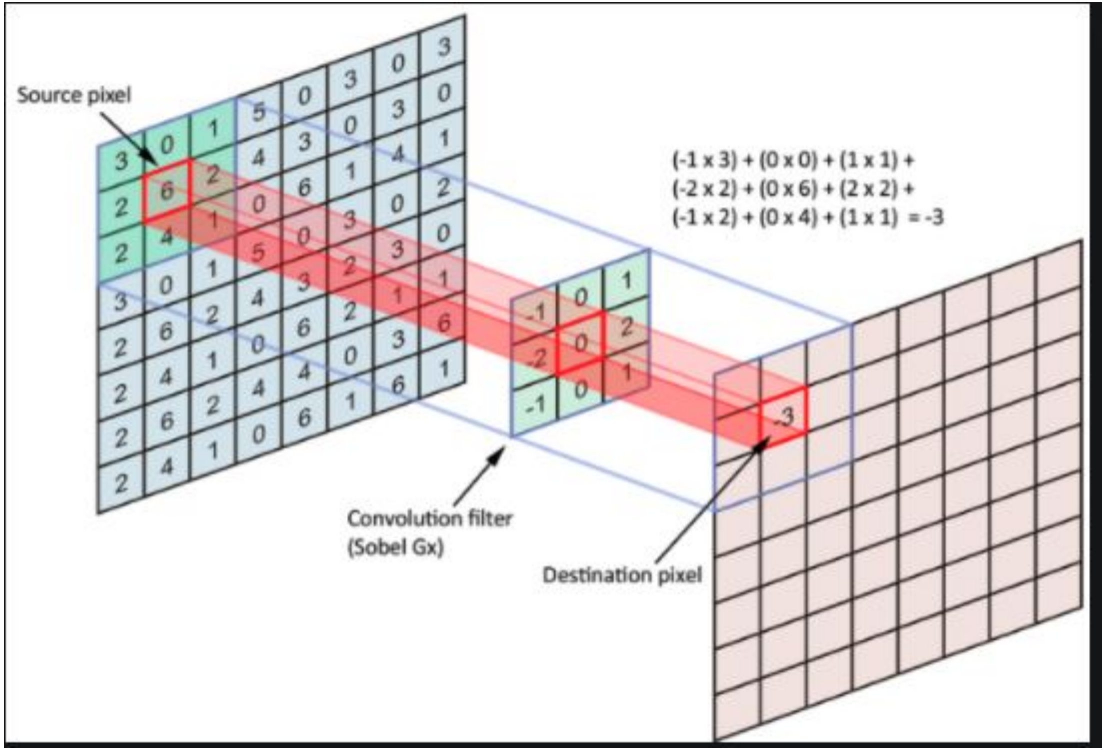
TP 7
Transfer Learning

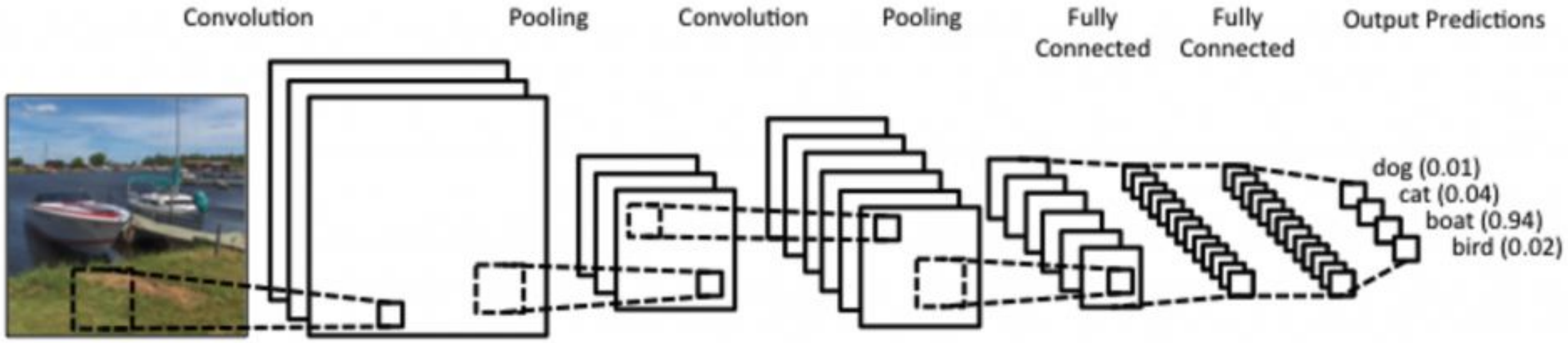
Convolutional Layer

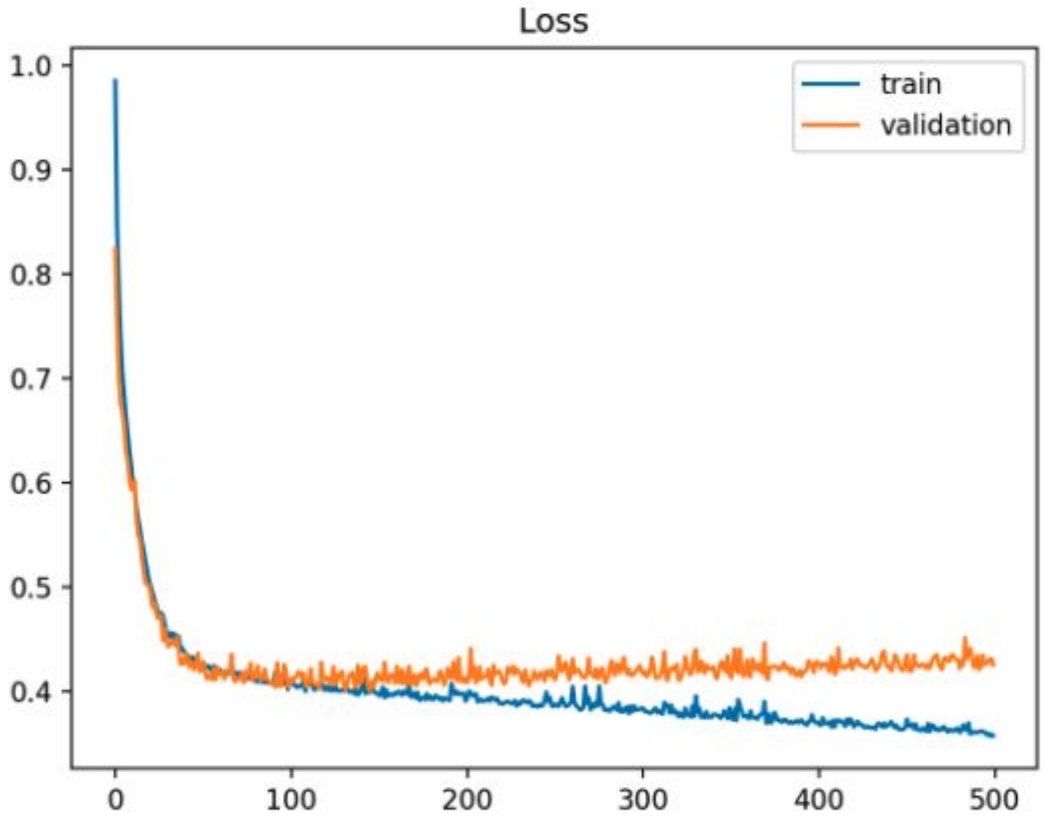


Different operations

Operation	Kernel ω	Image result g(x,y)
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 5 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	

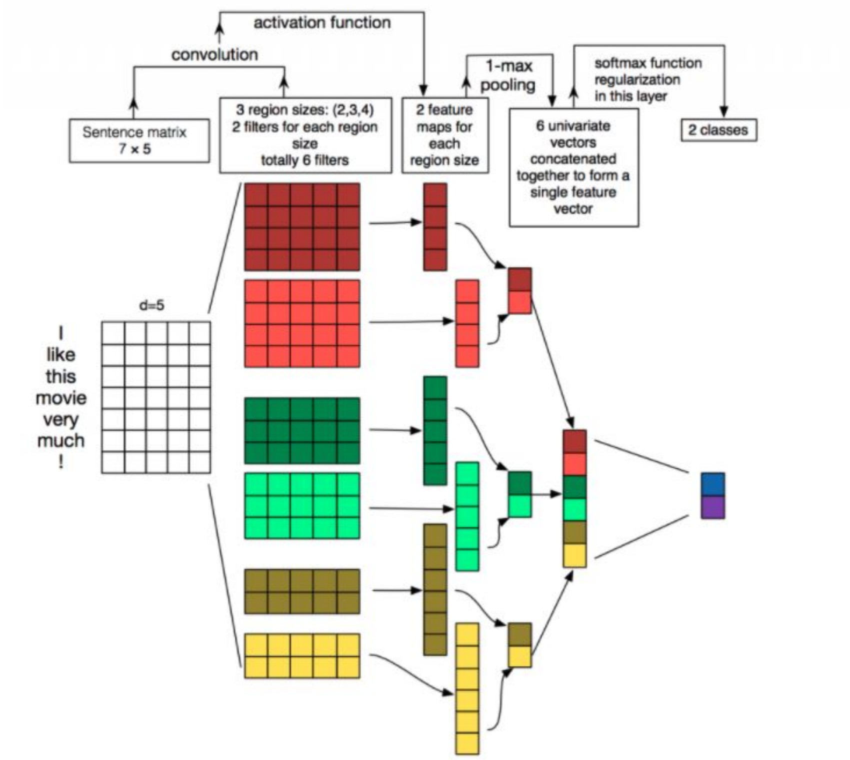
Convolutional Networks



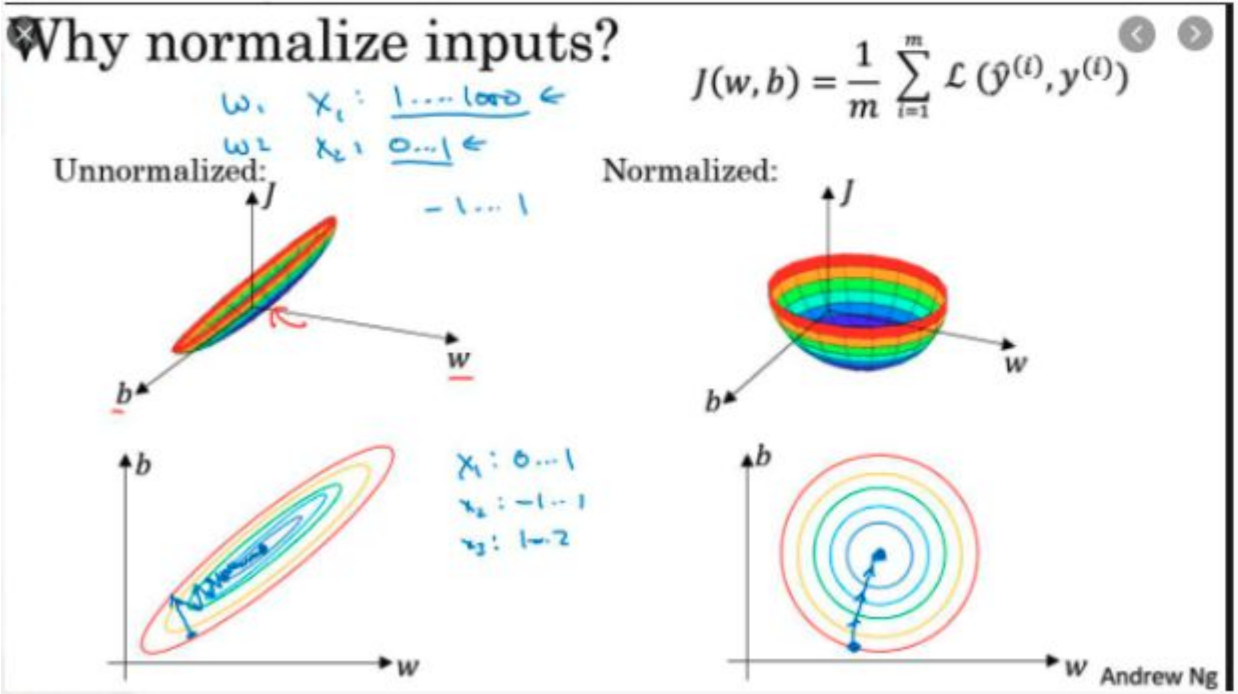


Example of Train and Validation Learning Curves Showing an Overfit Model

ConvNets for NLP



Tips & Tricks


Why normalize inputs?
$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$
w_1 x_1: 1....1000
w_2 x_2: 0...1
-1...1
Unnormalized:
Normalized:
J
b
w
x_1: 0...1
x_2: -1...1
x_3: 1...2
Andrew Ng

Data Augmentation

Different augmentations

Tranable Data Augmentation

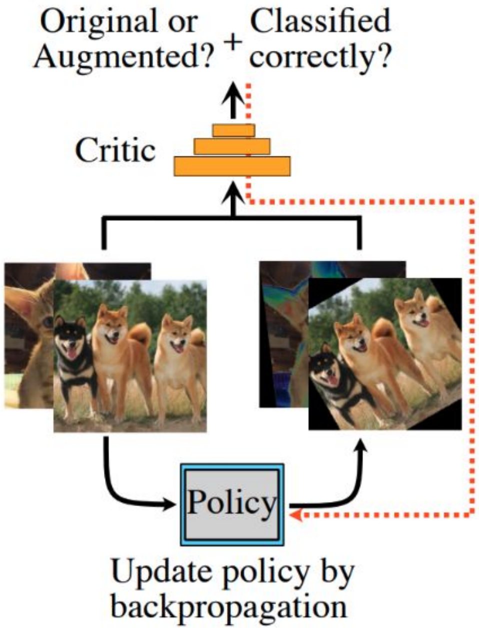


Figure 1. Overview of our proposed model. We propose to use a **differentiable data augmentation pipeline** to achieve faster policy search by using adversarial learning.

Strong Data Augmentation

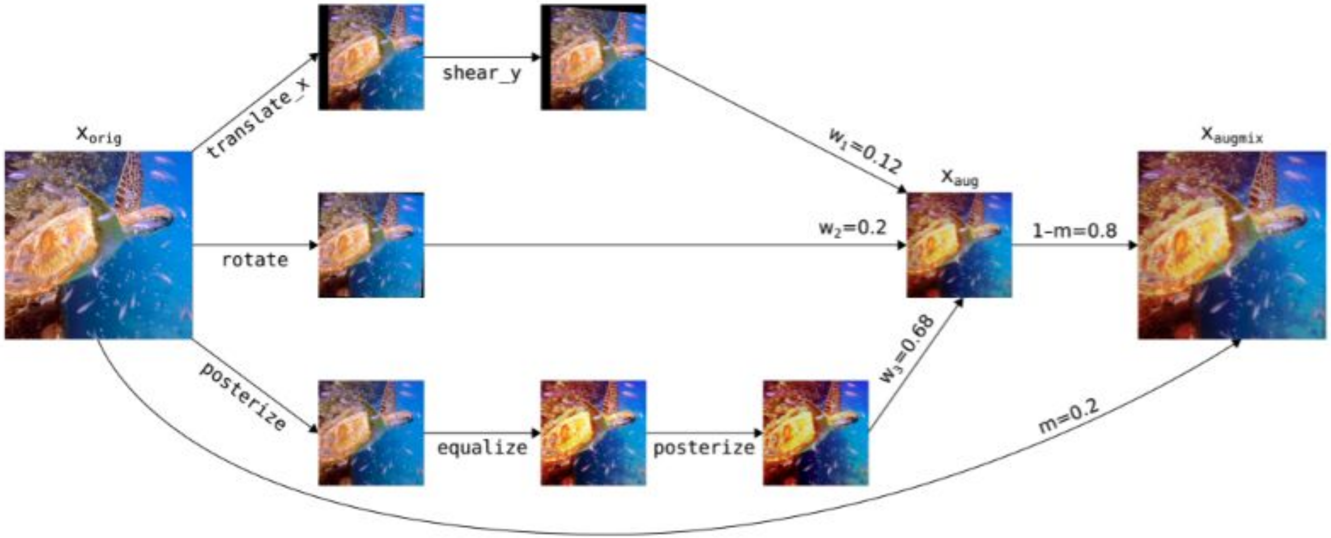
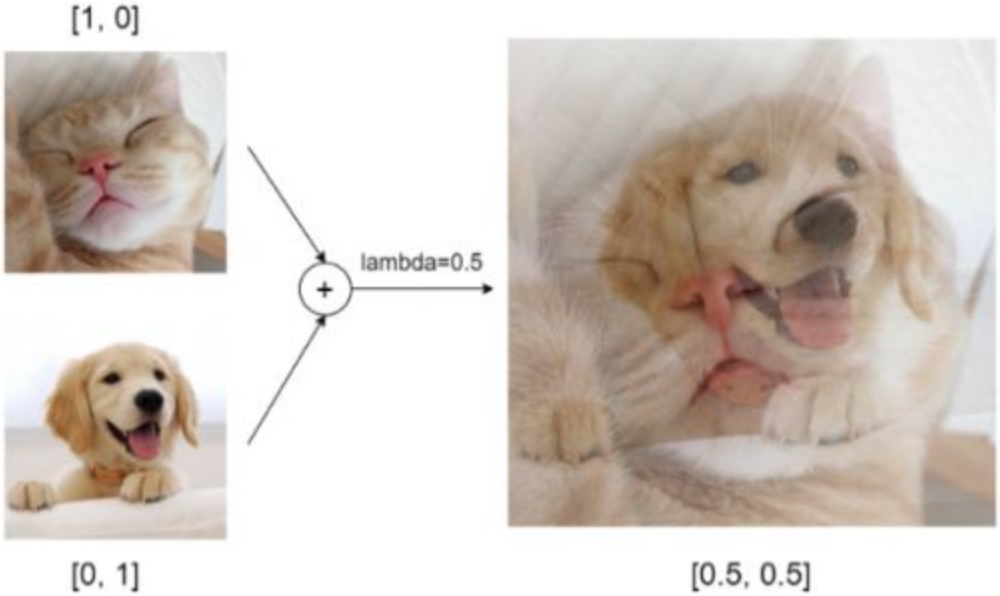


Figure 4: A realization of AUGMIX. Augmentation operations such as translate_x and weights such as m are randomly sampled. Randomly sampled operations and their compositions allow us to explore the semantically equivalent input space around an image. Mixing these images together produces a new image without veering too far from the original.

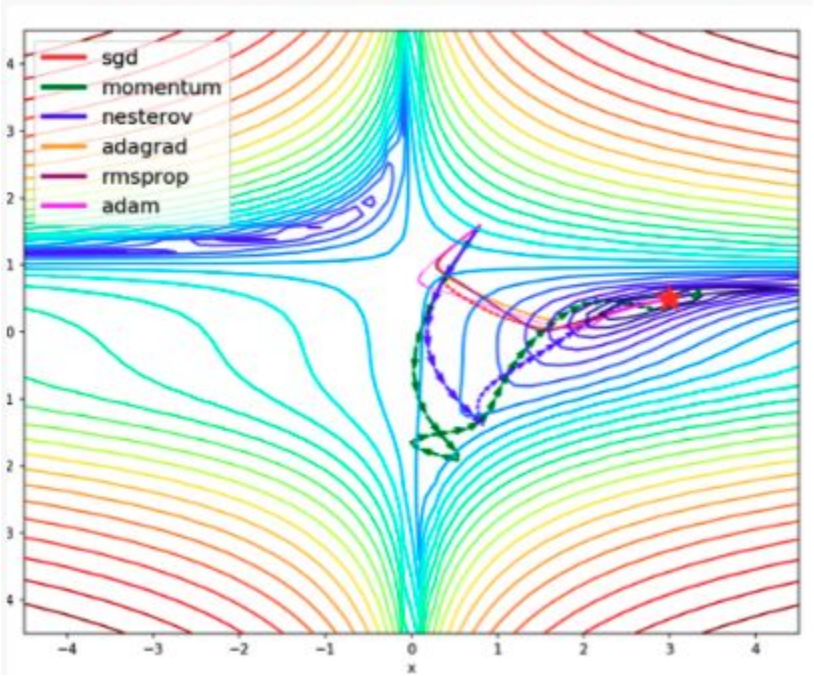
Mixed Data Augmentation

Mixup in Action

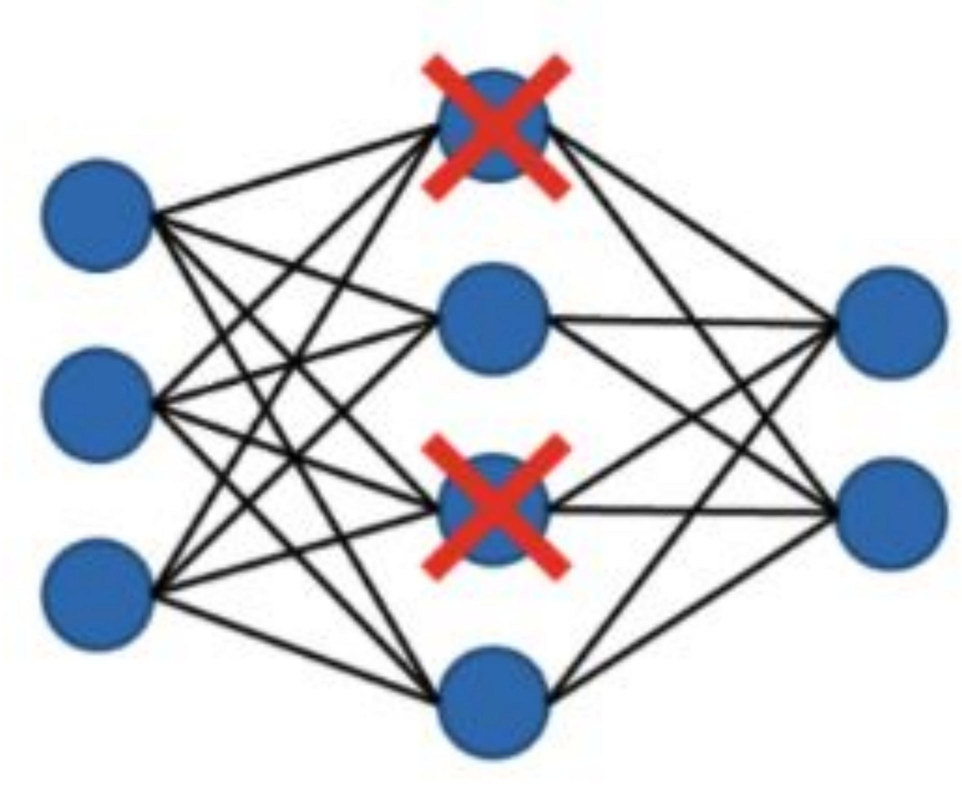


Mix-up works by blending 2 images with alpha % from image_1 and (1-alpha) % from image_2

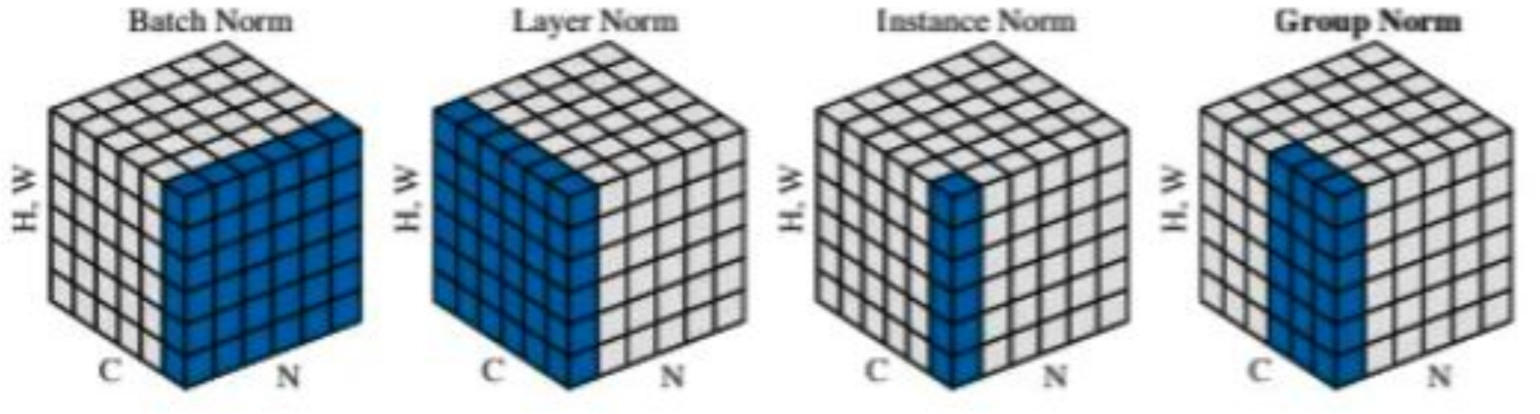
Optimizer



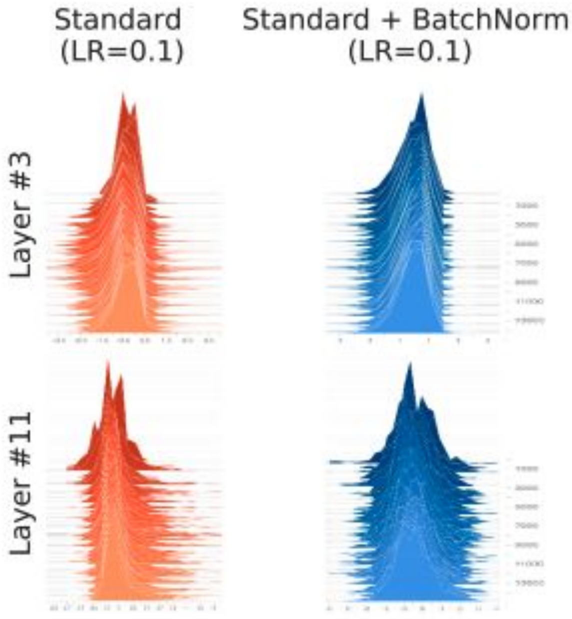
Dropout



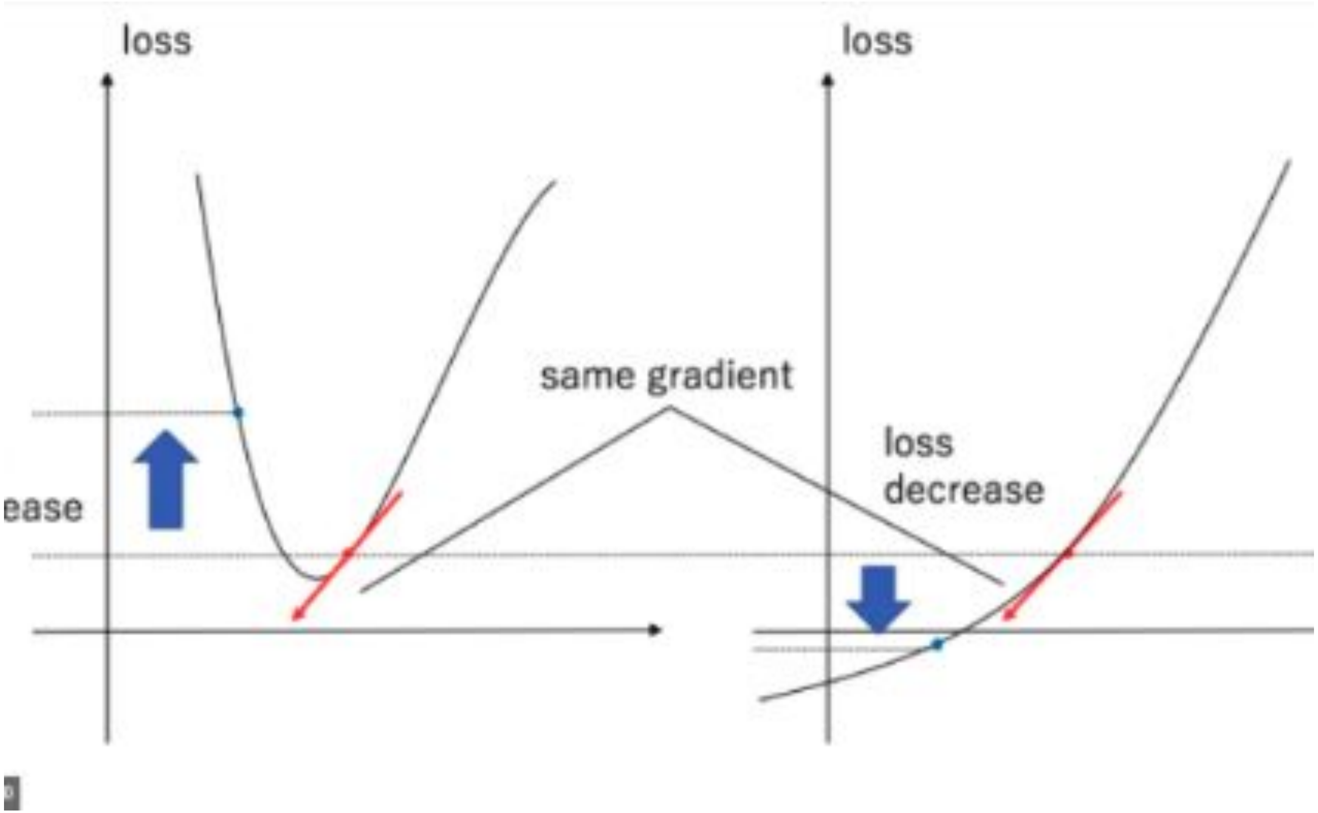
Batch normalization



internal covariate shift ?



Training loss smoother



VGG and Imagenet

VGG16

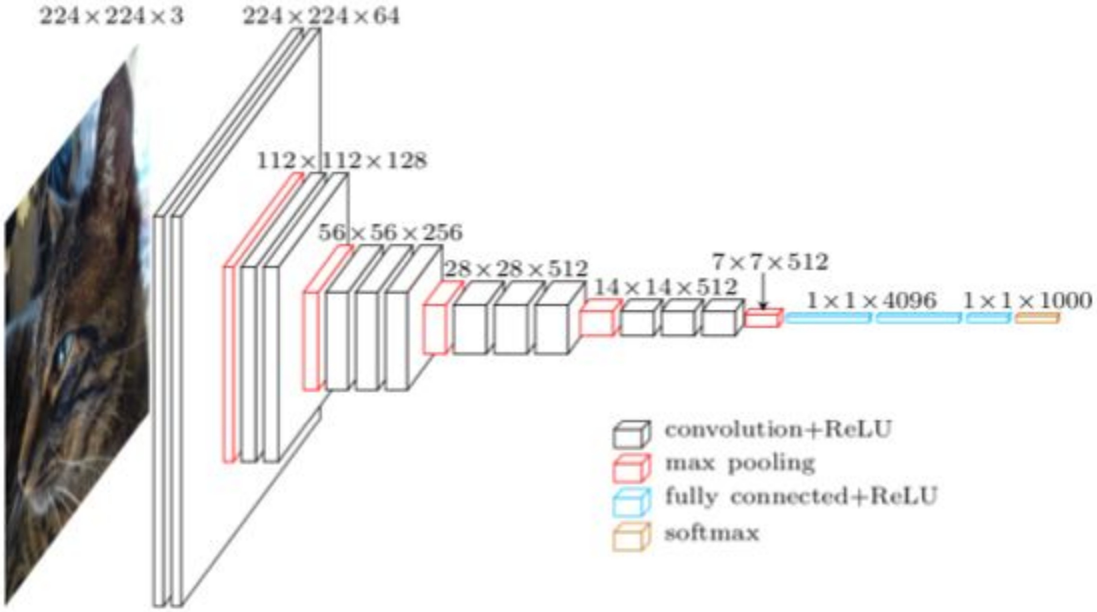


FIGURE 1 – Réseau VGG16

Summary of VGG16 Architecture

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

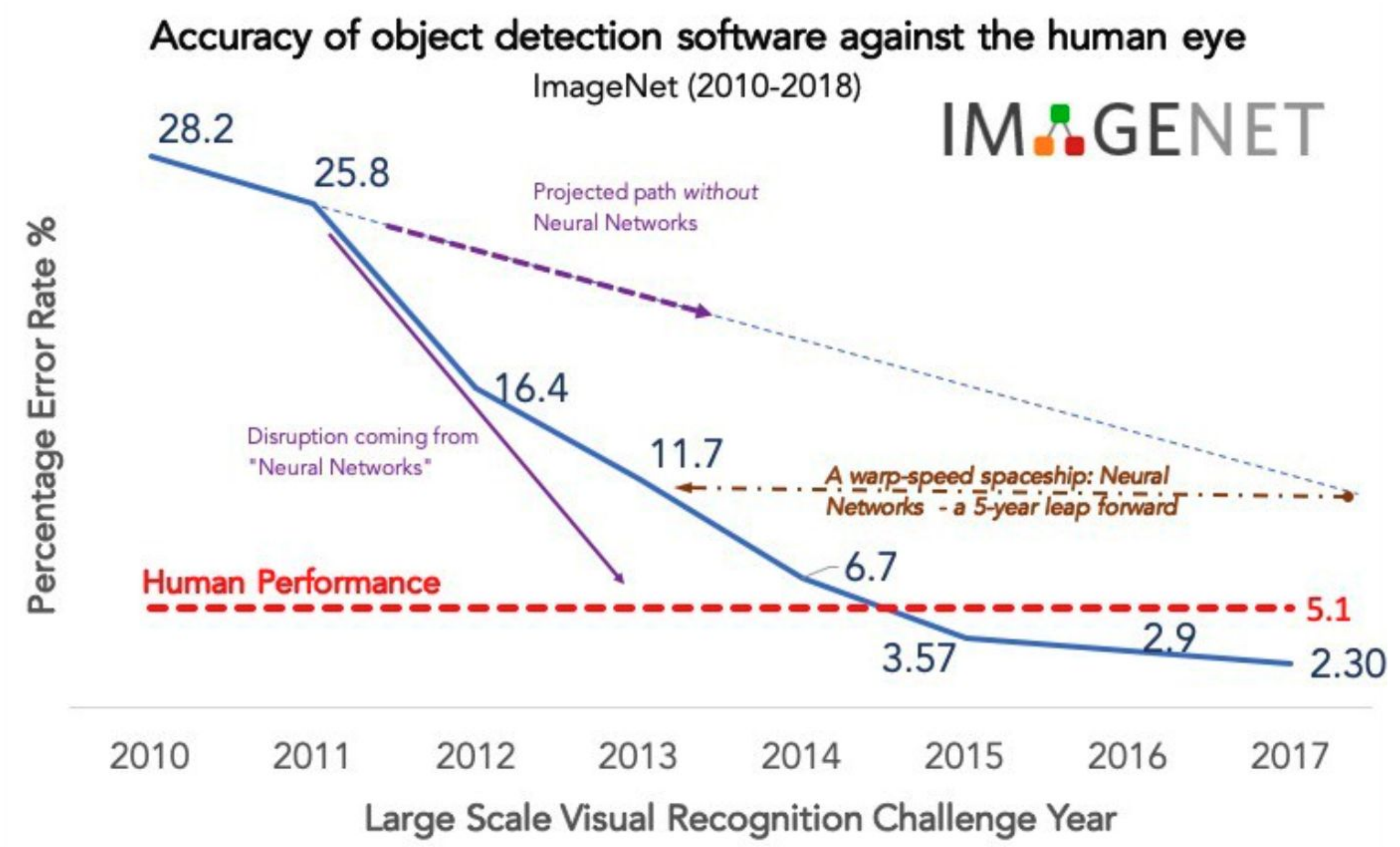
Imagenet

ImageNet Challenge

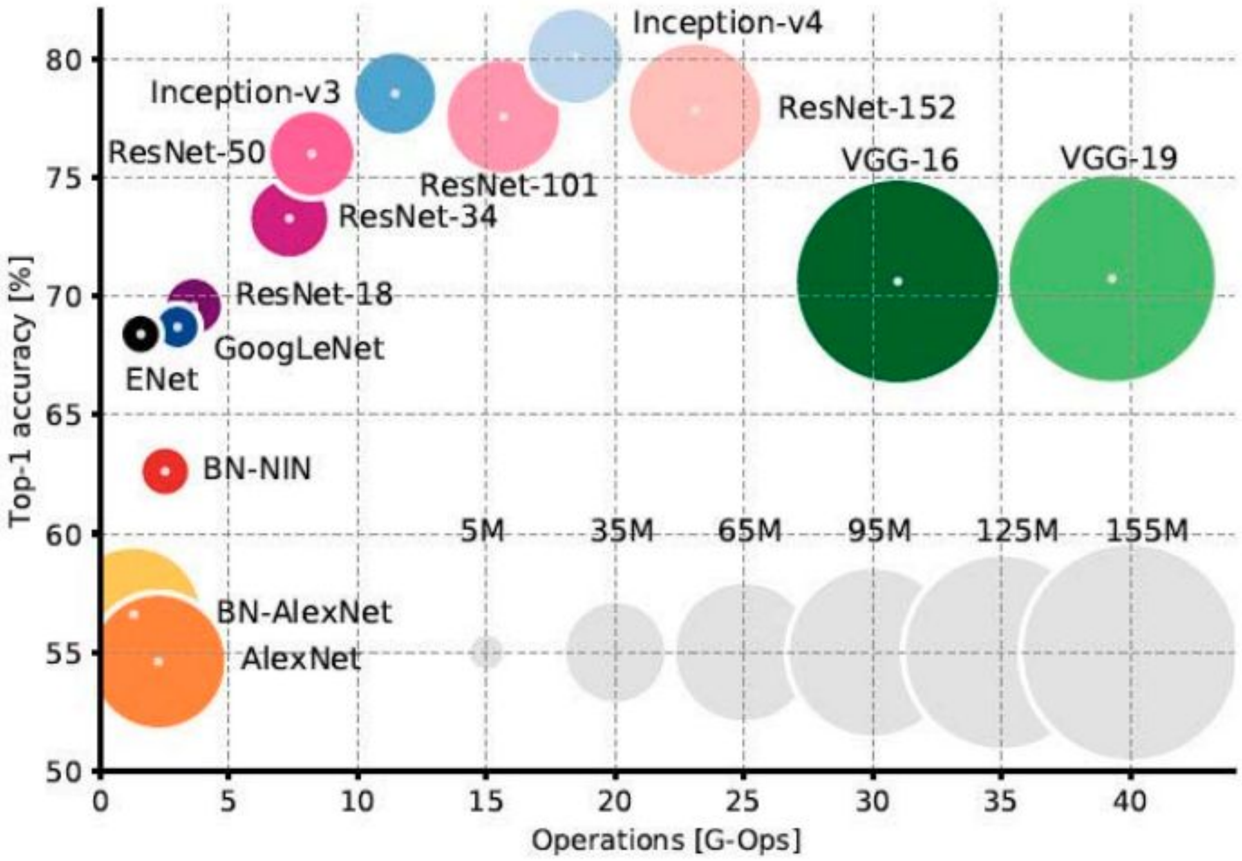
IMAGENET

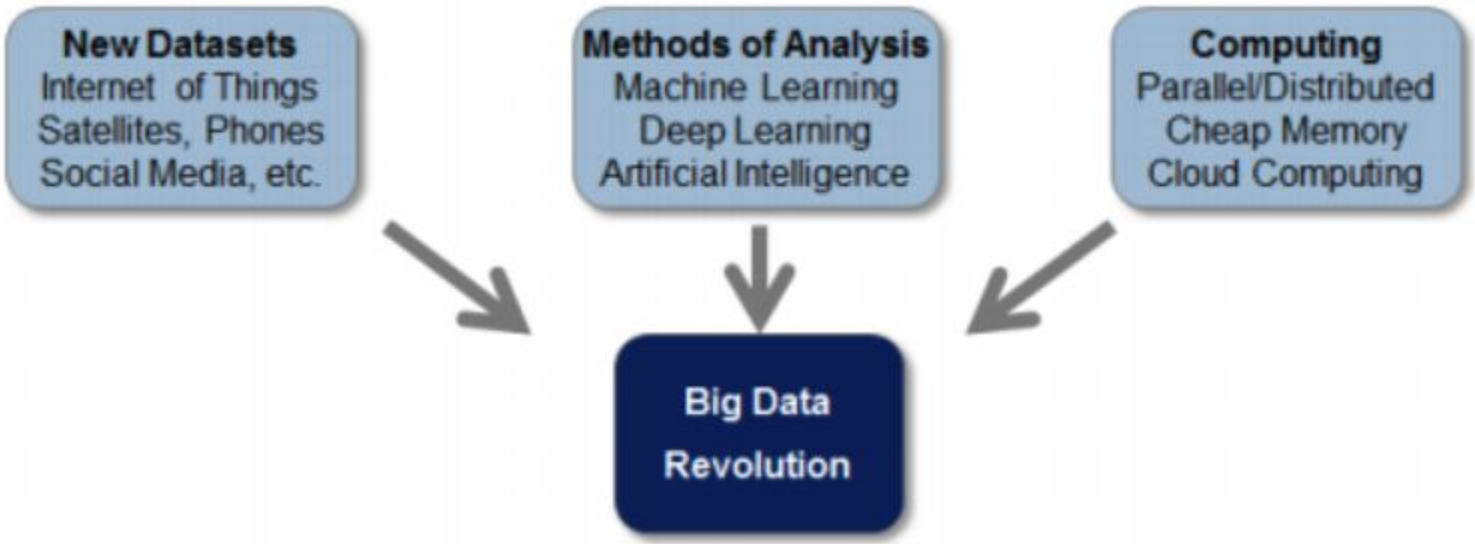
- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

The popular ImageNet Challenge based on the ImageNet Database


Accuracy of object detection software against the human eye
ImageNet (2010-2018)
IMAGENET
28.2
25.8
Projected path without
Neural Networks
16.4
Disruption coming from
"Neural Networks"
11.7
A warp-speed spaceship: Neural
Networks - a 5-year leap forward
6.7
Human Performance
5.1
3.57
2.9
2.30
Percentage Error Rate %
2010
2011
2012
2013
2014
2015
2016
2017
Large Scale Visual Recognition Challenge Year

Network Complexity by Operations and Weights Sizes

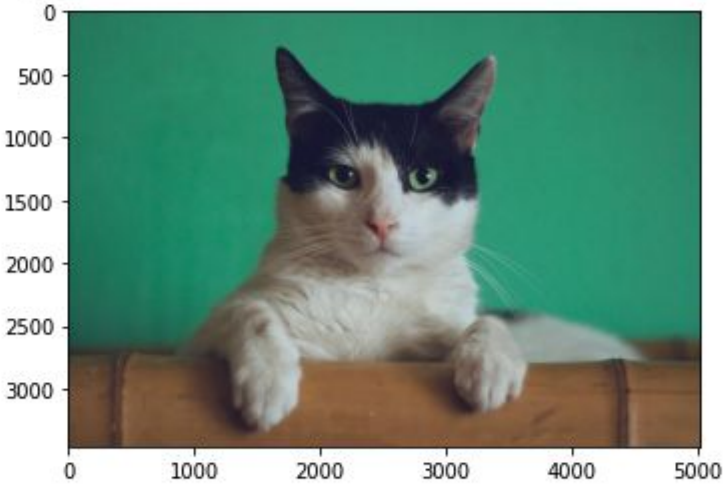



New Datasets
Internet of Things
Satellites, Phones
Social Media, etc.
Methods of Analysis
Machine Learning
Deep Learning
Artificial Intelligence
Computing
Parallel/Distributed
Cheap Memory
Cloud Computing
Big Data
Revolution

```
vgg16 = torchvision.models.vgg16(pretrained=True)
vgg16.eval()
```

```
pred("cat.jpg")
```

'Egyptian cat'



Imagenet

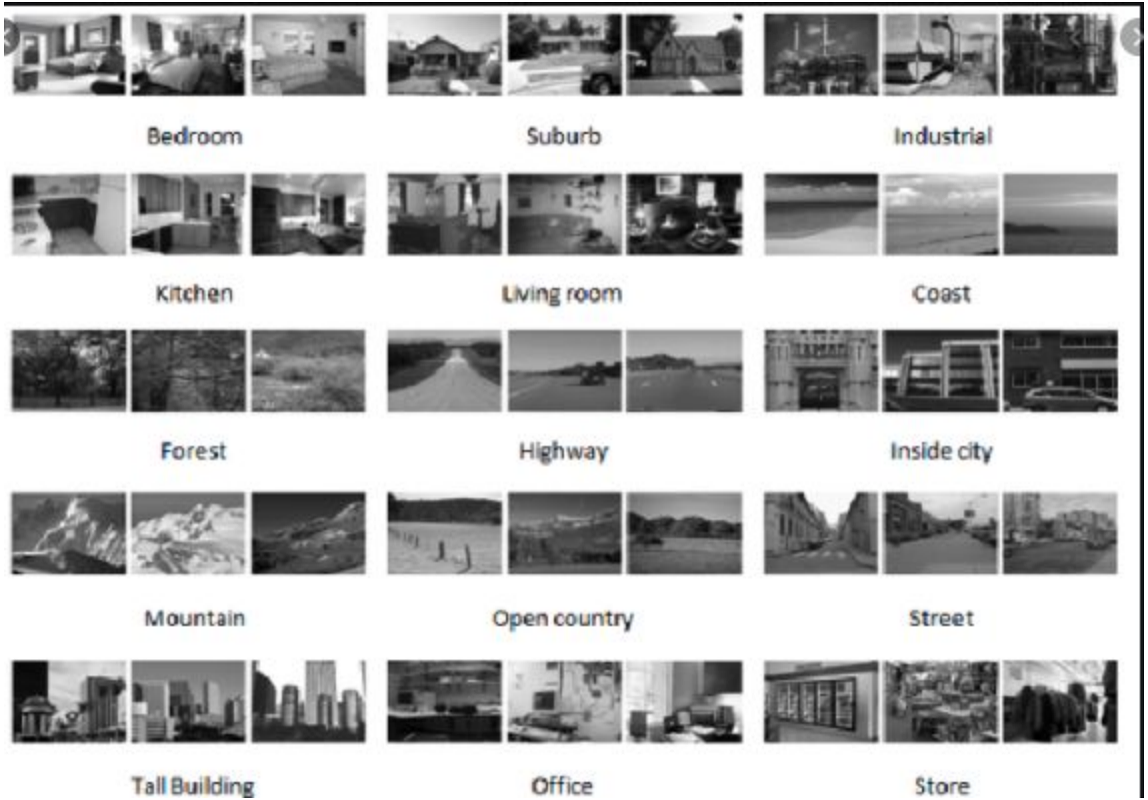
ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

The popular ImageNet Challenge based on the ImageNet Database

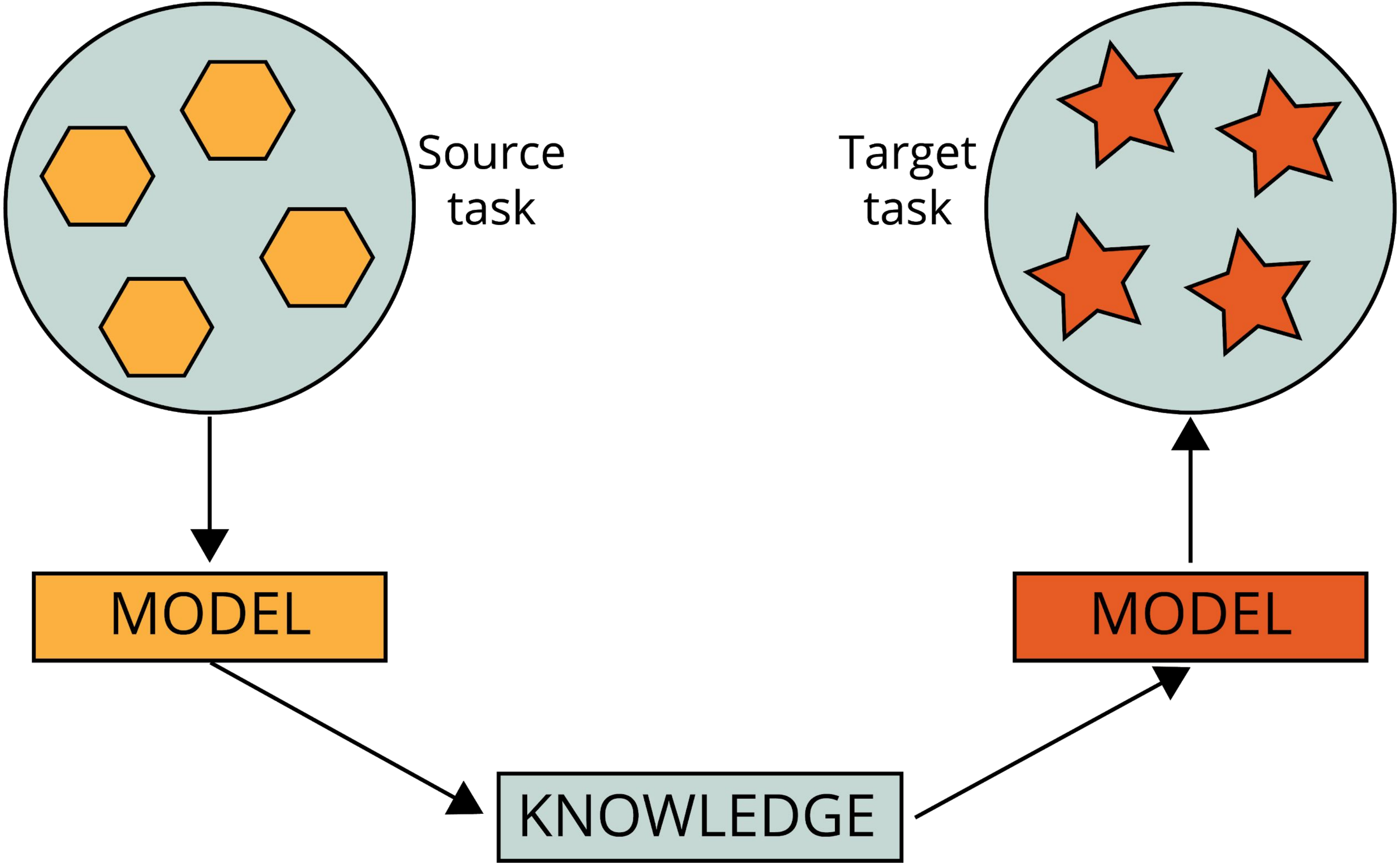
15 scene dataset

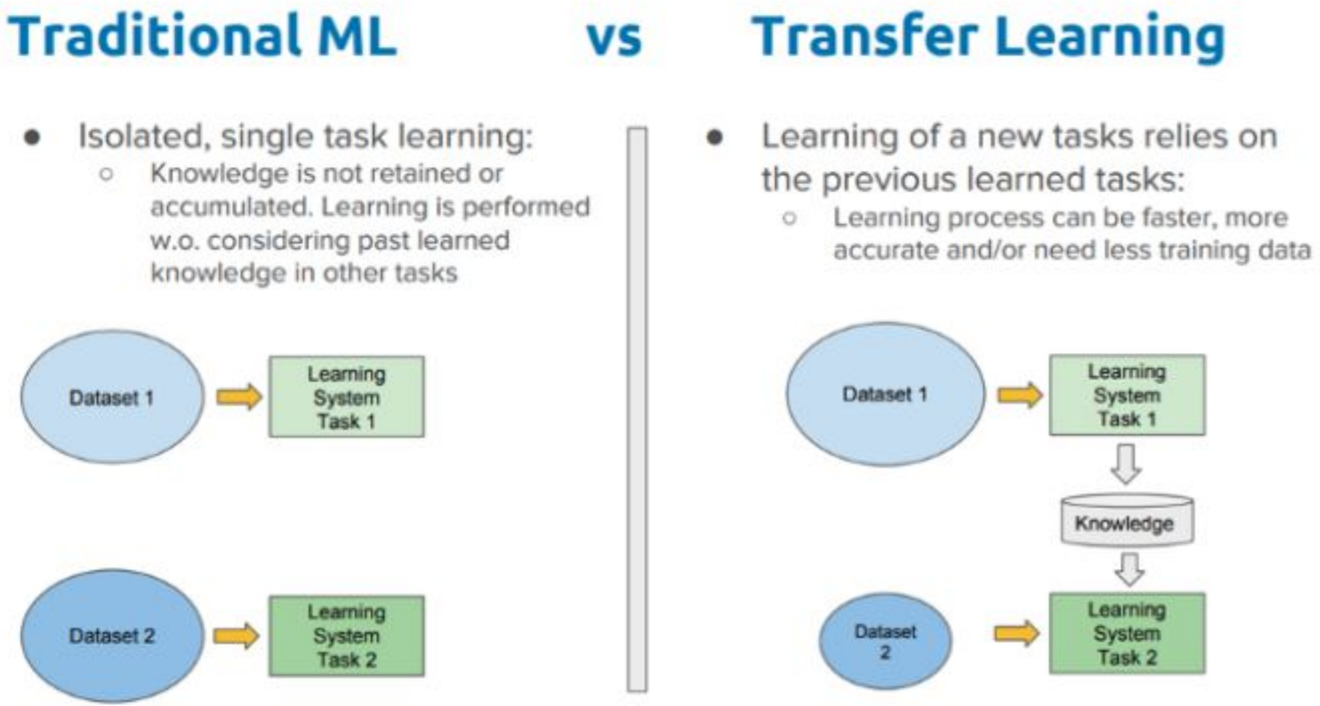


Transfer Learning

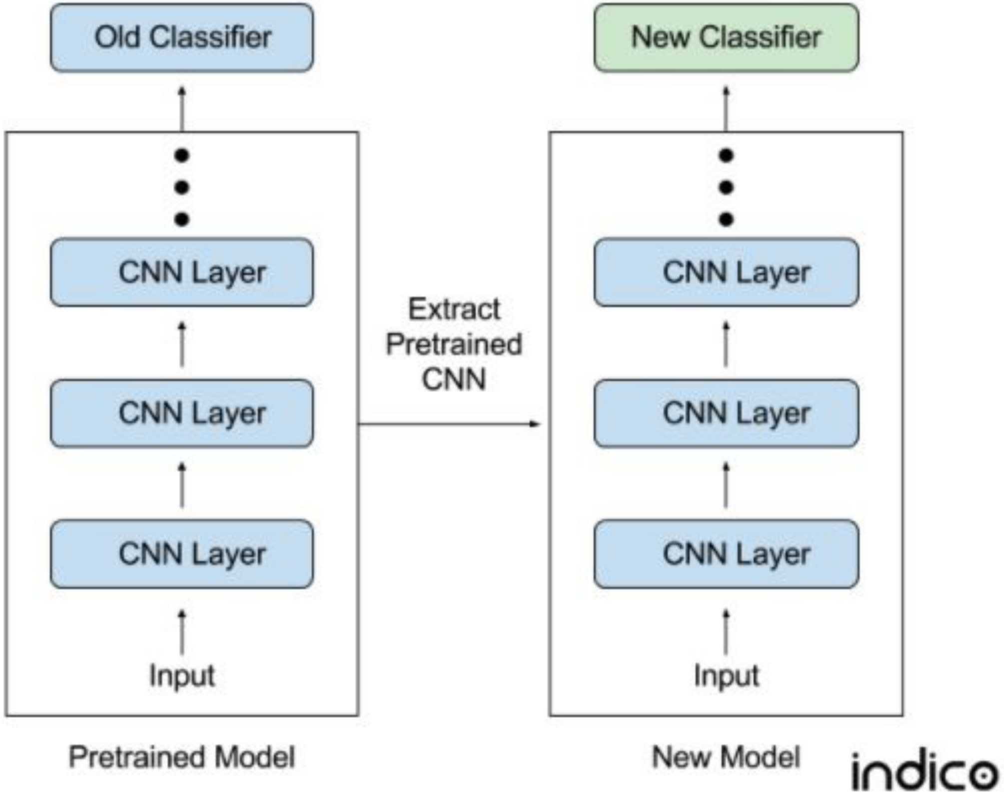
> *Transfer learning and domain adaptation refer to the situation where what has been learned in one setting … is exploited to improve generalization in another setting*

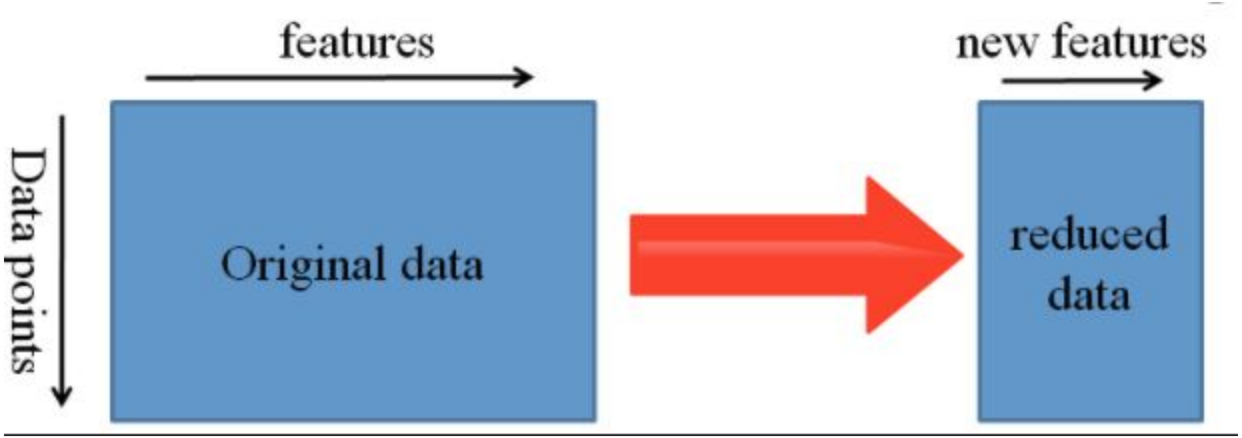
— Page 526, Deep Learning, 2016.


Source
task
Target
task
MODEL
MODEL
KNOWLEDGE



Traditional Learning vs Transfer Learning


Old Classifier
CNN Layer
CNN Layer
CNN Layer
Input
Pretrained Model
Extract
Pretrained
CNN
New Classifier
CNN Layer
CNN Layer
CNN Layer
Input
New Model
indico


features
new features
Data points
Original data
reduced
data

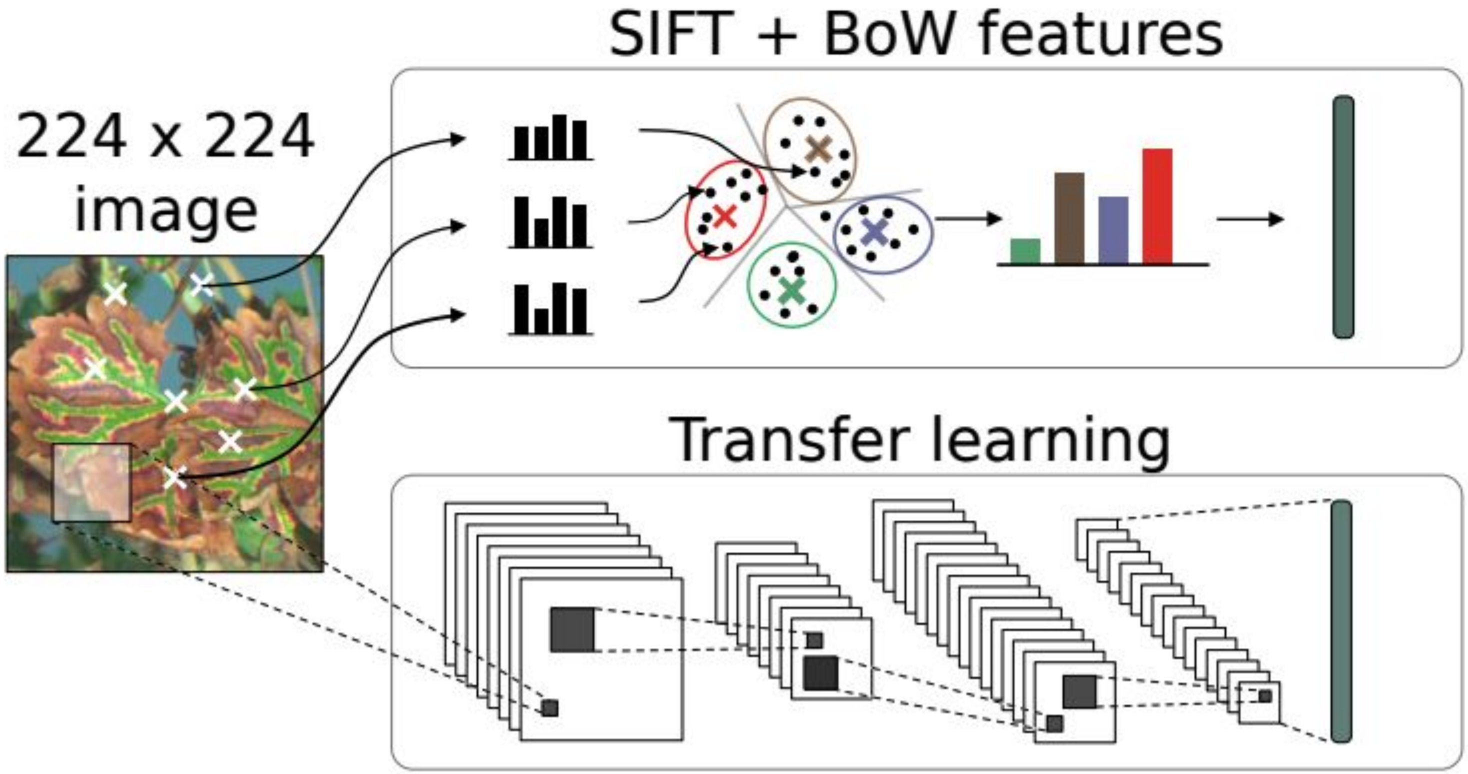
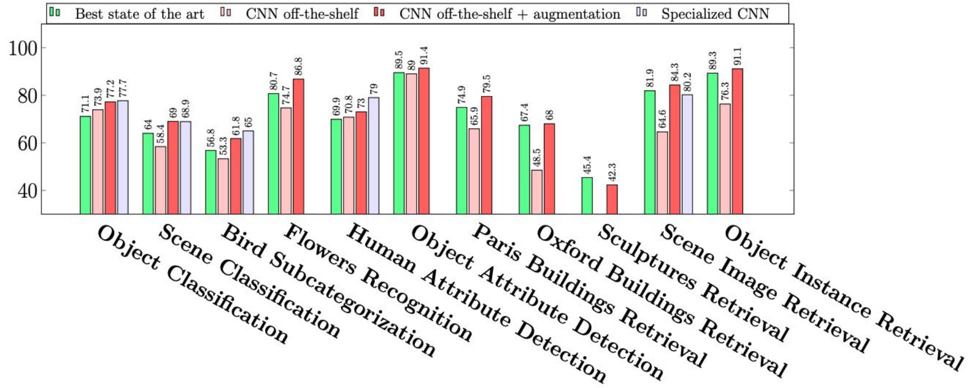


Figure 7. Comparative description of SIFT encoding and CNN transfer learning for the construction of informative features.

CNN Features off-the-shelf: an Astounding Baseline for Recognition

Ali Sharif Razavian Hossein Azizpour Josephine Sullivan Stefan Carlsson
CVAP, KTH (Royal Institute of Technology)
Stockholm, Sweden
{razavian,azizpour,sullivan,stefanc}@csc.kth.se

sculptures dataset. The results are achieved using a linear SVM classifier (or L2 distance in case of retrieval) applied to a feature representation of size 4096 extracted from a layer in the net. The representations are further modified using simple augmentation techniques e.g. jittering. The



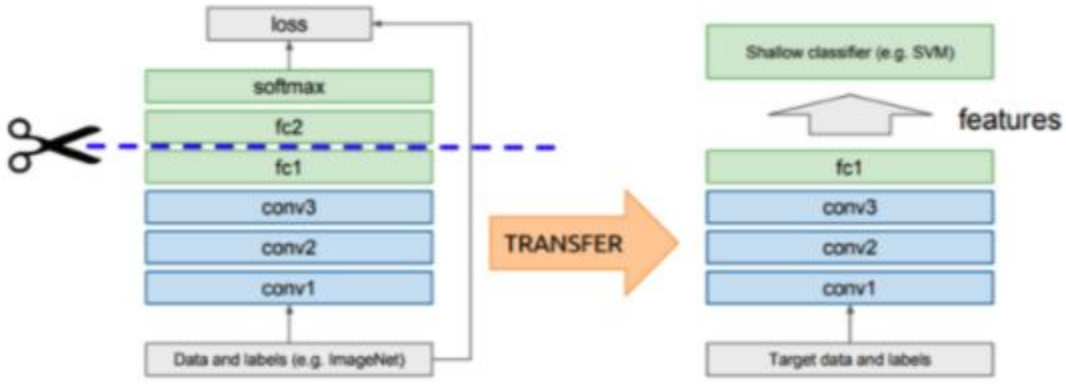
Faster training process: As discussed in the previous point, the training time for transfer learning is less, compared to training from scratch: transfer learning reduces the number of trainable parameters.

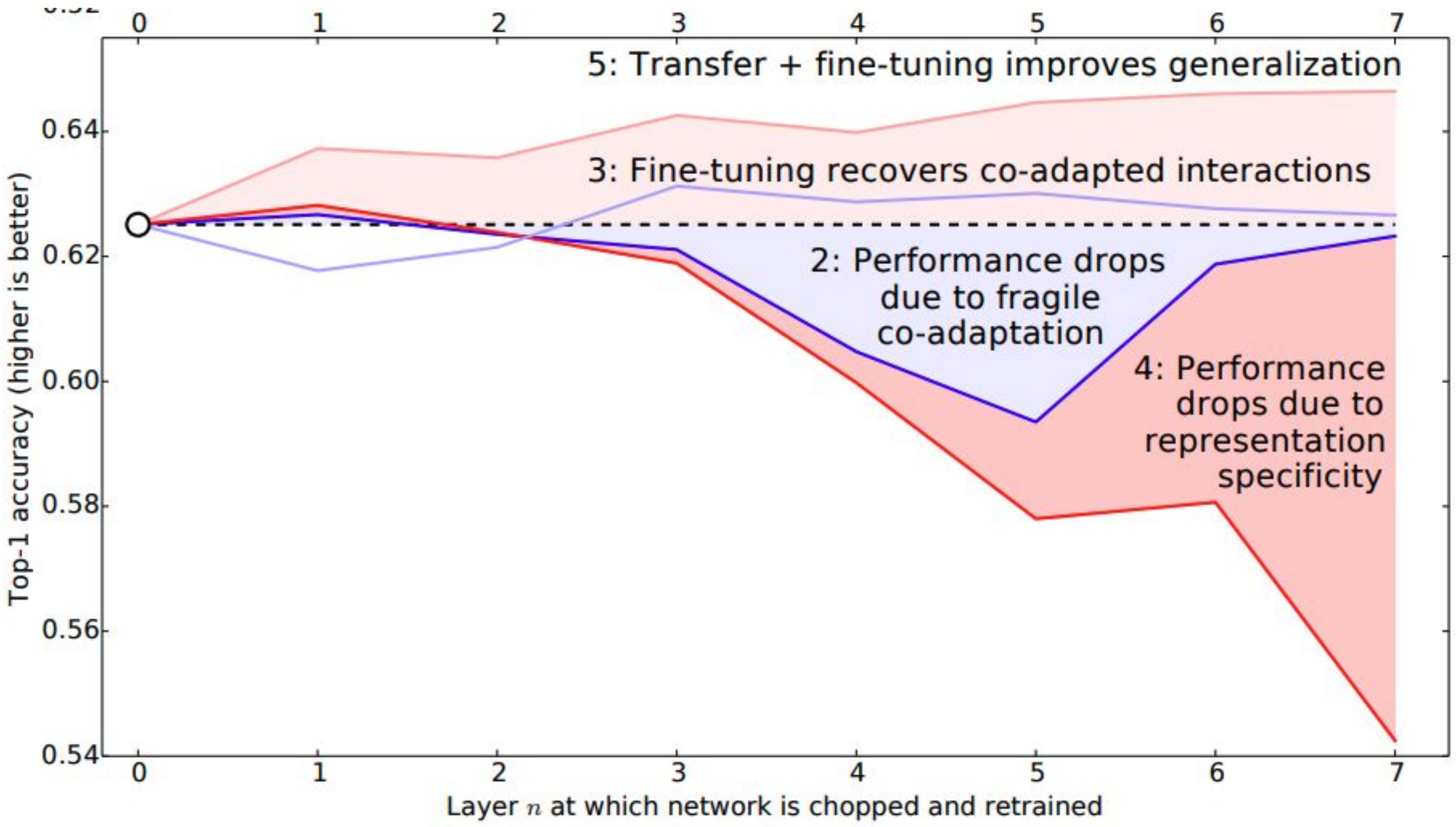
Model generalizes better: Transfer learning allows the model to perform better on new, unseen data, because state-of-the-art pretrained models are trained to learn generic features, thus allowing them to avoid overfitting.

Which layers ?

> *[Convolutional Neural Networks] features are more generic in early layers and more original-dataset-specific in later layers*

— Transfer Learning, CS231n Convolutional Neural Networks for Visual Recognition



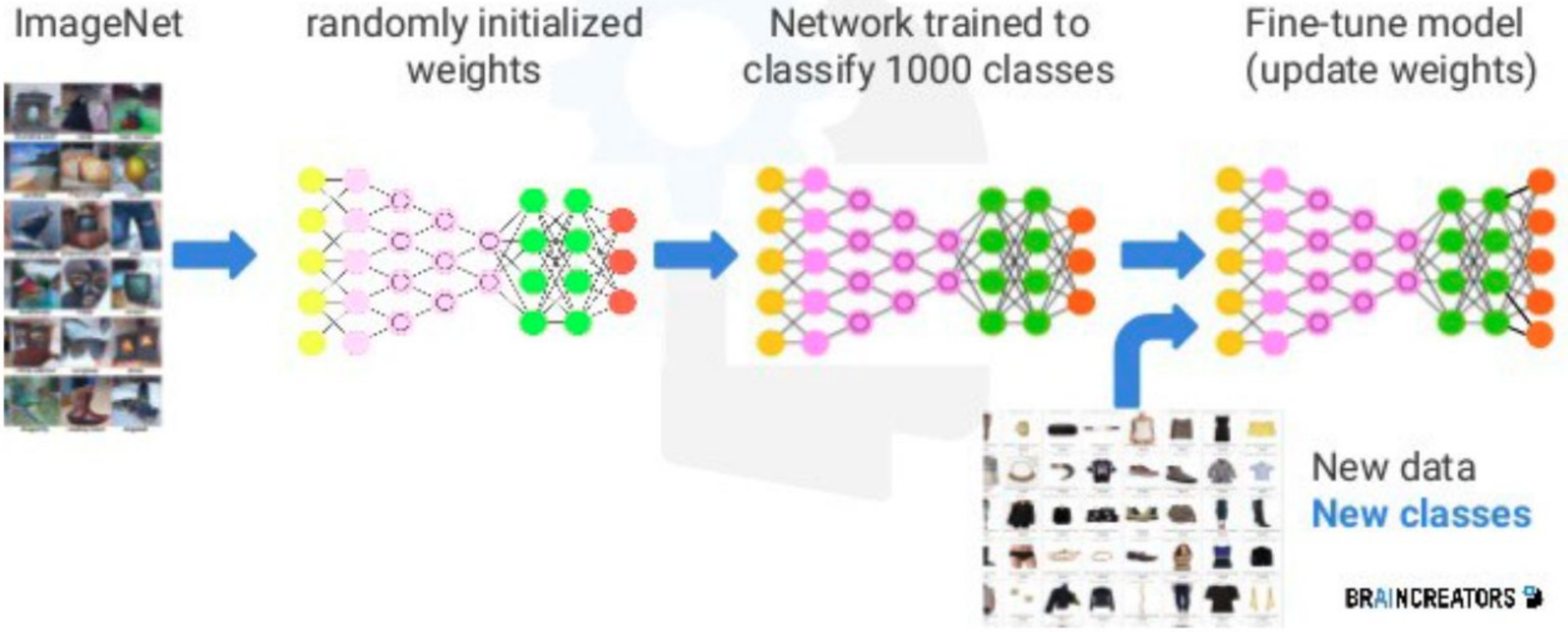


5: Transfer + fine-tuning improves generalization
3: Fine-tuning recovers co-adapted interactions
2: Performance drops due to fragile co-adaptation
4: Performance drops due to representation specificity
Top-1 accuracy (higher is better)
Layer n at which network is chopped and retrained
0.54
0.56
0.58
0.60
0.62
0.64
0
1
2
3
4
5
6
7

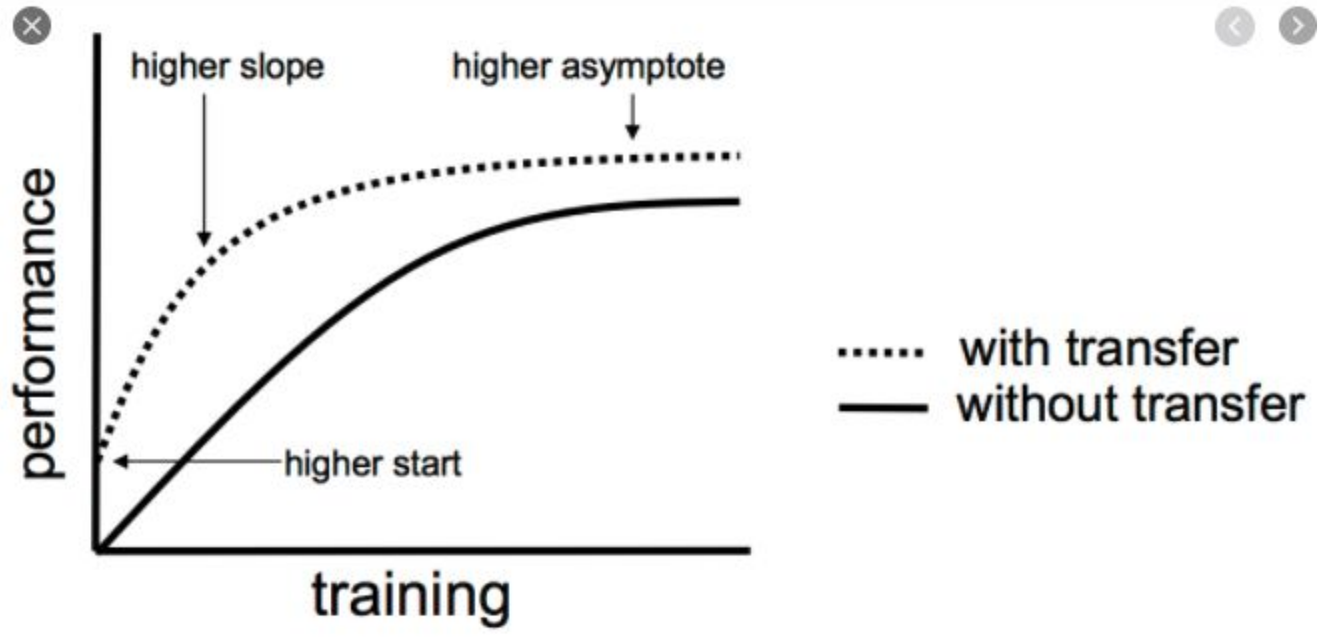
- co-adaptation: Gradient descent was able to find a good solution the first time, but this was only possible because the layers were jointly trained.

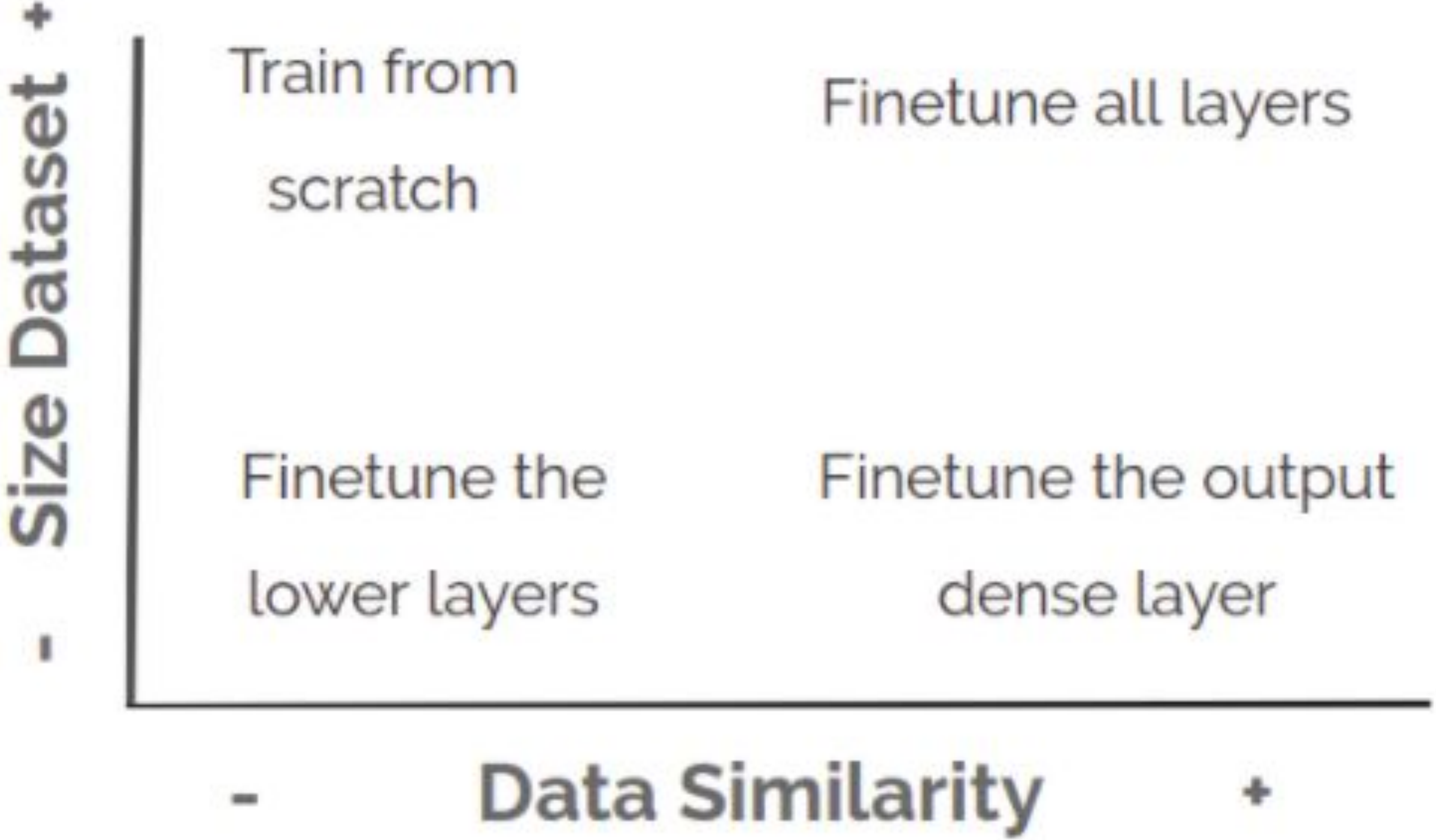
- specificity: the specialization of higher layer neurons to their original task at the expense of performance on the target task, which was expected

Transfer Learning



Performances




Size Dataset
+
-
Train from scratch
Finetune all layers
Finetune the lower layers
Finetune the output dense layer
-
Data Similarity
+

Tips

- extraires features différentes résolutions

- petit learning rate quand fine tune

- attention à la batch normalization ! quelles statistiques utilisées ?