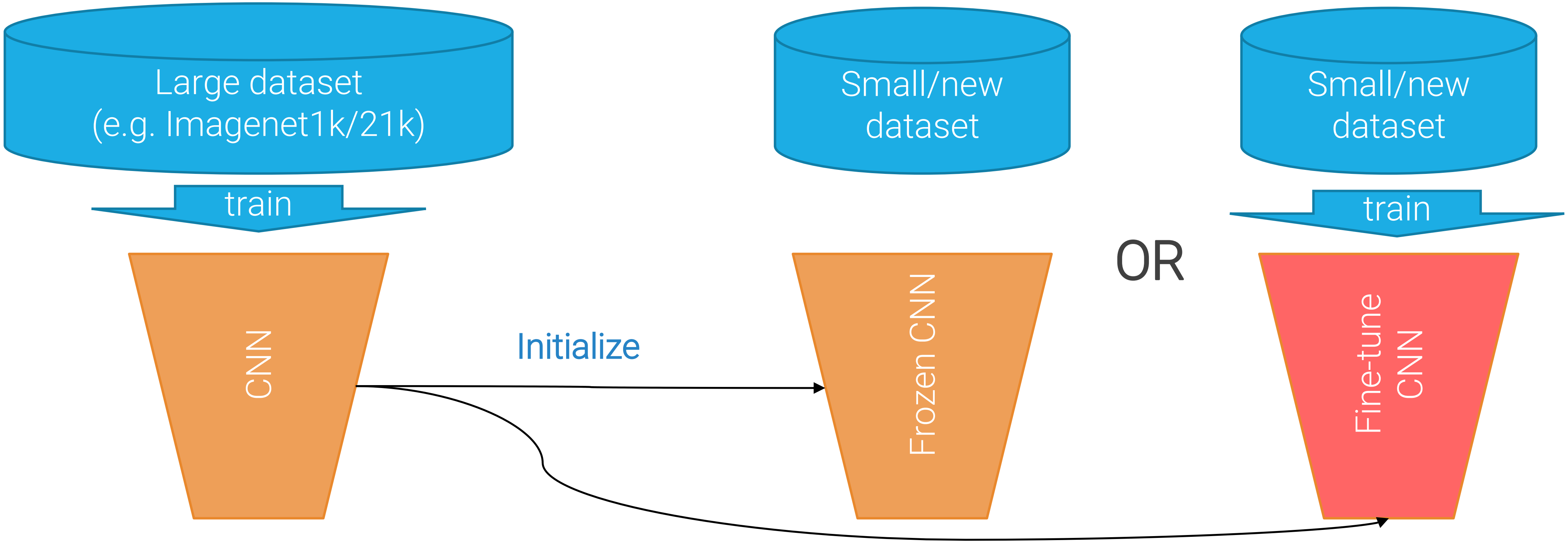
Residual blocks allow us to train deep networks. When properly trained, deep networks outperform shallower network as expected

Won all 2015 competitions by a large margin, still the standard baseline/backbone for most tasks today.

Transfer Learning

We normally want to run CNNs on new classification datasets, not on ImageNet.

One of the most important features, from a practical point of view, of learned representations is that they can be effectively transferred to new datasets. Transfer learning is the process of using and adapting a pre-trained NN to new datasets. Usually, we pre-train on large datasets, and then we use it as frozen feature extractor or fine-tune it on the new dataset.



Object detection

Problem definition

Input: RGB Image of size $W \times H$

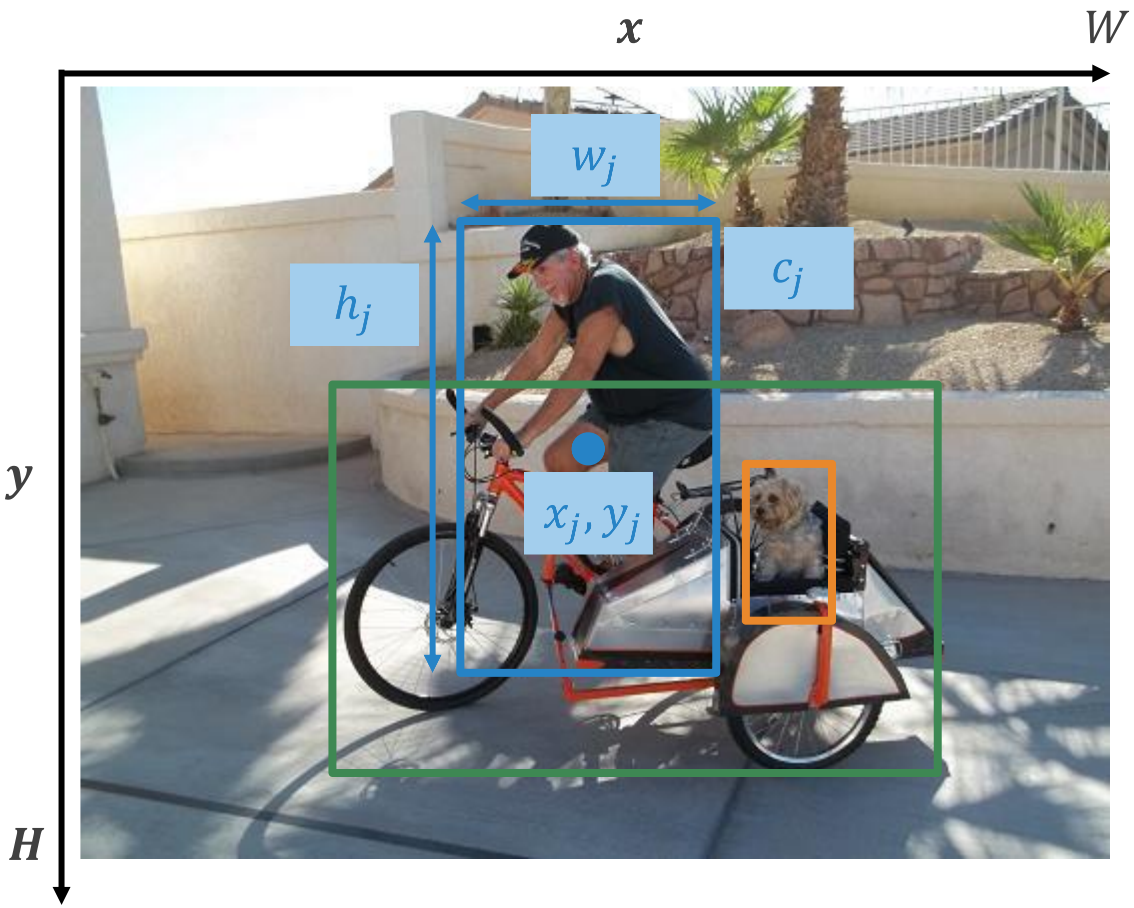
Output: a set of "objects".

For each object $\mathbf{o}_j$:

- category $c_j \in [1, \ldots, C]$ (from a fixed list of categories, as in image classification)

- bounding box $BB_j = [x_j, y_j, w_j, h_j]$,

$x_j, w_j \in [0, W-1]$, $y_j, h_j \in [0, H-1]$



Challenges:

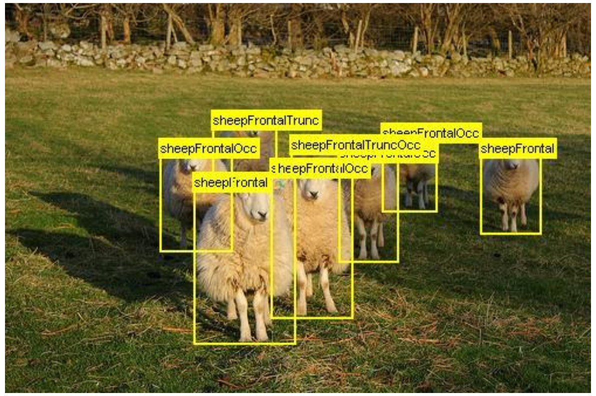
- output with variable length

- output with categorical ("what") as well as spatial ("where") information

- usually images processed at higher resolution than in image classification to have enough details

Datasets

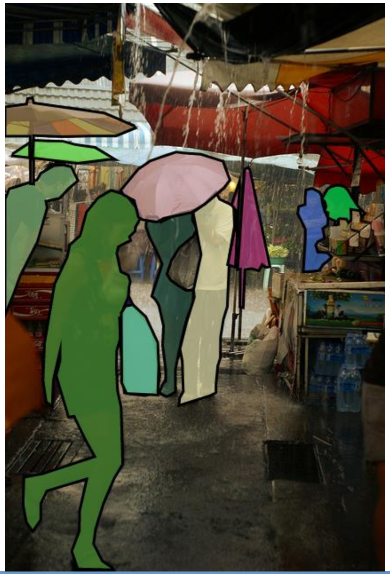
PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning



Trainval images: 11540 (27450 objects)
20 categories

http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html

COCO
Common Objects in Context



train/val images: 118K/5K
80 categories

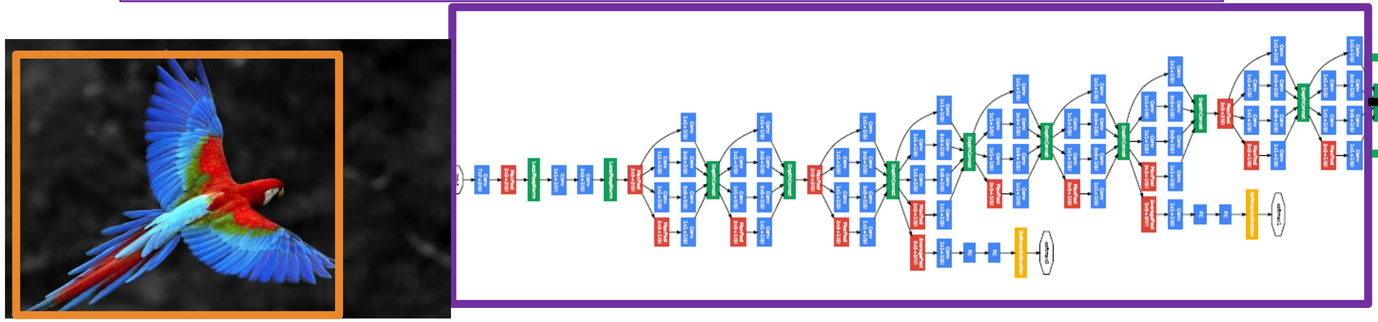
https://cocodataset.org/

Object localization

To see how deep CNNs we studied for image classification can be extended to the problem of object detection, let's first consider a simpler problem. If we can assume that only one object is present in the image, object detection simplifies to object localization, i.e., predicting one class and one bounding box per image.

To solve it, we can reuse any of the architectures seen for image classification, adding a regression head predicting the bounding box (i.e., 4 numbers) next to the standard classifier. Usually, the number of classes in object detection is smaller than 1000, so we retrain also the FC layer of the classification head.

Feature extractor (aka backbone network) is usually pre-trained on ImageNet: Transfer Learning



1024x7x7 activations

Global Pool

1024x1x1 activations

FC-80 ~~FC-1000~~

80 ~~1000~~x1x1 scores

argmax

Class $\hat{c}$ label out of ~~1000~~ 80 classes e.g. bird

FC-4

4x1x1

$\widehat{BB}^{(i)} = (x, y, w, h)$

$c^{(i)}, BB^{(i)}$ → $L_{loc}^{(i)}(\widehat{BB}^{(i)}) = \left\|\widehat{BB}^{i} - BB^{(i)}\right\|_2^2$

Total loss $L^{(i)} = CE\left(\text{softmax}(scores^{(i)}), \mathbb{I}(c^{(i)})\right) + \lambda\, L_{loc}^{(i)}(\widehat{BB}^{(i)})$ → Multi-task learning

Detecting Multiple Objects

Idea: we can apply a classification CNN as a sliding window detector

Problems:

1. we need a background class to discard background patches: how should we train it? Add a background class when fine-tuning the network on the detection dataset. Background patches are far more frequent: be sure to include positive samples in the training mini-batch (e.g., 32 positive boxes and 96 negative ones to reach 128 batch size). Total loss becomes:

Indicator function

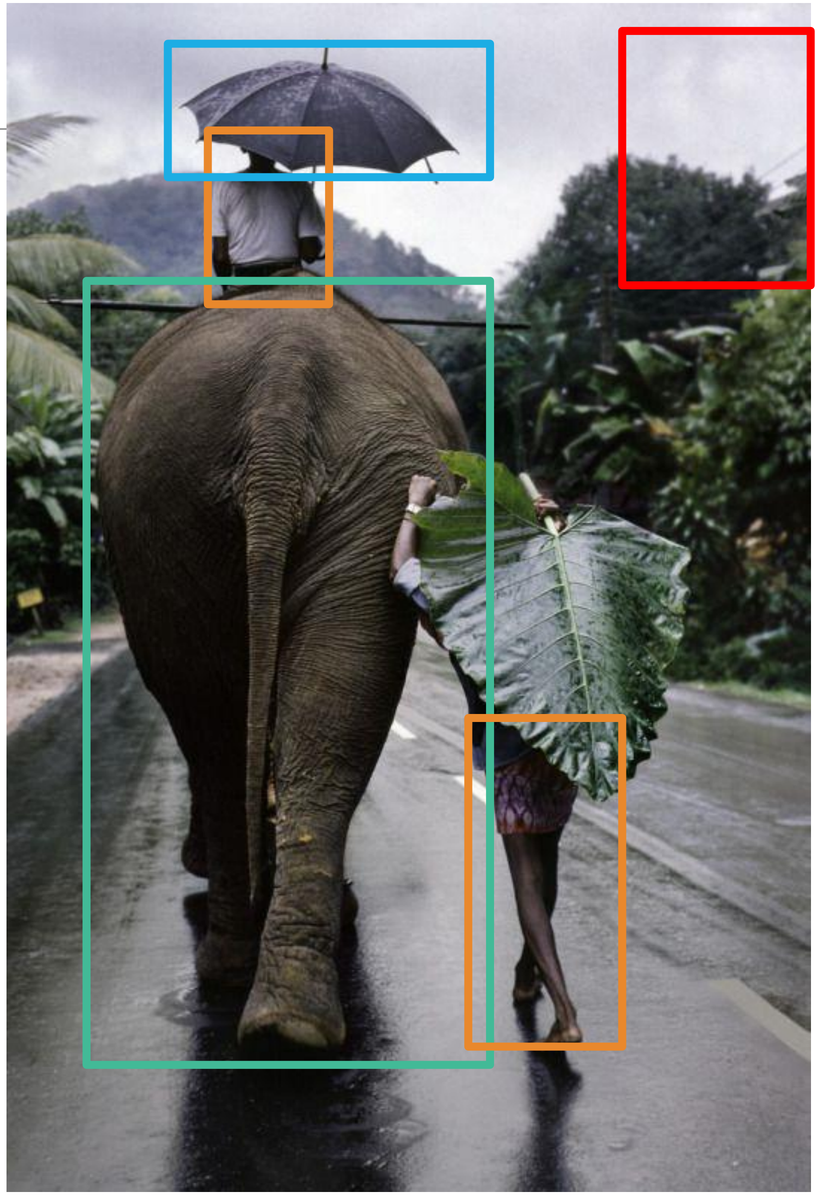
$$L^{(i)} = CE\left(\mathrm{softmax}(scores), \mathbb{1}(c^{(i)})\right) + \lambda\, I[c^{(i)} \neq bg] L_{loc}^{(i)}(\widehat{BB}^{(i)})$$

2. there are too many boxes to try: for a $w \times h$ window, there are $(W - w + 1) \times (H - h + 1)$ possible positions, but we have to try all (or most of) the scales and aspect ratios, hence

$$\#windows = \sum_{w=1}^{W} \sum_{h=1}^{H} (W - w + 1) \times (H - h + 1)$$

$$= \frac{H(H+1)}{2} \frac{W(W+1)}{2} = O(W^2 H^2)$$

Solution: use region proposals



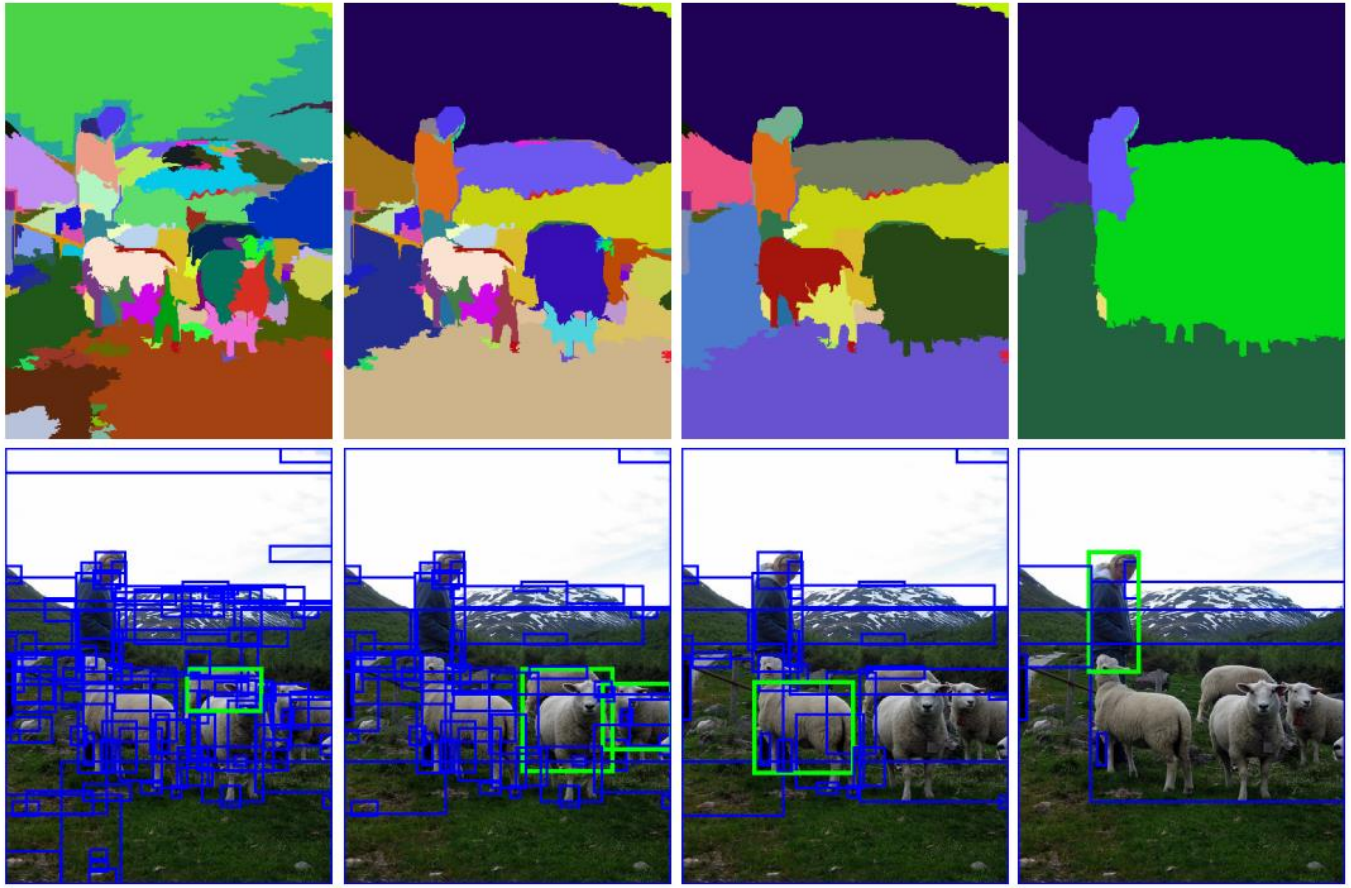
Region proposals

Region proposal are classical computer vision algorithms like Selective Search that inspect the image and attempt to find regions that likely contain an object.

It first oversegments the image into highly uniform regions (i.e. "superpixels").

Then, based on similarity scores of color, texture and size iteratively aggregates them: the two most similar regions are grouped together, and new similarities are calculated between the resulting region and its neighbors, until the whole image becomes a single region. Each aggregation is a region.

It aims for high recall but low precision while drastically reducing the number of boxes to be evaluated.



P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. IJCV 2004
Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

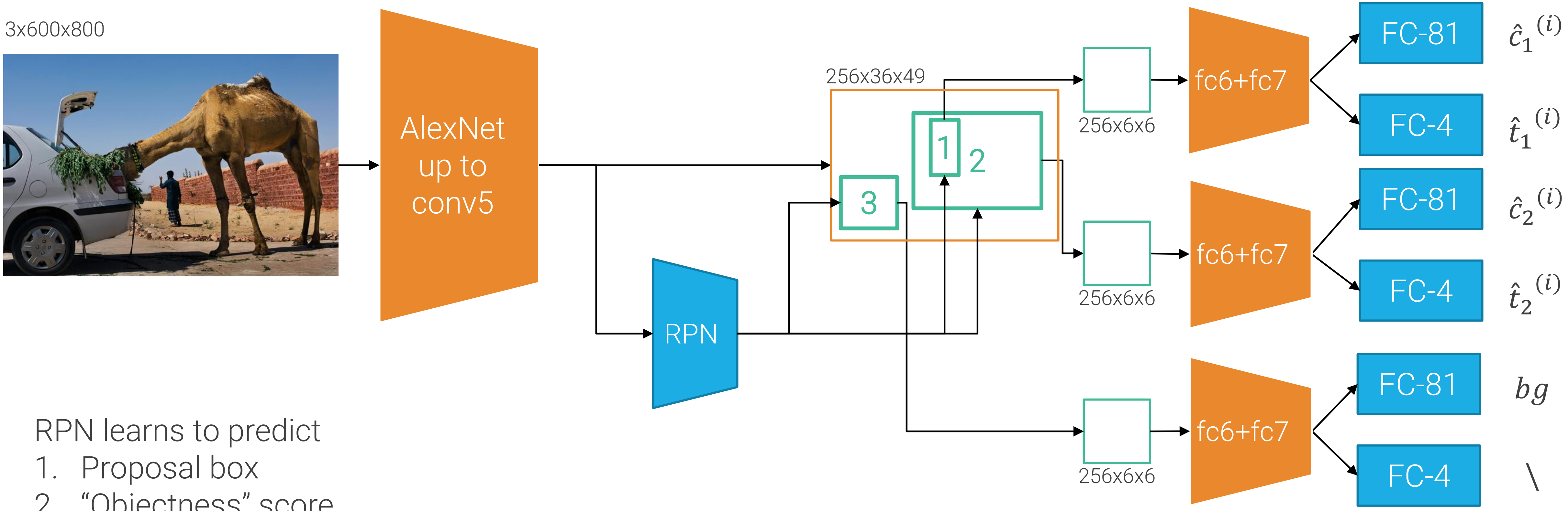
Faster R-CNN

Run expensive backbone feature extractor once on the full image

Region proposal network generates proposals

RoIPool layer crops and warps conv features according to proposals

Per-region network computes output class and BB correction



RPN learns to predict
1. Proposal box
2. "Objectness" score

Shaoqing Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NeurIPS 2015.

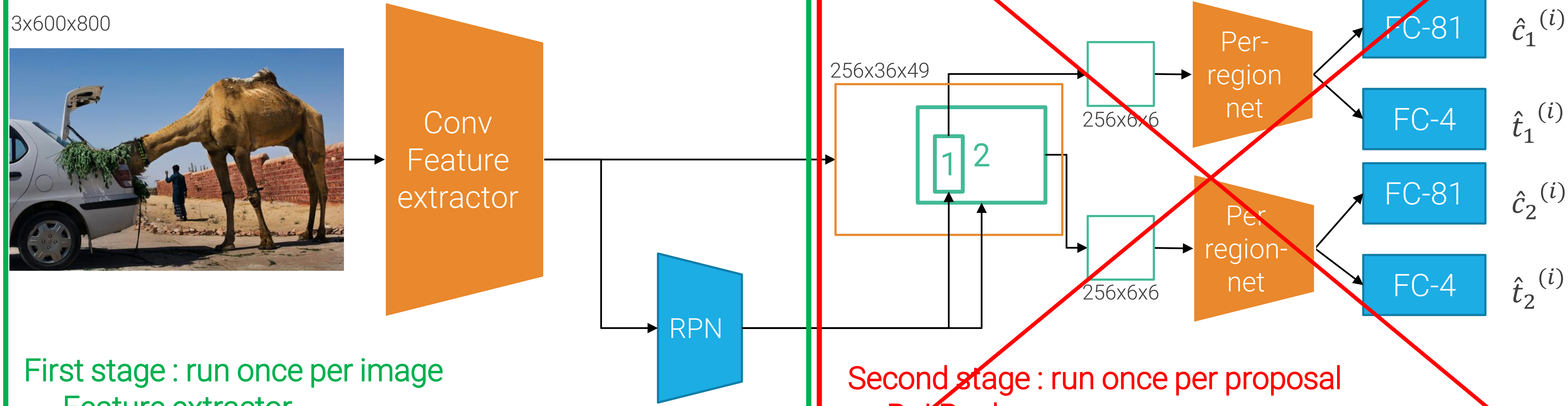
One-stage detectors: simplified view

Run expensive backbone feature extractor once on the full image

Region proposal network generates ~~proposals~~ output

RoIPool layer crops and warps conv features according to proposals

Per-region network computes output class and BB correction



First stage : run once per image
- Feature extractor
- "Region Proposal Network" <u>that also classifies boxes</u>

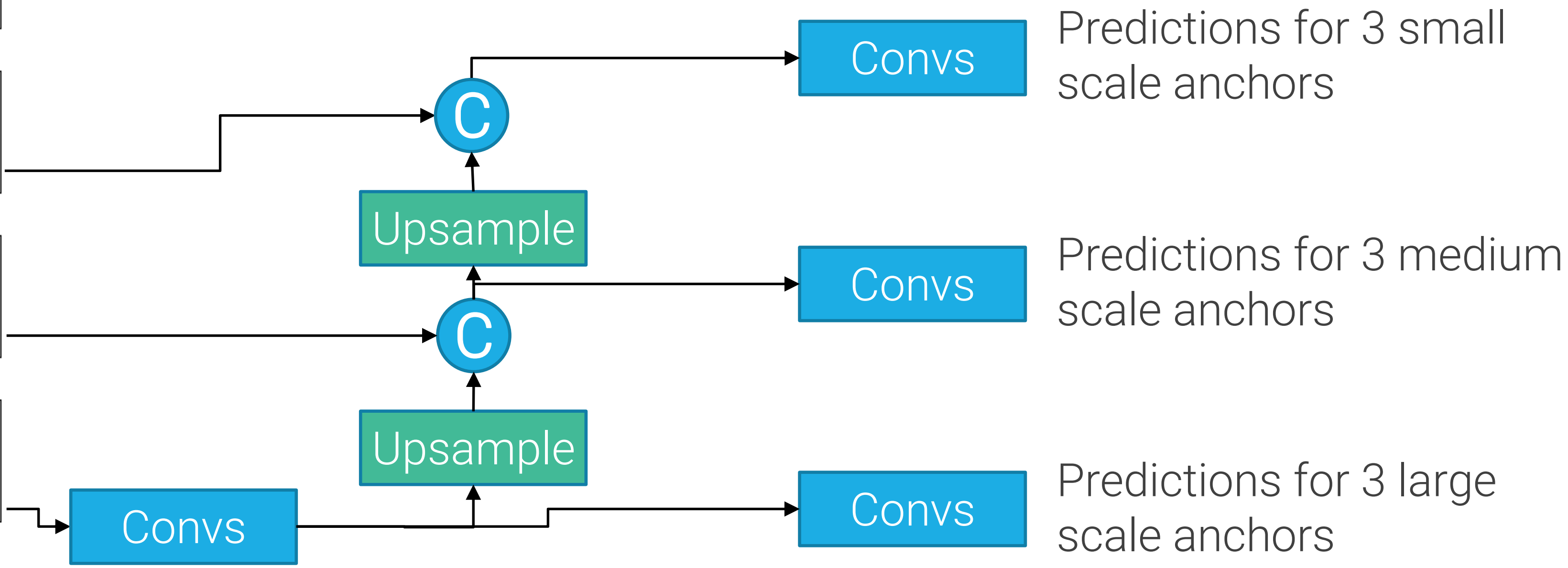
Second stage : run once per proposal
- RoI Pool
- Per-region classification and correction

YOLOv3

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
	Convolutional	32	1 × 1	
1×	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
	Convolutional	64	1 × 1	
2×	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
	Convolutional	128	1 × 1	
8×	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
	Convolutional	256	1 × 1	
8×	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
	Convolutional	512	1 × 1	
4×	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

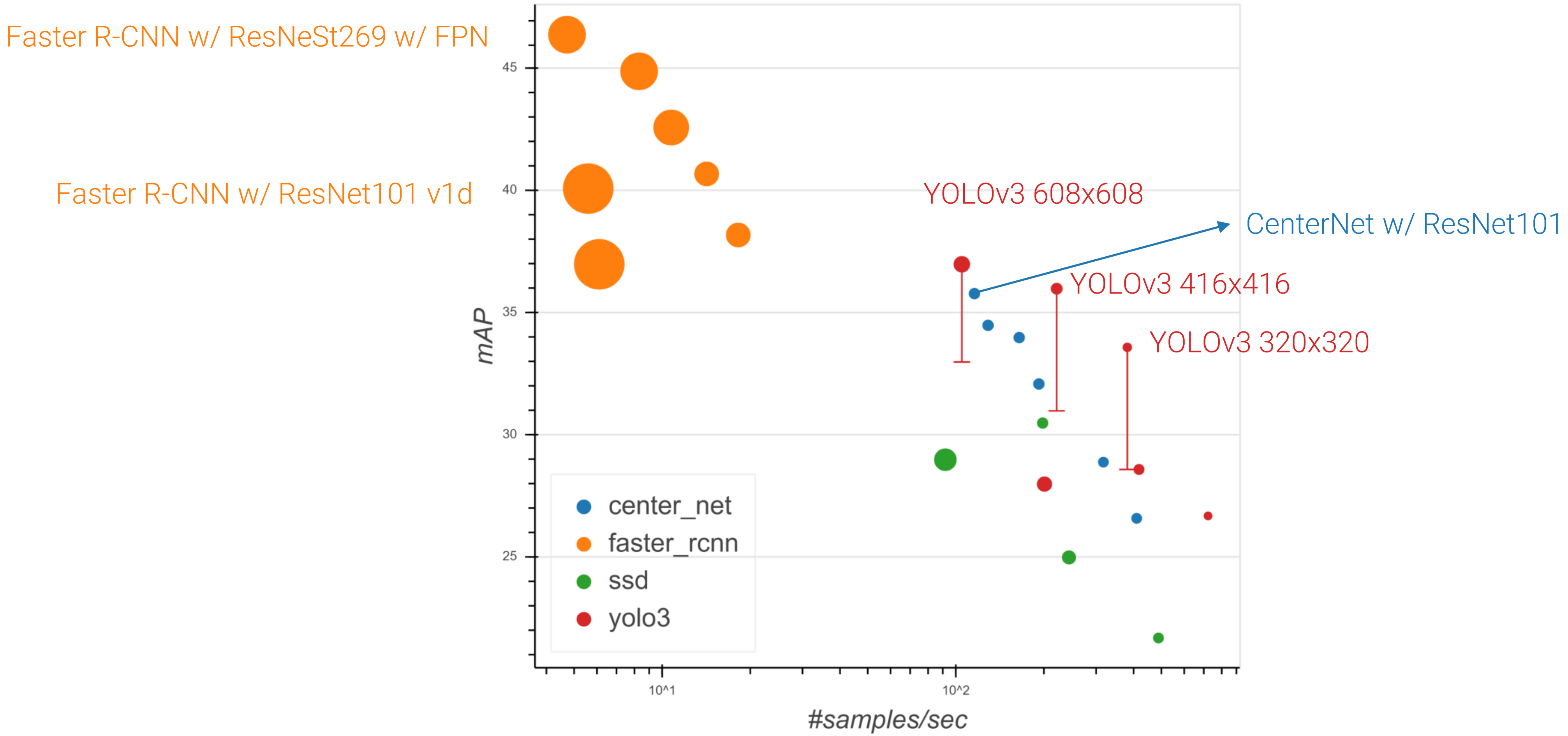
It uses a custom backbone (DarkNet-53) optimized to have a good trade-off between classification accuracy and speed.

It uses the idea of multi-scale detections on features with different spatial resolutions, as in FPN. It concatenates activations from different stages instead of summing them.



Joseph Redmon et Ali Farhadi., "YOLO9000: better, faster, stronger", CVPR 2017.
Joseph Redmon et Ali Farhadi., "YOLOv3: An Incremental Improvement", arXiv 2018.
https://pjreddie.com/darknet/yolo/

Comparison on COCO by GluonCV



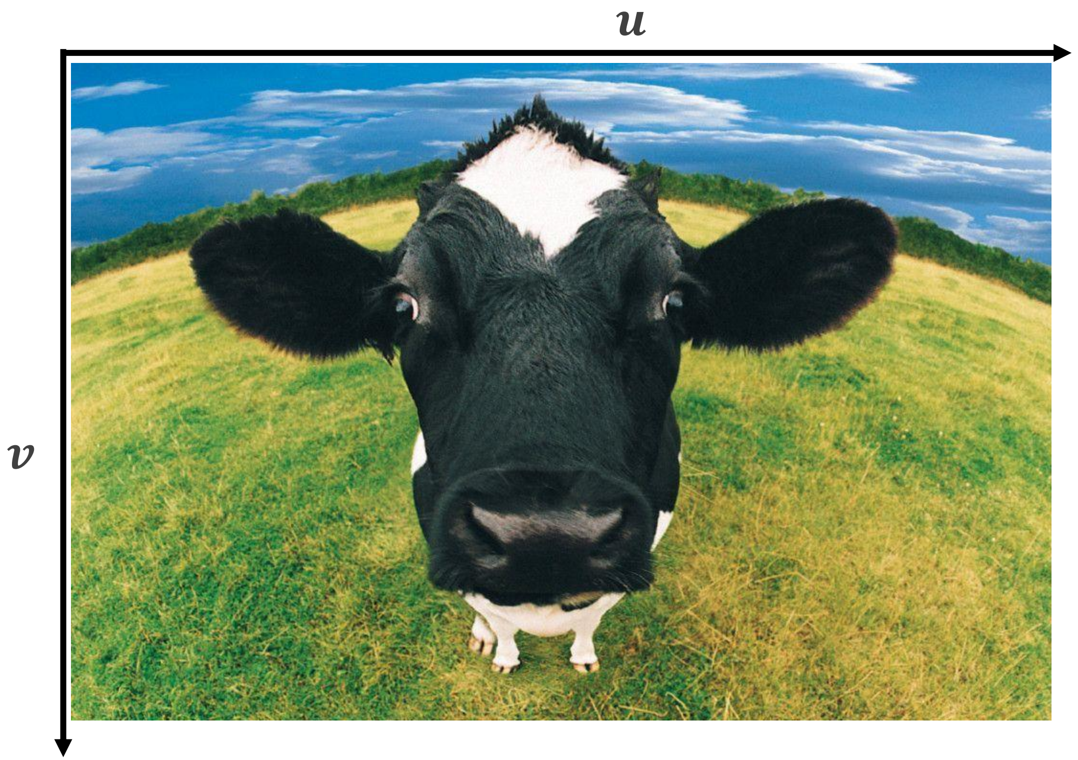
https://cv.gluon.ai/model_zoo/detection.html

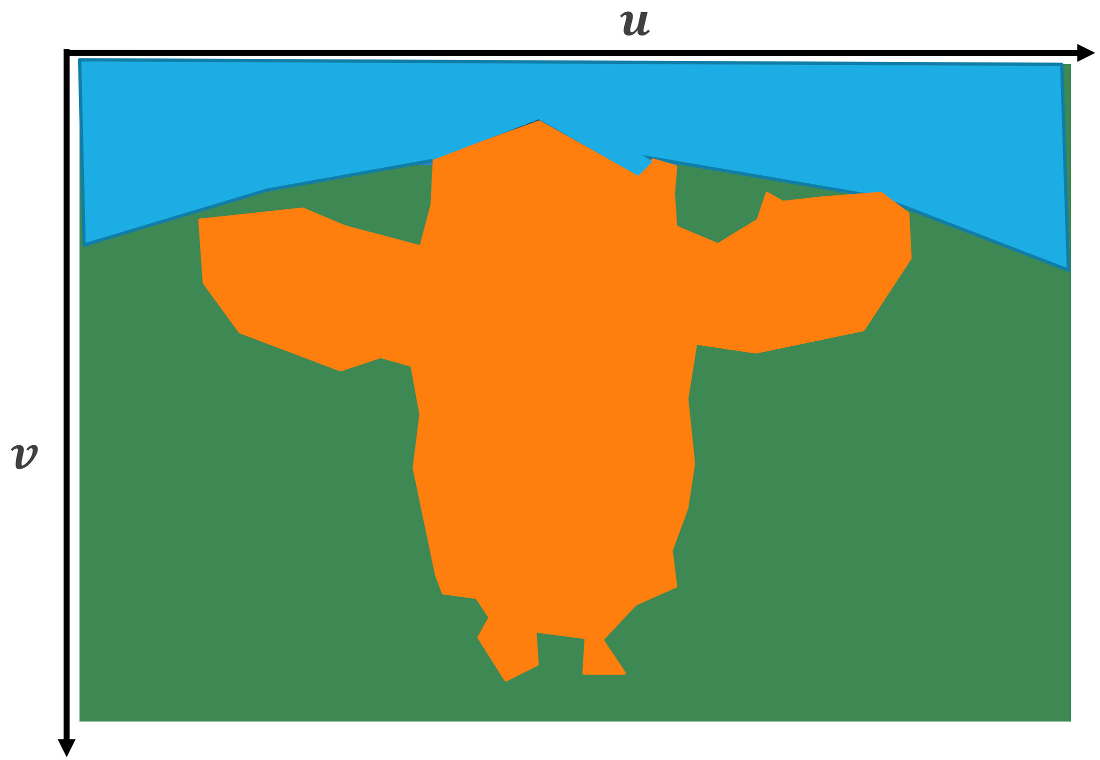
Semantic Segmentation

Problem definition

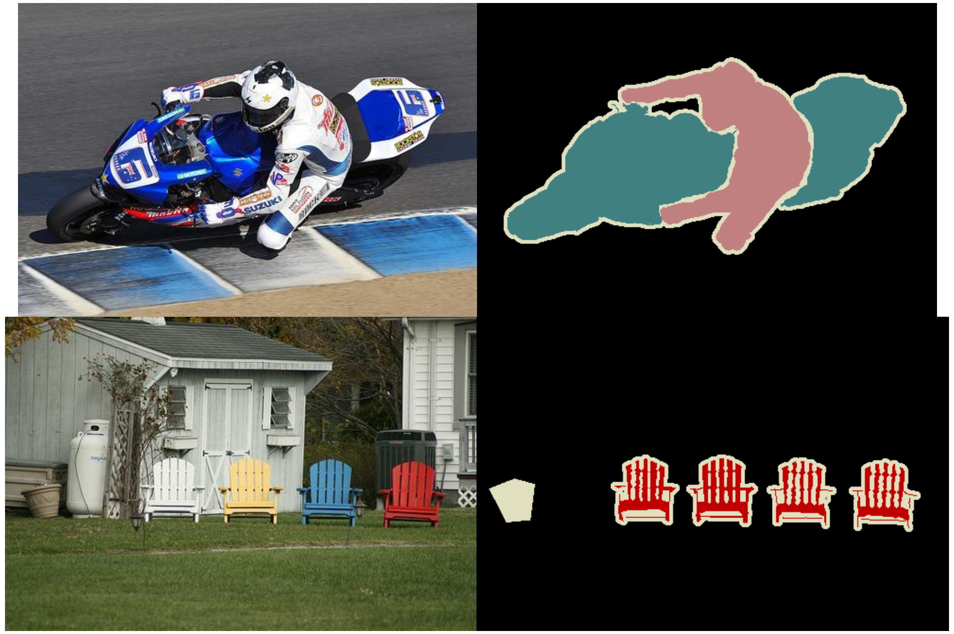
Input: RGB Image of size $W \times H$

Output: a category c_{uv} for each pixel $p = (u, v)$, $c_{uv} \in [1, \ldots, C]$ (a fixed list of categories, as in image classification)





Datasets



Trainval images: 11540 (6,929 segmentation masks)
20 categories

http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html

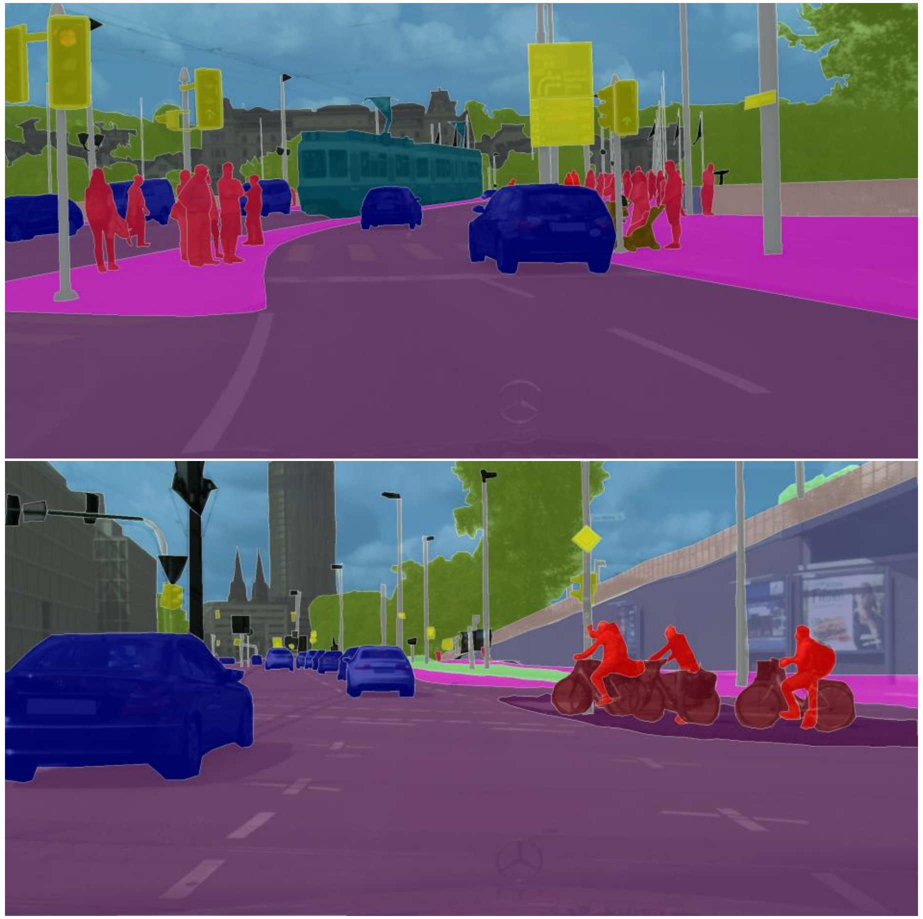
train/val images: 118K/5K
>100 categories

https://cocodataset.org/

Datasets

ADE20K

CITYSCAPES DATASET



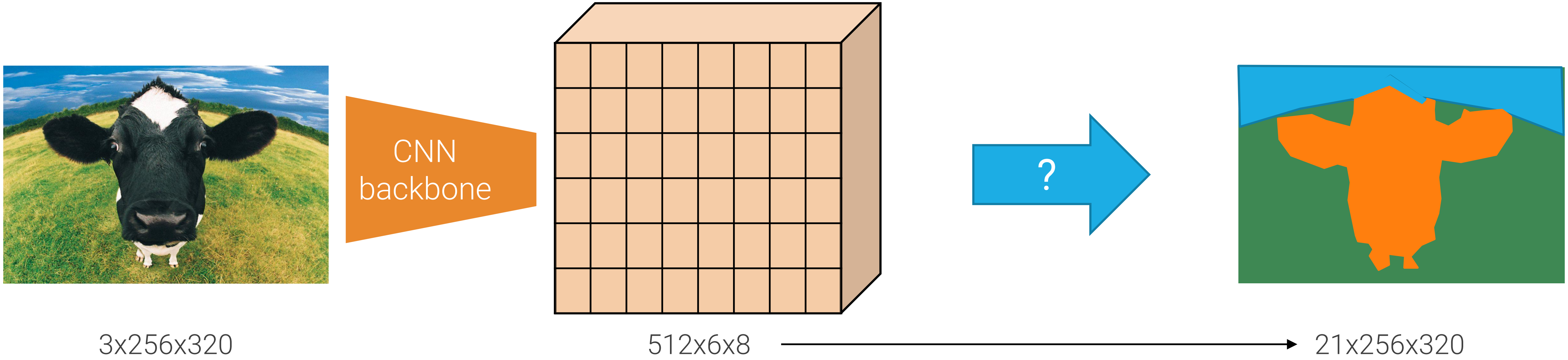
https://groups.csail.mit.edu/vision/datasets/ADE20K/

train/val images: 25K/2K
150 categories

https://www.cityscapes-dataset.com/

train/val images: 2750/500
30 categories, 19 used

Fully Convolutional Network (FCN)

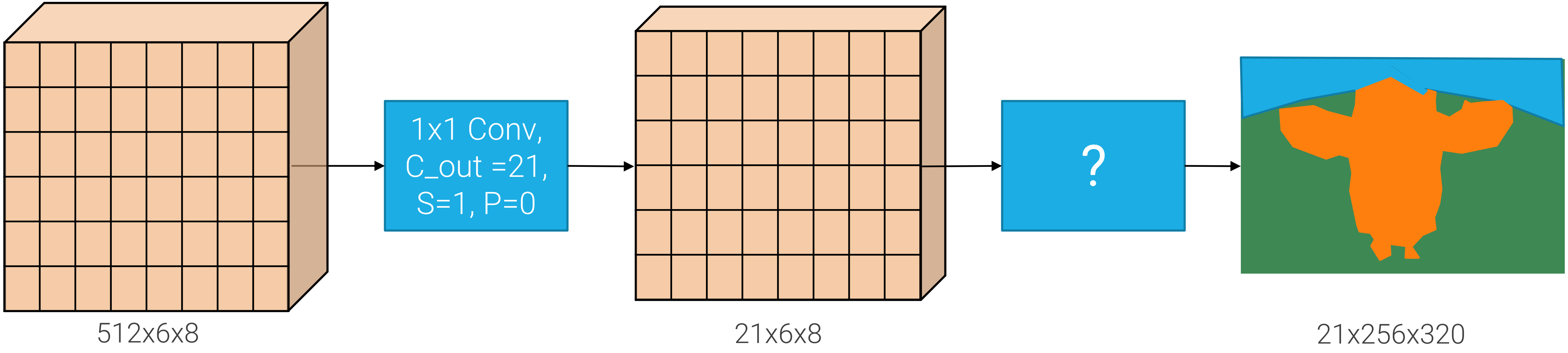


Long et al, “Fully convolutional networks for semantic segmentation”, CVPR 2015.
Evan Shelhamer, et al., “Fully convolutional networks for semantic segmentation”, PAMI 2017.

Fully Convolutional Network (FCN)

Fix channels to be equal to number of classes C

We need to convert coarse spatial class scores into fine grained scores with an upsampling operation



Long et al, "Fully convolutional networks for semantic segmentation", CVPR 2015

Upsampling

One way to perform upsampling can be to use **standard, not-learned image processing operators**

Input

1	2
3	4

Cx2x2

Nearest Neighbor

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

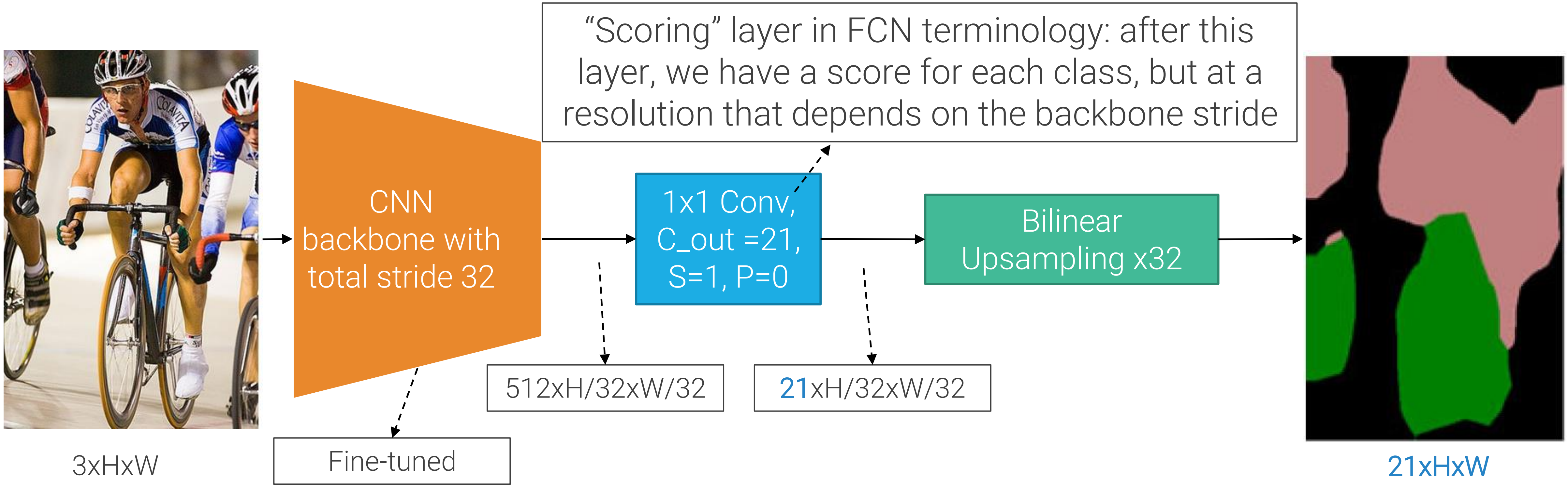
Cx4x4

Bilinear interpolation

1	1.25	1.75	2
1.50	1.75	2.25	2.5
2.5	2.75	3.25	3.5
3	3.25	3.75	4

Cx4x4

FCN-32s

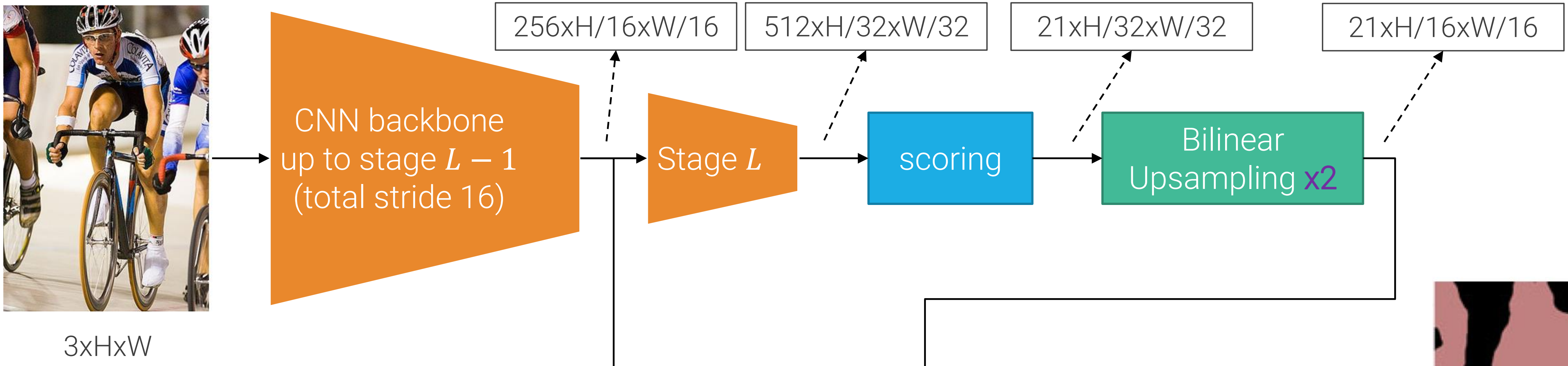


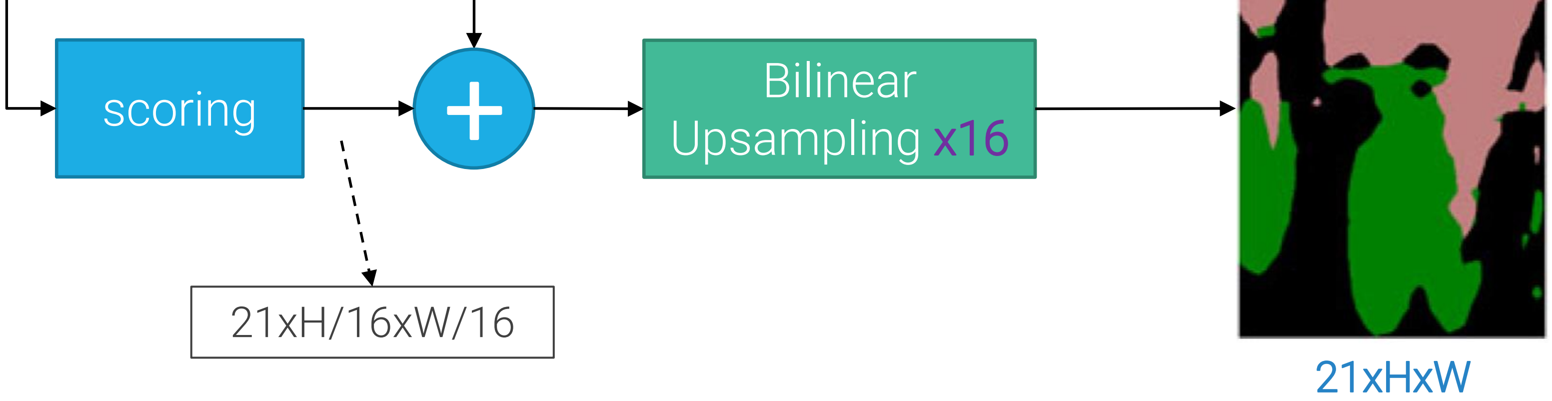
Problem: without learning a non-linear upsampling transformation, we can only uniformly spread the coarse info in the final convolutional activation, obtaining very coarse masks.

Solution: upsample multiple activations at different resolutions

Long et al, "Fully convolutional networks for semantic segmentation", CVPR 2015.
Evan Shelhamer, et al., "Fully convolutional networks for semantic segmentation", PAMI 2017.

FCN-16s



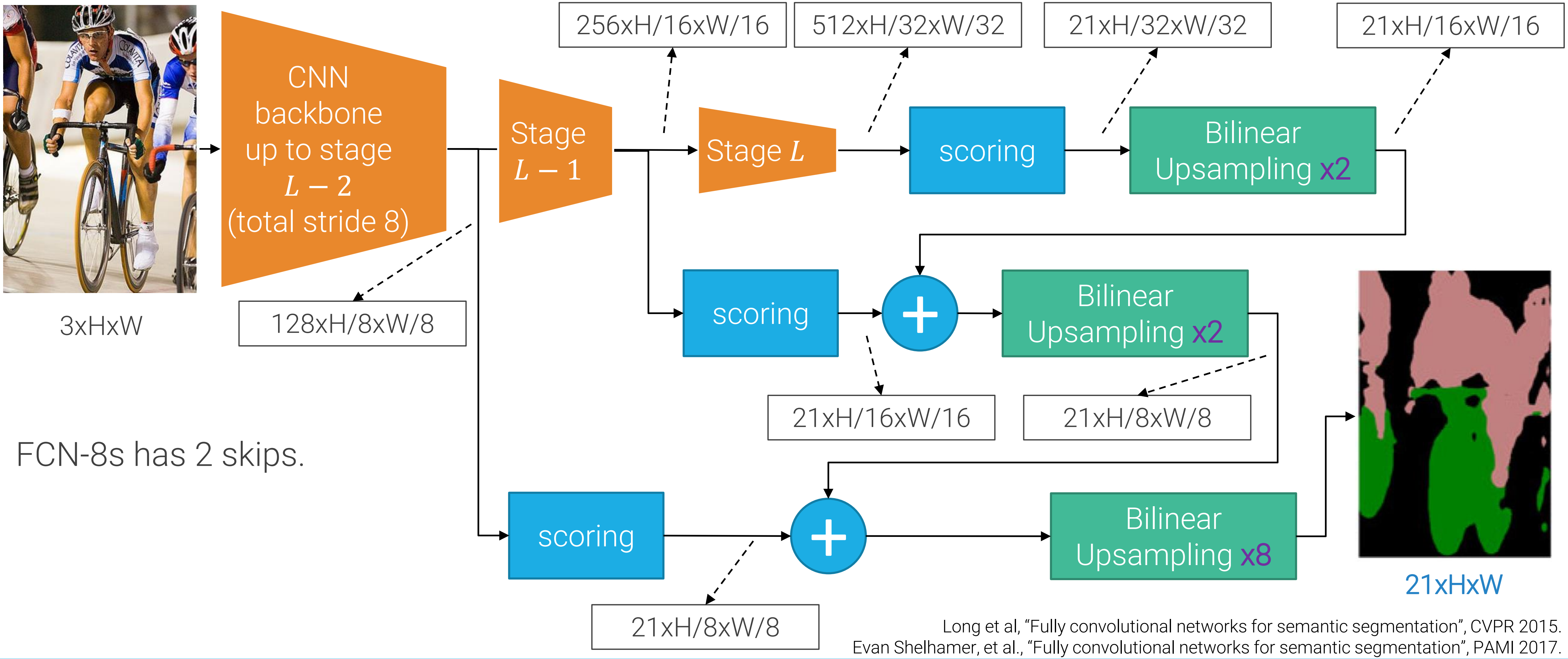


The additional connections between output and internal layers are referred to as skips.

FCN-16s has 1 skip.

Long et al, "Fully convolutional networks for semantic segmentation", CVPR 2015.
Evan Shelhamer, et al., "Fully convolutional networks for semantic segmentation", PAMI 2017.

FCN-8s



FCN-8s has 2 skips.

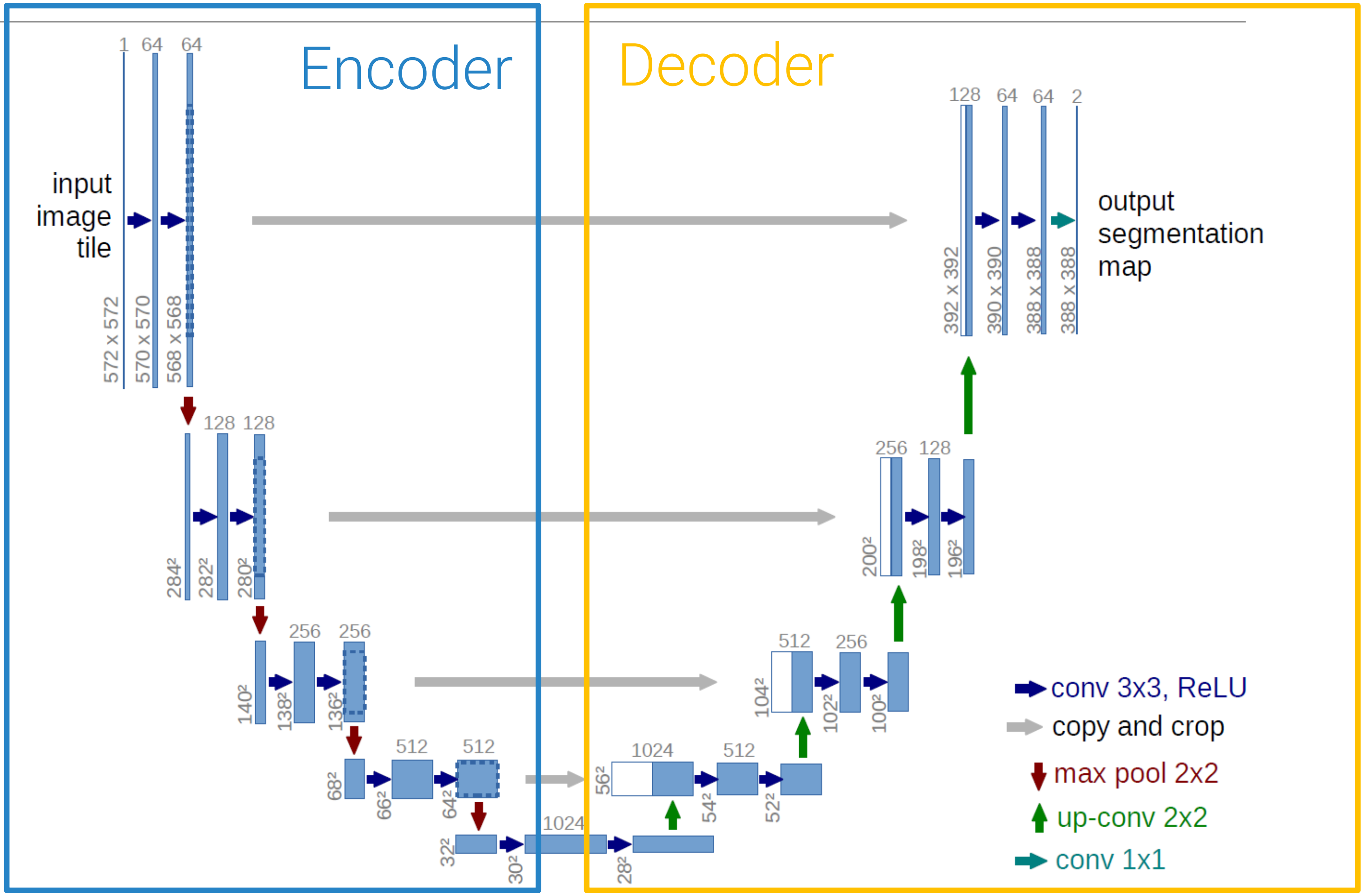
Long et al, "Fully convolutional networks for semantic segmentation", CVPR 2015.
Evan Shelhamer, et al., "Fully convolutional networks for semantic segmentation", PAMI 2017.

U-net

It extends the idea of skips from FCNs to create a full-fledged **decoder**, which has roughly a symmetric structure with respect to the encoder.

Every activation produced by a stage of the backbone (or **encoder**, or "contracting path") has **a skip connection** with the corresponding level of the decoder (or "expansive path").

Scoring layer only at the end to project onto the desired number of classes



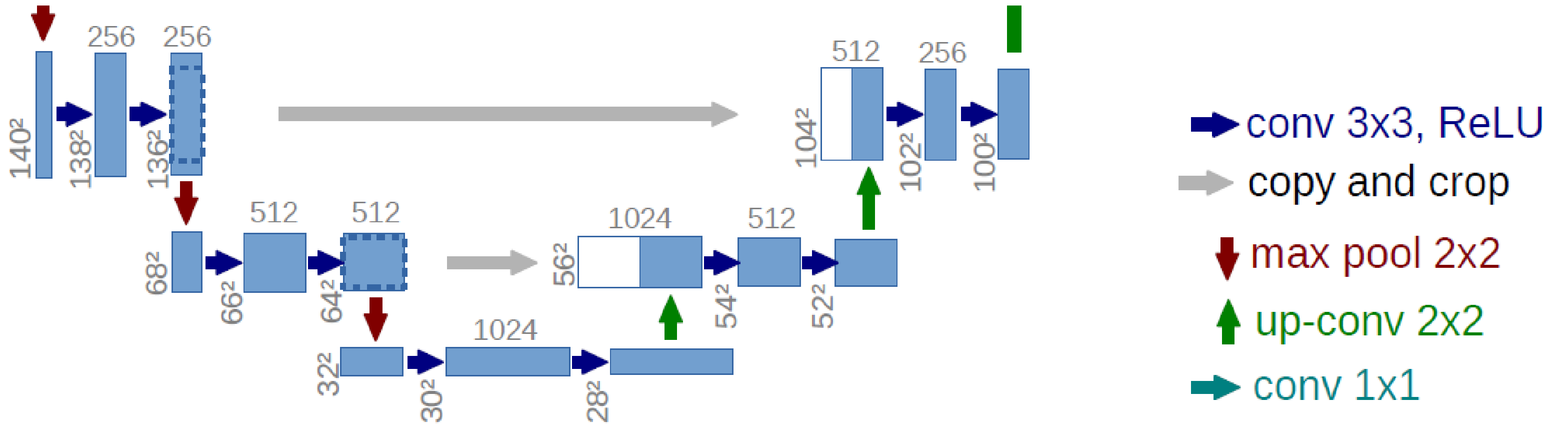
Ronneberger et al., "U-Net: Convolutional Networks for Biomedical Image Segmentation", MICCAI 2015

U-net

Skip connections use concatenation instead of summation as in FCN.

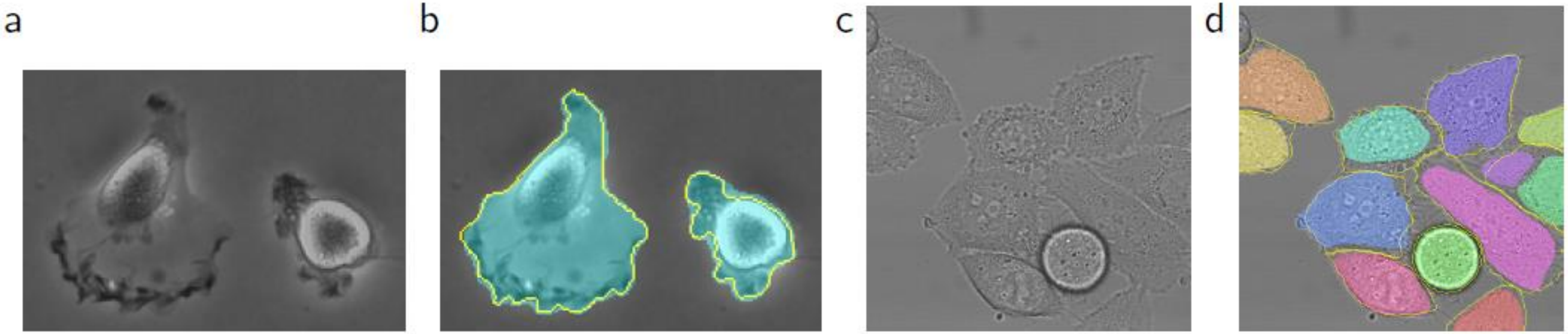
2x2 stride-2 transposed convolutions ("up-convolutions") are used to upsample the activations in the decoder, while halving the number of channels.

Normal 3x3 convolutions are used in the decoder as well: with further processing, even initial layers of the backbone can effectively contribute to the final segmentation mask, as opposed to what happened in FCN.



Ronneberger et al., "U-Net: Convolutional Networks for Biomedical Image Segmentation", MICCAI 2015

Unet - results



Recent topics

Supervised learning is great but…

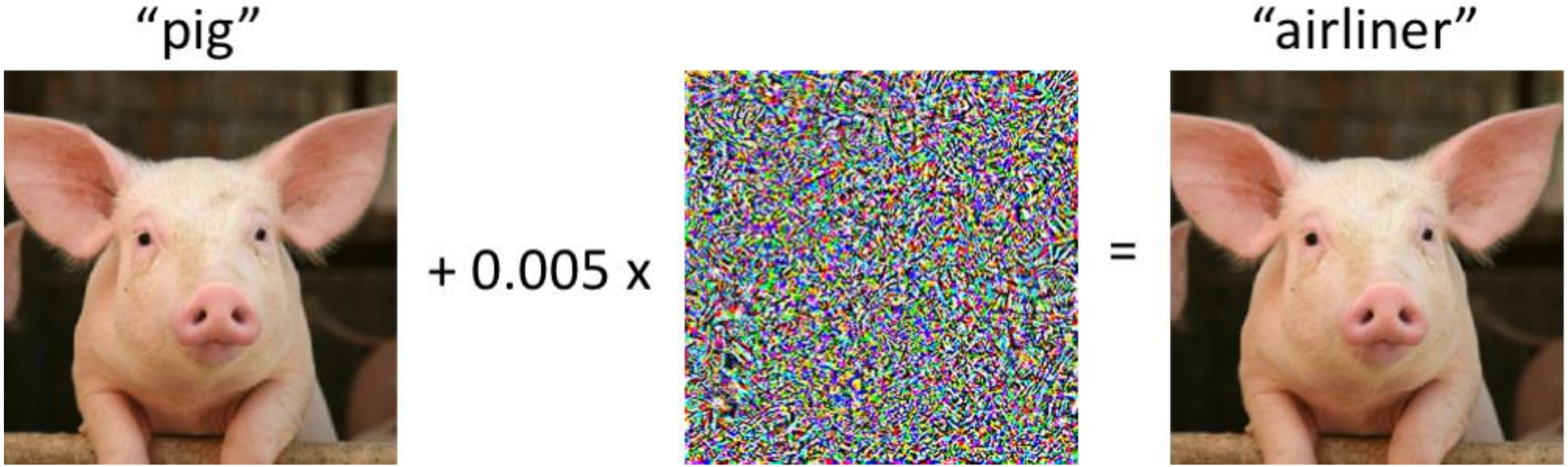
Labels are expensive

- ImageNet 21k took 22 human years, and it "only" contains 21k concepts

Ground-truth is a convenient fiction

- Sometimes labels are ambiguous
- Not all the visual tasks allow for easy collection of labels

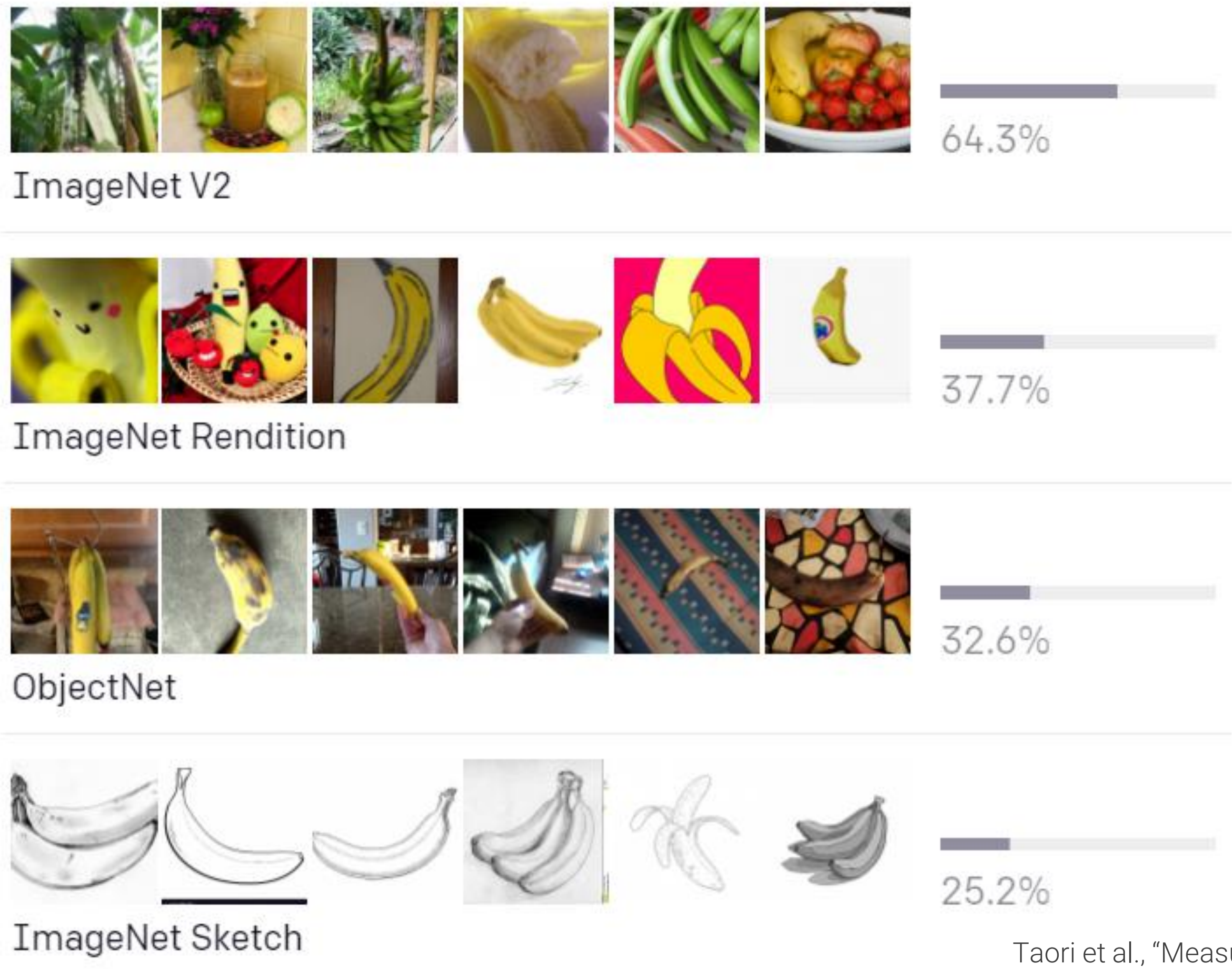
…it is far from perfect…

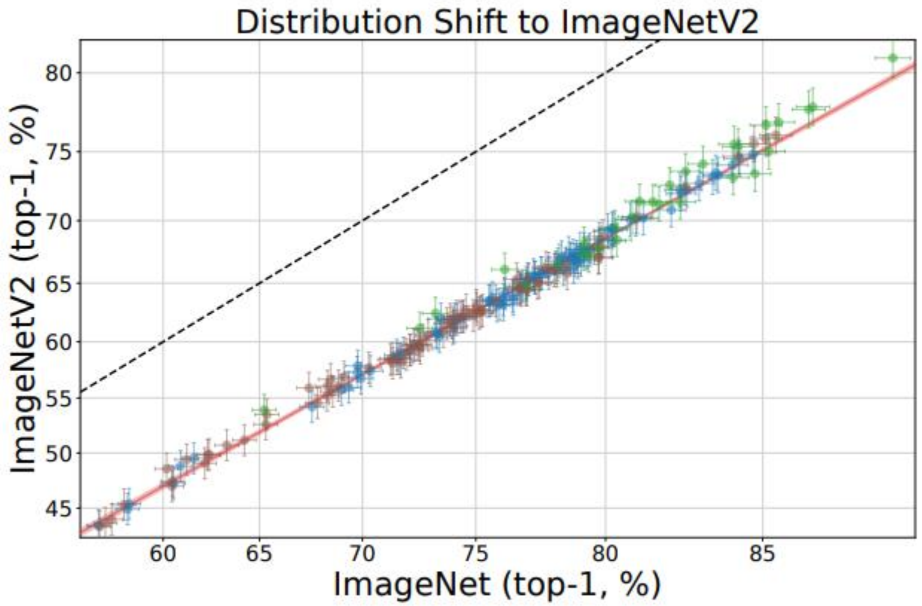


Performance of supervised methods is brittle: for instance, there exist adversarial examples, i.e. images that are wrongly classified when they are imperceptibly modified for humans

Ian J. Goodfellow et al. : "Explaining and Harnessing Adversarial Examples". ICLR 2015

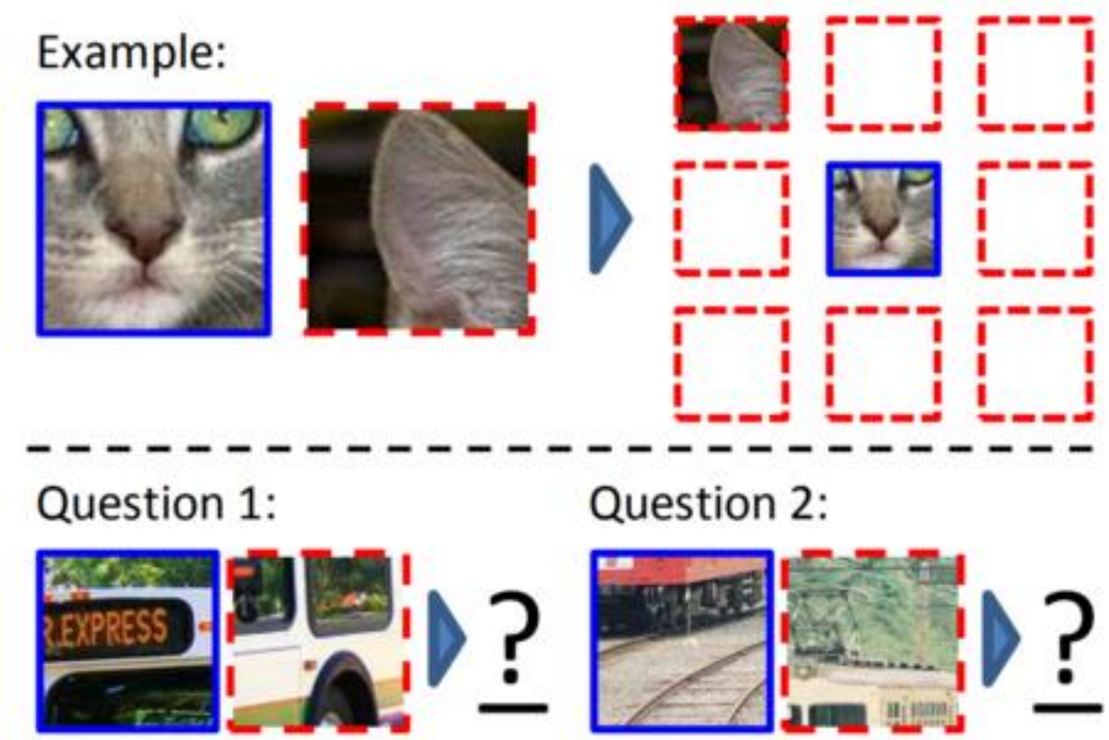
… and overfits the benchmarks (lack "common sense")

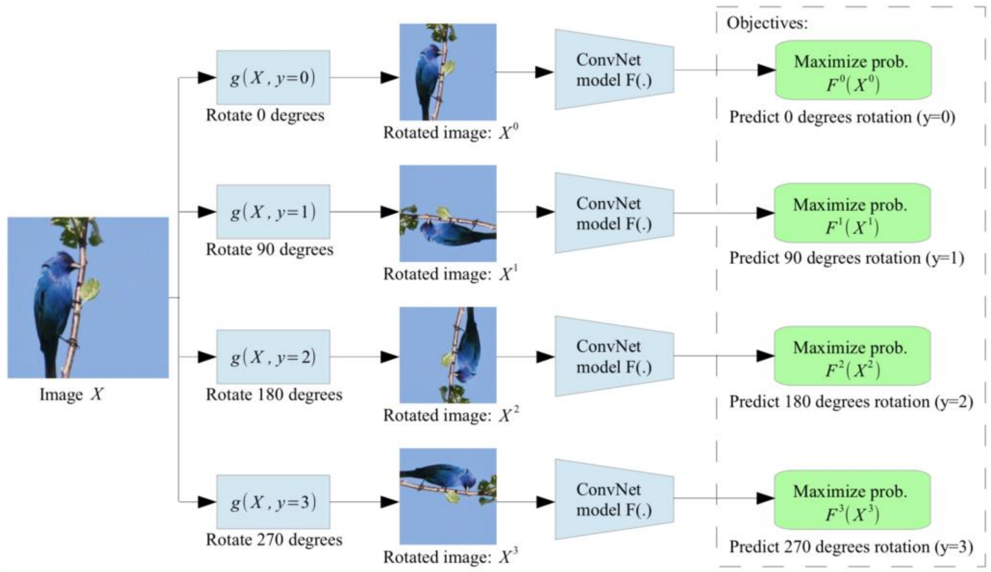


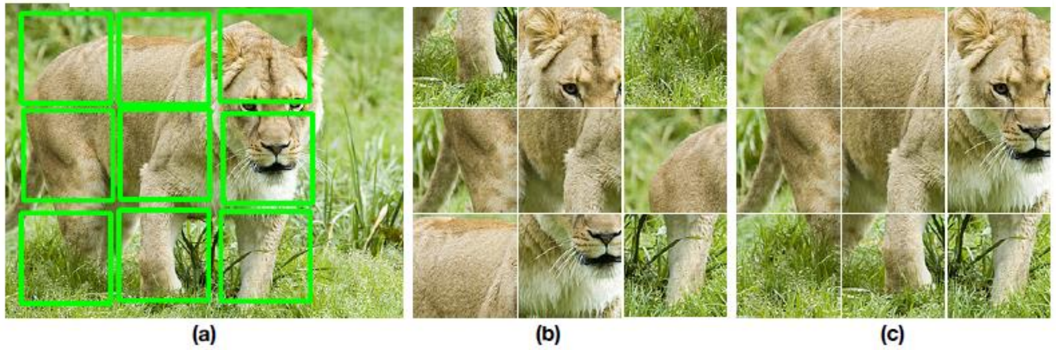


Taori et al., "Measuring Robustness to Natural Distribution Shifts in Image Classification", NeurIPS 2020.

Self-supervised learning



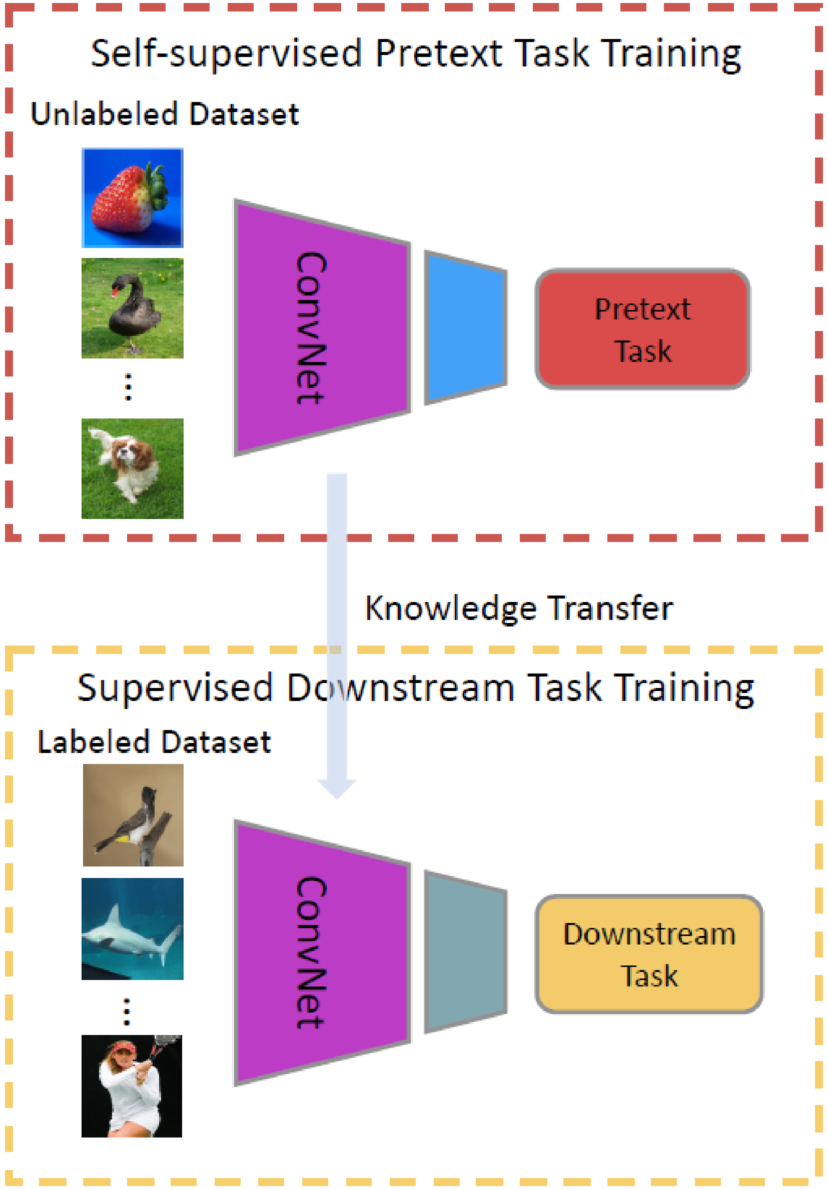




S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations" in ICLR 2018.
Doersch et al. "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015
M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles", ECCV 2016.

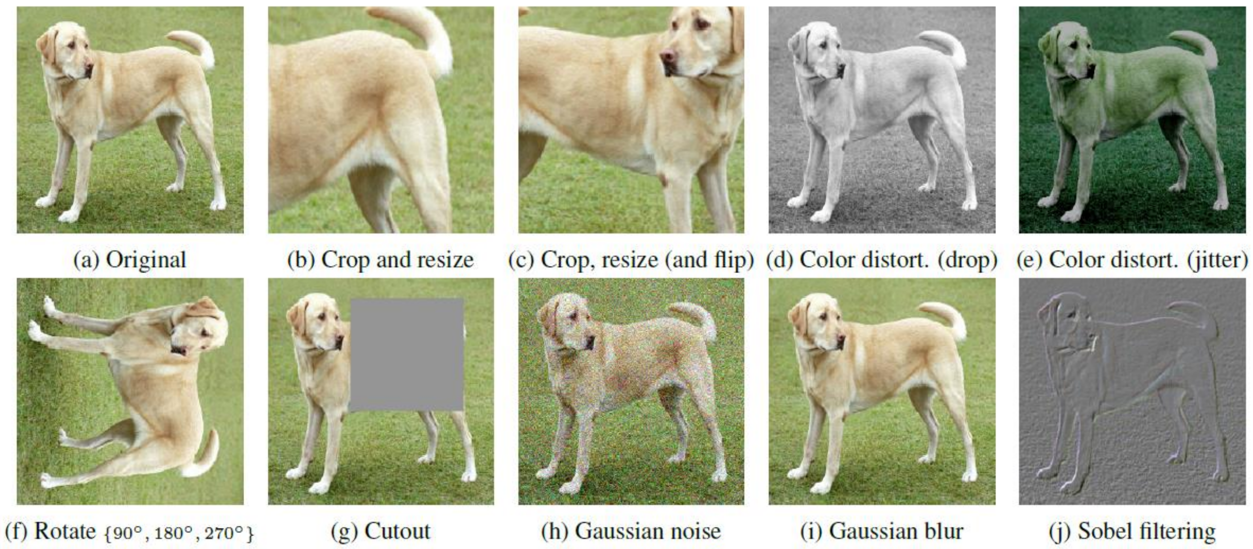
Self-supervised learning

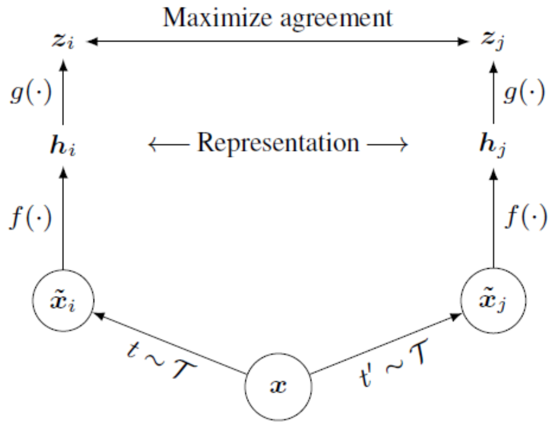
- When used to learn effective representations to bootstrap/improve supervised learning, unsupervised learning is (often) referred to as self-supervised learning.
- "Self-supervised learning is a subset of unsupervised learning methods […] in which [neural networks] are explicitly trained with automatically generated labels (pseudo-labels)."
- The task solved while performing self-supervised learning is often referred to as pretext task



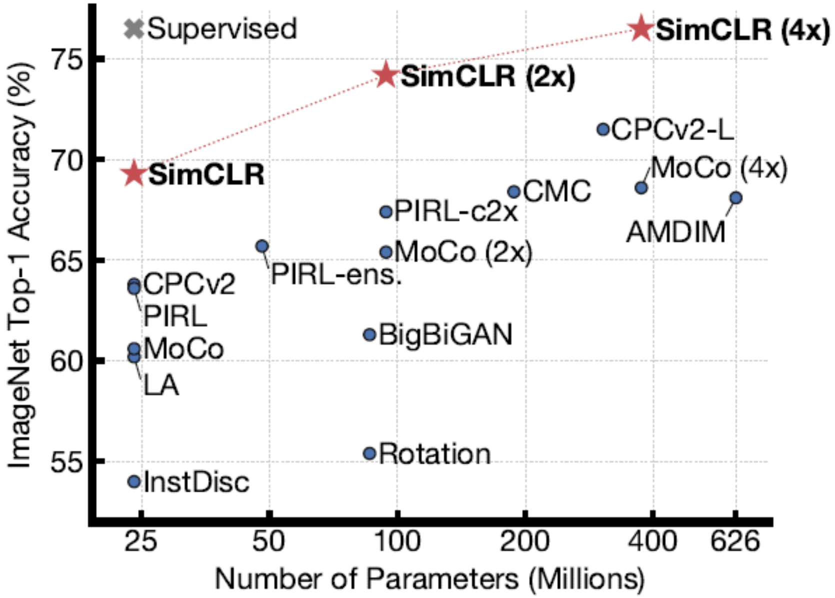
Longlong Jing and Yingli Tian, "Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey", https://arxiv.org/abs/1902.06162

Self-supervision by contrastive learning





State-of-the-art self-supervised methods **are closing the gap** with respect to the supervised counterpart in some tasks.



Ting Chen et al. , "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020
Ting Chen et al. , "Big Self-Supervised Models are Strong Semi-Supervised Learners", NeurIPS 2020

SSL is the “dark matter” of AI

Supervised learning is a **bottleneck** for building more intelligent generalist models that can do multiple tasks and acquire new skills without massive amounts of labeled data. […]

A working hypothesis is that **generalized knowledge about the world, or common sense,** forms the bulk of biological intelligence in both humans and animals. This common sense ability is taken for granted in humans and animals but has remained an open challenge in AI research since its inception. In a way, common sense is the dark matter of artificial intelligence.

Common sense helps people learn new skills without requiring massive amounts of teaching for every single task. For example, if we show just a few drawings of cows to small children, they’ll eventually be able to recognize any cow they see. By contrast, AI systems trained with supervised learning require many examples of cow images and might still fail to classify cows in unusual situations, such as lying on a beach. How is it that humans can learn to drive a car in about 20 hours of practice with very little supervision, while fully autonomous driving still eludes our best AI systems trained with thousands of hours of data from human drivers? The short answer is that humans rely on their previously acquired background knowledge of how the world works.

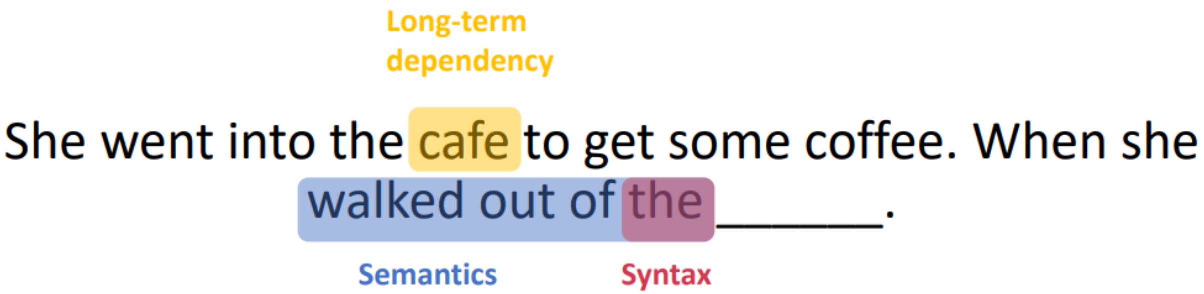
How do we get machines to do the same? We believe that **self-supervised learning (SSL)** is one of the most promising ways to build such background knowledge and approximate a form of common sense in AI systems.

https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

Self-supervised learning in NLP

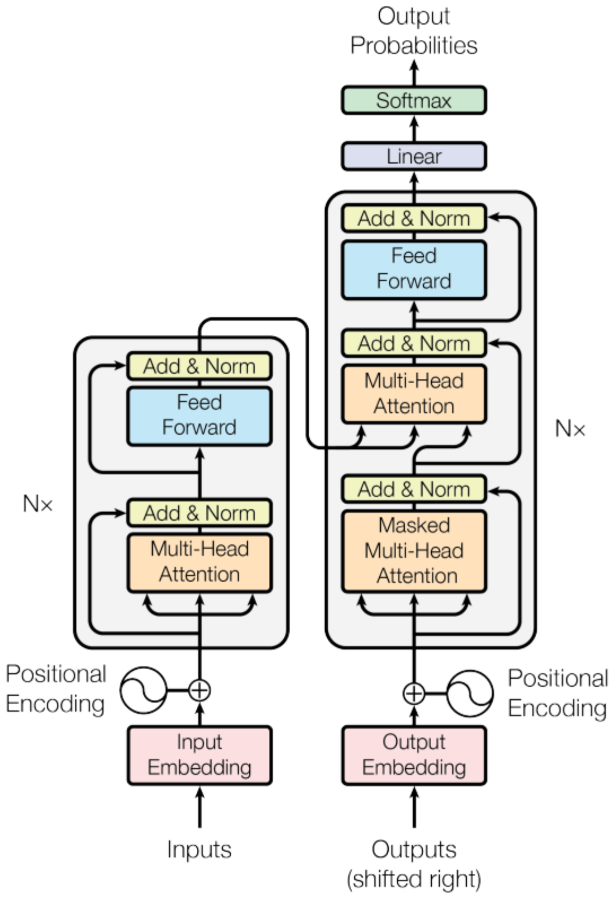
Self-supervised learning is routinely used in Natural Language Processing to learn language models, i.e. probability for the next word in a sentence.

Why is language modelling a good pretext task?



http://cs229.stanford.edu/notes2021spring/notes2021spring/cs229_lecture_selfsupervision_final.pdf

Transformers



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

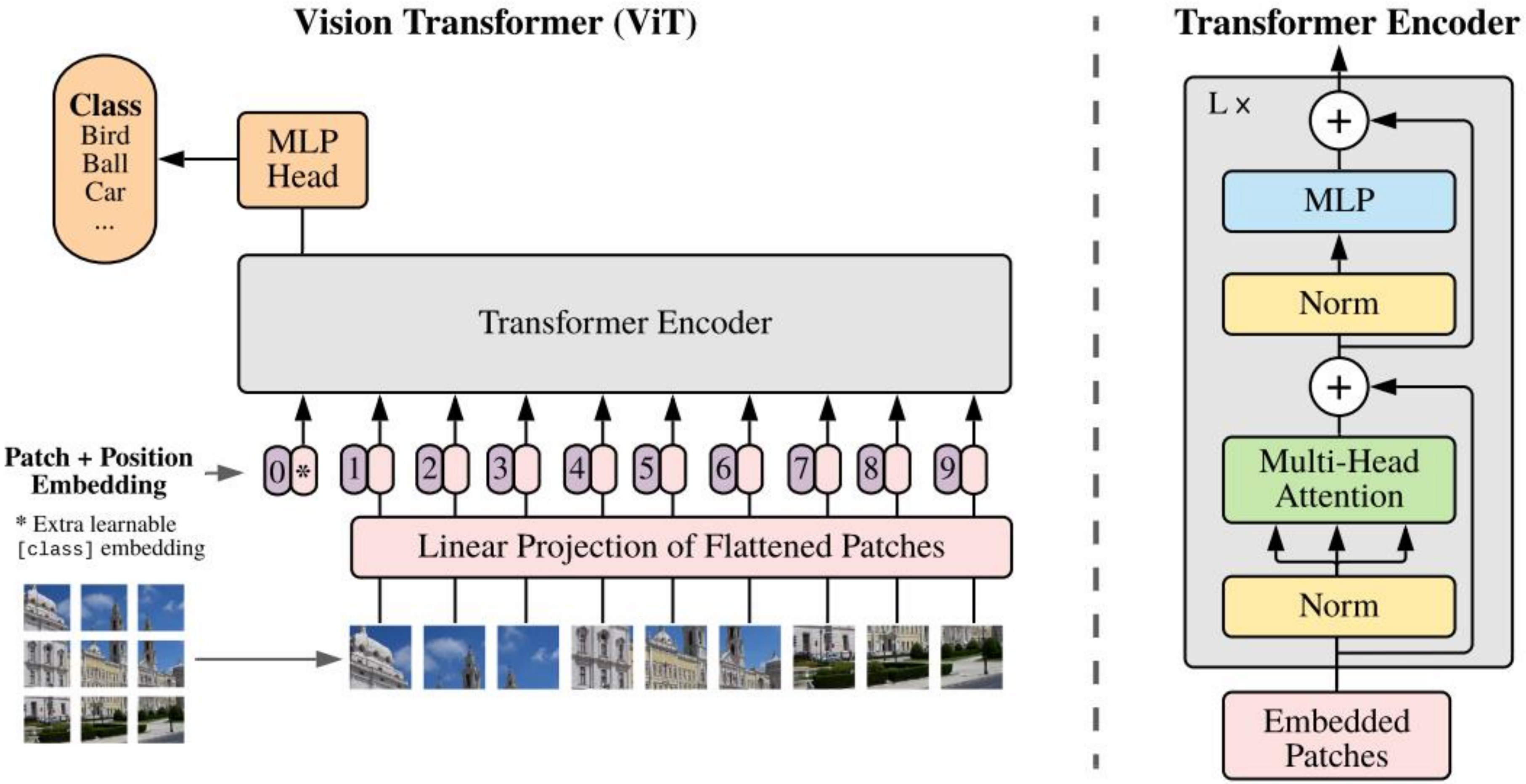
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

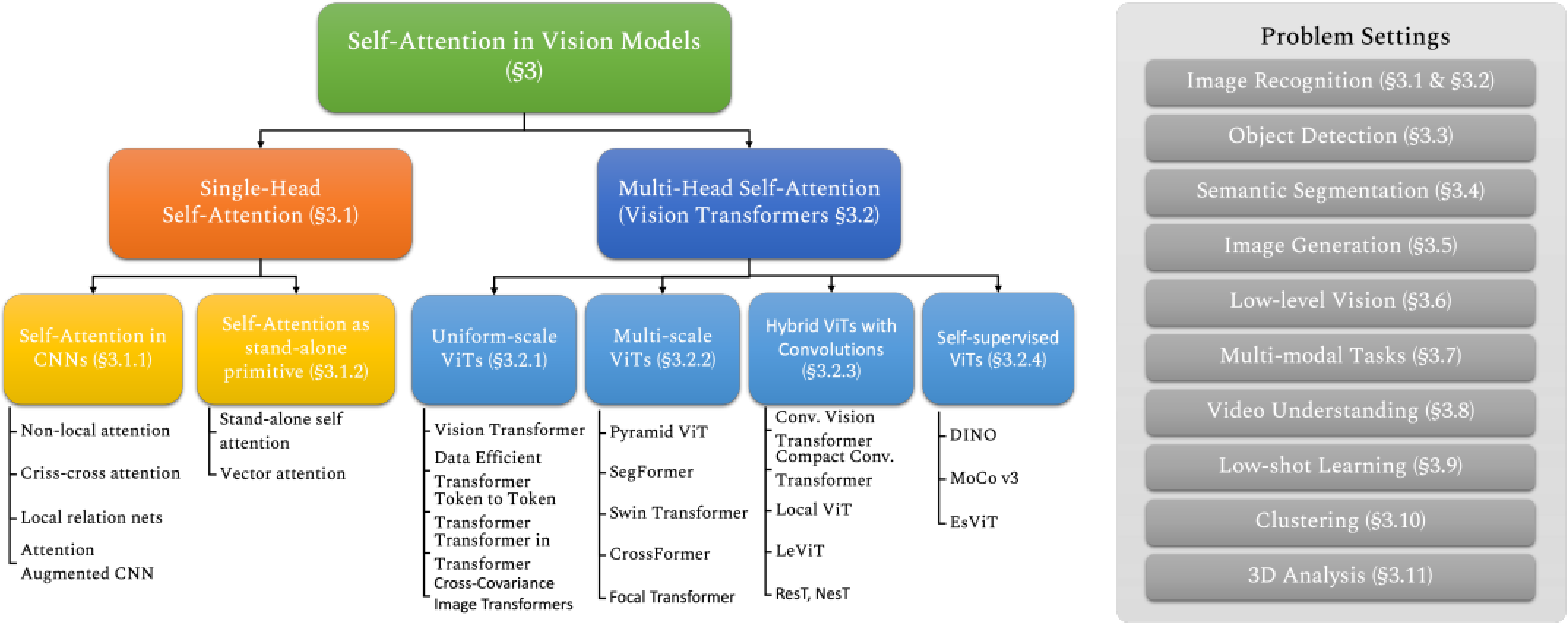
Ashish Vaswani et al., "Attention Is All You Need", NIPS 2017.

Vision Transformer (ViT)



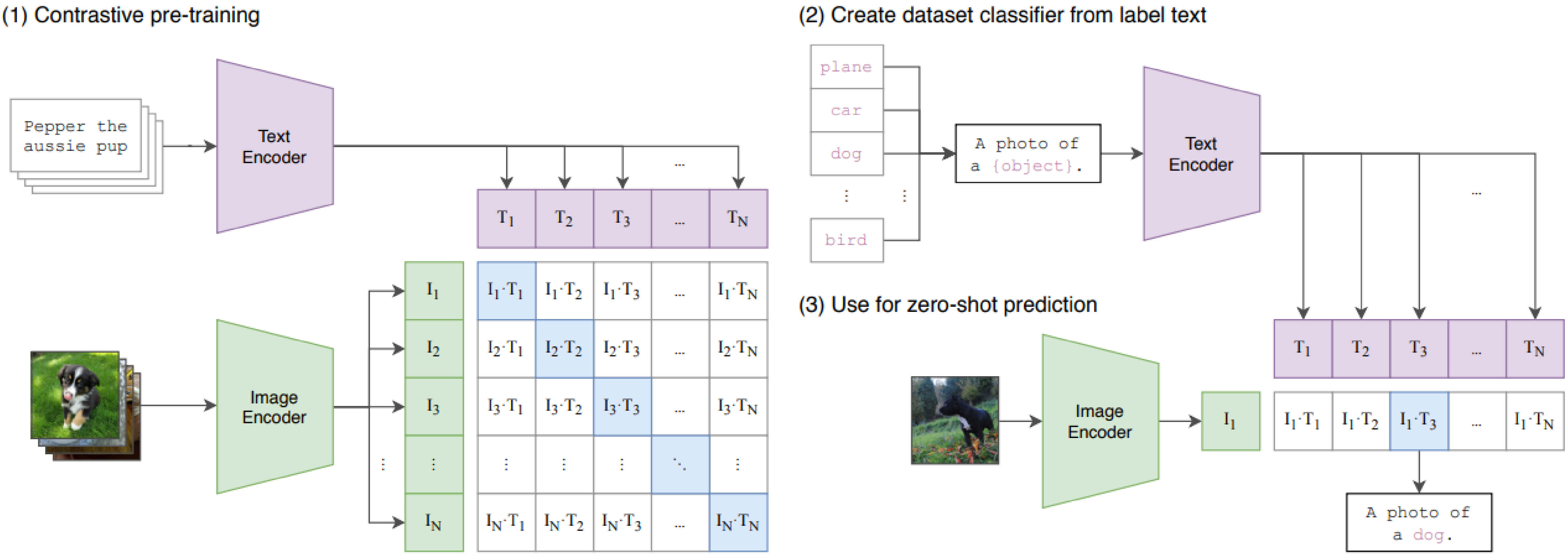
A. Dosovitskiy et al., "An Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale", ICLR 2020.

Transformers everywhere



Khan et al., "Transformers in Vision: A Survey", arXiv 2021.

CLIP (Contrastive Language–Image Pre-training)



A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision”, ICML 2021

CLIP – Results and beyond

DATASET	IMAGENET RESNET101	CLIP VIT-L
 ImageNet	76.2%	76.2%
 ImageNet V2	64.3%	70.1%
 ImageNet Rendition	37.7%	88.9%
 ObjectNet	32.6%	72.3%
 ImageNet Sketch	25.2%	60.2%
 ImageNet Adversarial	2.7%	77.1%

Prompt: "Colourful cubist painting of a parrot in a cage"

https://openai.com/blog/clip/
https://creator.nightcafe.studio/text-to-image-art