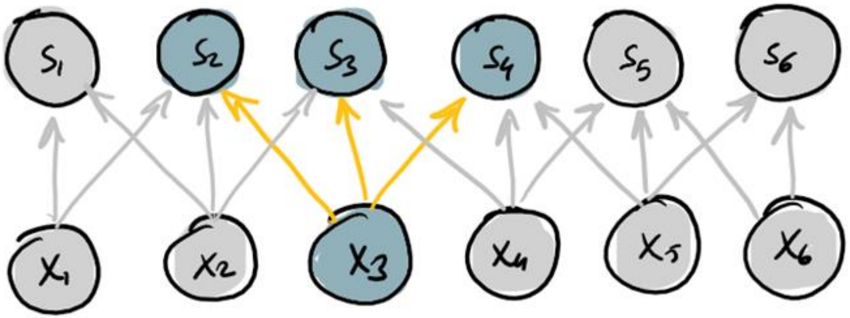
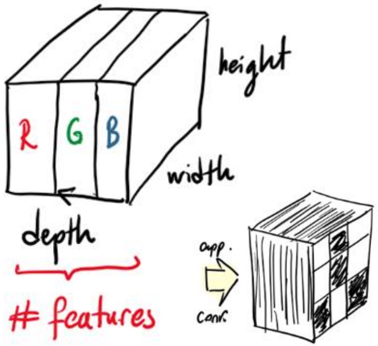
Karlsruhe Institute of Technology

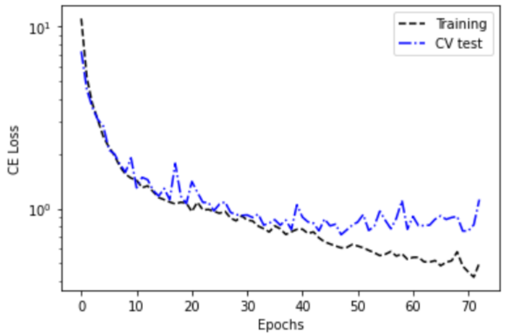
Data Driven Engineering II: Advaced Topics

Image processing and analysis

Institute of Thermal Turbomachinery
Prof. Dr.-Ing. Hans-Jörg Bauer







Term Projects

Welcome to DDE II projects!

If you are interested in the group projects for fun or planning to take the final exam for credits, you need to register to a topic before 14.05.021.Note that each topic has a number of maximum participants. You may find the details in Lecture 1.

Particle Image Density Analysis in PIV Recordings

An object detection study for PIV analysis

Free places: 1

Physical interpretation of LCSs

Data driven model discovery in air blast atomizers

Free places: 5

Time resolved flow field analysis in film cooling

PIV data will be used for flow analysis.

Free places: 5

Others

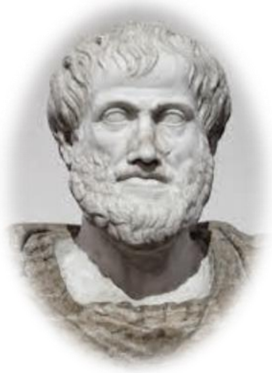
for HPC access

Period of Event: Today - 14. May 2021

Outline of the week :

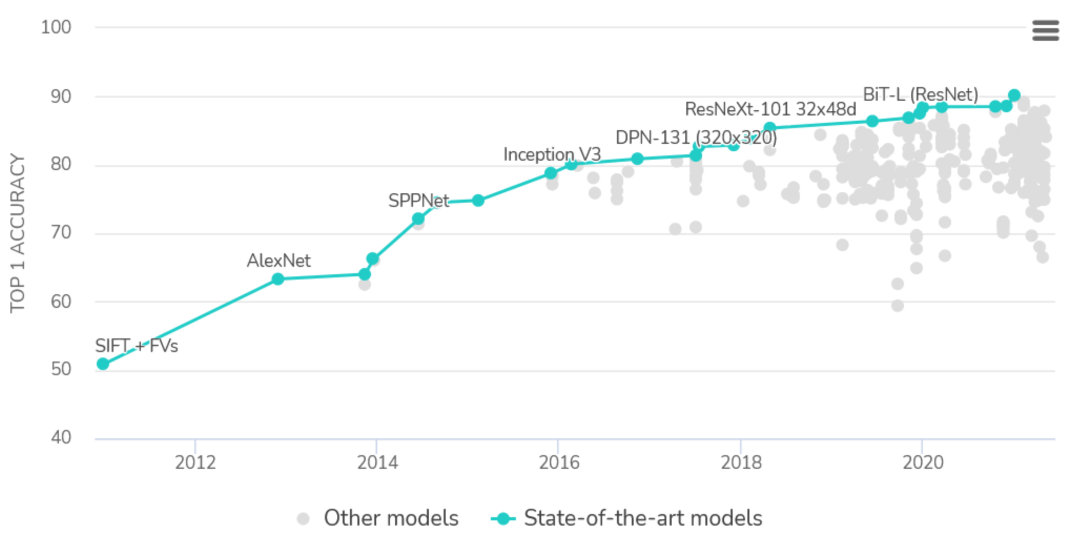
Conv. Neural Networks

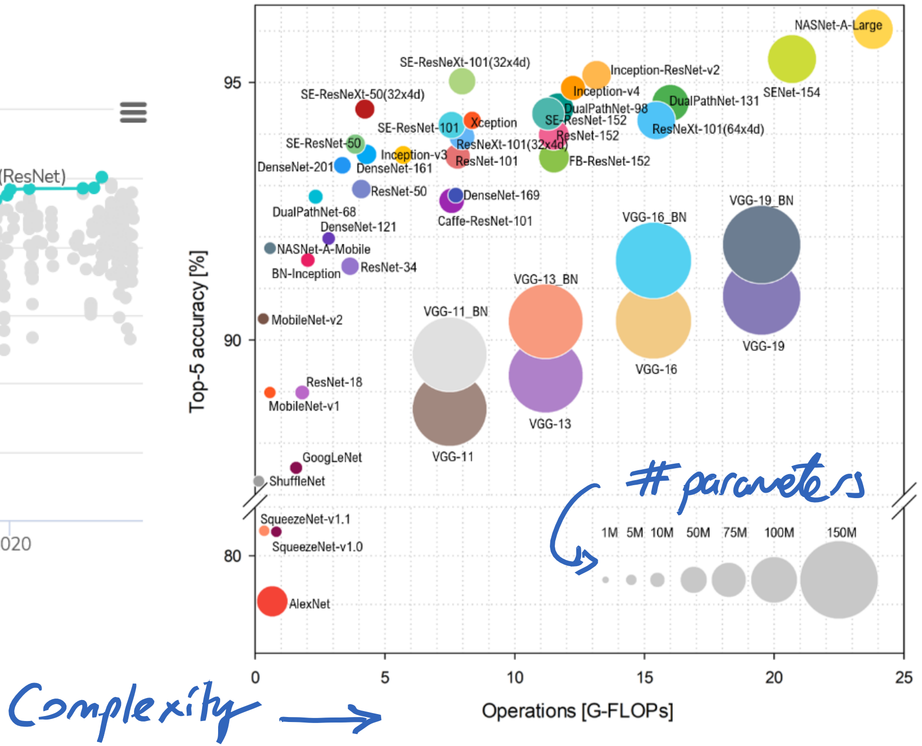
* What is CNN ?
* Why convolution is useful ?
* Where is it useful ?
* CNN – How does it work ?
* "Hall of Fame" : Popular Arch.
* Transfer Learning with CNN



"The soul never thinks without a picture."

where to begin ...





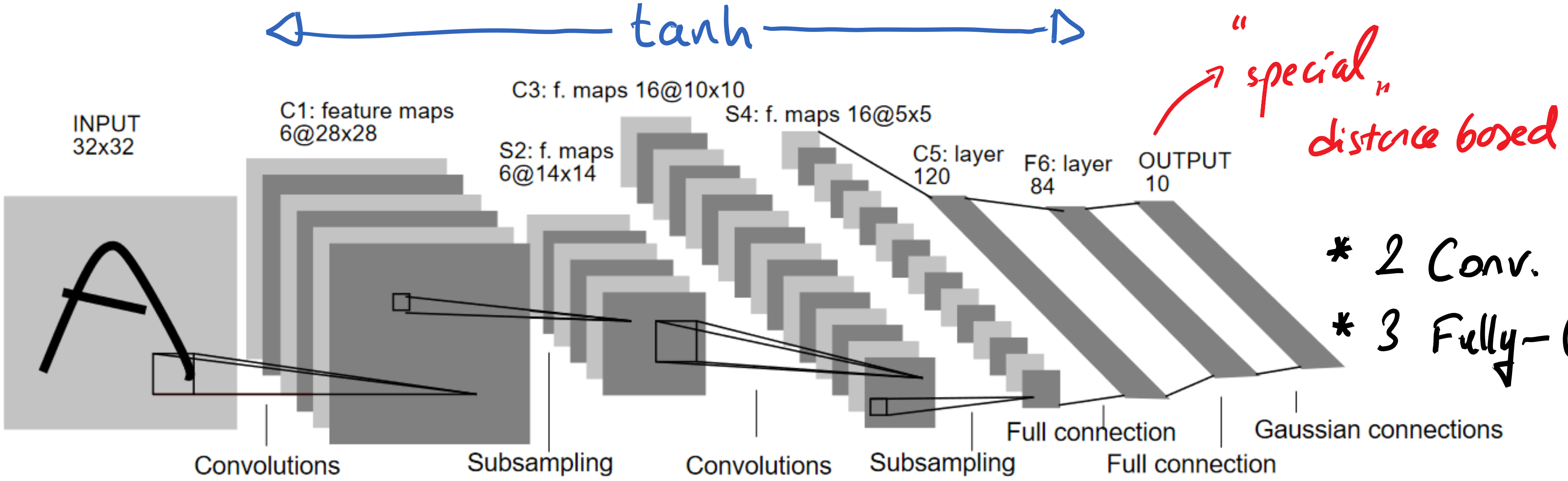
what is different?

Karlsruhe Institute of Technology

* Increase in depth
* increase in width ⇒ more filters ⇒ filter hacks
* Regularization & fine tuning
* Training hacks
* Data augmentation & tr. learning
* Increased image resolution

Who does not like classics?

LeNet-5 (1998)

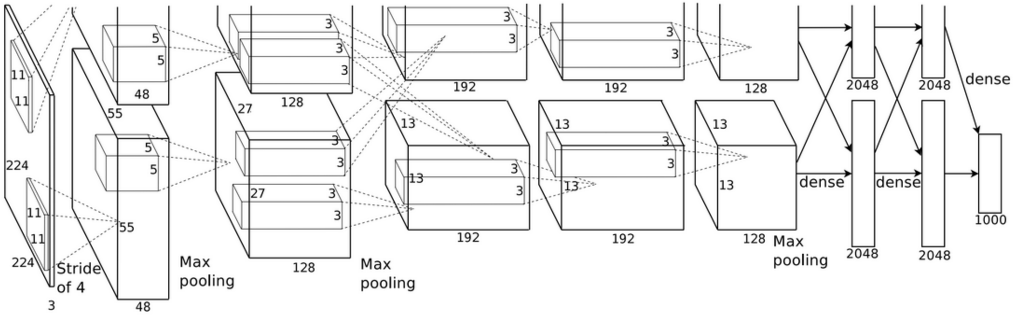


* 2 Conv.
* 3 Fully-Con.

One small step for men, one giant leap for CNN

AlexNet (2012)

* Deeper := 5 Conv. + 3 Fully Con.
  ↳ 60 M parameters
* Stack conv. layers
* Use ReLu + Dropouts
* Data agumentation

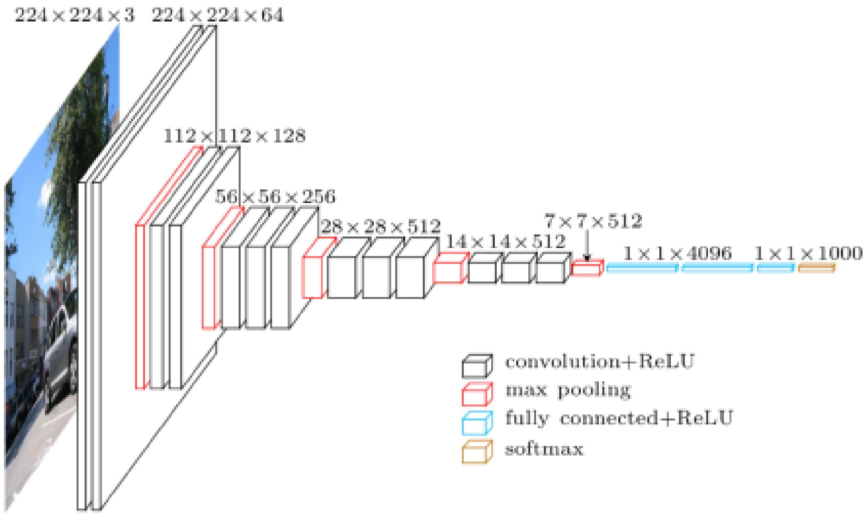


Deep Learning ...

VGG-16 (2014)

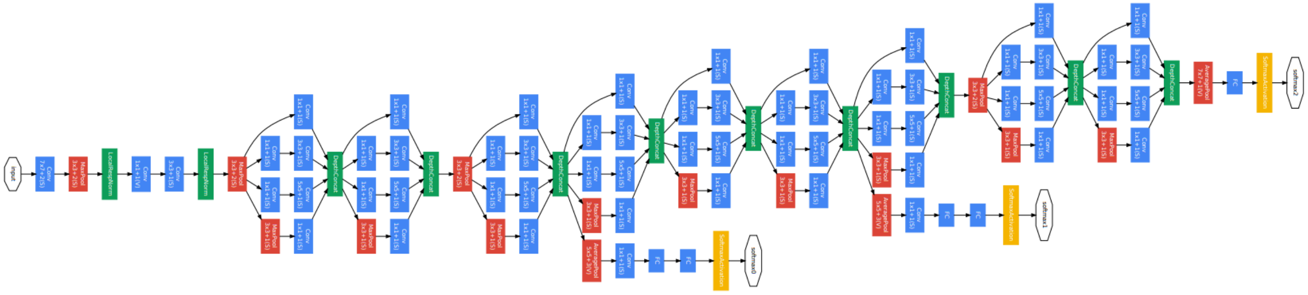
* 13 Conv. + 3 Fully-connected.
* 138 M parameters
* ReLu + Smaller kernels (2x2, 3x3)

↳ Deeper variant "VGG-19"



Things get multi-layered:

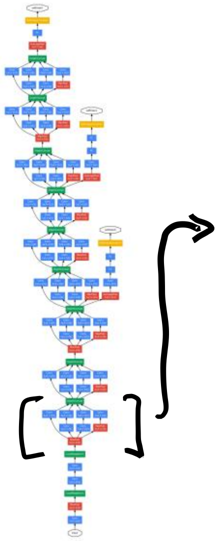
inception-v1 // GoogLeNet (2014)

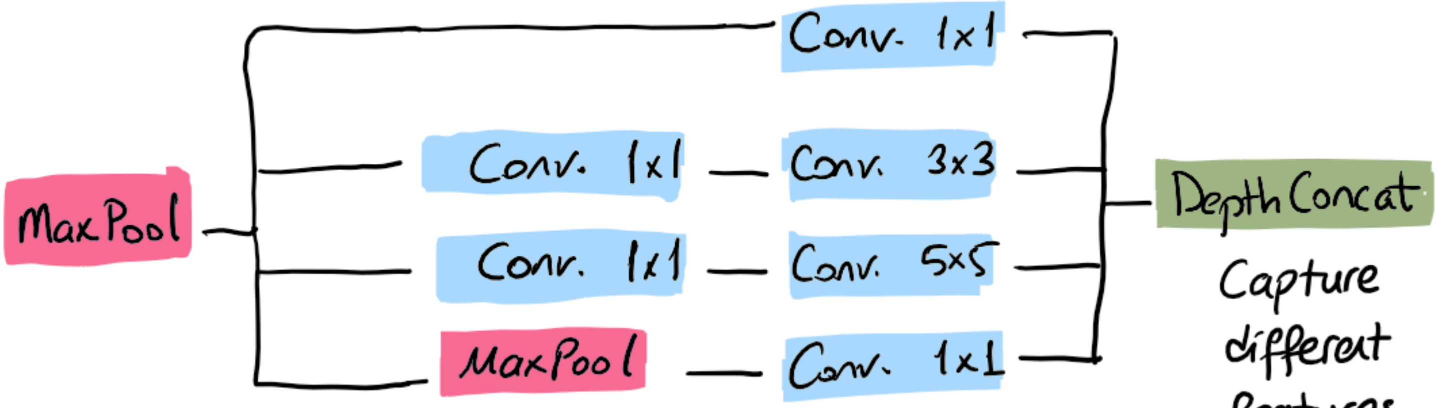


* 22 Layers ⇒ 5M parameters
* inception module ⇒ "network in network"
  ↳ enables much deeper network

Things get multi-layered:

inception-v1 // GoogLeNet (2014)

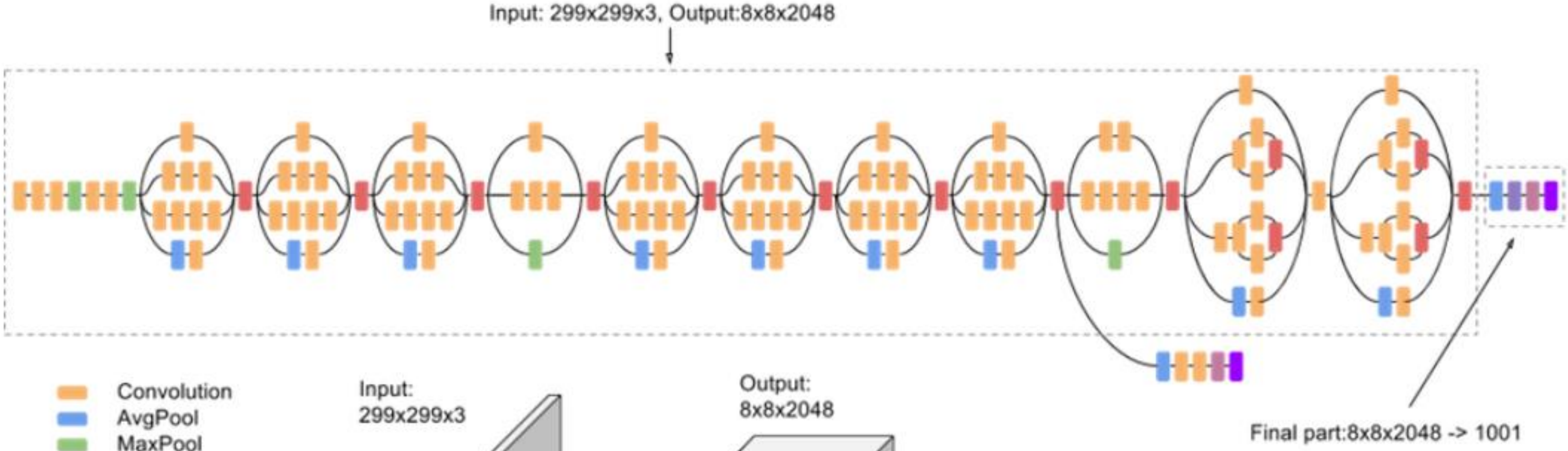


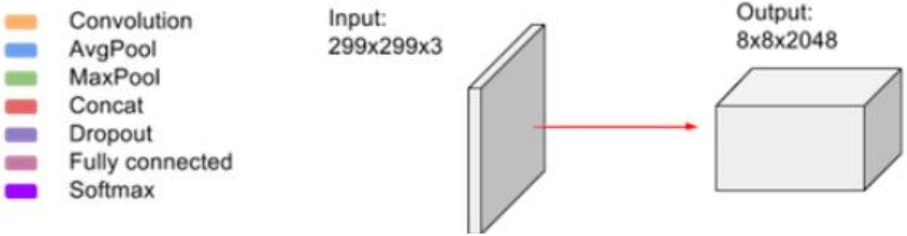


Capture different features

Inception module

Things get multi-layered: v3 (2015)





⇨ 24M parameters

+ Batch Norm.

- [7x7] ⇒ [3x3]
- [5x5] ⇒ [3x3]
- [nxn] ⇒ [1xn, nx1]

(v4 → 2016)

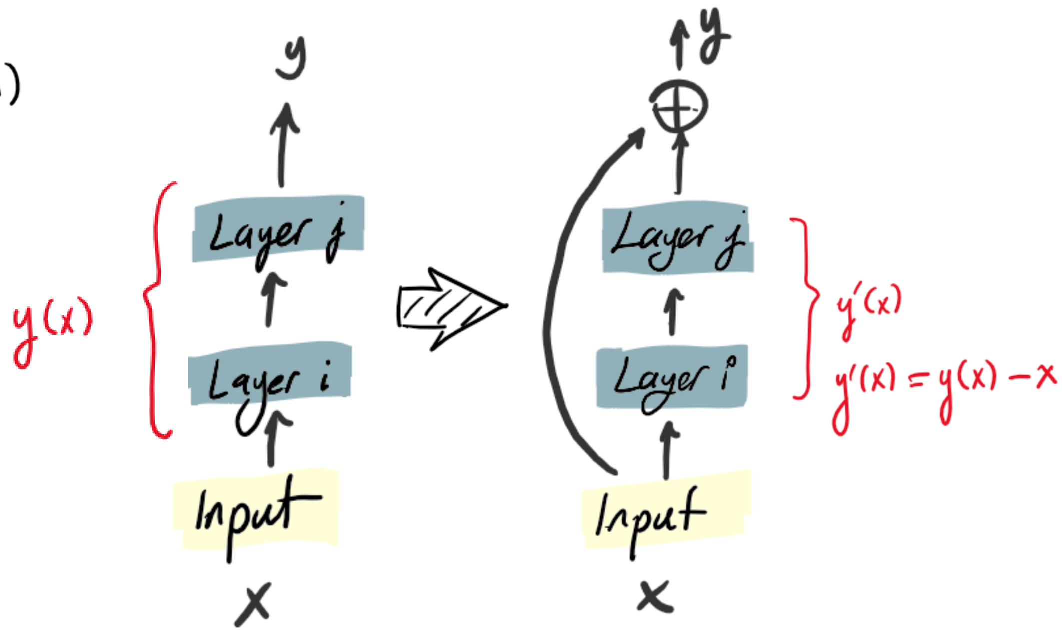
A smart touch: ResNet (2015)

* much deeper network → 152 layers
  ↳ fewer parameters (26M)

Exp./Van. Gradient problem

Skip Connections
+ BatchNorm

* Residual learning
* LSTM ?



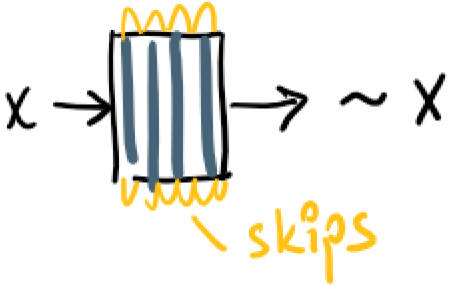
A smart touch: ResNet (2015)

Karlsruhe Institute of Technology

How does it help training?

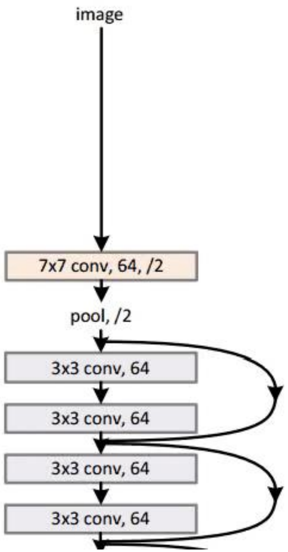
① when network is initialized;

$\bar{w} \rightarrow \emptyset$



} identity function ↑↑↑ training

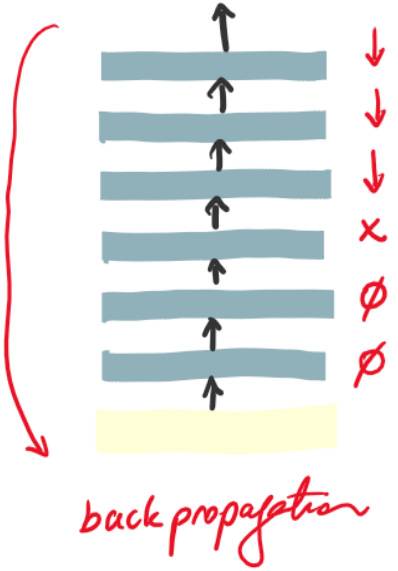
34-layer residual

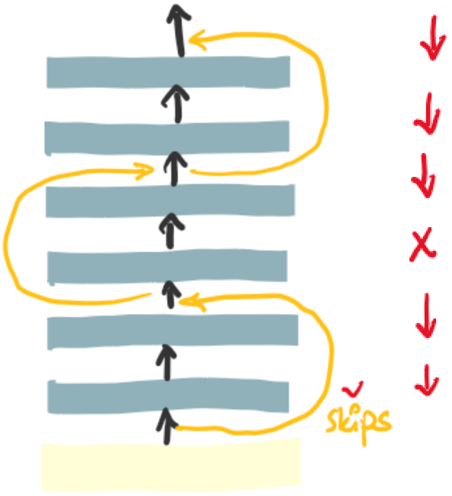


A smart touch: ResNet (2015)

How does it help training?

② Learning bottlenecks





information can propagate!

It looks nice. Lets use it everywhere !

DenseNet 2017

* use skip connections in dense blocks → Re-use features with less parameters
* use "concatenate" rather than adding ! Needs proper padding

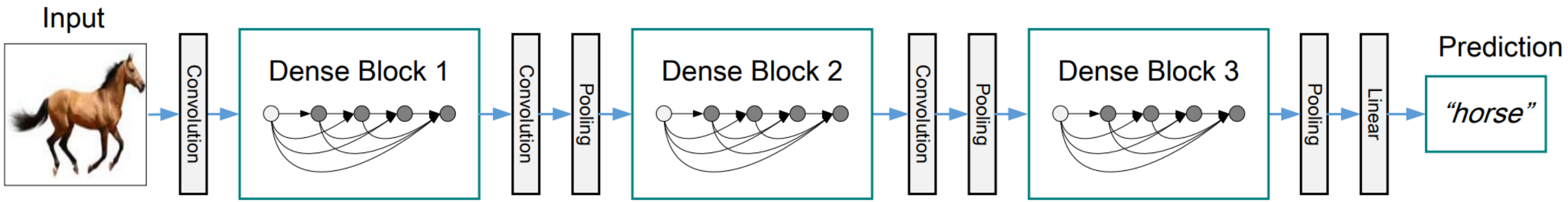


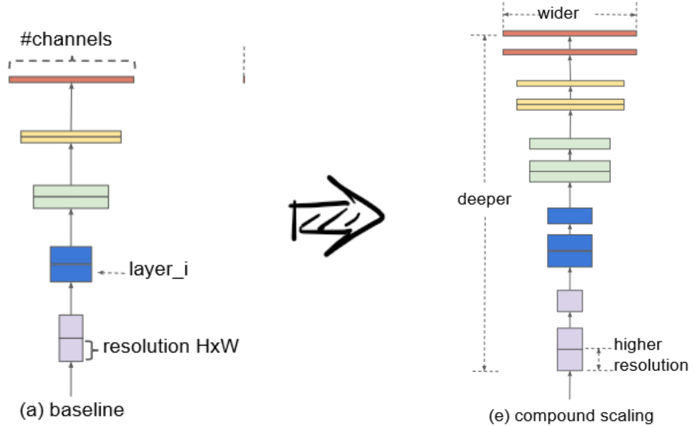
Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

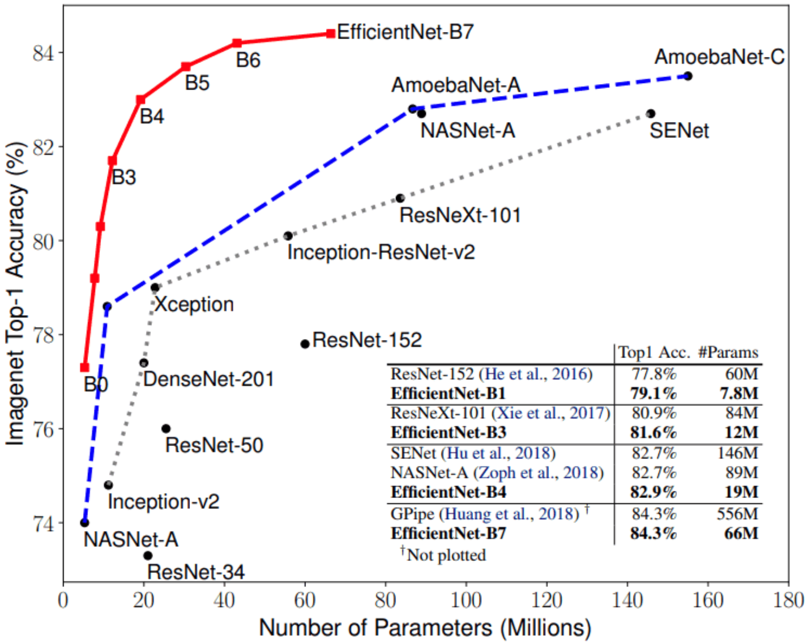
EfficientNet: "Rethinking model scaling"

* balancing network { depth, width, resolution }

↓

scaling method





EfficientNet: "Rethinking model scaling"

KIT Karlsruhe Institute of Technology

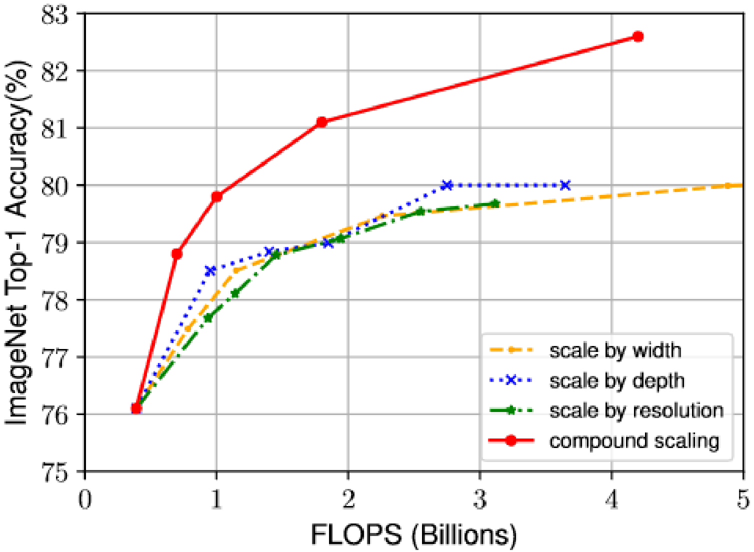
* balancing network { depth, width, resolution
  ↓
  scaling method

- depth = $d = \alpha^{\phi}$; $\alpha \beta^2 \gamma^2 \approx 2$ (FLOPs)
- width = $w = \beta^{\phi}$
- res. = $r = \gamma^{\phi}$; $\alpha, \beta, \gamma > 1$

(i) $\phi = 1$; do grid search on α, β, γ
eg. (1.2, 1.2, 1.15)

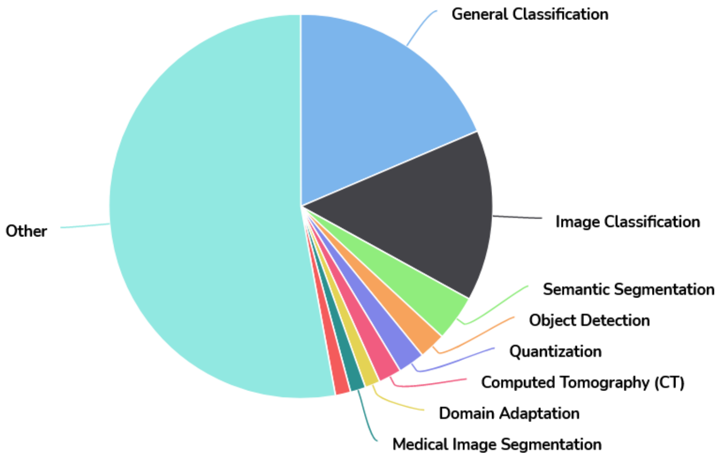
(ii) Fix α, β, γ ; scale ϕ with respect to hardware

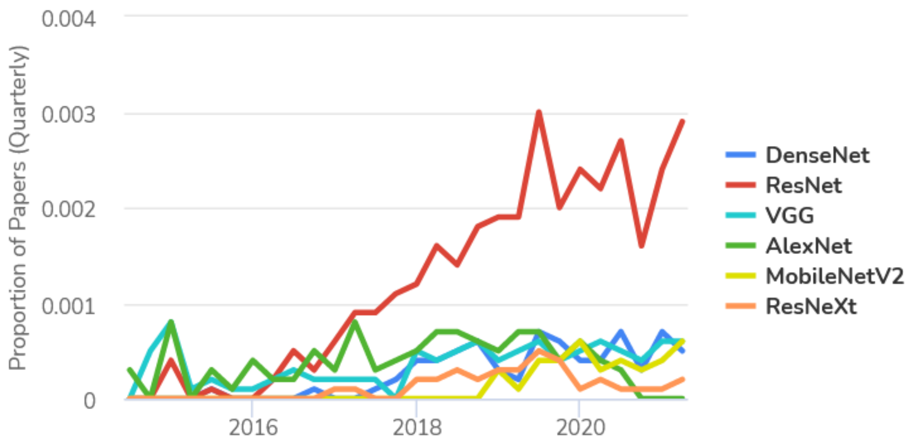
⇒ Start small; gradually scale it



Too many options ...

* Models including CNN ~ 100 accessible
* Utilized in various fields



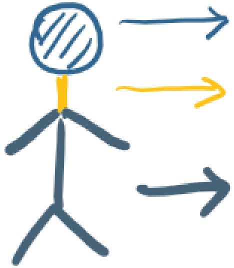


Case study: Object Detection :

* {Classify + Location + many objects}

Model anatomy reflects this aim !

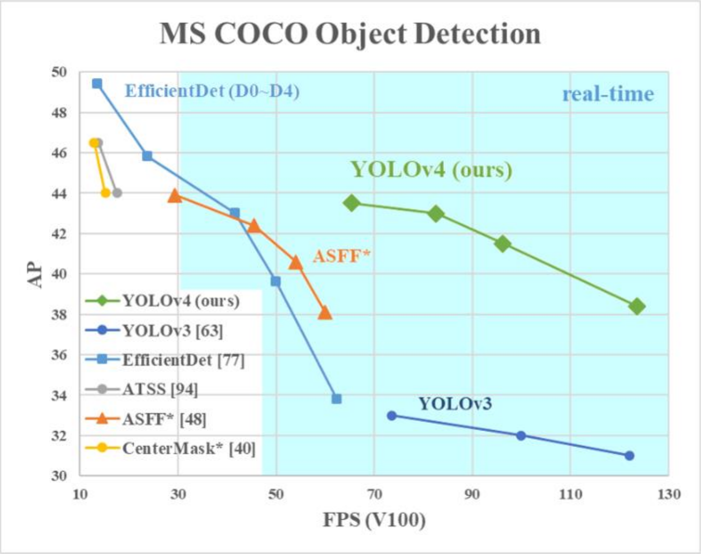
* YOLO := "You only look once"



Head : Decision-unit

Neck : collect features

Backbone : extracts features

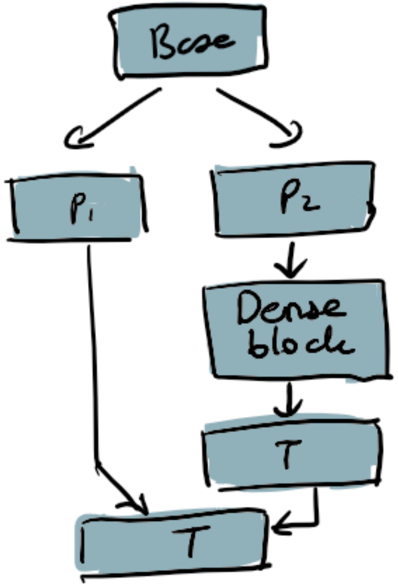


Case study: Object Detection:

Backbone: Dense block + Cross-Stage-Partial connections

+ $\underbrace{\text{Darknet 53}}$

higher acc. > Resnet



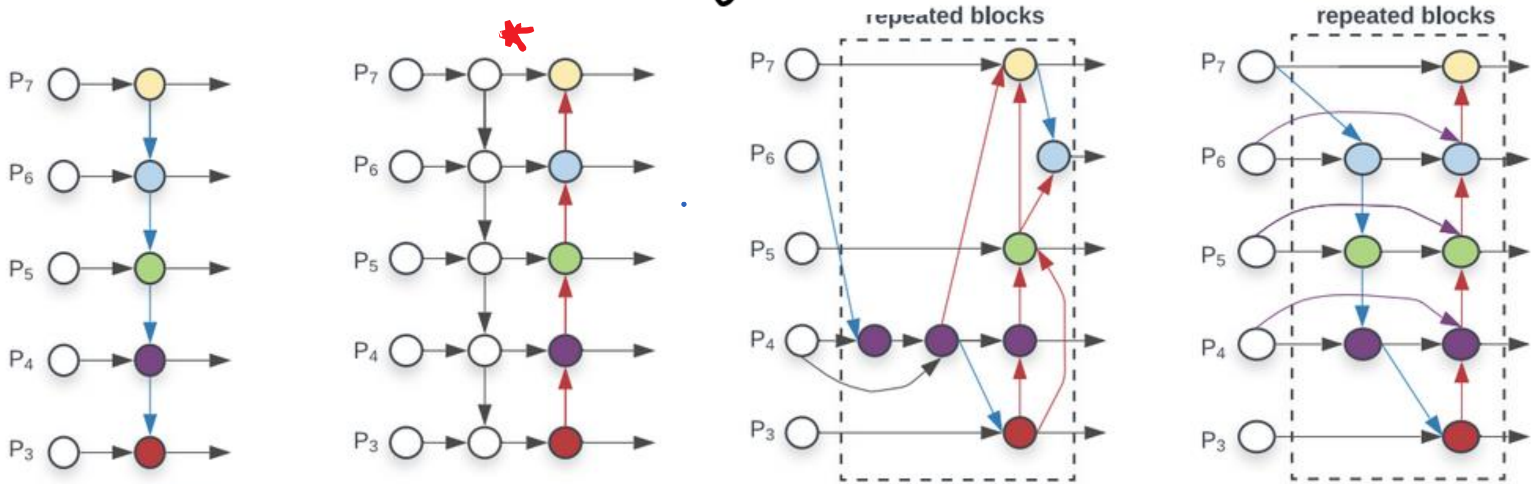
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3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

Case study: Object Detection:

Neck

* Combination of features managed here

P → feature layer



(a) FPN (b) PANet (+ SPP) (c) NAS-FPN (d) BiFPN

Eff. Net

Case study: Object Detection:

Head

* YOLOv4 ← YOLOv3
* anchor-based detection
* 3 levels of granularity

Bag of Specials

* Mish activation
* Cross Stage Partial connections
* Multi-input-weighted residual connections

Bag of Freebies

* DropBlock regularization
* Class label smoothing
* CutMix & Mosaic data aug.

Case study: Object Detection:

Anchor Boxes

(i) Create 'many' boxes for each predictor

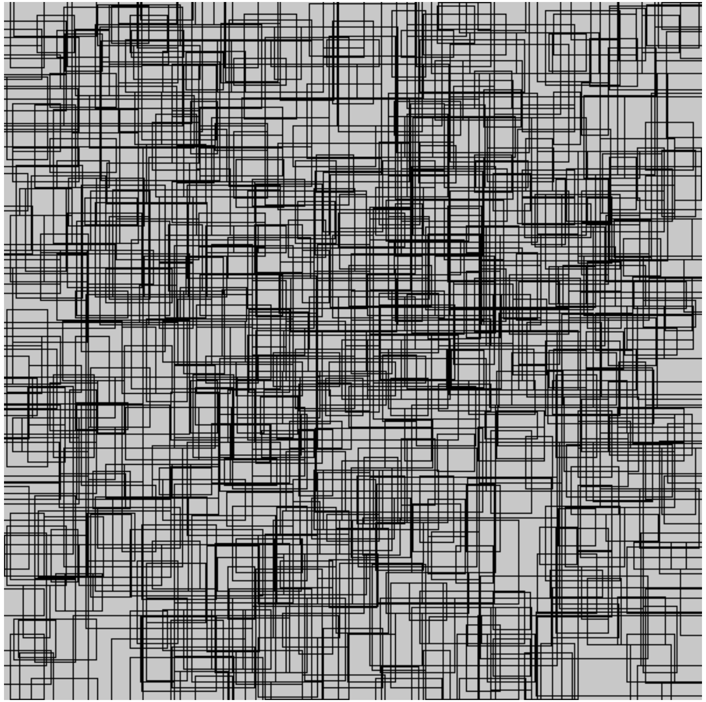
(ii) For each box, calculate which objects bounding box has the highest overlap/non-overlap ratio. (IOU)

(iii) If highest (IOU) > 50% ⇒ "Detect object"

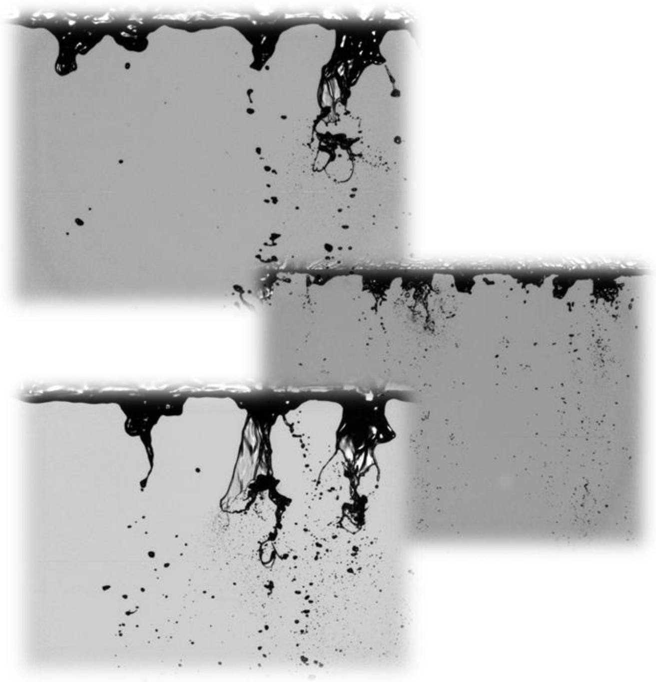
(iv) ~40% – 50% ⇒ ambiguous ⇒ "not learn from this case"

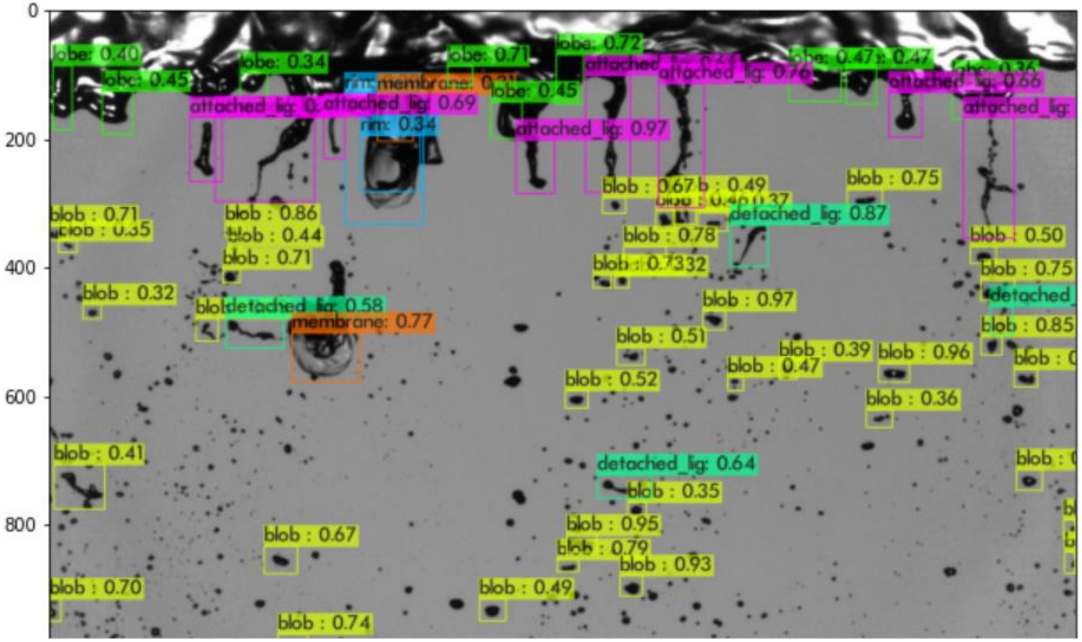
(v) < 40% ⇒ No object here !?

? Box dimensions ⇒ YOLO; K-means clustering on training data



Case study: Object Detection :





colab

Methods > Computer Vision

Computer Vision

544 methods • 43959 papers with code

Image Feature Extractors

Method	Papers with code
Convolution	8632 papers with code
1x1 Convolution	2692 papers with code
Grouped Convolution	452 papers with code
Pointwise Convolution	450 papers with code
Depthwise Convolution	441 papers with code

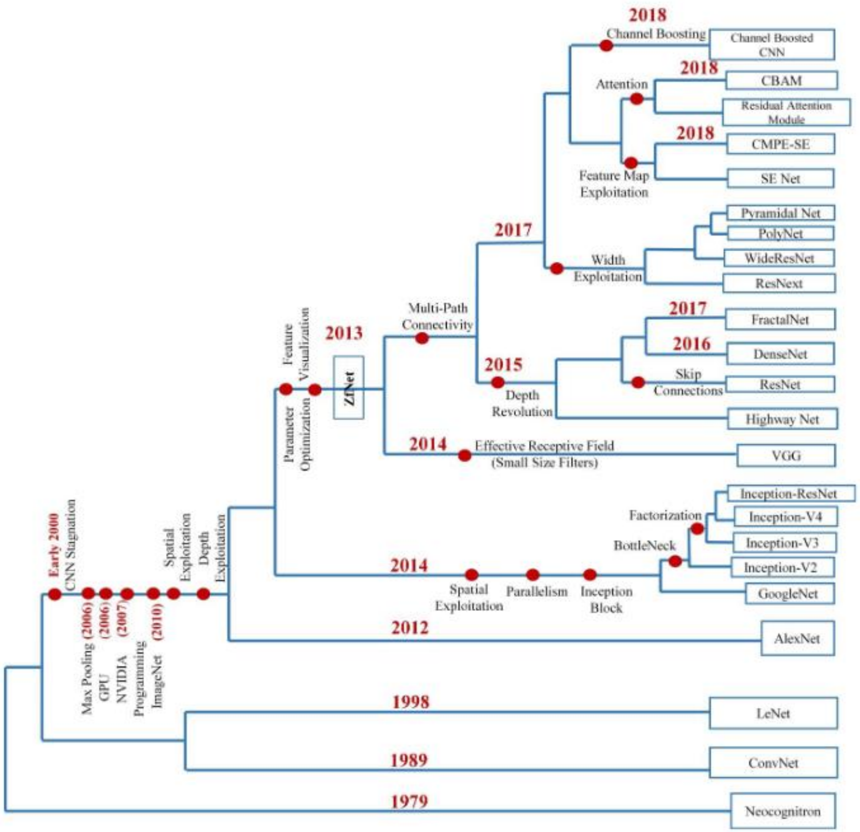
See all 39 methods

Convolutions

Method	Papers with code
Convolution	8632 papers with code
1x1 Convolution	2692 papers with code
Grouped Convolution	452 papers with code
Pointwise Convolution	450 papers with code
Depthwise Convolution	441 papers with code

Convolutional Neural Networks

METHOD	YEAR	PAPERS
ResNet	2015	1139
VGG	2014	293
AlexNet	2012	278
DenseNet	2016	248
MobileNetV2	2018	144
ResNeXt	2016	107
GoogLeNet	2014	106
EfficientNet	2019	101



Karlsruhe Institute of Technology

Artificial Intelligence Review

A Survey of the Recent Architectures of Deep Convolutional Neural Networks

ImageNet Challenge :