Lecture Notes for

Machine Learning in Python

⟦👨‍🏫, 👨‍💻, 🐍, 👨‍🔬⟧

Professor Eric Larson

An Ongoing History of Convolutional Networks

History of Convolutional Neural Networks

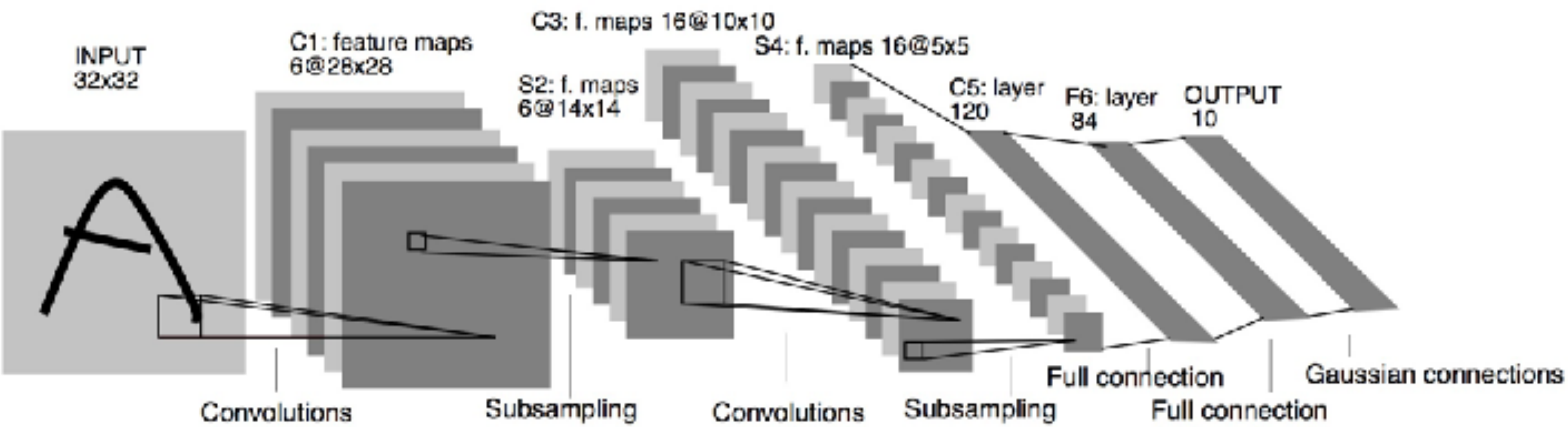
Machine Learning 101

Types of CNN, 1988-1998

Heads Facebook AI Team

- **LeNet-1** (1988)
  - ~2600 params, not many layers
- **LeNet-5** (1998)
  - 7 layers, gets excellent MNIST performance
- Major contribution, general structure:
  - conv=>pool=>non-linearity=> …=>MLP

avg **tanh or sigmoid**



https://culurciello.github.io/tech/2016/06/04/nets.html

CNN History

- List of major breakthroughs from 1998 through 2010 in convolutional networks:

¯_(ツ)_/¯

- 2010

Types of CNN, 2010

AI Researcher
IDSA, Switzerland

- **Ciresan Net**
- Publishes code for running CNN via GPU
  - Subsequently wins 5 international competitions
  - from stop signs => cancer detection
- Major contribution: NVIDIA parallelized training algorithms

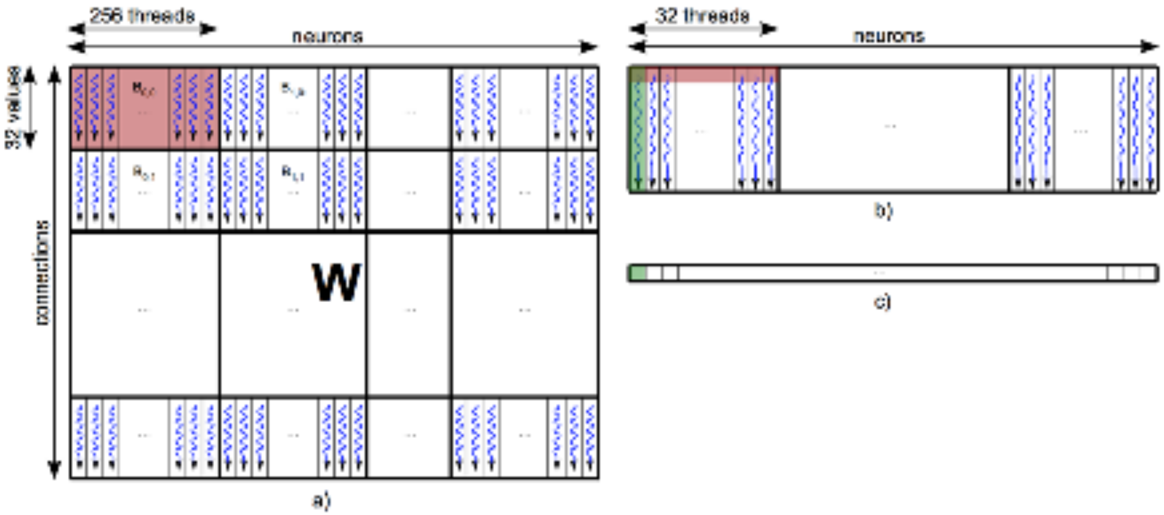
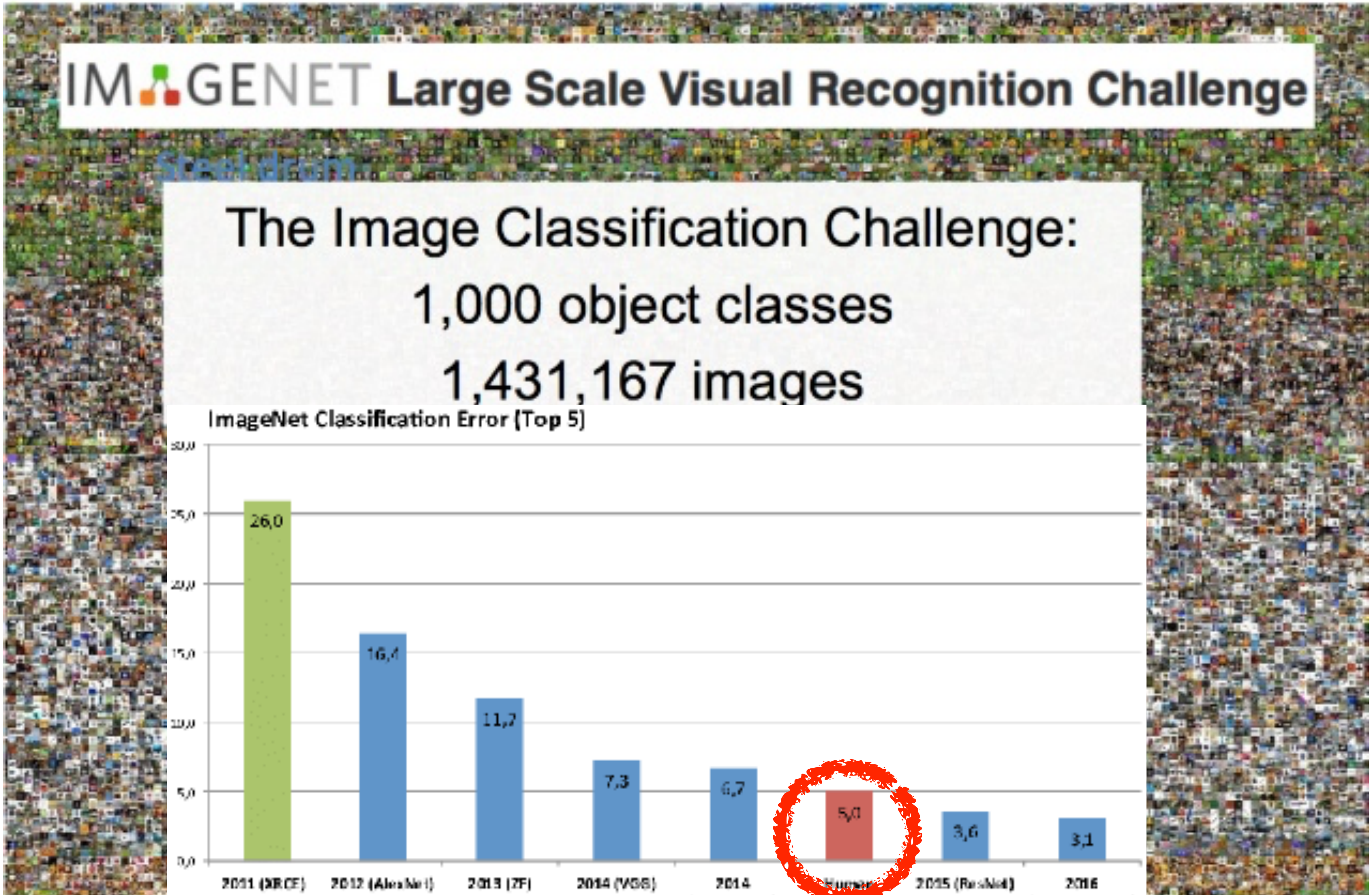


Figure 2: Forward propagation: a) mapping of kernel 1 grid onto the padded weight matrix; b) mapping the kernel 2 grid onto the partial dot products matrix; c) output of forward propagation.

https://arxiv.org/pdf/1003.0358.pdf

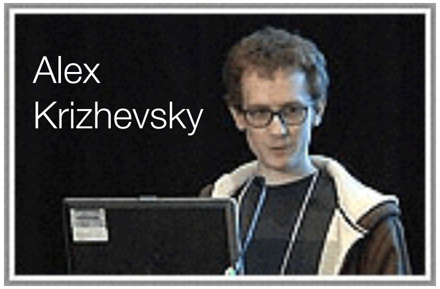
ImageNet Competition (2010-2016)



https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

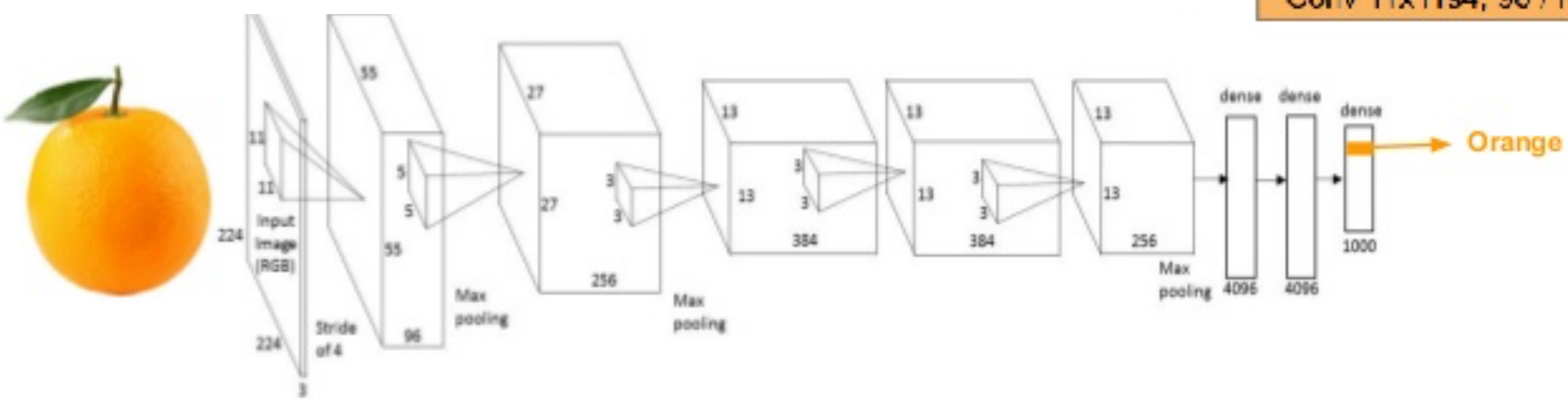
https://www.slideshare.net/nmhkahn/case-study-of-convolutional-neural-network-61556303

Types of CNN, 2012



Google

- **AlexNet**, Hinton is mentor
  - wins ImageNet competition
- Major contributions:
  - dropout for regularization
  - systematic use of ReLU
  - data expansion
  - ***overlapping max pool***

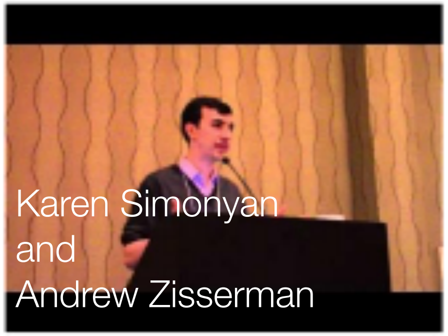


https://culurciello.github.io/tech/2016/06/04/nets.html

Warning


DON'T TRY WHAT YOU'RE ABOUT TO SEE AT HOME
WE'RE WHAT YOU CALL
EXPERTS
WeKnowMemes

Types of CNN, 2013



UNIVERSITY OF OXFORD

Google DeepMind

- Oxford **VGG Net** (Visual Geometry Group)
- Major contributions:
    - small cascaded kernels
    - way more layers (19 versus ~7)
    - “emulates” biology “better”
    - trained on NVIDIA GPUs for 2-3 weeks

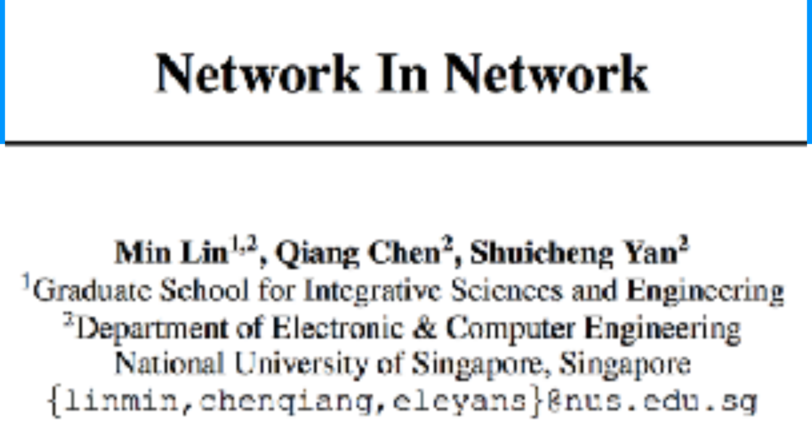
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128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

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

http://www.robots.ox.ac.uk/~vgg/practicals/cnn/

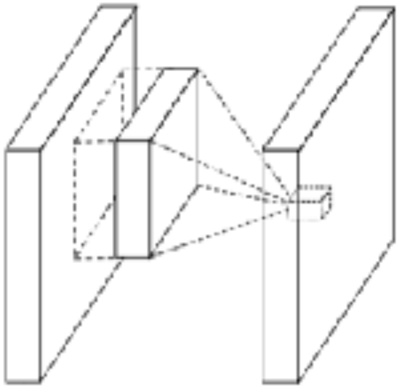
Types of CNN, 2014



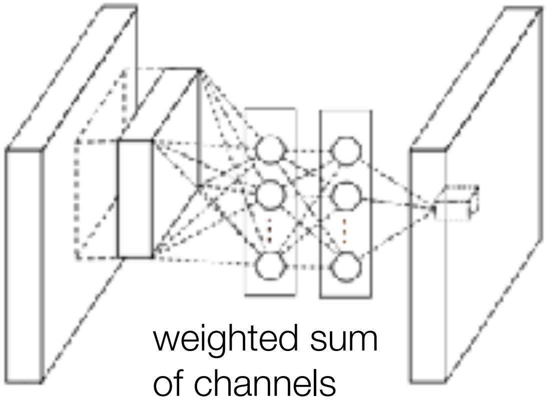
Network In Network

Min Lin[1,2], Qiang Chen[2], Shuicheng Yan[2]
[1]Graduate School for Integrative Sciences and Engineering
[2]Department of Electronic & Computer Engineering
National University of Singapore, Singapore
{linmin,chenqiang,eleyans}@nus.edu.sg

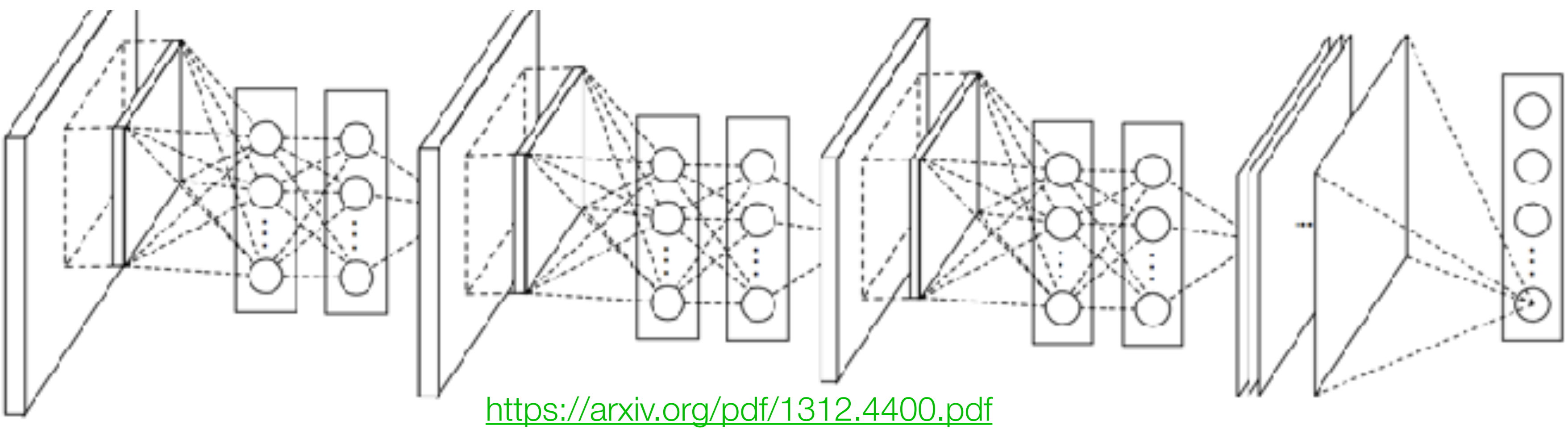
- Network in Network **NiN**
  - or MLPConv



(a) Linear convolution layer



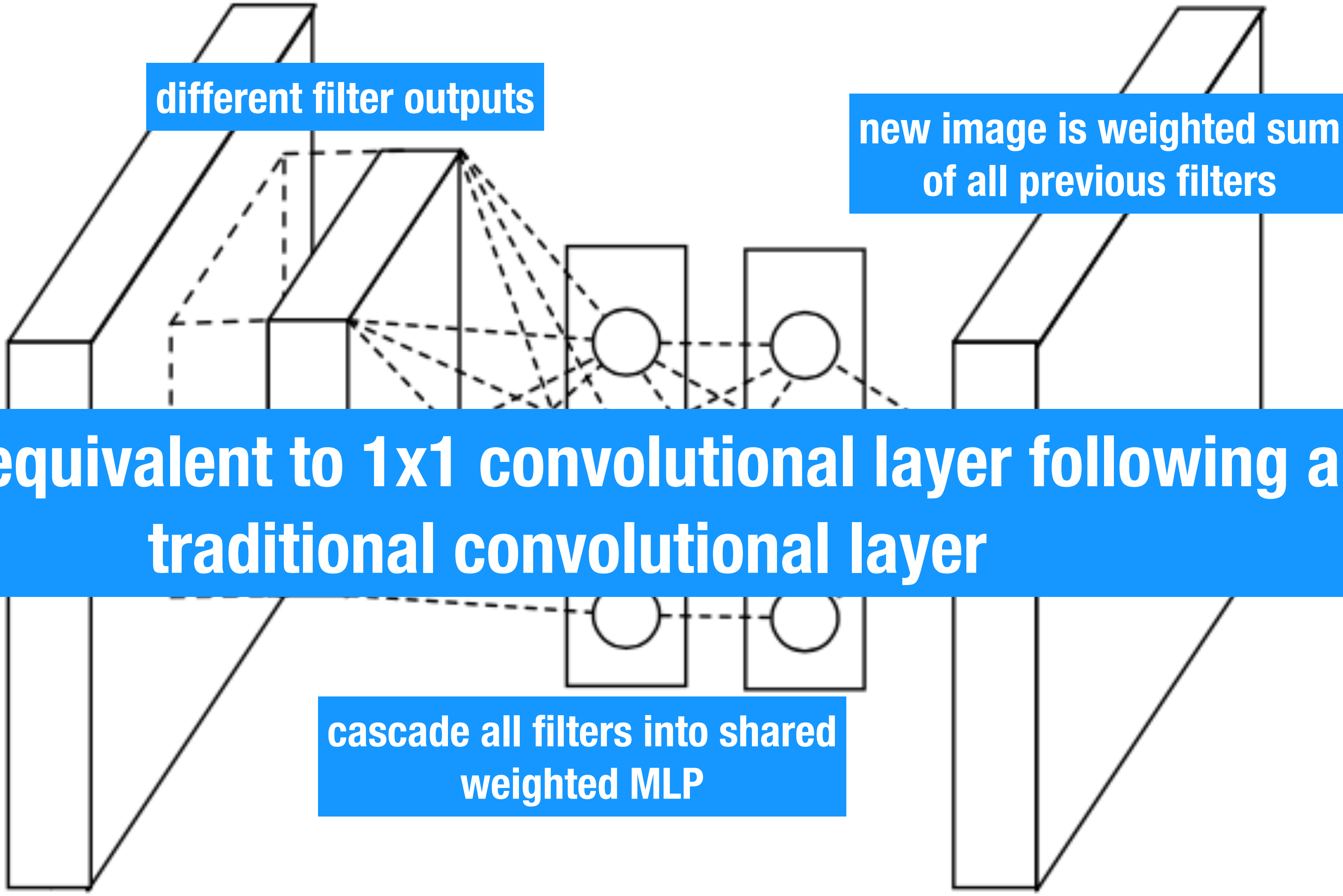
(b) Mlpconv layer



https://arxiv.org/pdf/1312.4400.pdf

Types of CNN, 2014

- Network in Network



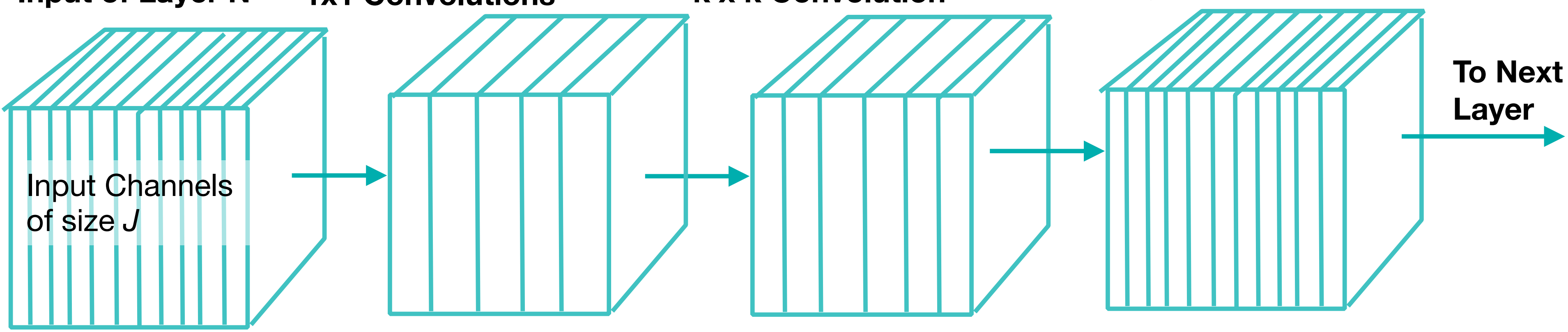
this is equivalent to 1x1 convolutional layer following a traditional convolutional layer

https://arxiv.org/pdf/1312.4400.pdf

NiN, expanded view

J and ***M*** >> ***K*** and ***L***

Common Choice: ***J***==***M*** and ***K***==***L***

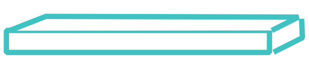


Convolve with *K* filters
1x1 filters

Convolve with *L* filters
k x k filters

Convolve with *M*, 1x1 Filters
Optional Output Bottleneck

Convolve with *K*, 1x1 Filters
Equivalently: each new channel is weighted sum of convolutions complete control of channels size



one 1x1 filter input filter

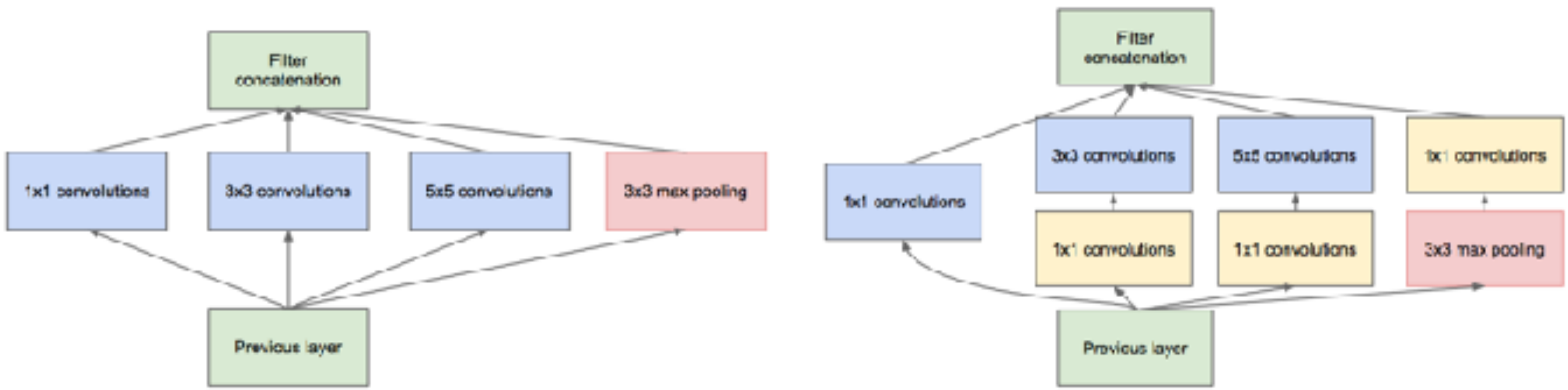
one full filter, *k* x *k*

optional one 1x1 filters to control output size

Structure of Each Tensor: Channels x Rows x Columns

Types of CNN, 2014

- **GoogLeNet**
  - or **Inception V1**
- Major contribution:
  - bottleneck layering
  - parallel NiN

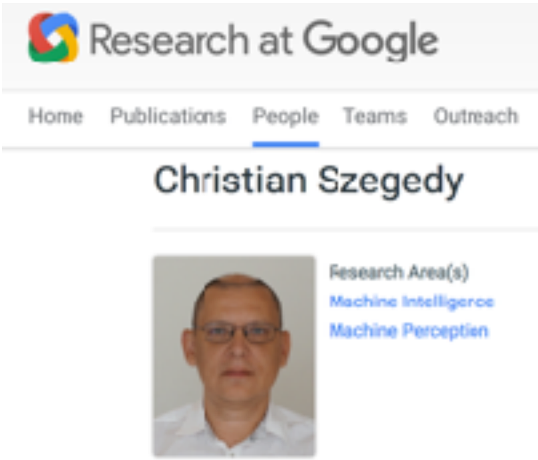


(a) Inception module, naïve version (b) Inception module with dimension reductions

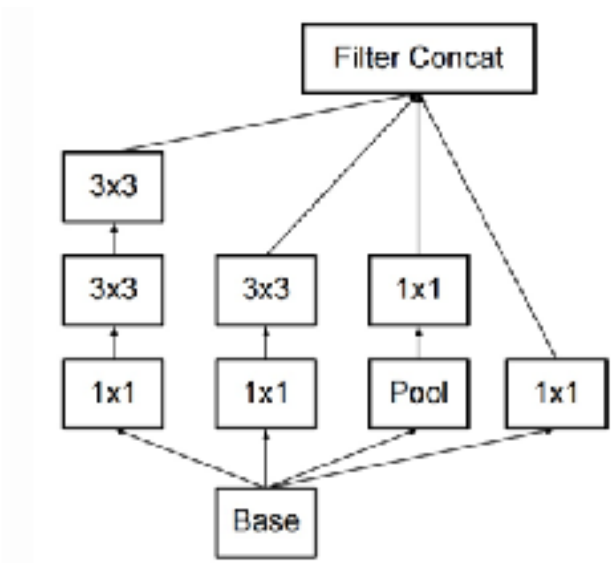
Figure 2: Inception module

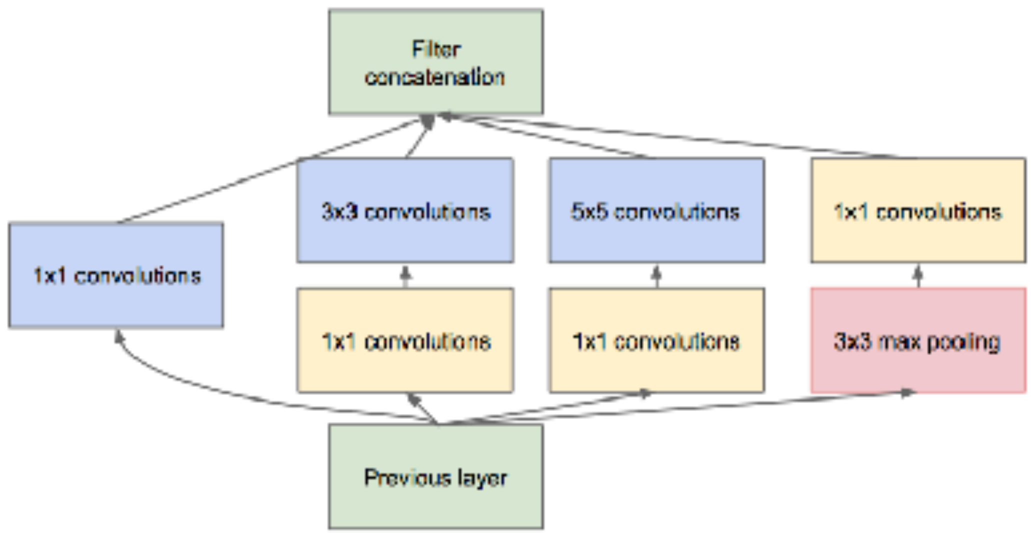
https://arxiv.org/pdf/1409.4842.pdf

Types of CNN, 2015 February and December



- **Inception V2**, Inception V1 with batch normalization
- **Inception V3**:
  - replace 5x5 with multiple 3x3



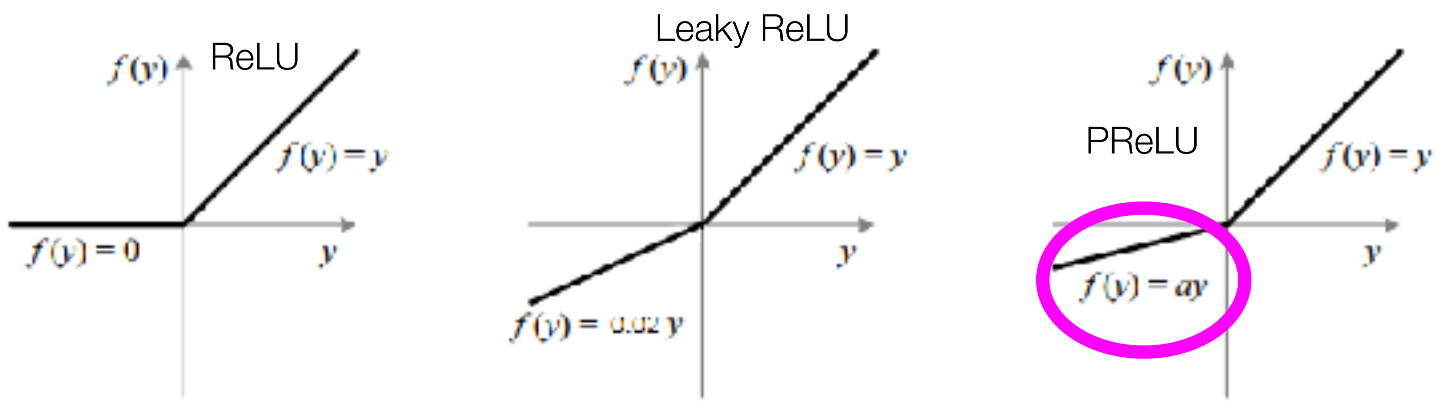


https://arxiv.org/pdf/1512.00567.pdf

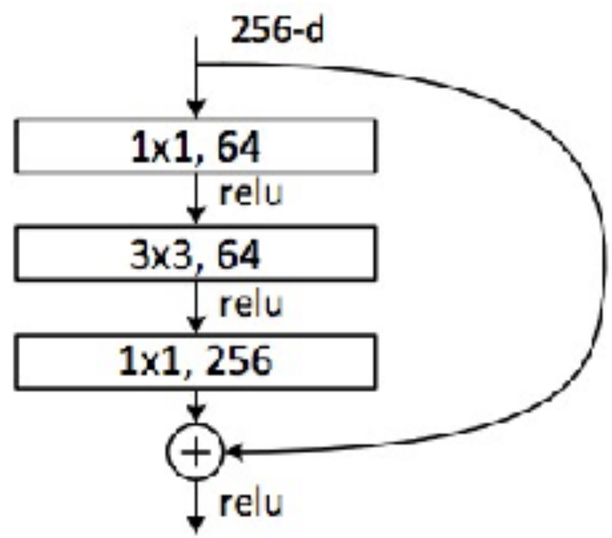
Types of CNN, 2015 December

- Major Contributions:
  - “ensembles” not strictly sequential
  - “bio-plausible” with feedback
- **ResNet**
  - Parametric ReLU

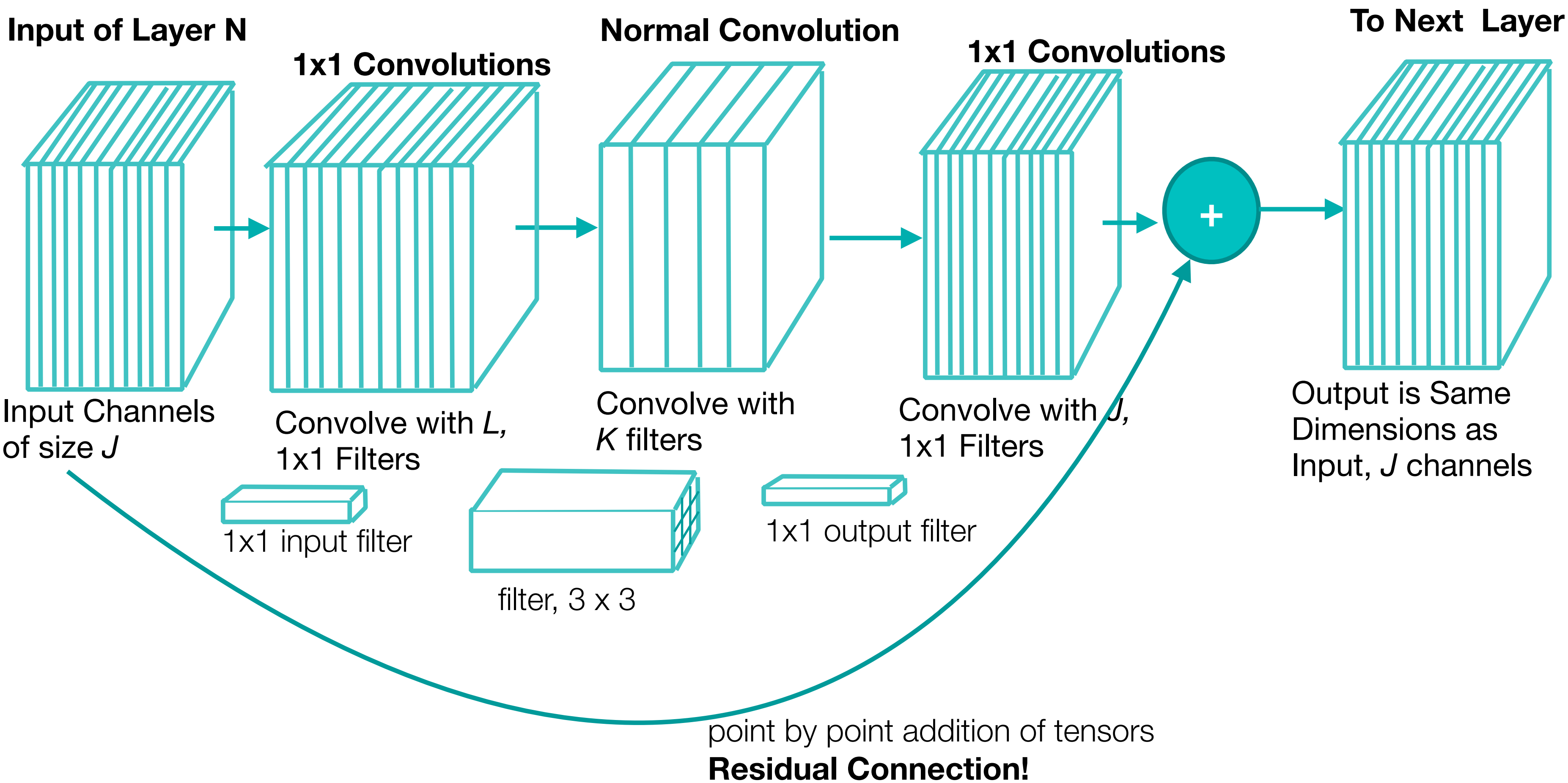
    PReLU: adaptive trained slope



- NiN: triple bypass layer
  - similar to bottelneck



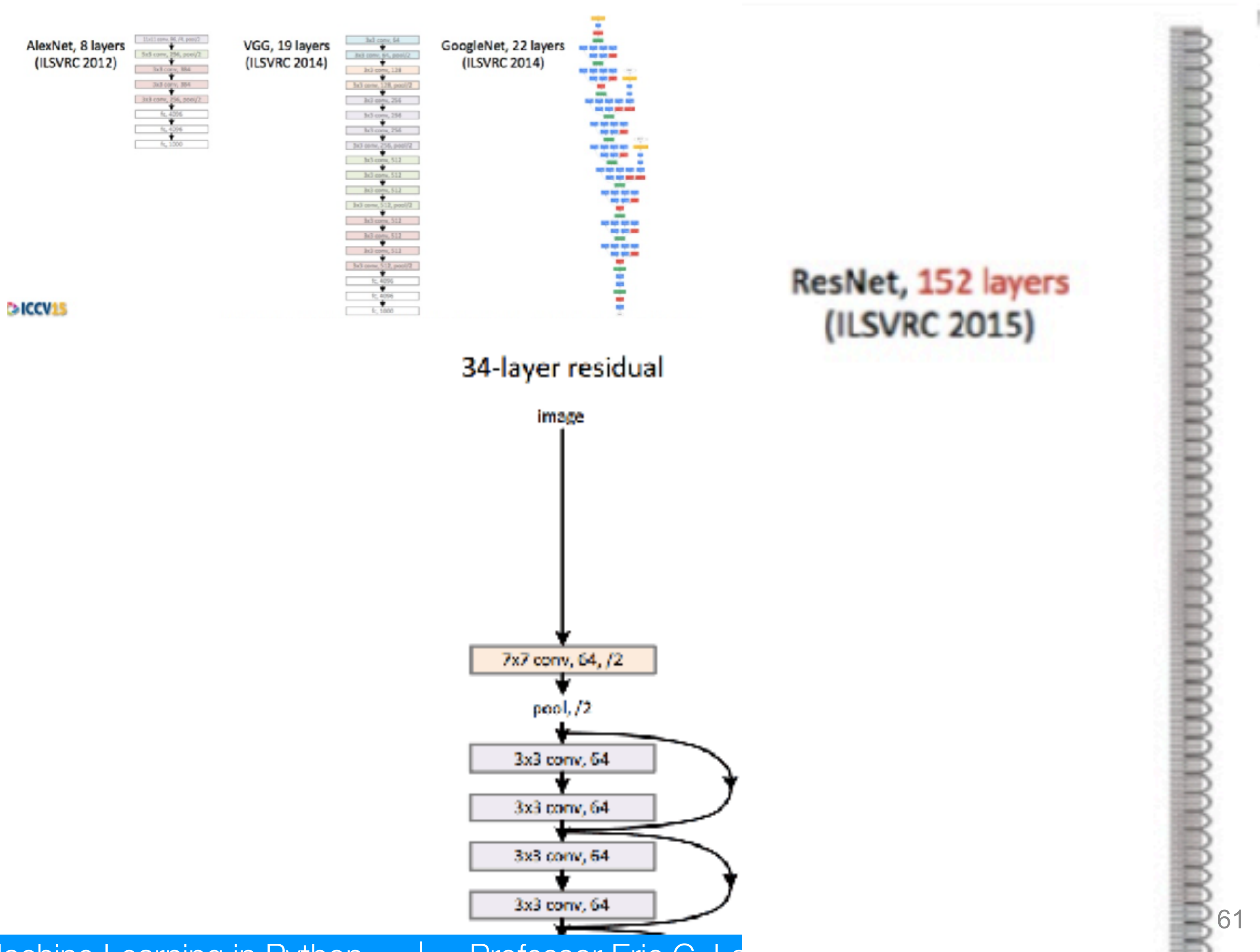
Residual Connection, expanded view



Back Propagation: Two paths, including one without ANY operations that cause the gradient to vanish…

How big are these networks?

How big are these networks?



Transition Period in Convolutional Networks

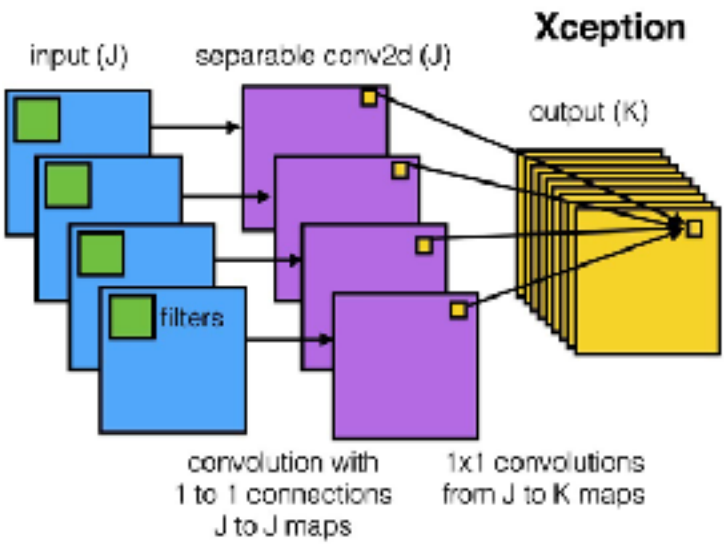
- 2012 - 2017:
  - Add more layers! 🤪
  - How can we train it even deeper? 🤓
  - Can we run out of memory? Let’s try! 🫣
- 2017-present:
  - How can we get similar performance with reduced parameters? 🧐
  - How should the number of parameters scale for competing resource? Is there an optimum scaling for a given set of resources? 📉

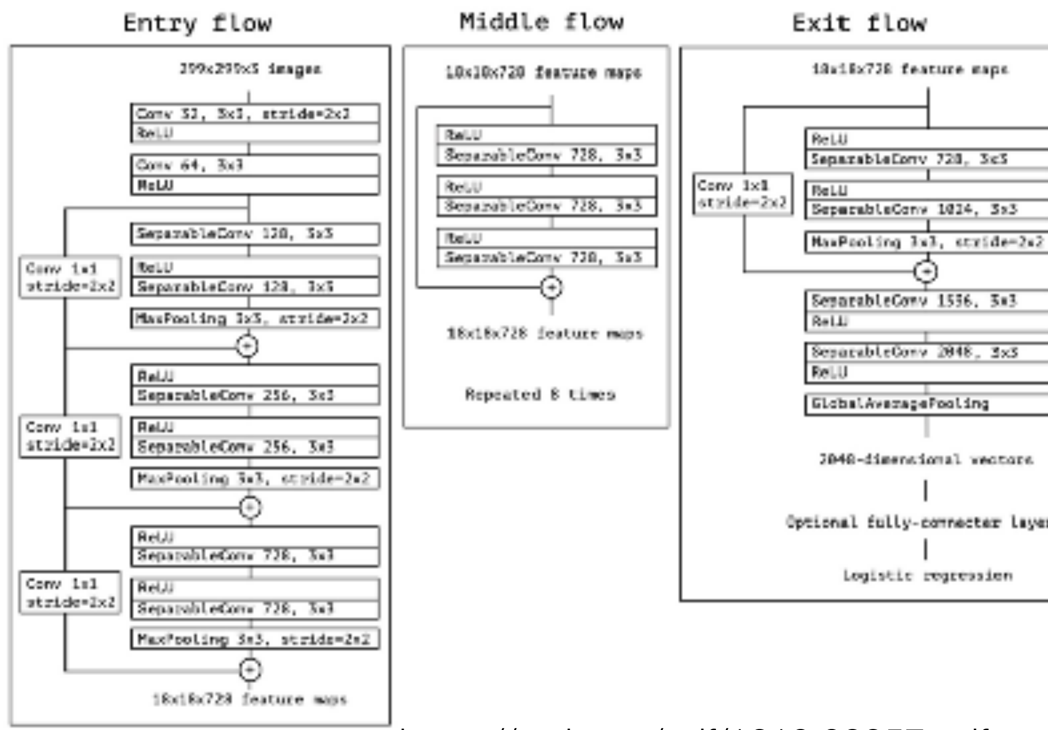
Types of CNN, 2017

Xception · Major Contributions:

- combining branching / residual blocks
- separable convolutions (fewer trainable params)

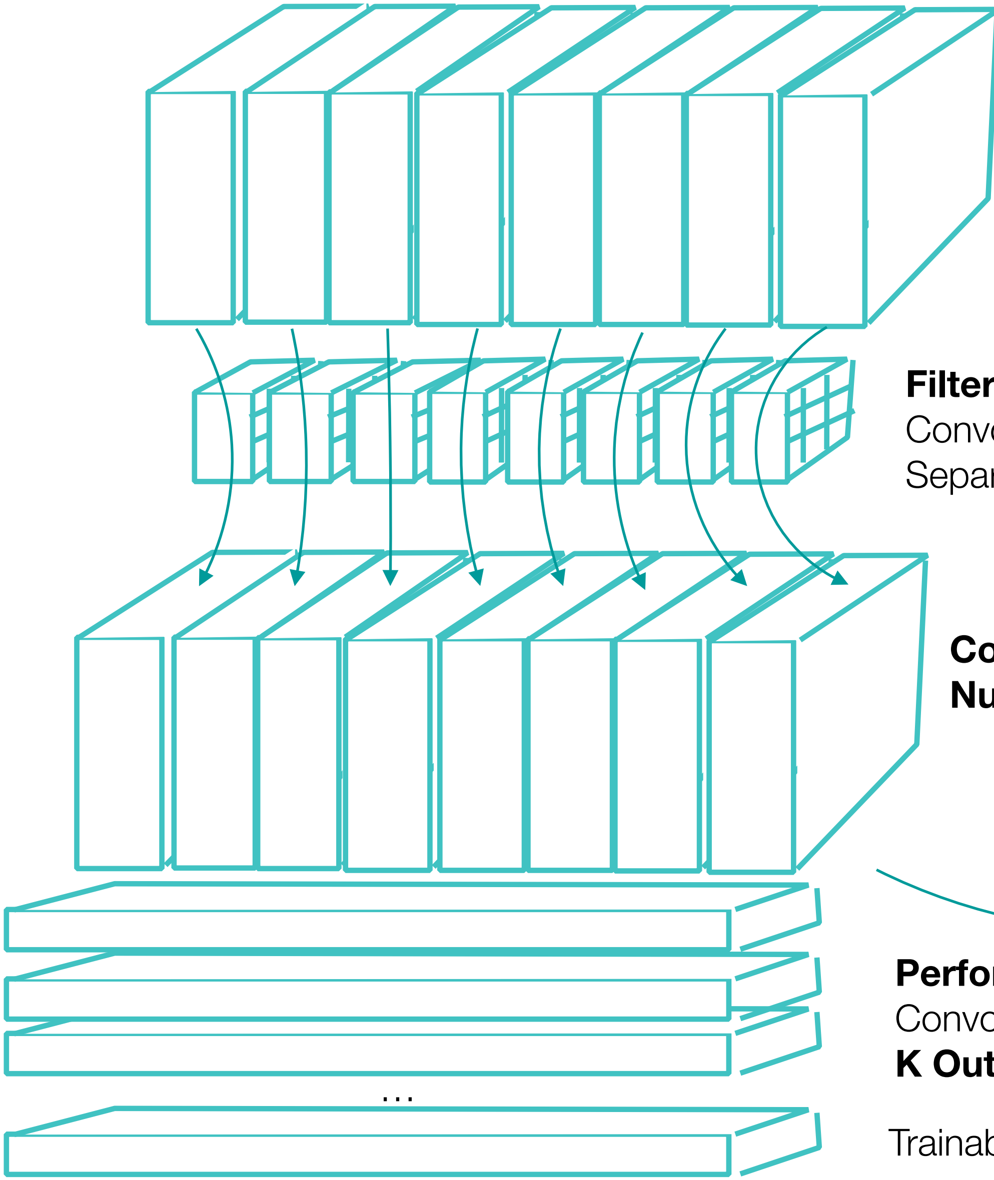
Francois Chollet
Google





https://arxiv.org/pdf/1610.02357.pdf

Separable Convolution Explanation



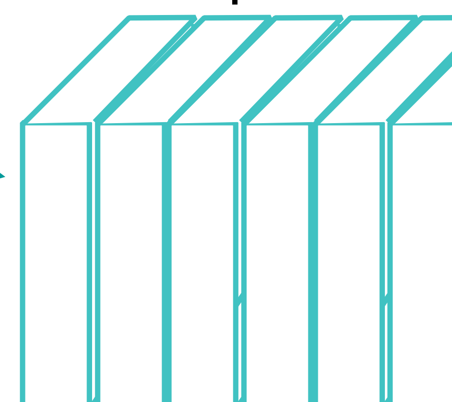
Inputs, From Layer N-1
Num Channels = J

Filters, Layer N
Convolve Each Channel Separately

Trainable params:
Same as one filter in traditional convolution!

Concat Outputs
Num Channels = J

K Outputs

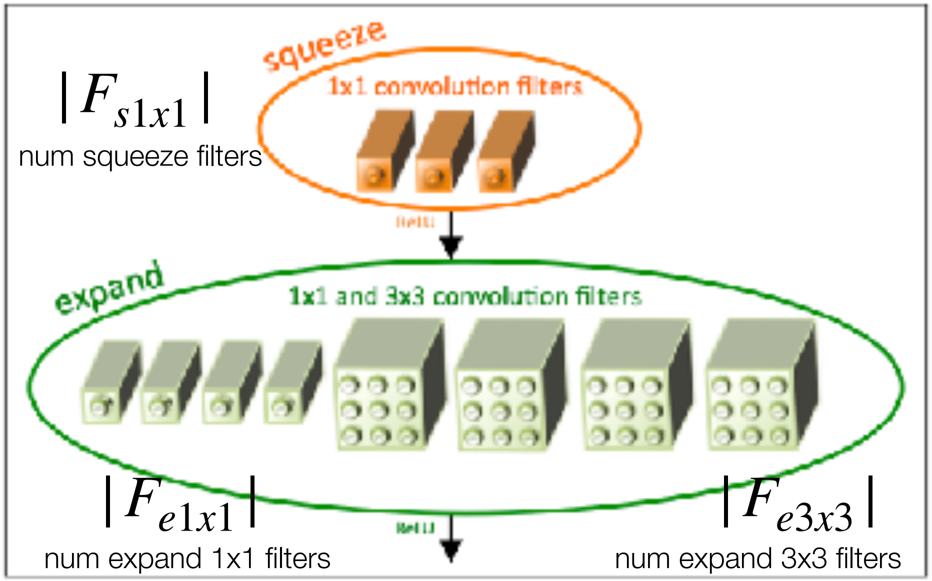


Perform K, 1x1 Traditional Convolutions
K Outputs

Trainable params: **K** x **J**

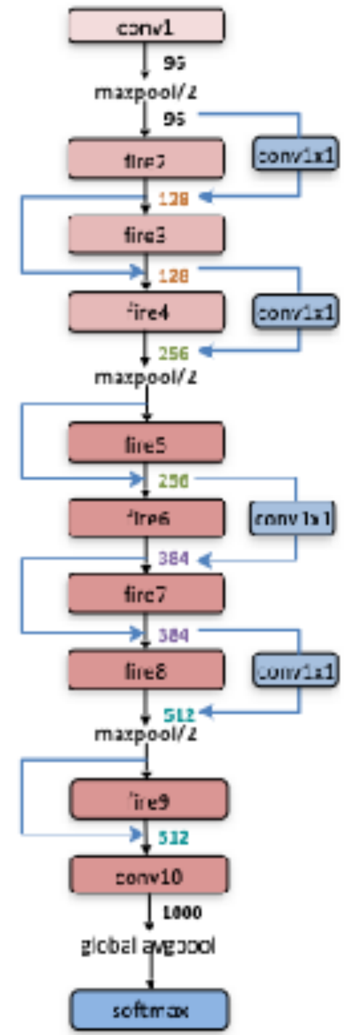
SqueezeNet (2018)

- Idea: squeeze and expand in each layer
  - Use mostly 1x1 filters
  - downsample later in network



$$SR = \frac{|F_{s1x1}|}{|F_{e1x1}| + |F_{e3x3}|}$$

$$PCT_{3x3} = \frac{|F_{e3x3}|}{|F_{e1x1}| + |F_{e3x3}|}$$



SQUEEZENET: ALEXNET-LEVEL ACCURACY WITH 50X FEWER PARAMETERS AND <0.5MB MODEL SIZE

Forrest N. Iandola[1], Song Han[2], Matthew W. Moskewicz[1], Khalid Ashraf[1], William J. Dally[2], Kurt Keutzer[1]
[1]DeepScale* & UC Berkeley [2]Stanford University
{forresti, moskewcz, kashraf, keutzer}@eecs.berkeley.edu
{songhan, dally}@stanford.edu

In paper:

- Good SR = 12.5% up to 100%
- Good PCT_{3x3} from 25% up to 100%

Start with so

Observation 1 – Scaling up any
width, depth, or resolution improve
racy gain diminishes for bigger mo

Observation 2 – In order to purs
efficiency, it is critical to balance al
width, depth, and resolution during

Depth Scalin

Resolution Scaling: If we use larger resolu

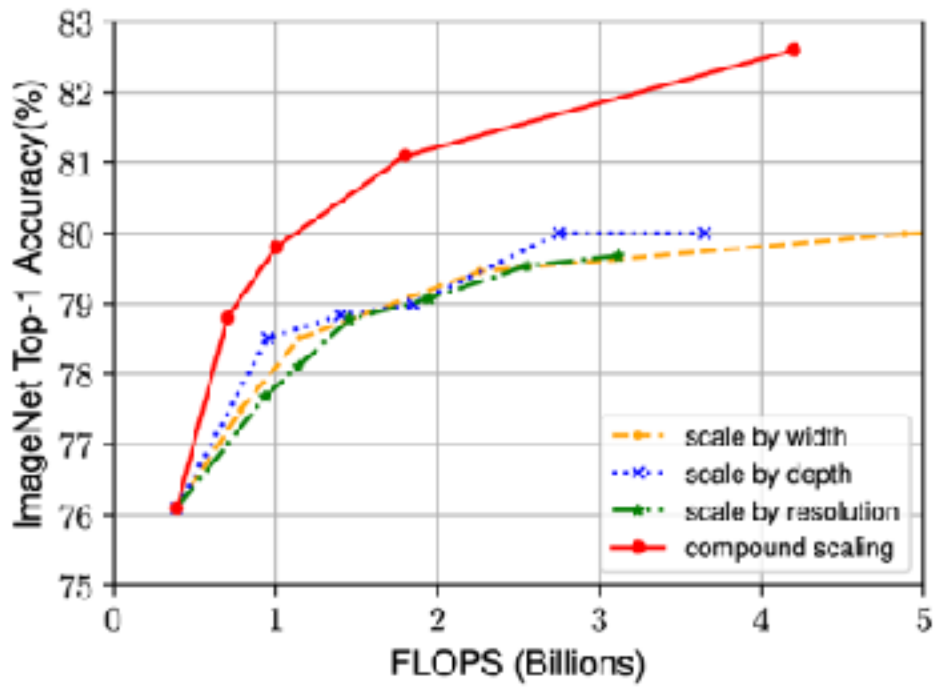


Figure 8. **Scaling Up EfficientNet-B0 with Different Methods.**

depth: $d = \alpha^{\phi}$

width: $w = \beta^{\phi}$

res.: $r = \gamma^{\phi}$

s.t. $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$

$\alpha, \beta, \gamma \geq 1$

ϕ user specified scaling coefficient

$$\alpha = 1.2$$
$$\beta = 1.1$$
$$\gamma = 1.15$$

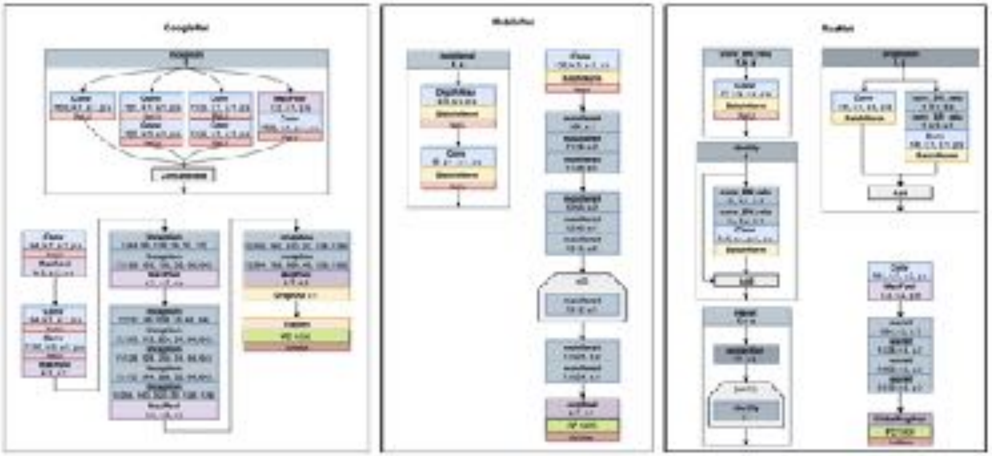
optimal values found in paper!

where α, β, γ are constants that can be determined by a small grid search. Intuitively, ϕ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution re-

https://arxiv.org/pdf/1905.11946v5.pdf

Even more Convolutional Neural Networks
…in TensorFlow
…with Keras

Self Guided Demo



`12. More Advanced CNN Techniques as TFData.ipynb`

Next Time:

- Intro to Sequential Neural Network Architectures
  - Word Embeddings, 1D CNNs, Transformers
  - Ethics by Case Study