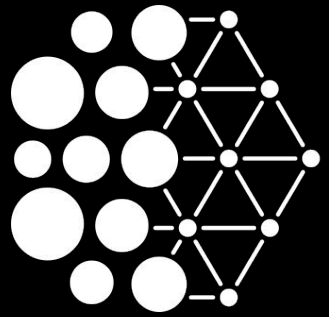
Quebec
Artificial
Intelligence
Institute



Mila

Convolutional Neural Network Architectures

Jeremy Pinto
Applied Research Scientist, Mila
jeremy.pinto@mila.quebec

Contents

- Architecture evaluation and benchmarking
- Introduction to ImageNet / ILSVRC
- Review of key CNN architectures
- Transfer learning
- Data augmentation
- Structured outputs

Object Detection:
“Bicycle” + Bounding Box

Architectures

So far, we have seen the most common **building blocks** that compose CNNs. There are many ways in which they can be **combined** and stacked. We will look at the more **common** architectures used for images.

https://commons.wikimedia.org/wiki/File:Lego_blocks.jpg

Benchmarking

How do we know which model is “**better**”?

Is a “**better**” model:

- More accurate?
- Lightweight?
- Fast?
- Generic?
- Adaptable?
- Easily Trainable?

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters
Xception	88 MB	0.790	0.945	22,910,480
VGG16	528 MB	0.713	0.901	138,357,544
VGG19	549 MB	0.713	0.900	143,667,240
ResNet50	98 MB	0.749	0.921	25,636,712
ResNet101	171 MB	0.764	0.928	44,707,176
ResNet152	232 MB	0.766	0.931	60,419,944
ResNet50V2	98 MB	0.760	0.930	25,613,800
ResNet101V2	171 MB	0.772	0.938	44,675,560
ResNet152V2	232 MB	0.780	0.942	60,380,648
ResNeXt50	96 MB	0.777	0.938	25,097,128
ResNeXt101	170 MB	0.787	0.943	44,315,560
InceptionV3	92 MB	0.779	0.937	23,851,784
InceptionResNetV2	215 MB	0.803	0.953	55,873,736
MobileNet	16 MB	0.704	0.895	4,253,864
MobileNetV2	14 MB	0.713	0.901	3,538,984

Comparison of different model accuracies and sizes

https://keras.io/applications/

Typically, we seek to find the best trade-offs between all these categories.

Comparing Models

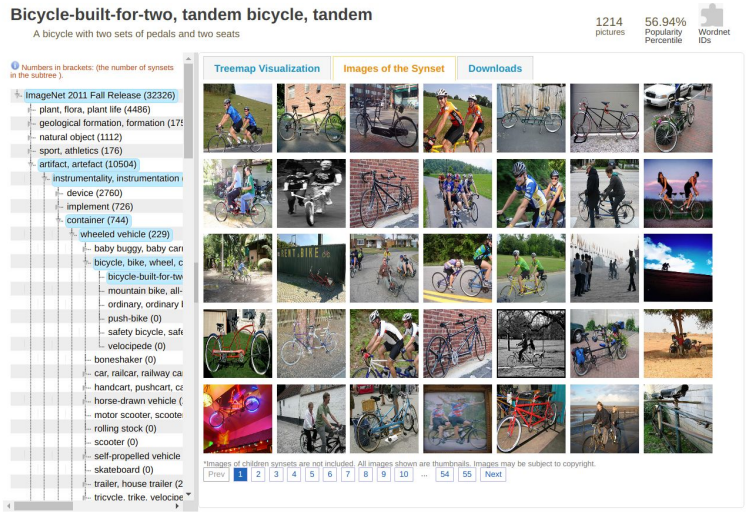
In the deep learning community, claims of **better** models can (somewhat) easily be verified. There are publicly available **datasets** (and competitions) to train networks on and compare metrics. The model with the highest points wins (and usually earns a publication).

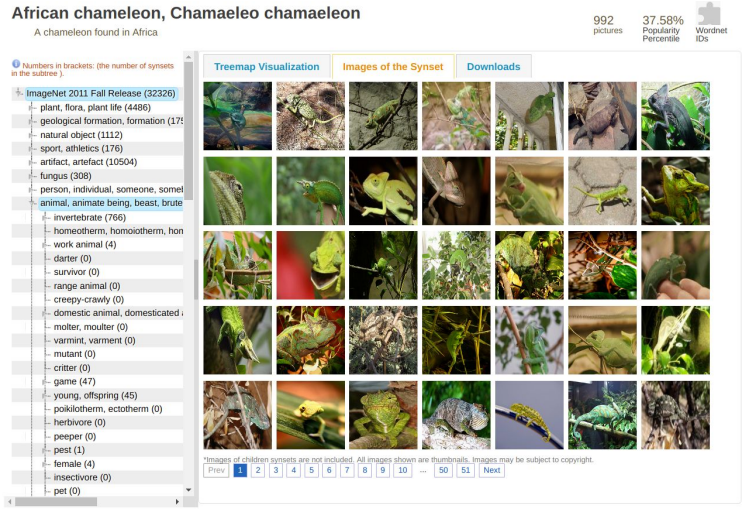
	A Metric	Another Metric
Our Model	**41.57**	**32.45**
Their Model	37.45	-
Their Other Model	38.74	29.45

Table 1 - Our model outperforms all other models and is definitely the best model.

ImageNet

ImageNet is one of the most widely used datasets in image classification. It has paved the way for many important discoveries in CNN architectures. It consists of **millions of images** manually labelled with thousands of categories.



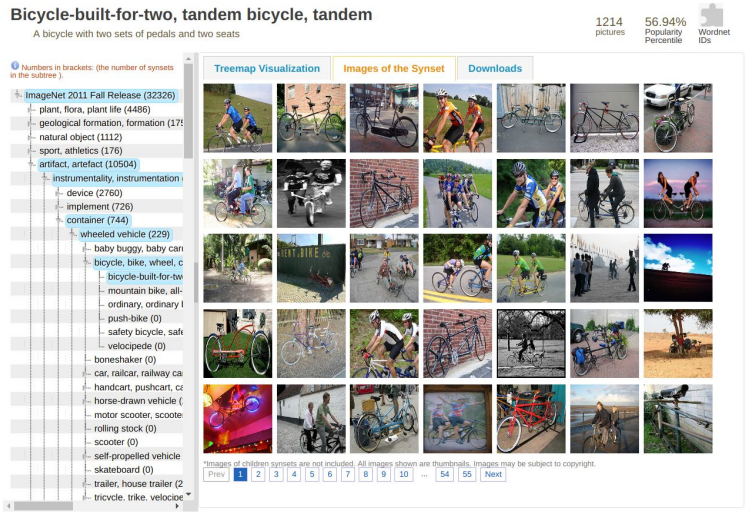


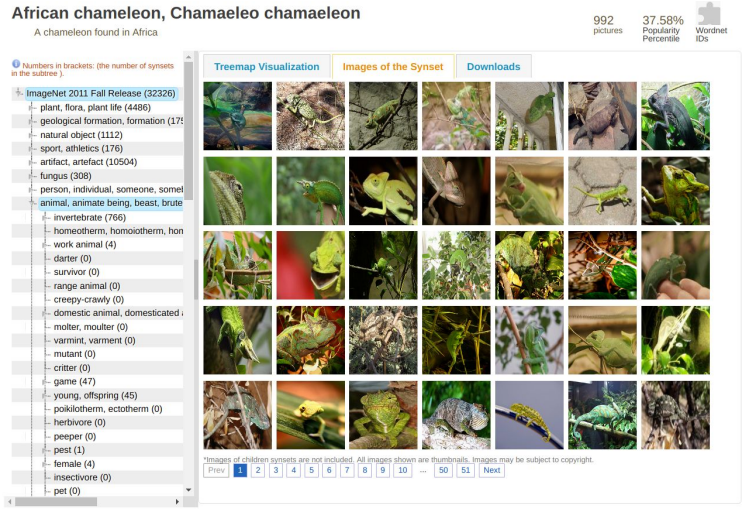
http://imagenet.stanford.edu/

ImageNet history

ILSVRC

ImageNet Large Scale Visual Recognition Challenge (**ILSVRC**) is a competition which evaluates classification models on a subset of ImageNet (1000 categories). It is an important benchmark in computer vision.





http://imagenet.stanford.edu/

ImageNet

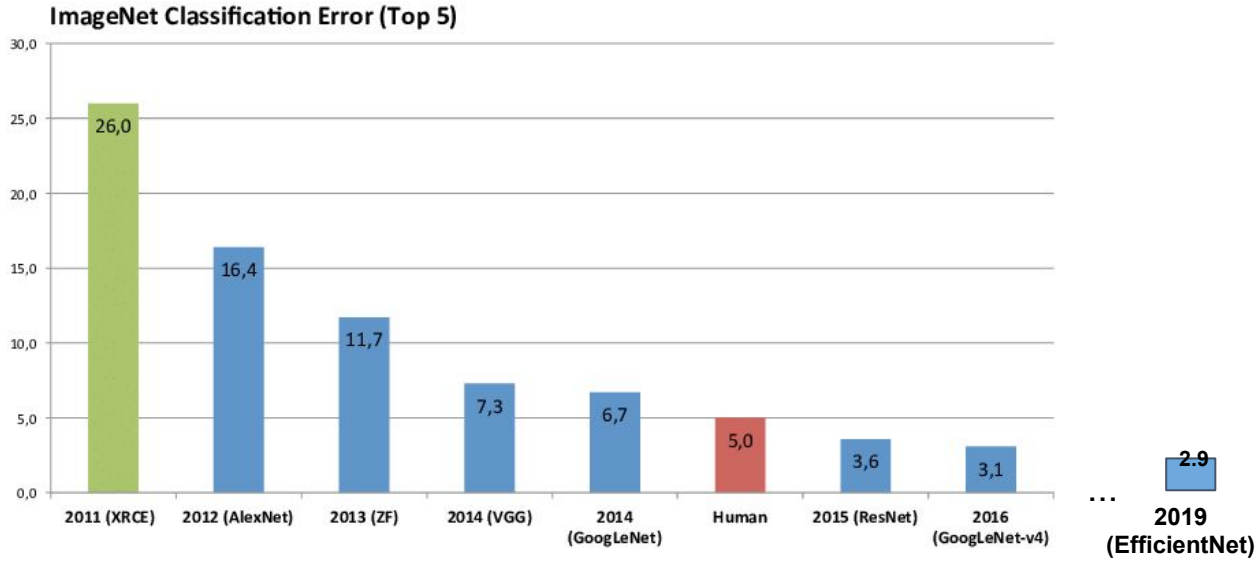
"Don't get fooled by people who claim to have a solution to Artificial General Intelligence [...] Ask them what error rate they get on **MNIST** or **ImageNet**."
- Yann Lecun

https://research.fb.com/people/lecun-yann/
https://www.reddit.com/r/MachineLearning/comments/25lnbt/ama_yann_lecun

ImageNet

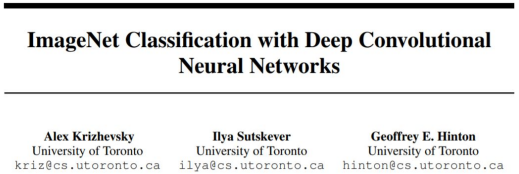
ImageNet and ILSVRC **paved** the way for computer vision research and deep learning in particular. It allowed researchers to compare various algorithms in a fair and principled way.

ILSVRC: Annual competition with 1000 categories and 1.2 million images



ImageNet

AlexNet was a **turning point** for deep learning research. Up until that point, the main bottleneck for neural networks was lack of labelled data and compute power. They were among the first to use **GPUs** for convolutions.

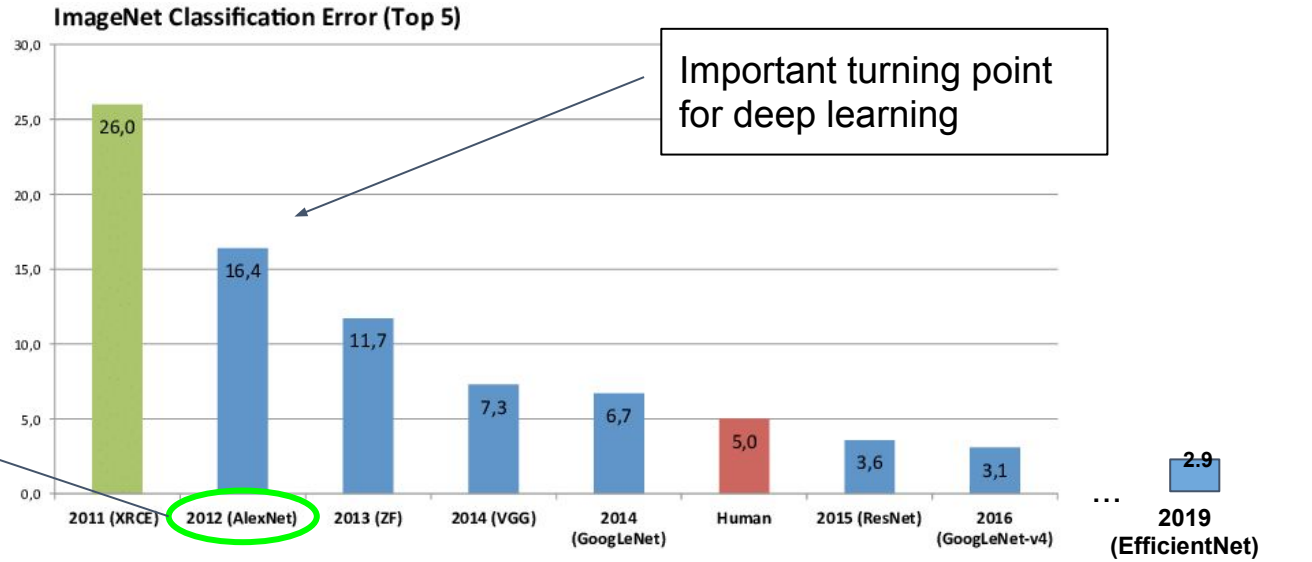


ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

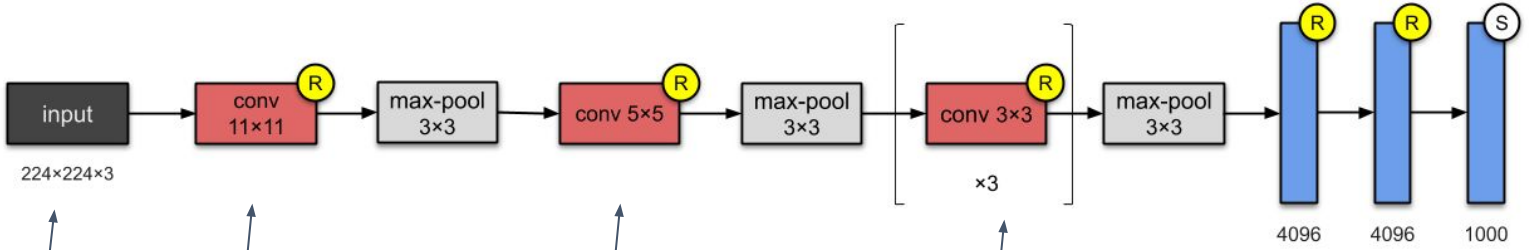
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



AlexNet - Architecture

AlexNet remains close in style to LeNet. It consists of 5 convolutional layers, each followed by a ReLU (R). Not all were followed by max-pooling. 2 fully connected (FC) layers followed by a softmax (S) layer generate a probability distribution. AlexNet also uses **dropout** in the FC layers.



Cropped RGB image

D = 96
F = 11
S = 4

D = 96
F = 5

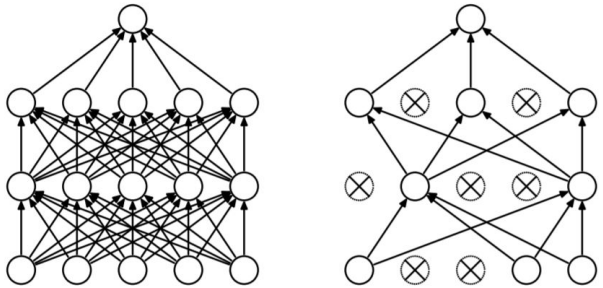
D = [384, 384, 256]
F = 3

	LeNet	AlexNet
Parameters	~30 x 10^3	~60 x 10^6
Layers	5	8

https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

Dropout

AlexNet also used **dropout** in its architecture. This reduces the risk of overfitting and acts as a form of **regularization**. It can be thought of as training many models simultaneously, i.e. an **ensemble** of networks.



(a) Standard Neural Net (b) After applying dropout.

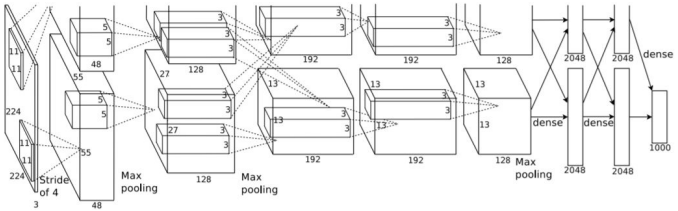
Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

AlexNet

ImageNet Classification with Deep Convolutional ... - NIPS Proceedings
https://papers.nips.cc/.../4824-imagenet-classification-with-deep-convolutional-neural-...
by A Krizhevsky - 2012 - Cited by 43769 - Related articles
Electronic Proceedings of Neural Information Processing Systems.

"All of our experiments suggest that our results can be improved simply by waiting for **faster** GPUs and **bigger datasets** to become available."
- Krizhezvsky et al.

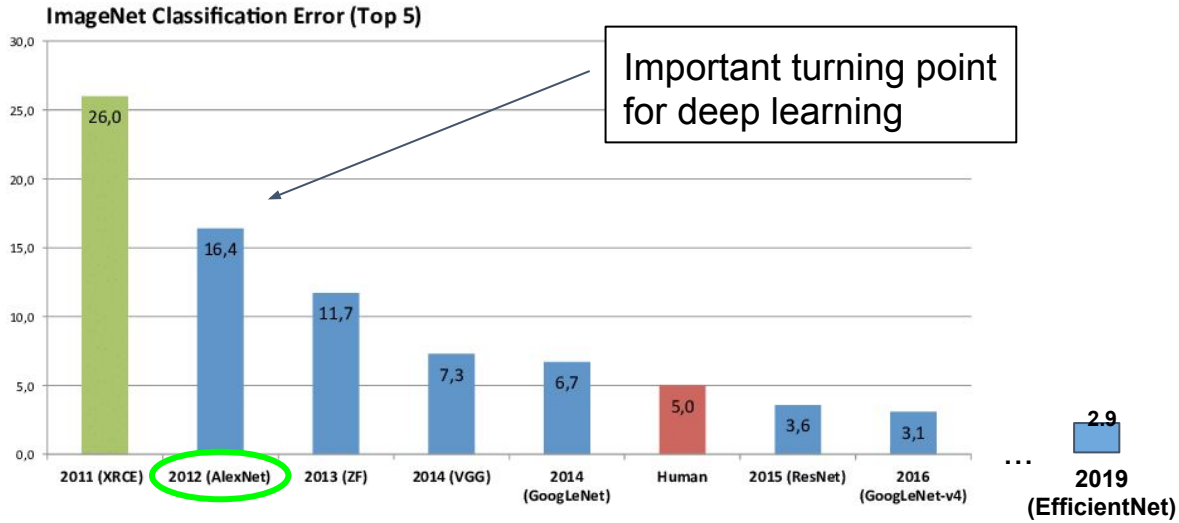


https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

ImageNet

Following the success of AlexNet, the next logical step was to attempt to build **deeper** networks. This was challenging both from an engineering perspective (hardware limitations, long to train) as well as from an algorithmic perspective (overfitting, underfitting, lots of hyperparameters, etc.)

ILSVRC: Annual competition with 1000 categories and 1.2 million images



VGG

IMAGENET

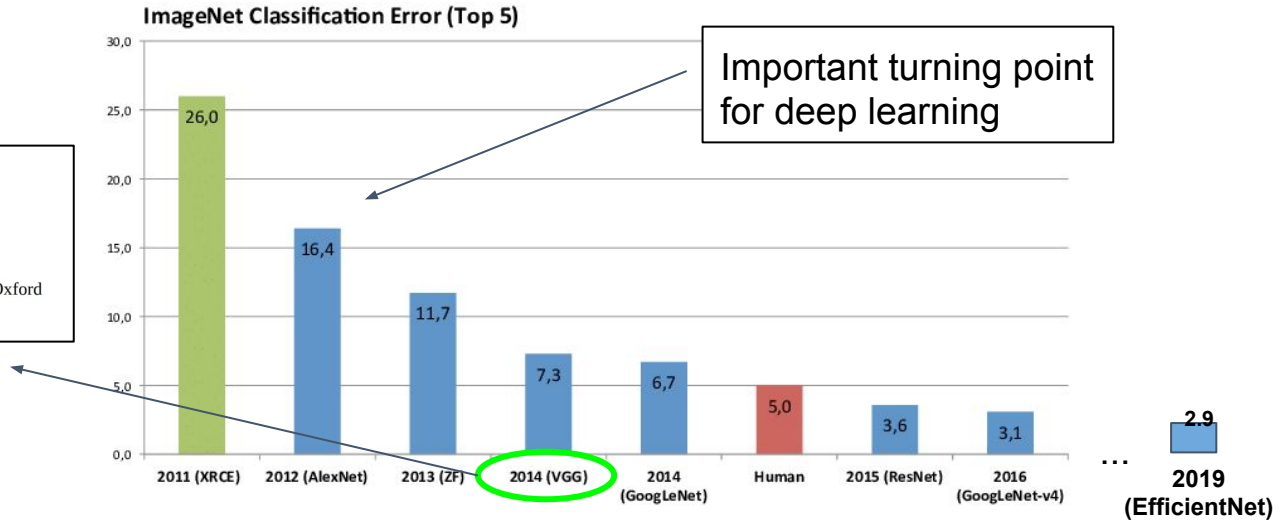
https://knowyourmeme.com/memes/we-need-to-go-deeper

VGG

VGG managed to train **deeper** networks and obtain better accuracy than previous state-of-the-art.

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* **& Andrew Zisserman+**
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk



https://arxiv.org/pdf/1409.1556.pdf

VGG

IMAGENET

"Notably, we did not depart from the classical ConvNet architecture of LeCun et al. (1989), but improved it by substantially increasing the **depth** [of the network]." - Simonyan et al.

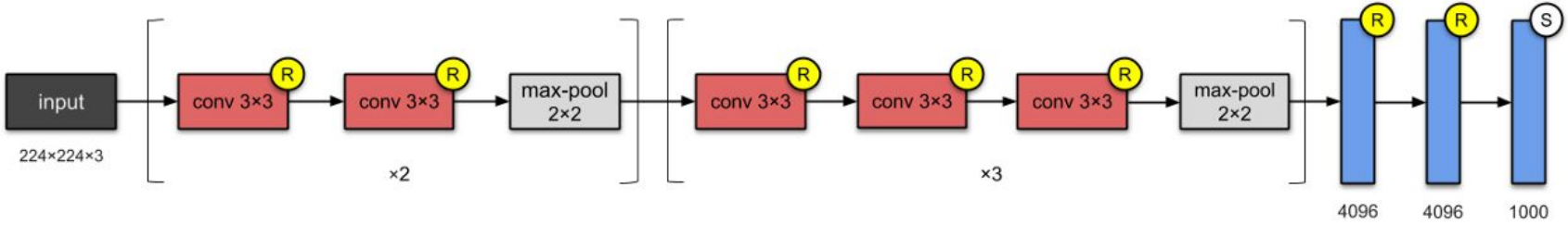
https://knowyourmeme.com/memes/we-need-to-go-deeper

https://arxiv.org/pdf/1409.1556.pdf

VGG

VGG, from the **V**isual **G**eometry **G**roup at Oxford, is very similar in structure to what we have seen so far. It had 2 main contributions:

- smaller kernels everywhere (all 3x3)
- **more layers**



Here we have a diagram for **VGG-16**, where 16 indicates the total number of layers. Spatial resolution is preserved after every convolution layer and downsampling only occurs at max-pooling stages.

https://arxiv.org/pdf/1409.1556.pdf

VGG

Here we see the **different configurations** suggested in the original paper. They are very similar in structure with more or less convolutional layers.

	LeNet	AlexNet	**VGG-16**	**VGG-19**
Parameters	~30 x 10^3	~60 x 10^6	**~138 x 10^6**	**~144 x 10^6**
Layers	5	8	**16**	**19**

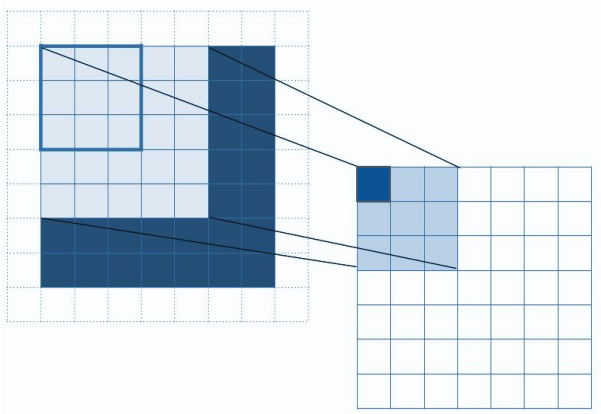
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	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					

https://arxiv.org/pdf/1409.1556.pdf

VGG

They noticed that using multiple chains of 3x3 convolutions was more efficient than a single 7x7 convolution. Why?

"It is easy to see that a stack of two 3×3 conv. layers (without spatial pooling in between) has an **effective receptive field** of 5×5; three such layers have a 7 × 7 effective receptive field." - Simonyan et al.



Field of view for 2D system

http://cs231n.github.io/convolutional-networks/

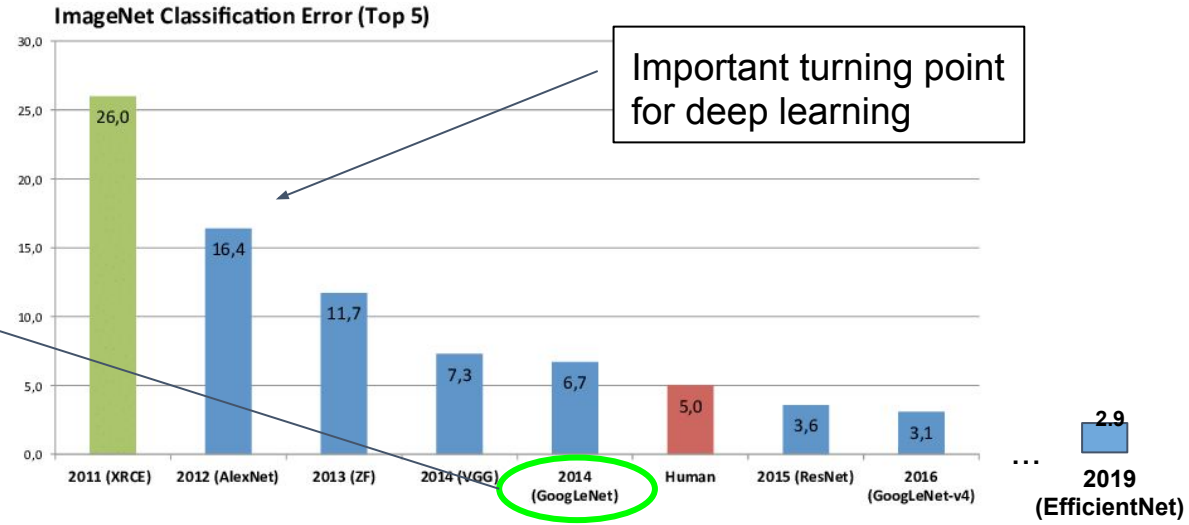
https://arxiv.org/pdf/1409.1556.pdf

GoogLeNet

The same year, GoogLeNet was published, a **deeper** model with less parameters and **higher** accuracy.

Going Deeper with Convolutions

Christian Szegedy[1], Wei Liu[2], Yangqing Jia[1], Pierre Sermanet[1], Scott Reed[3], Dragomir Anguelov[1], Dumitru Erhan[1], Vincent Vanhoucke[1], Andrew Rabinovich[4]
[1]Google Inc. [2]University of North Carolina, Chapel Hill
[3]University of Michigan, Ann Arbor [4]Magic Leap Inc.



https://arxiv.org/abs/1409.4842

GoogLeNet

"In this paper, we will focus on an efficient deep neural network architecture for computer vision, codenamed **Inception**, which derives its name from the Network in network paper by Lin et al [12] in conjunction with the famous "we need to go deeper" internet **meme** [1]"

https://arxiv.org/abs/1409.4842

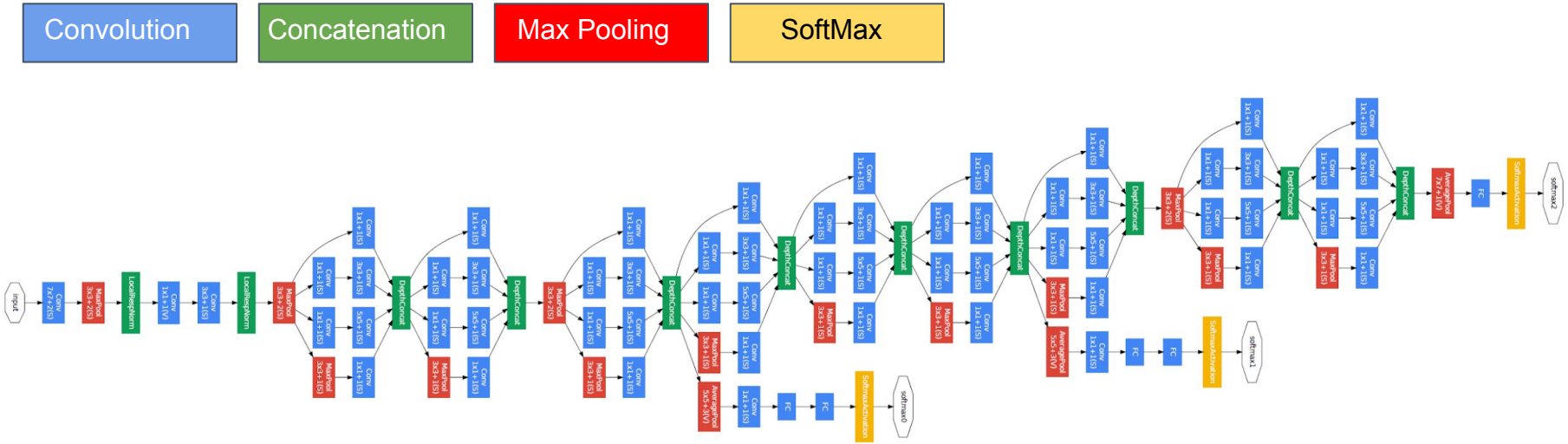
References

[1] Know your meme: We need to go deeper. http://knowyourmeme.com/memes/we-need-to-go-deeper. Accessed: 2014-09-15.

Mila

GoogLeNet

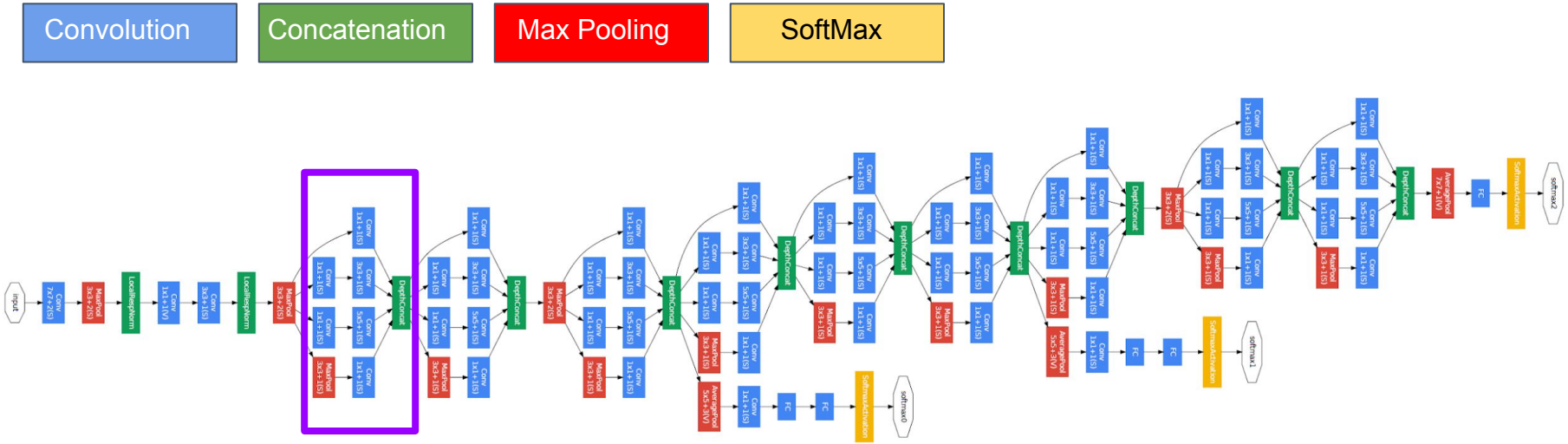
Here is the schematic of the full network. Notice that there are now multiple **parallel** convolution operations per layer.



https://arxiv.org/abs/1409.4842

GoogLeNet

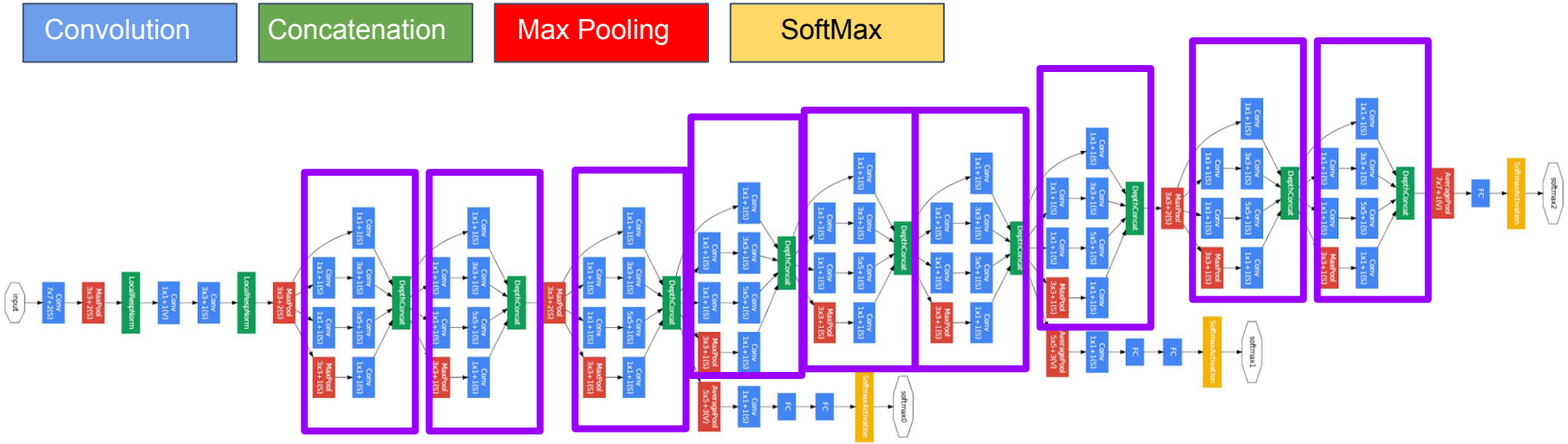
Here is the schematic of the full network. Notice that there are now multiple **parallel** convolution operations per layer. They introduced the **inception module**,a block of operations reused at many layers through the network.



https://arxiv.org/abs/1409.4842

GoogLeNet

Here is the schematic of the full network. Notice that there are now multiple **parallel** convolution operations per layer. They introduced the **inception module**,a block of operations reused at many layers through the network. Here we have **9** inception modules.

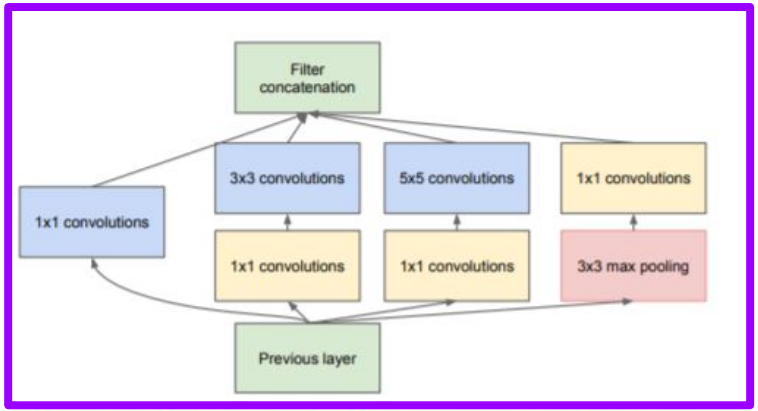


https://arxiv.org/abs/1409.4842

Inception Module

The inception module computes multiple **convolutions in parallel**. It also uses 1x1 convolutions to reduce the number of parameters needed.

"1×1 convolutions are used to compute reductions before the expensive 3×3 and 5×5 convolutions. Besides being used as **reductions**, they also include the use of **rectified linear activation** making them **dual-purpose**." - Szegedy et al.



(b) Inception module with dimension reductions

https://arxiv.org/abs/1409.4842

GoogLeNet

GoogLeNet managed to squeeze **more layers** (22) with significantly less parameters while taking advantage of parallel convolutional layers.

	LeNet	AlexNet	VGG-16	VGG-19	**GoogLeNet**
Parameters	~30 x 10^3	~60 x 10^6	~138 x 10^6	~144 x 10^6	**~7 x 10^6**
Layers	5	8	16	19	**22**

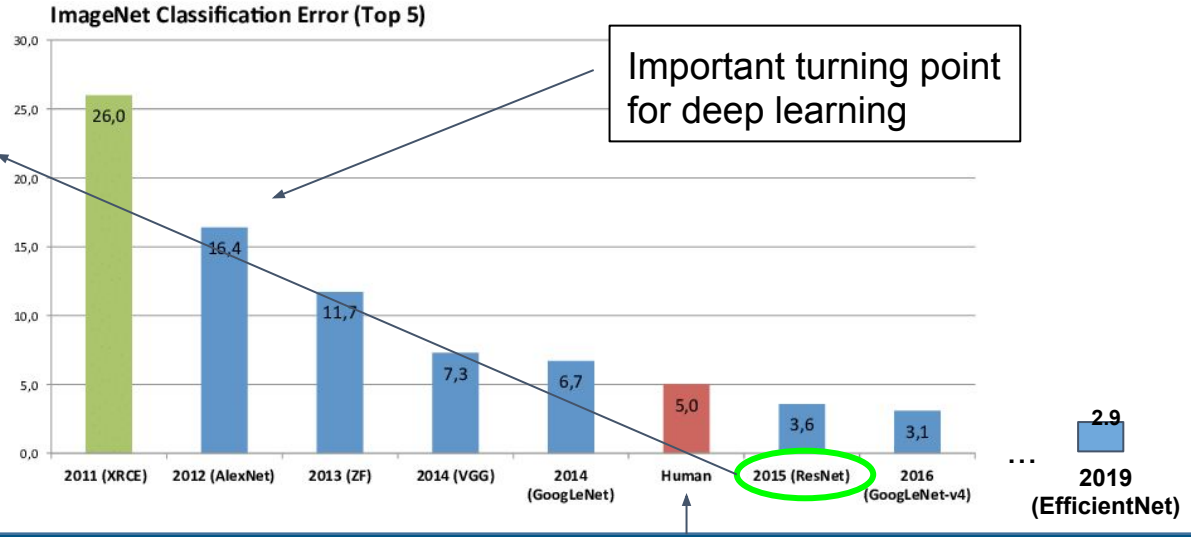
https://arxiv.org/abs/1409.4842

ResNet

Following GoogLeNet, it was noticed that performance didn't always increase when simply going deeper.

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com



https://arxiv.org/pdf/1512.03385.pdf

ResNet

ResNet

"Driven by the significance of **depth**, a question arises: Is learning better networks as easy as **stacking more layers**? [...] Our 152-layer residual net is the **deepest** network ever presented on ImageNet, while still having **lower complexity** than VGG nets"

- He et al.

https://arxiv.org/pdf/1512.03385.pdf

ResNet

It was noticed that as networks of similar architectures got deeper, **accuracy decreased**. However, since a network could simply learn to pass all the info through via identity mapping, two networks that differ by depth only should in theory be able to produce the same accuracy. This was not observed experimentally and meant there was an **information bottleneck**.

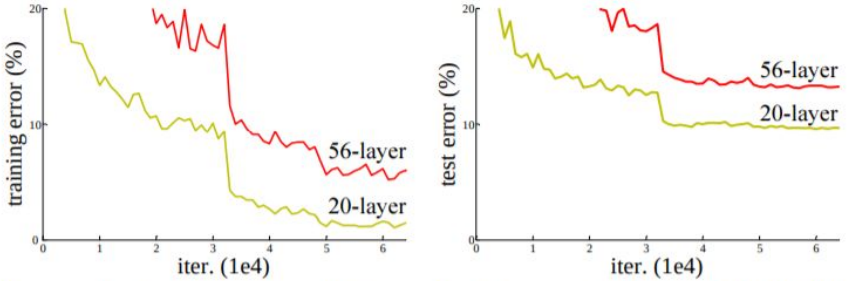


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

https://arxiv.org/pdf/1512.03385.pdf

ResNet

In ResNet, the idea of ***Residual blocks*** is introduced. This allows the networks to get much deeper while avoiding the phenomenon of “accuracy degradation”. The main idea behind ResNet is that the network can always refer to identity mapping of the inputs from previous layers to facilitate information flow.

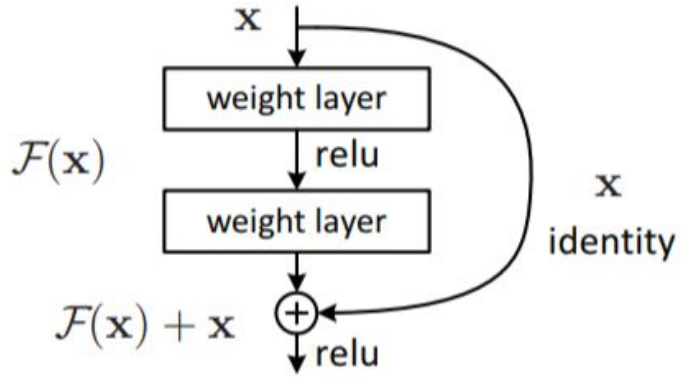


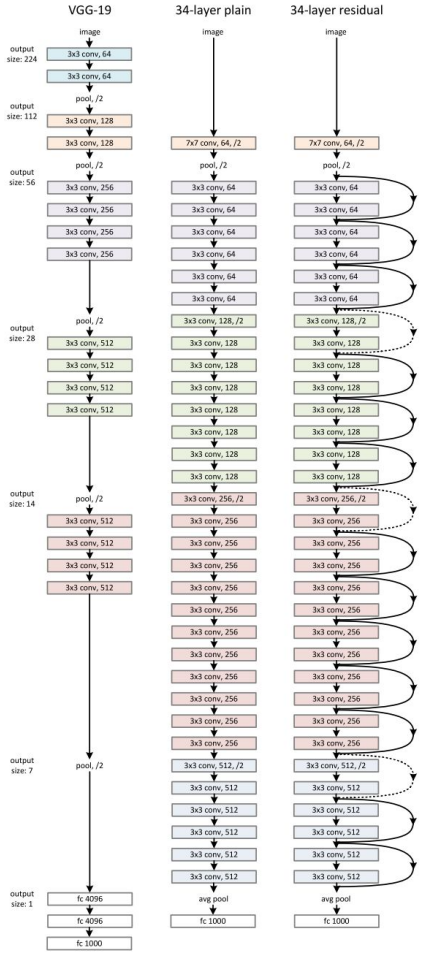
Figure 2. Residual learning: a building block.

https://arxiv.org/pdf/1512.03385.pdf

ResNet

The ResNet architecture consists of many **similar blocks** that are stacked. No pooling is used (except in the first and last layer), only convolutions with stride 2 for down-sampling.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9



https://arxiv.org/pdf/1512.03385.pdf

Mila

Batch Normalization

- Batch normalization was introduced as a form of **regularization** shortly after GoogleNet.
- Prevents the network from getting stuck in the saturated modes
- It is used after each convolution in ResNet
- The network learns at each layer “how much” to normalize its inputs.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

https://arxiv.org/pdf/1502.03167v3.pdf

ResNet

ResNet achieves **better accuracy** with more depth and has less parameters than VGG. It was the first model to achieve better performance than humans evaluated on the same task.

	LeNet	AlexNet	VGG-16	VGG-19	GoogLeNet	**ResNet-152**
Parameters	~30 x 10^3	~60 x 10^6	~138 x 10^6	~144 x 10^6	~7 x 10^6	**60 x 10^6**
Layers	5	8	16	19	22	**152**

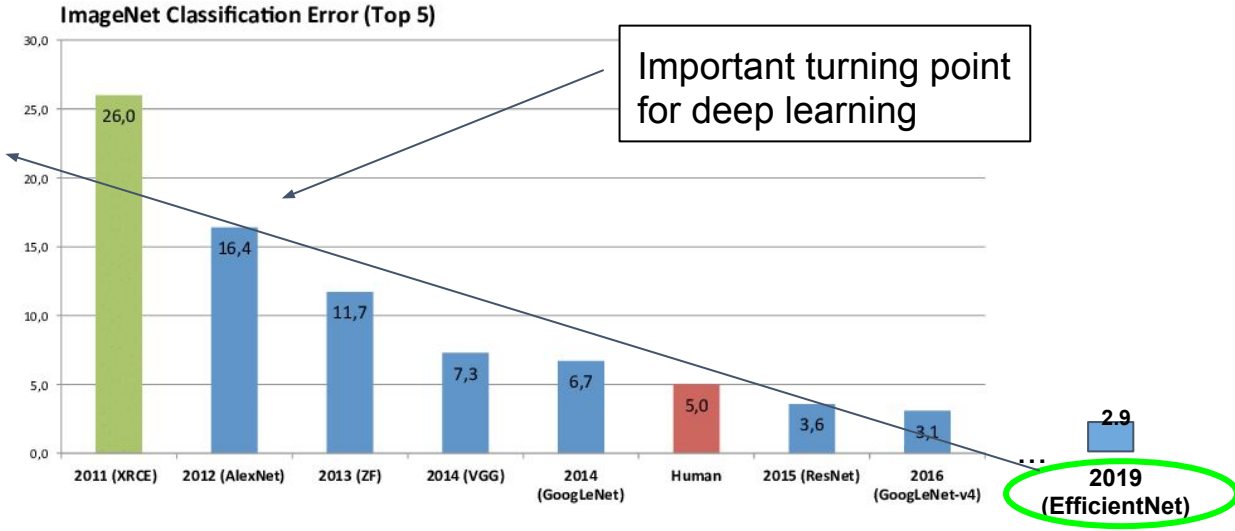
https://arxiv.org/pdf/1512.03385.pdf

EfficientNet

EfficientNet is currently the top performer on ILSVRC. It explores in a principled way how to scale networks in width, depth and resolution and proposes a new baseline architecture that is both faster and more accurate than previous SOTA.

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

Mingxing Tan [1] Quoc V. Le [1]



https://arxiv.org/pdf/1905.11946.pdf

EfficientNet

A base network is found using an exhaustive search of possible architectures. This base network can then be scaled up or down to efficiently trade off accuracy and number of parameters.

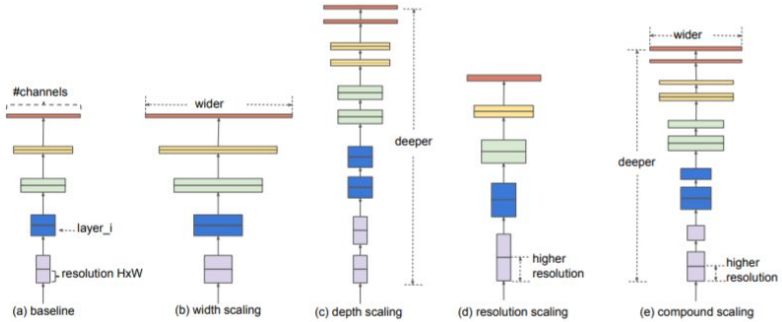
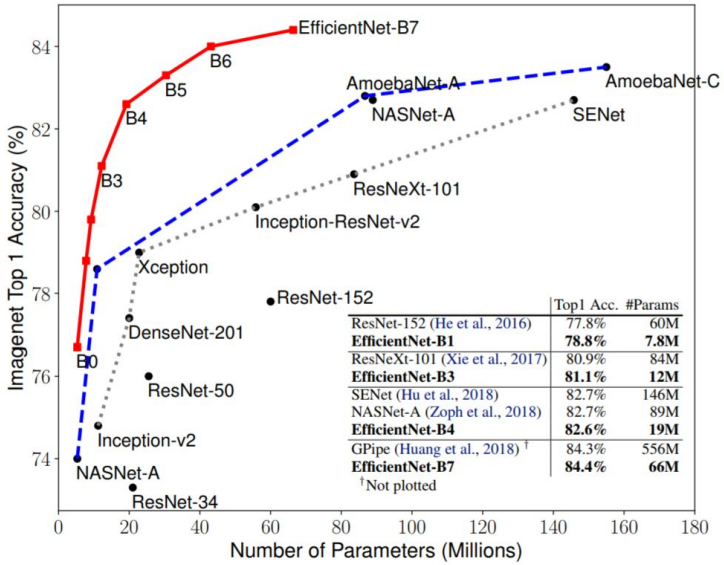


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

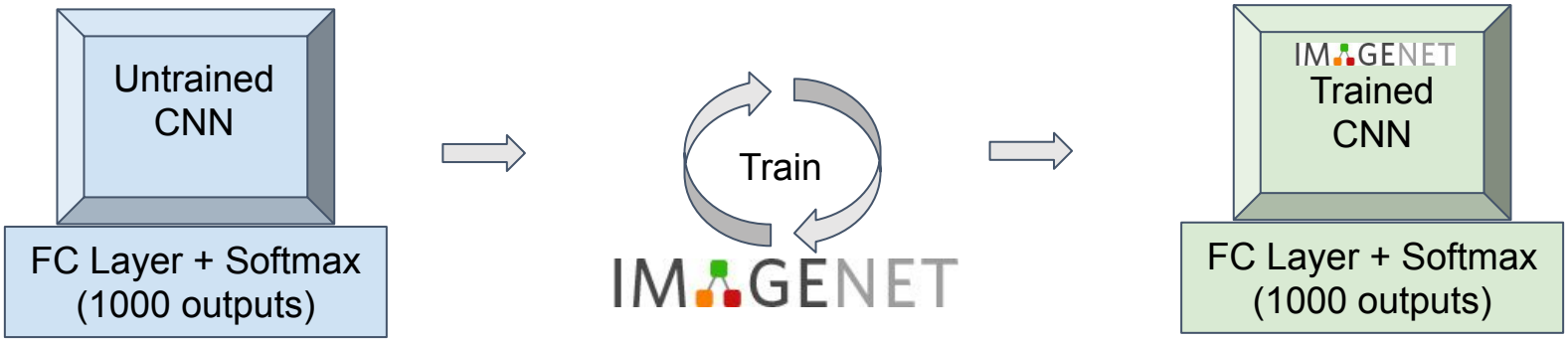


https://arxiv.org/pdf/1905.11946.pdf

Mila

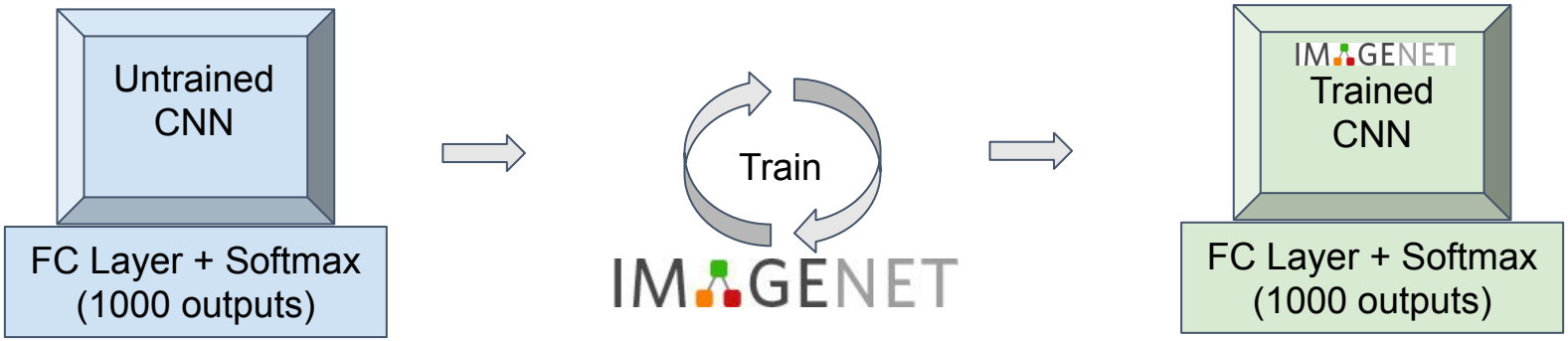
Recap

All the classification networks we have seen so far have been carefully crafted to perform best on **ImageNet**. They all follow a very similar design pattern:

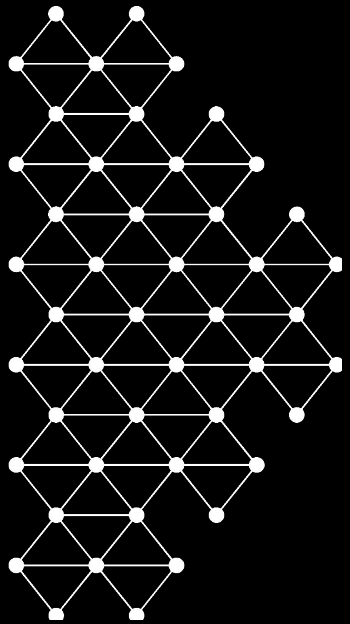


Recap

How can we use what we have learned on smaller datasets?



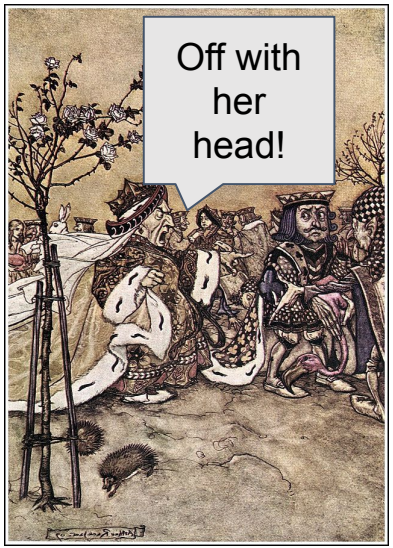
Quebec
Artificial
Intelligence
Institute



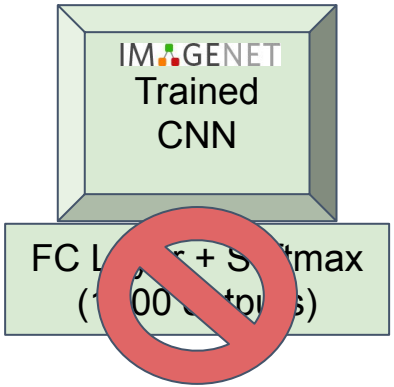
Transfer Learning

Transfer Learning

One method is **Transfer Learning**. We can freeze the trained CNN layers and retrain subsequent fully connected layers on our data. This allows us to leverage the representations learned by ImageNet.

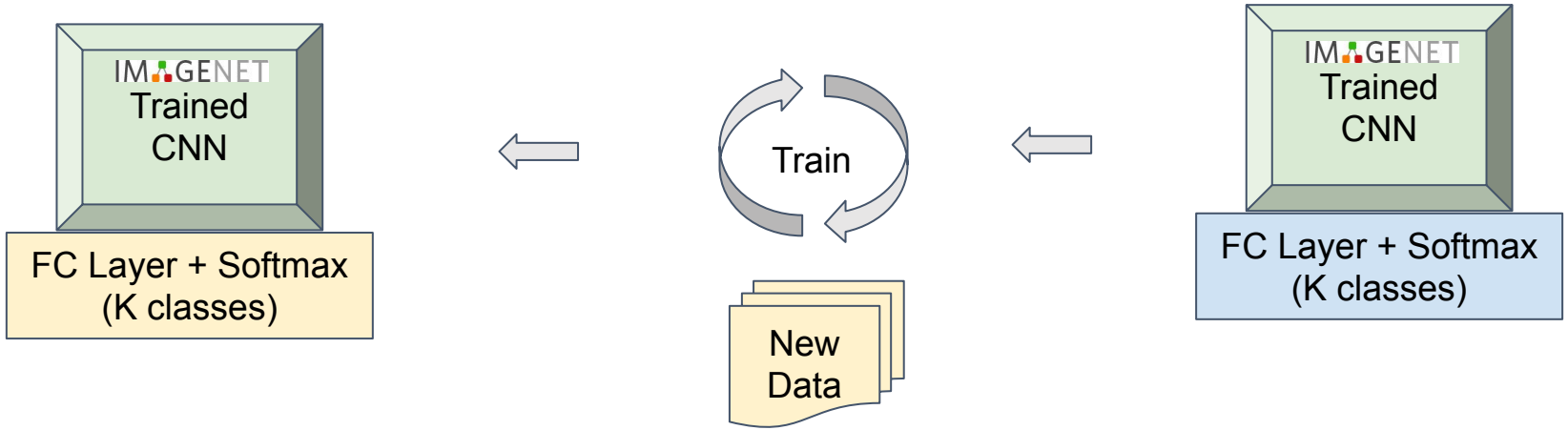


Source: Arthur Rackham



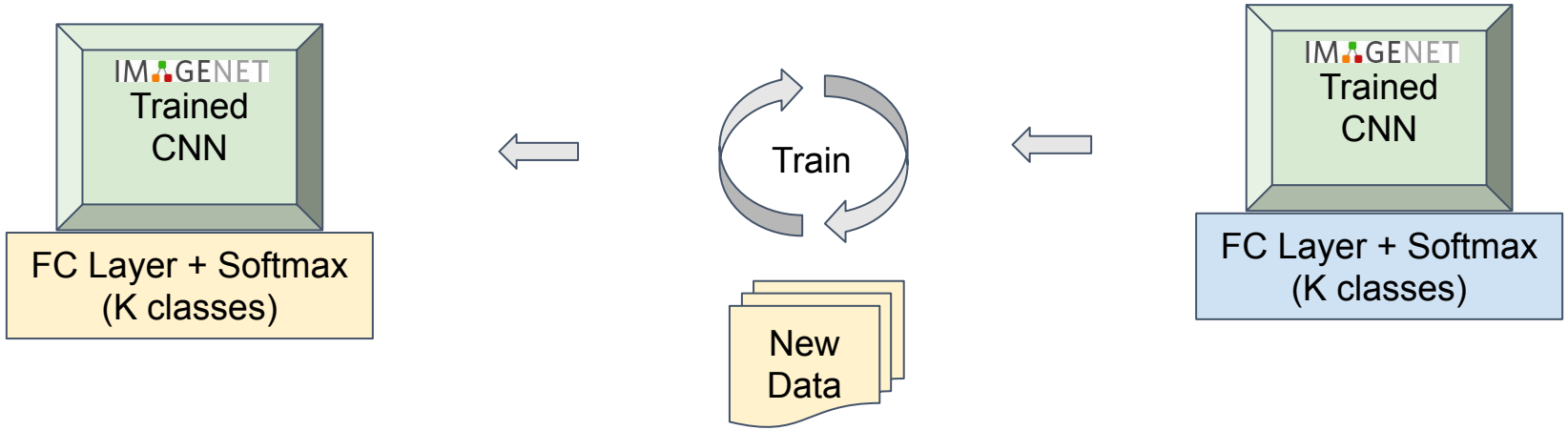
Transfer Learning

One method is **Transfer Learning**. We can freeze the trained CNN layers and retrain subsequent fully connected layers on our data. This allows us to leverage the representations learned by ImageNet.



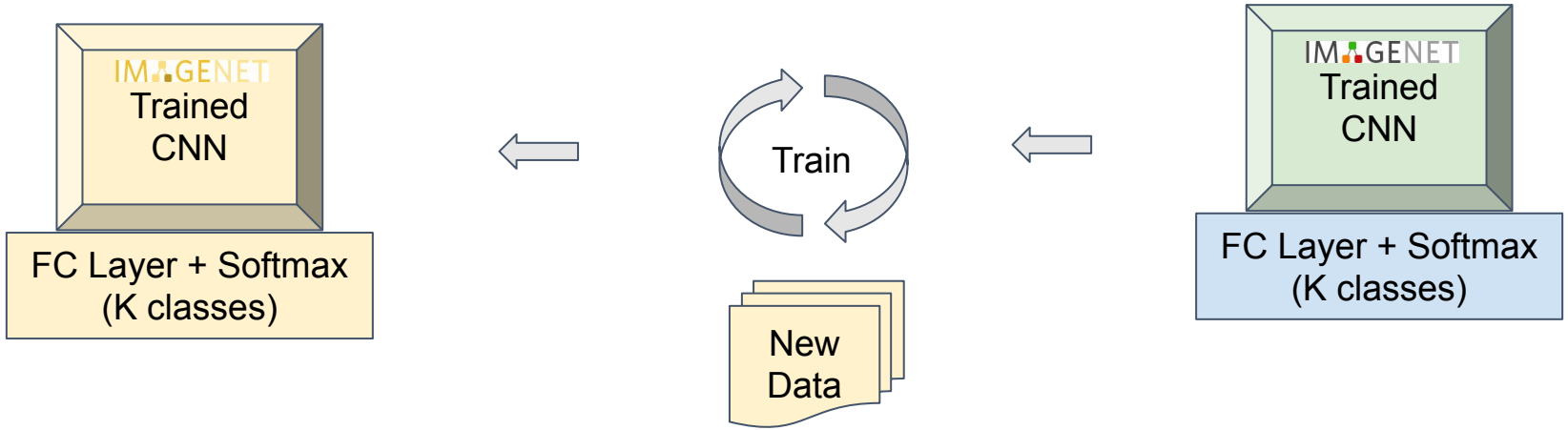
Transfer Learning

This method works particularly well if the domain of the new dataset is **similar** to ImageNet (i.e. if your dataset contains cars, cats, dogs, etc.).



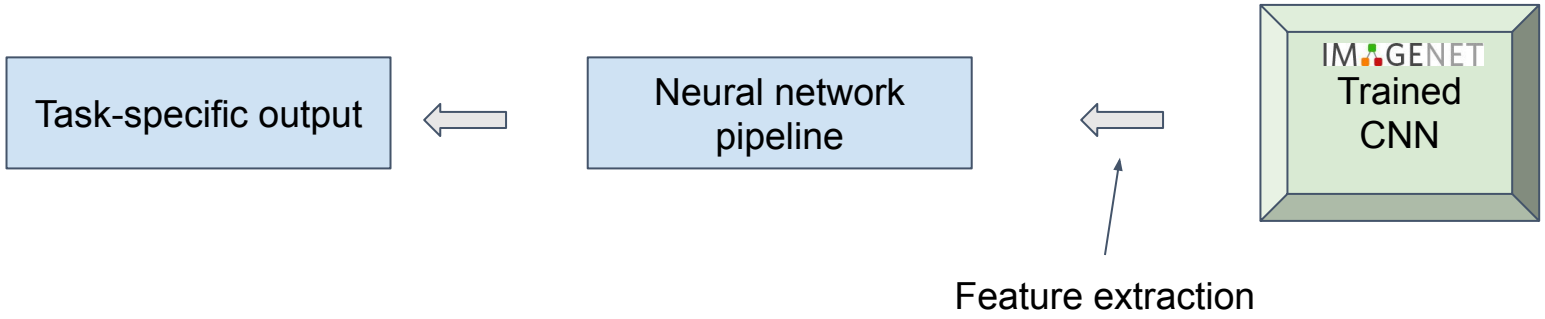
Transfer Learning

It is also possible to retrain the convolutional layers starting from pre-trained weights. Results will vary based on data availability, models, choice of hyper-parameters, etc.



Backbone Networks

A lot of models will use a CNN **backbone** pre-trained on ImageNet as the starting point for different computer vision tasks. It serves as a **generic feature extractor** and can help complicated architectures converge quicker.



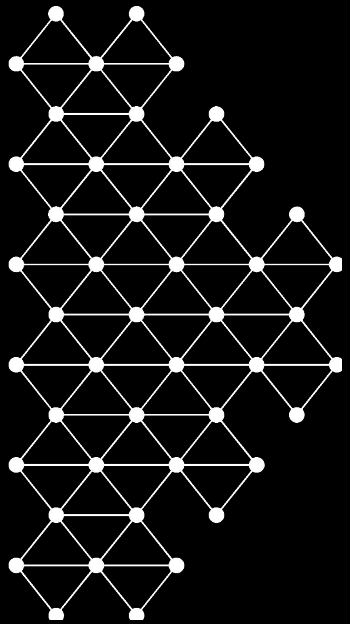
#Facebook

“Since a single machine would have taken **more than a year** to complete the model training, we created a way to distribute the task across up to **336 GPUs**, shortening the total training time to just a few weeks.”

They pretrained their model on 3.5 billion images and 17,000 hashtags

https://engineering.fb.com/ml-applications/advancing-state-of-the-art-image-recognition-with-deep-learning-on-hashtags/

Quebec
Artificial
Intelligence
Institute

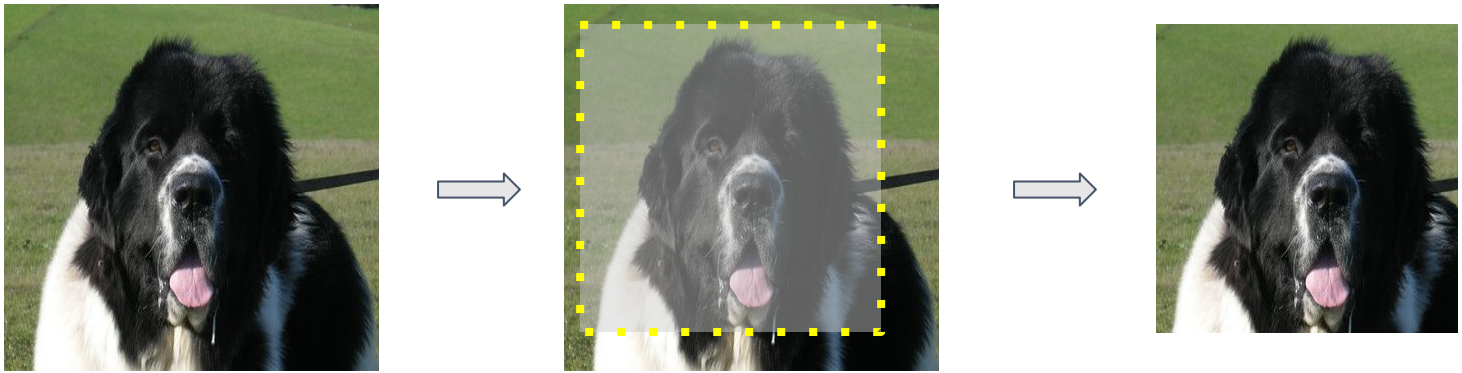


Data Augmentation

Data Augmentation

One trick to help artificially augment a dataset is **data augmentation**. We apply **transformations** to our inputs such that the associated labels remain **unchanged**.

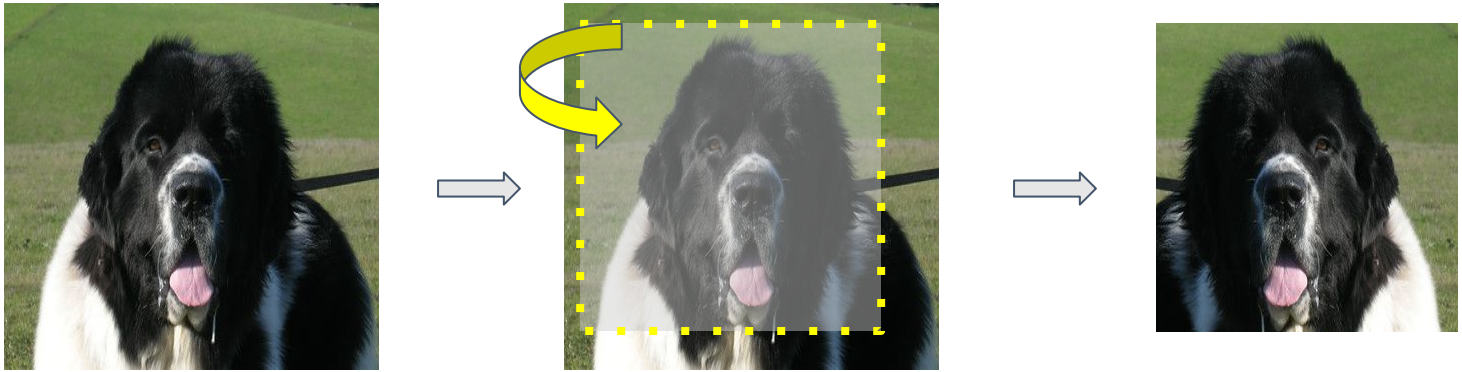
Cropping: We randomly pick a 224x224 patch in our image and use the crop as input to our network.



Data Augmentation

One trick to help artificially augment a dataset is **data augmentation**. We apply **transformations** to our inputs such that the associated labels remain **unchanged**.

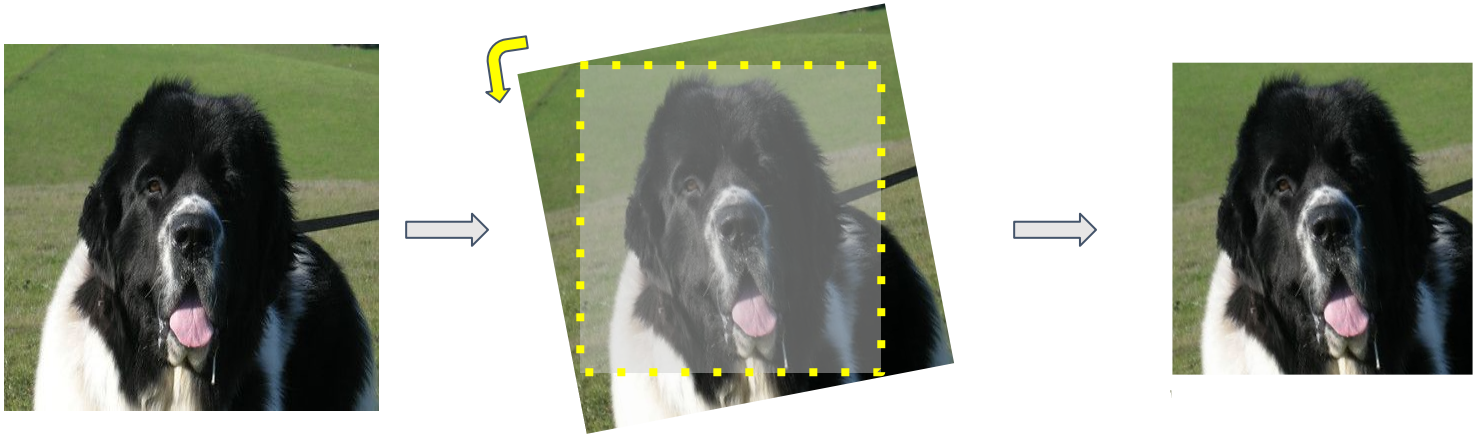
Mirroring: Here we first randomly crop then **mirror** a 224x224 patch in our image and use the crop as input to our network.



Data Augmentation

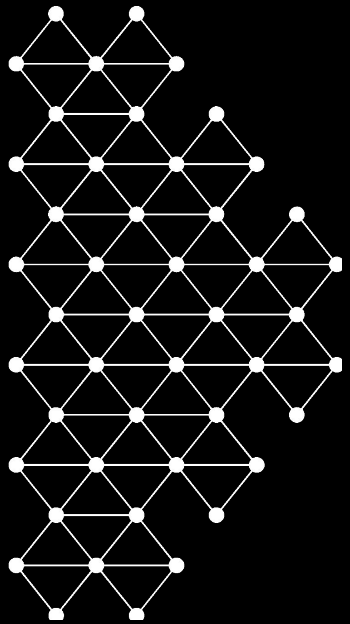
One trick to help artificially augment a dataset is **data augmentation**. We apply **transformations** to our inputs such that the associated labels remain **unchanged**.

Rotation: Here we first rotate our image then crop ta 224x224 patch in our image and use the crop as input to our network.



Quebec
Artificial
Intelligence
Institute

Mila



Structured Outputs

Structured Outputs

Structured outputs allow us to predict **multiple outputs** that **relate** to one another. For example, in the case of an image, a structured output could be a **bounding box** or a per-pixel **segmentation mask** of our input.

Classification:
“Bicycle”

Object Detection:
“Bicycle” + Bounding Box

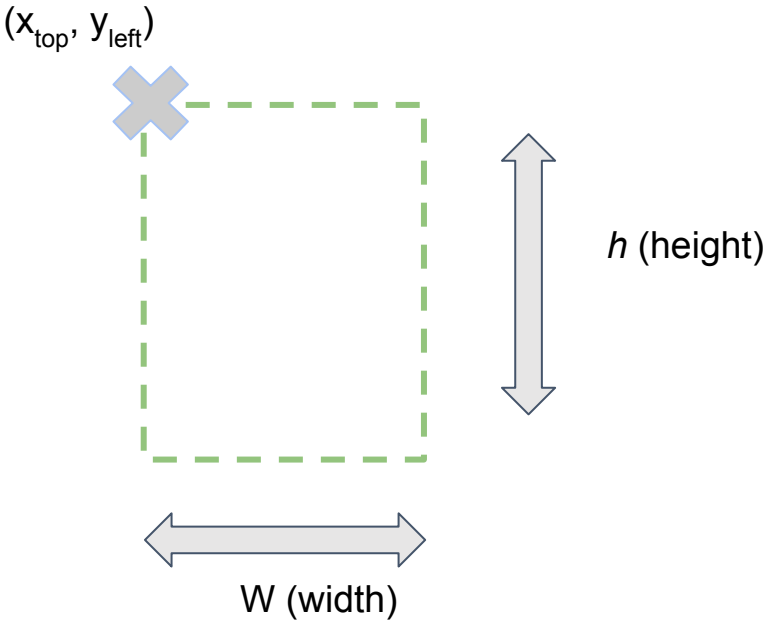
Segmentation:
“Bicycle” + Segmentation Mask

https://farm3.staticflickr.com/2578/4232522029_9dffb2a2df_z.jpg

Structured Outputs

Bounding boxes consist of 4*(x,y) pixel coordinates (8 total), but can be represented using only 4 coordinates. Here is an example of a representation of a bounding box using 4 coordinates.

Object Detection:
“Bicycle” + Bounding Box



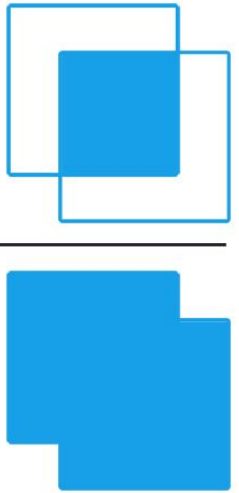
https://farm3.staticflickr.com/2578/4232522029_9dffb2a2df_z.jpg

Structured Outputs

A widely used **metric** to evaluate a bounding box is the **Intersection over Union** (IoU) metric. It measures the **positive overlap** between prediction and ground truth.

Object Detection:
“Bicycle” + Bounding Box

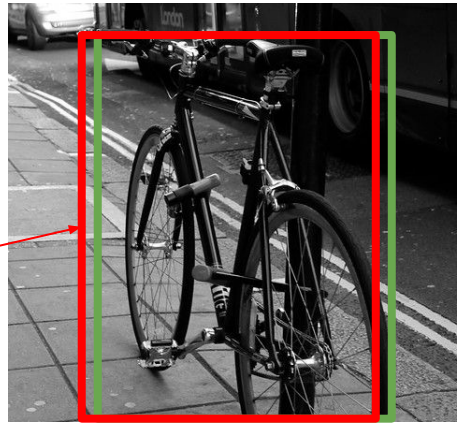
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



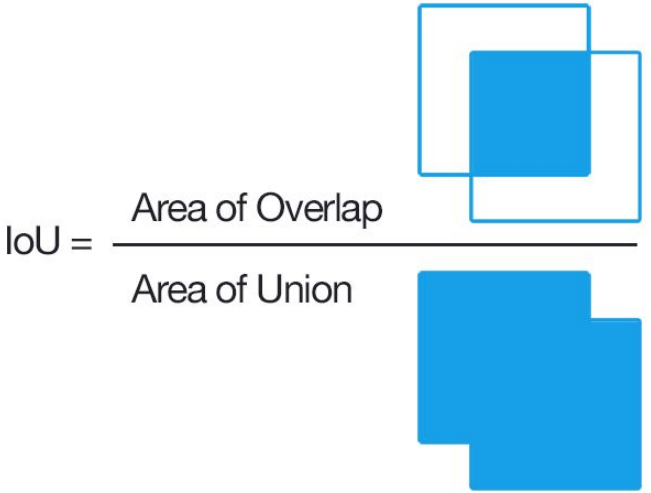
https://farm3.staticflickr.com/2578/4232522029_9dffb2a2df_z.jpg

Structured Outputs

A widely used **metric** to evaluate a bounding box is the **Intersection over Union** (IoU) metric. It measures the **positive overlap** between prediction and ground truth. IoU is a value between 0 (worst) and 1 (best).



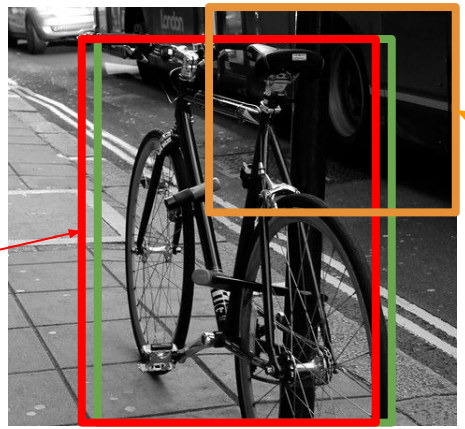
Object Detection:
"Bicycle" + Bounding Box



https://farm3.staticflickr.com/2578/4232522029_9dffb2a2df_z.jpg

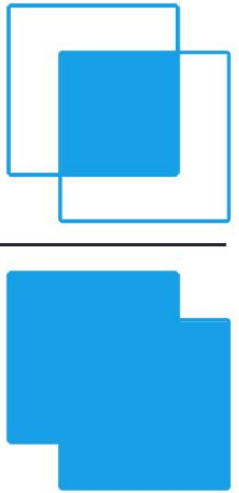
Structured Outputs

A widely used performance **metric** to evaluate a bounding box is the **Intersection over Union** (IoU) metric. It measures the **positive overlap** between prediction and ground truth. IoU is a value between 0 (worst) and 1 (best).



Object Detection:
"Bicycle" + Bounding Box

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



https://farm3.staticflickr.com/2578/4232522029_9dffb2a2df_z.jpg

Structured Outputs

A **segmentation mask** classifies pixels of the input image as belonging to a certain class or to the background. Classes are represented as integers. Segmentation masks can be overlaid on top of the original image.

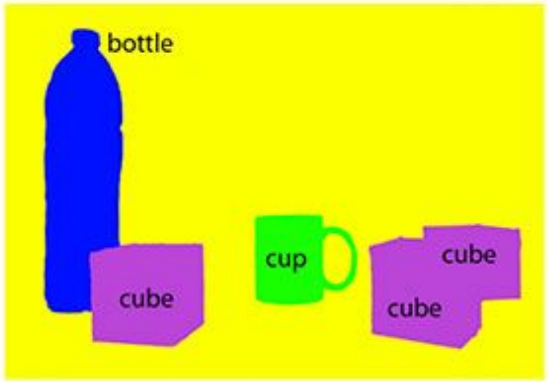
Segmentation:
“Bicycle” + Segmentation Mask

https://farm3.staticflickr.com/2578/4232522029_9dffb2a2df_z.jpg

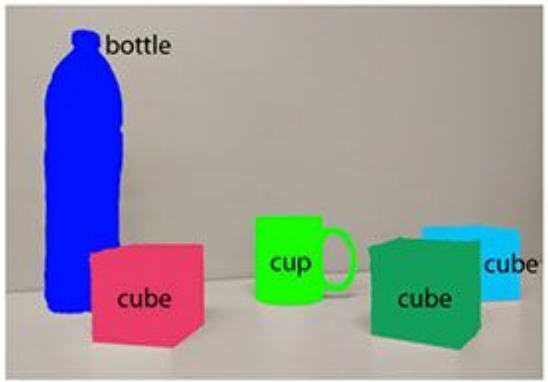
Structured Outputs

Segmentation is done at the **pixel level**. In **semantic segmentation**, the entire image is classified, pixel by pixel. In **instance segmentation**, only regions of interest are segmented. **Distinction** between objects of the same class can be made.

Segmentation:
“Bicycle” + Segmentation Mask



Semantic segmentation



Instance segmentation

https://www.pyimagesearch.com/2018/11/19/mask-r-cnn-with-opencv/
https://www.di.ens.fr/willow/events/cvml2010/materials/INRIA_summer_school_2010_Christoph.pdf

Faster R-CNN

Faster R-CNN can be used for **object detection**. It uses a VGG backbone pre-trained on ImageNet to extract feature maps. These feature maps are fed to the Region Proposal Network which suggests candidate **bounding boxes** that might contain objects. These are then classified accordingly and return for each box the associated labels, coordinates and confidence scores.

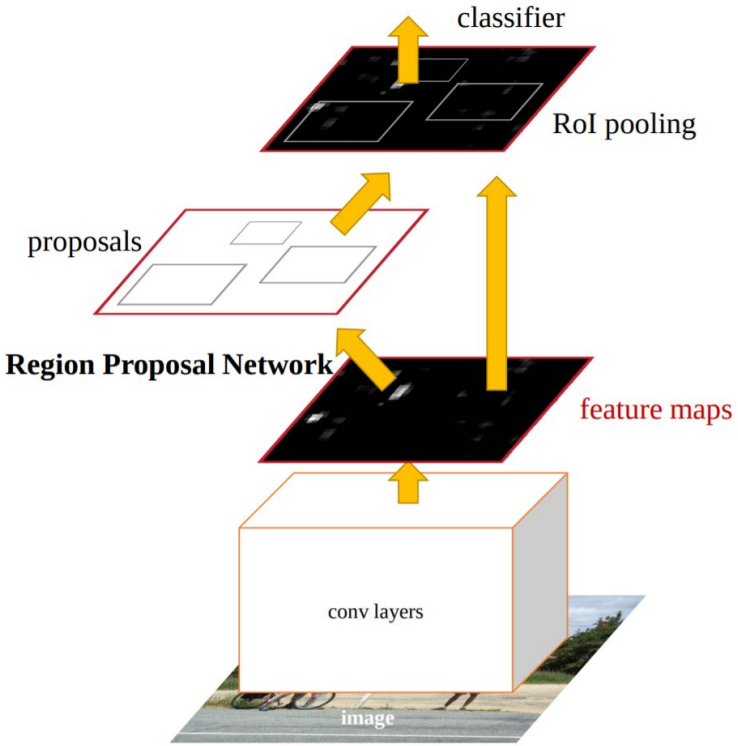
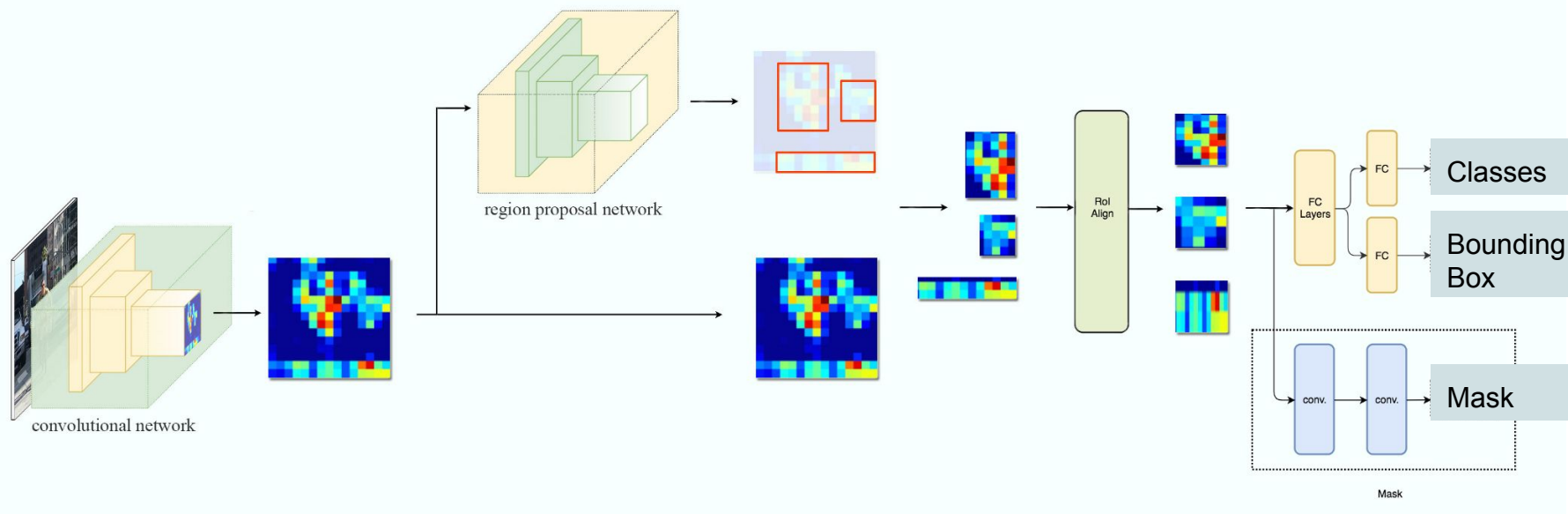


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

https://arxiv.org/pdf/1506.01497.pdf

Mask R-CNN

Mask R-CNN is a **popular** architecture for **instance segmentation**. It works by predicting in **parallel** regions of interest (bounding boxes), which class they belong to, and the **segmentation mask** of these objects.



https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

UNet

UNet is a model for **semantic segmentation** that was introduced in the context of biomedical image segmentation. It preserves higher level convolutions and concatenates them with lower level convolutional layers.

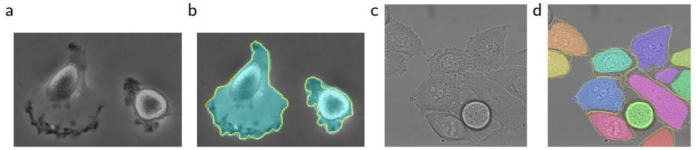
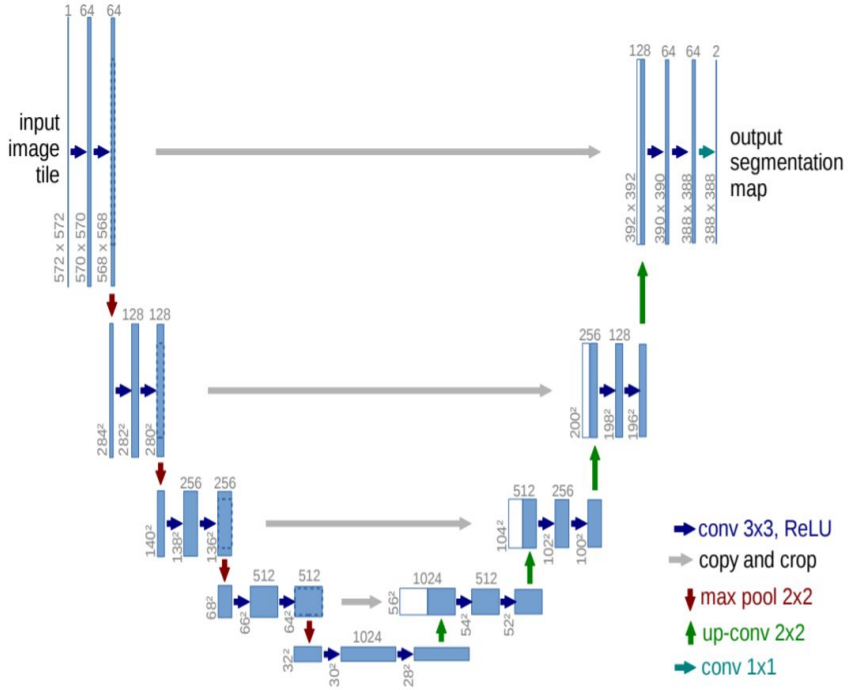


Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

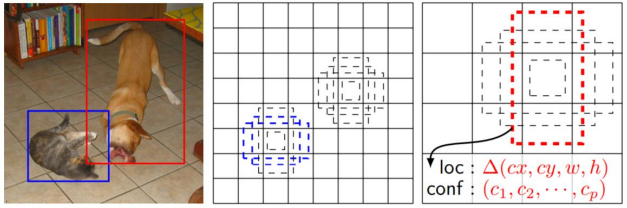


https://arxiv.org/pdf/1505.04597.pdf

Mila

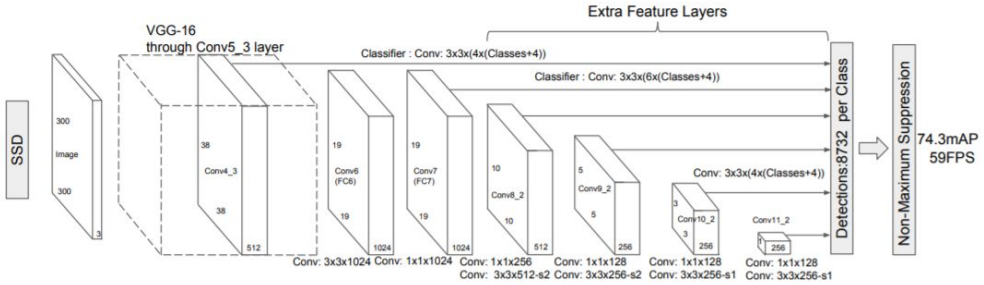
SSD

SSD (Single Shot Detector) is a model designed for **real-time** object detection. It can achieve high performance while maintaining **high frame rates**. It predicts the contents of multiple pre-scaled bounding boxes at different scales of the feature maps.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Method	mAP	FPS
Faster R-CNN (VGG16)	73.2	7
Fast YOLO	52.7	155
YOLO (VGG16)	66.4	21
SSD300	74.3	46
SSD512	76.8	19



https://arxiv.org/pdf/1512.02325.pdf

MobileNet

MobileNet is a backbone architecture that was introduced to **reduce** the computational complexity of the backbone models without sacrificing accuracy. It introduced the concept of *depthwise separable convolutions* which significantly **reduce the number of parameters** in a model.

Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

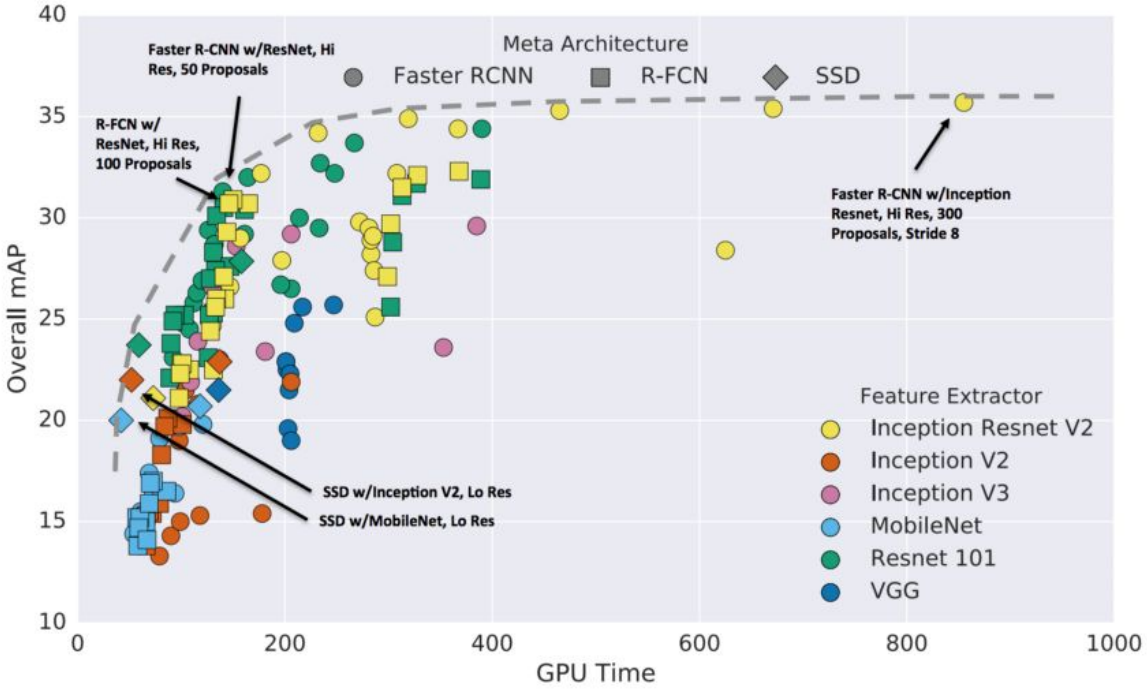
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters
Xception	88 MB	0.790	0.945	22,910,480
VGG16	528 MB	0.713	0.901	138,357,544
VGG19	549 MB	0.713	0.900	143,667,240
ResNet50	98 MB	0.749	0.921	25,636,712
ResNet101	171 MB	0.764	0.928	44,707,176
ResNet152	232 MB	0.766	0.931	60,419,944
ResNet50V2	98 MB	0.760	0.930	25,613,800
ResNet101V2	171 MB	0.772	0.938	44,675,560
ResNet152V2	232 MB	0.780	0.942	60,380,648
ResNeXt50	96 MB	0.777	0.938	25,097,128
ResNeXt101	170 MB	0.787	0.943	44,315,560
InceptionV3	92 MB	0.779	0.937	23,851,784
InceptionResNetV2	215 MB	0.803	0.953	55,873,736
MobileNet	16 MB	0.704	0.895	4,253,864
MobileNetV2	14 MB	0.713	0.901	3,538,984

https://arxiv.org/pdf/1704.04861.pdf

https://keras.io/applications/

Model Comparisons

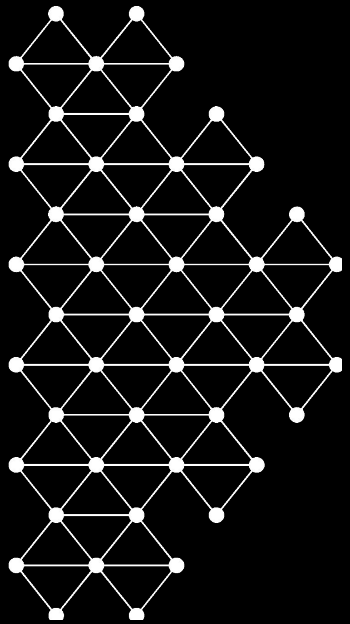
Speed vs. Accuracy



https://arxiv.org/pdf/1611.10012.pdf
https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359

Quebec
Artificial
Intelligence
Institute

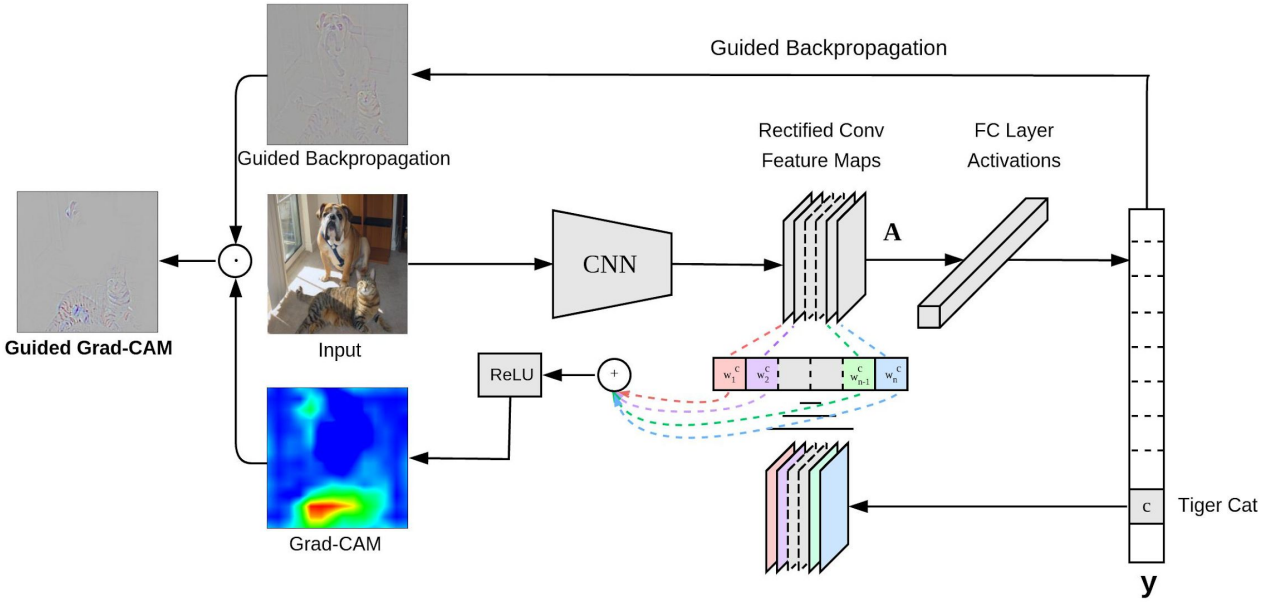
Mila



Questions

Interpreting networks

Gaining **insight** in to CNN predictions is **hard**. Some methods, such as **GradCam**, look at the **gradient** of a single-class prediction to interpret what the model is looking at.



https://arxiv.org/abs/1610.02391

FractalNet

Fractal Nets allow the training of deep networks without residuals.

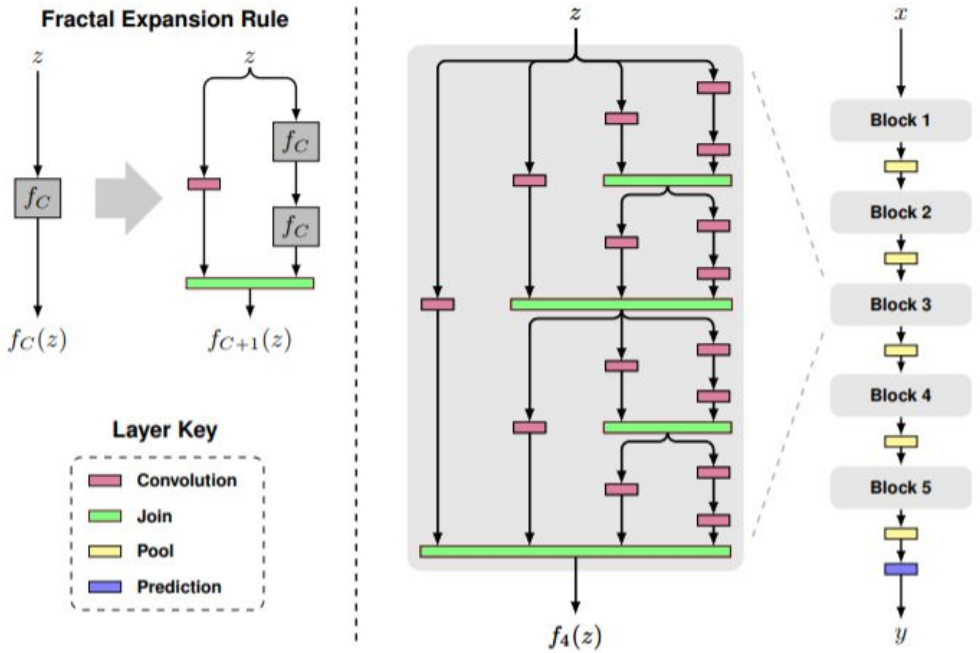


Figure 1: **Fractal architecture.** *Left:* A simple expansion rule generates a fractal architecture with C intertwined columns. The base case, $f_1(z)$, has a single layer of the chosen type (*e.g.* convolutional) between input and output. Join layers compute element-wise mean. *Right:* Deep convolutional networks periodically reduce spatial resolution via pooling. A fractal version uses f_C as a building block between pooling layers. Stacking B such blocks yields a network whose total depth, measured in terms of convolution layers, is $B \cdot 2^{C-1}$. This example has depth 40 ($B = 5$, $C = 4$).

https://arxiv.org/pdf/1605.07648.pdf