AI Training Sessions

Introduction to Convolutional Neural Networks

by Bartlomiej Borzyszkowski

AGENDA

- Computer Vision - problems
- Architecture of CNN
- What happens inside?
- Popular architectures
- Popular datasets
- What’s next?

____

- Hands-on

COMPUTER VISION

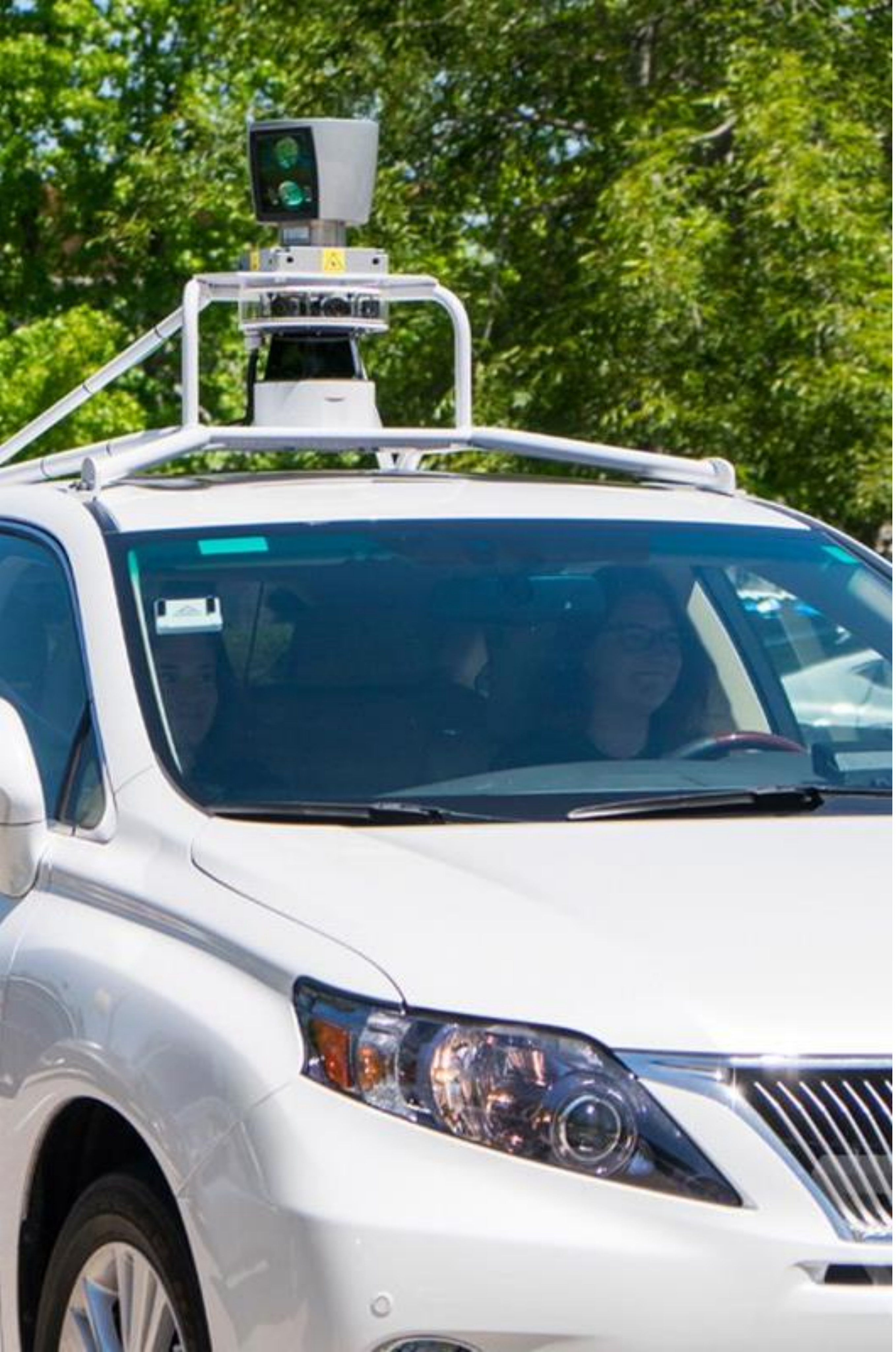
“

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. From the perspective of engineering, it seeks to **automate tasks** that the **human visual system** can do.

~Wikipedia

APPLICATIONS

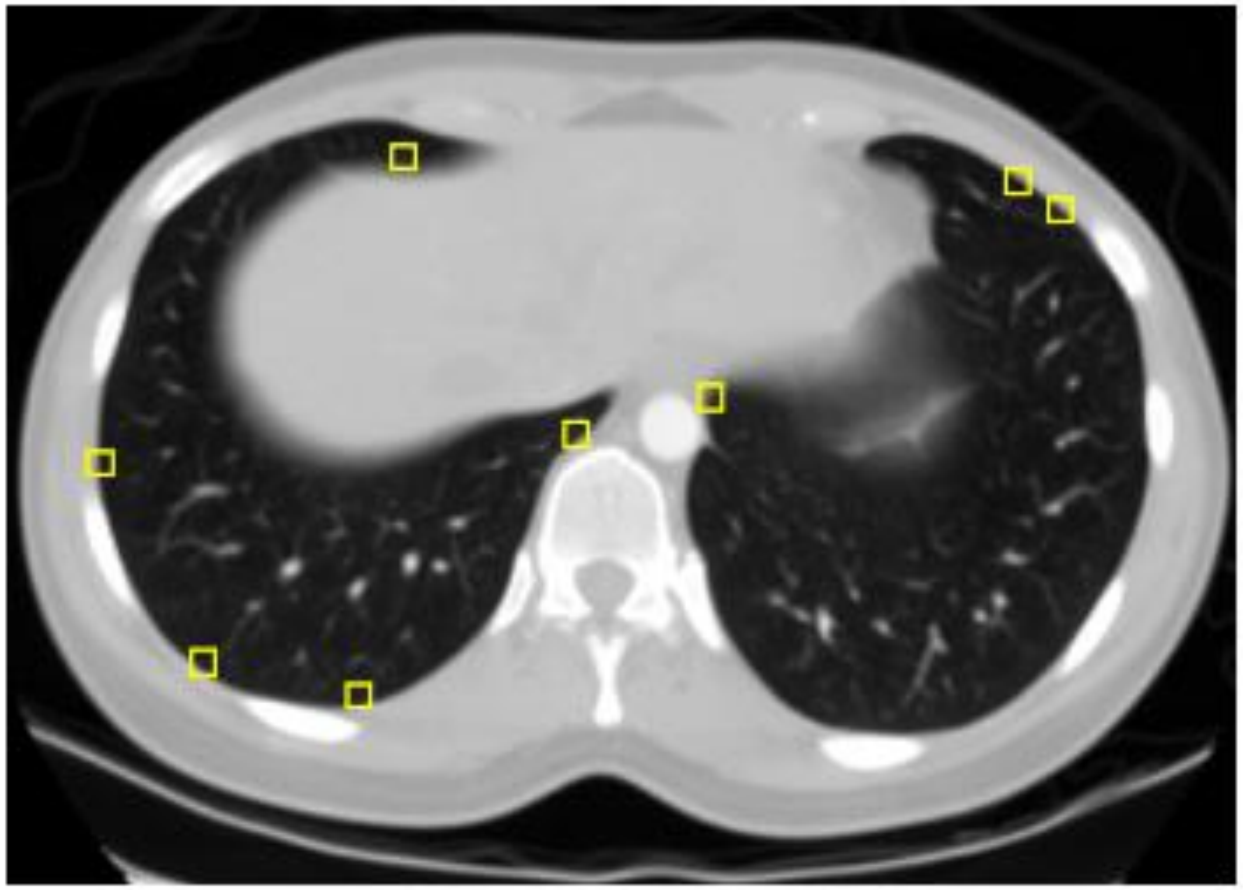
- agriculture
- **augmented reality**
- **autonomous vehicles**
- **biometrics**
- character recognition
- forensics
- quality inspection
- face recognition
- gesture analysis
- geoscience
- image restoration
- **medical image analysis**
- pollution monitoring
- process control
- remote sensing
- robotics
- security and surveillance
- transport
- ...

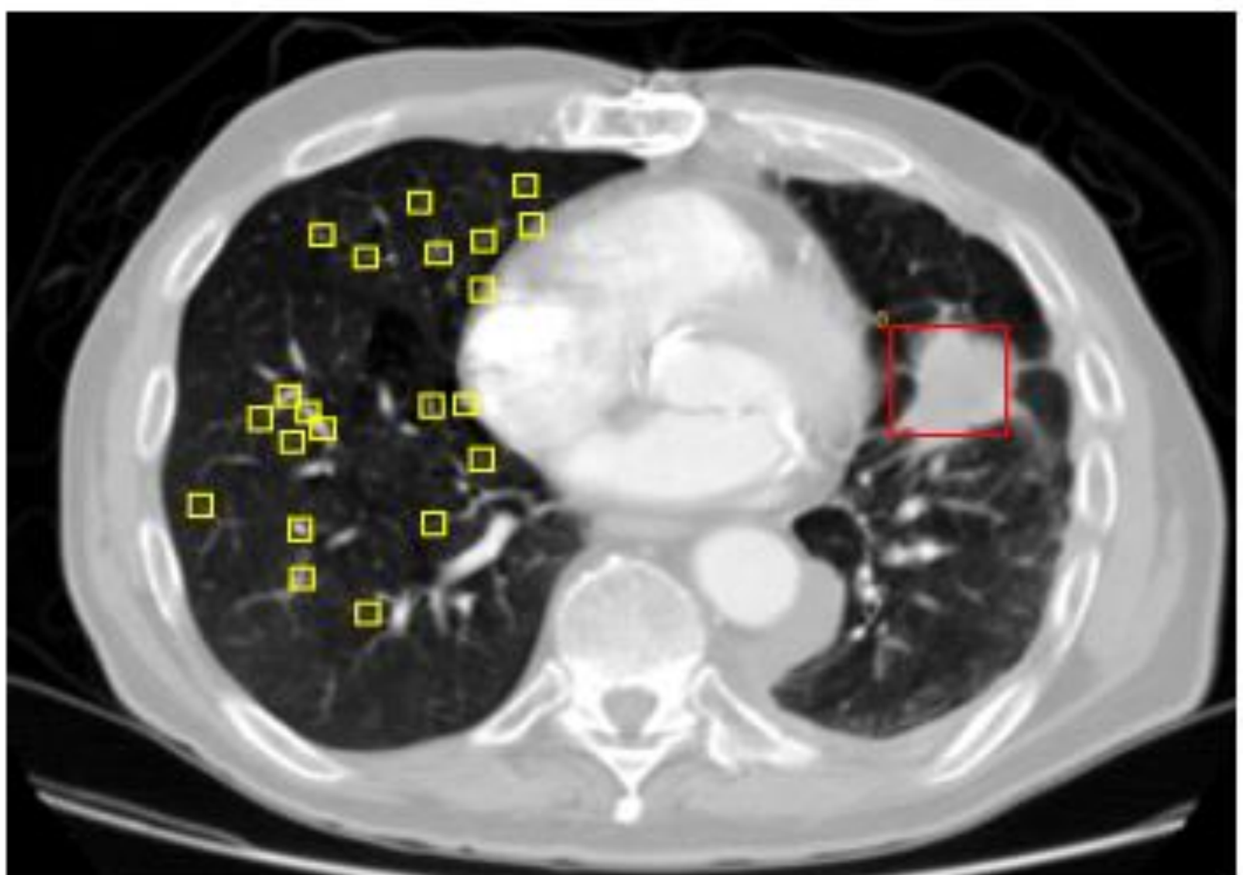


AUTONOMOUS VEHICLES

- Multiple sensors like:
  - Cameras
  - Lidars
  - Ultrasonic sensors
- Big players like:
  - Intel & BMW
  - Nvidia & Audi
  - Google
  - Tesla

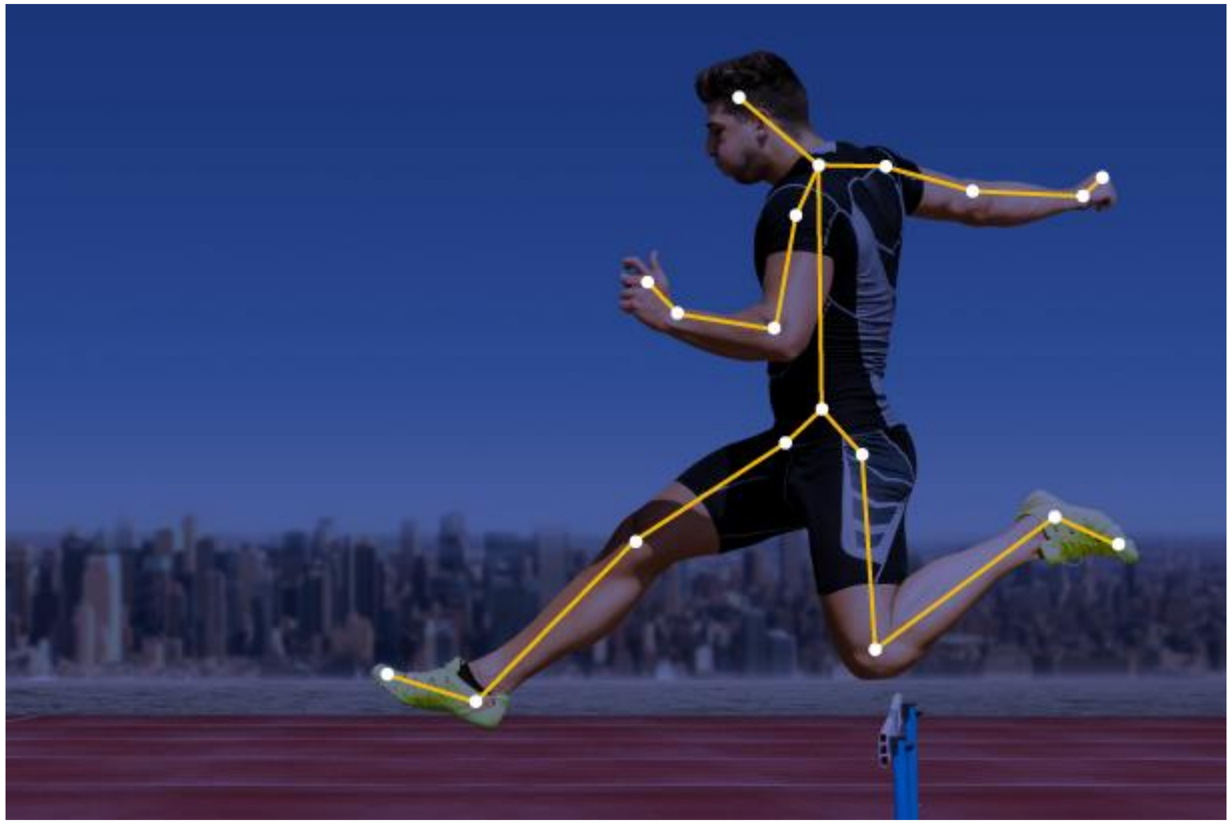
LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
MOTION FLOW
LANE LINES
LANE LINES
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS





MEDICINE

- Assistance for doctors,
- Cancer detection (Kaggle's Data Science Bowl 2017),
- Healthcare automation (future?).



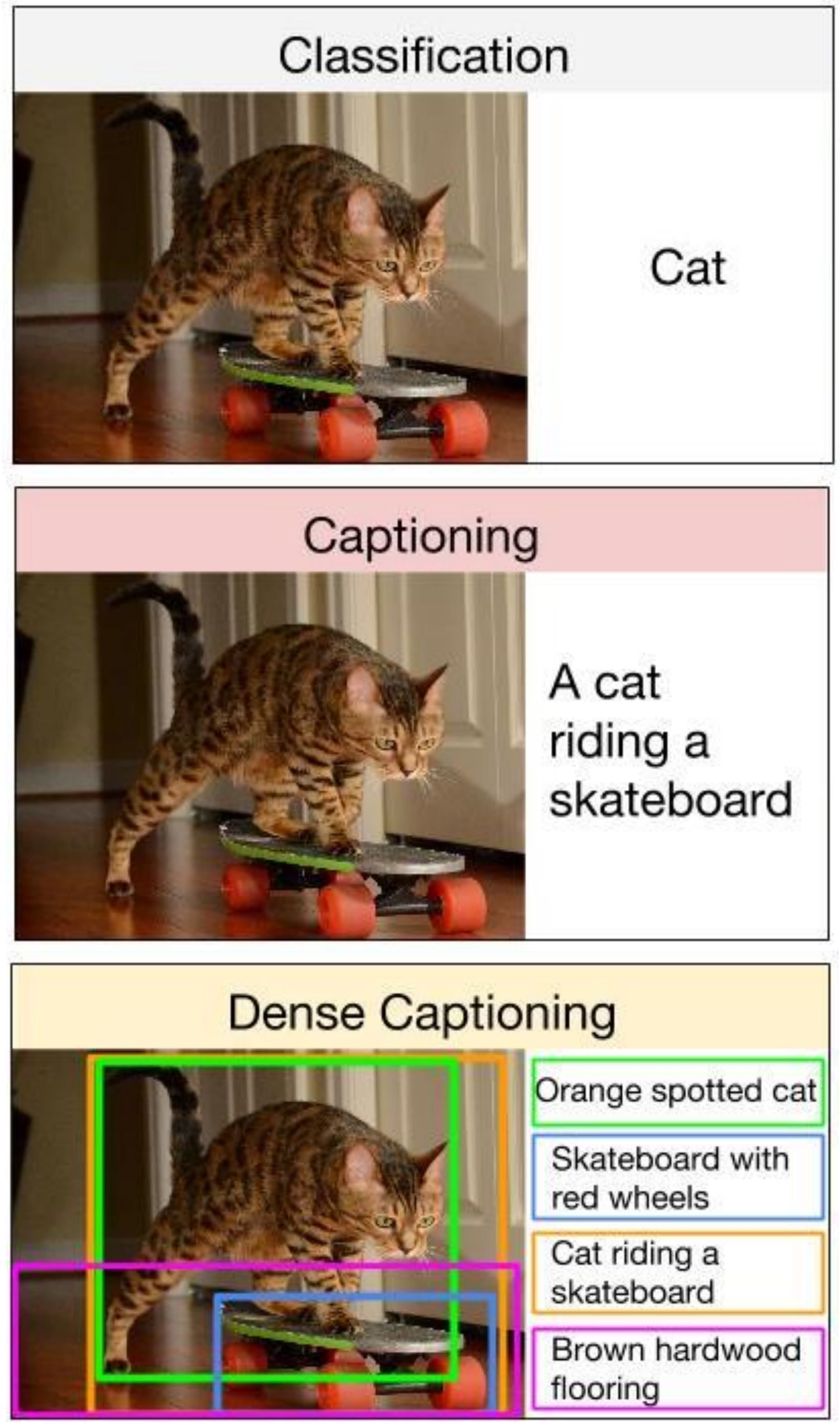
SPORT

- ➤ Intel Olympic Games!
- ➤ Statistics for audience
- ➤ Support for athletes

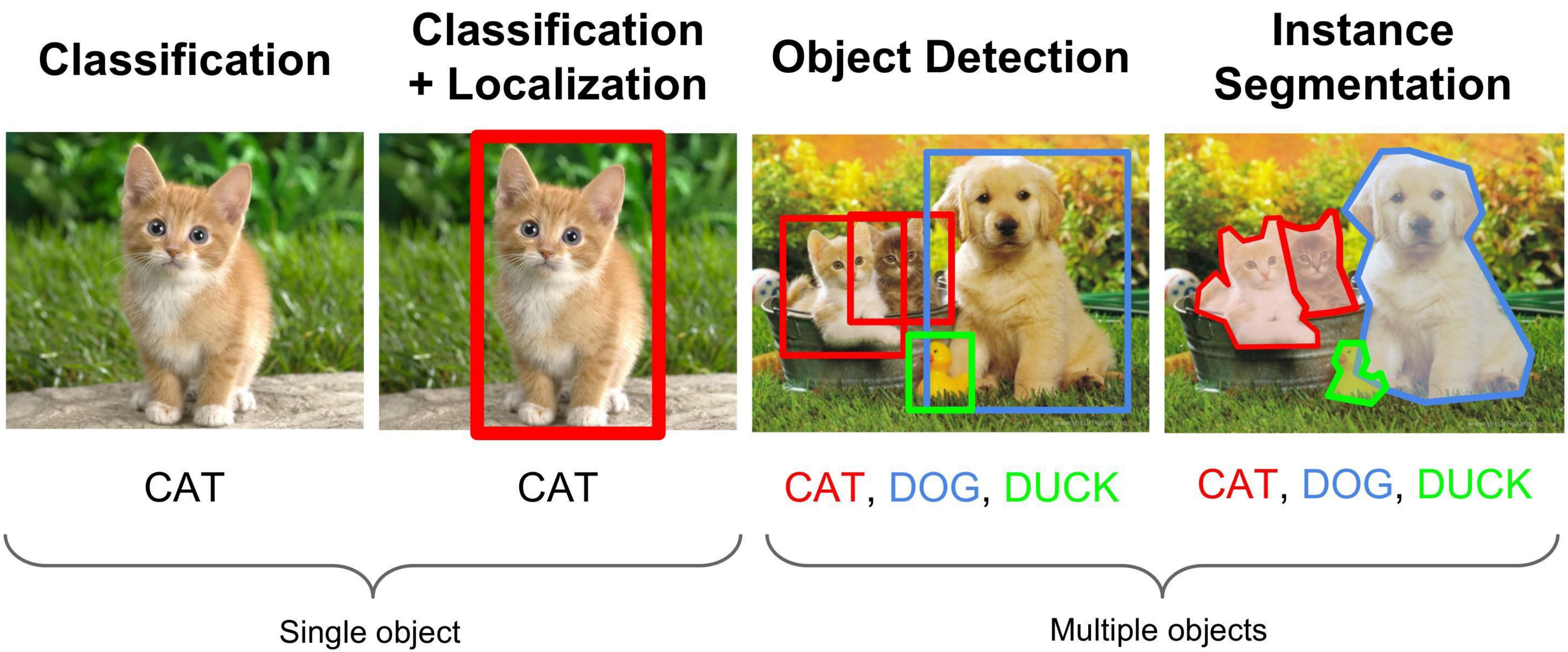
ROBOTICS

- CNNs in manufactures
- Robots with computer vision
- Reinforcement learning

WHAT WE CAN DO?



WHAT WE CAN DO?



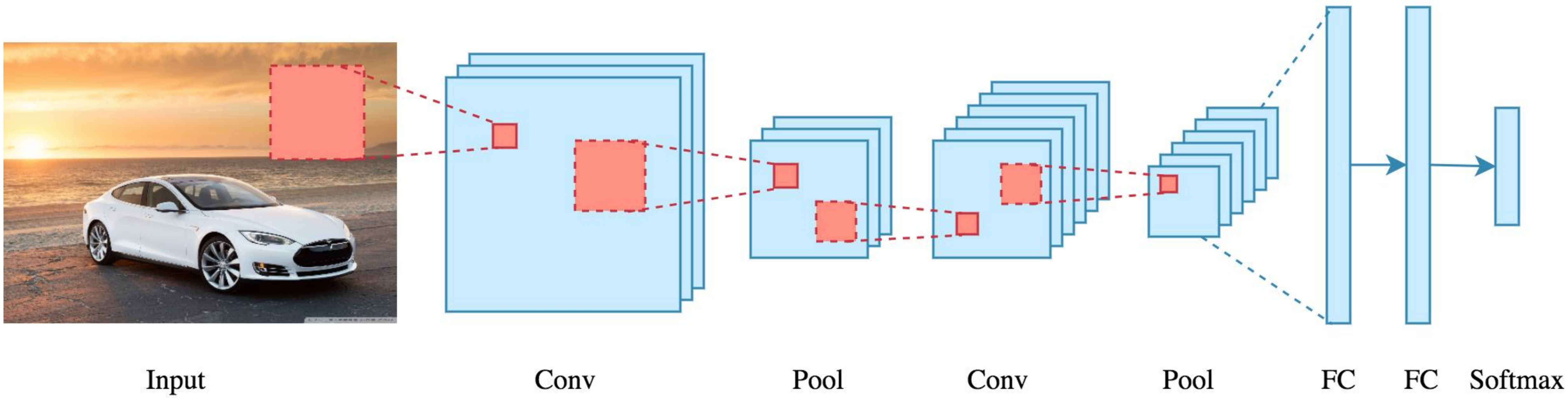
CNN ARCHITECTURE

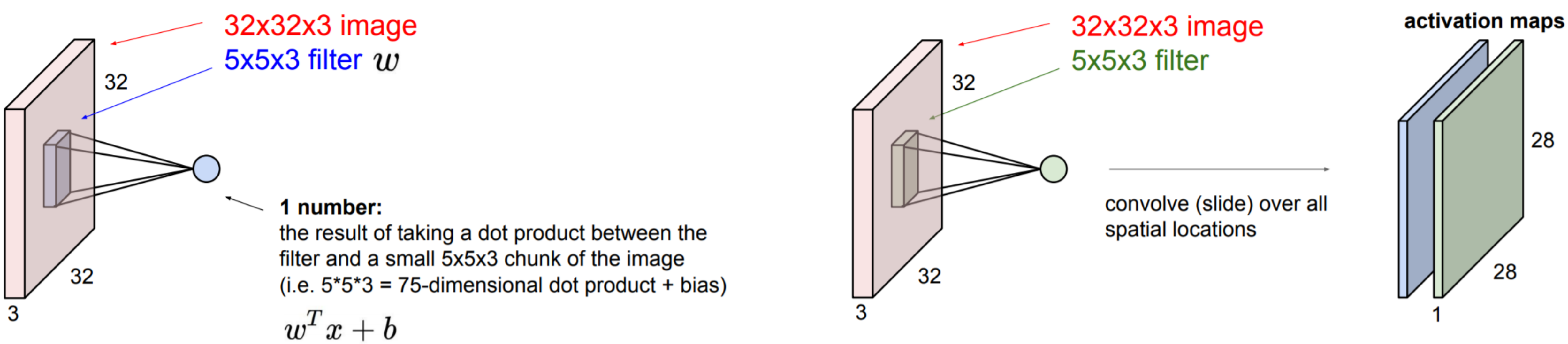
What's inside?

BASIC BLOCKS

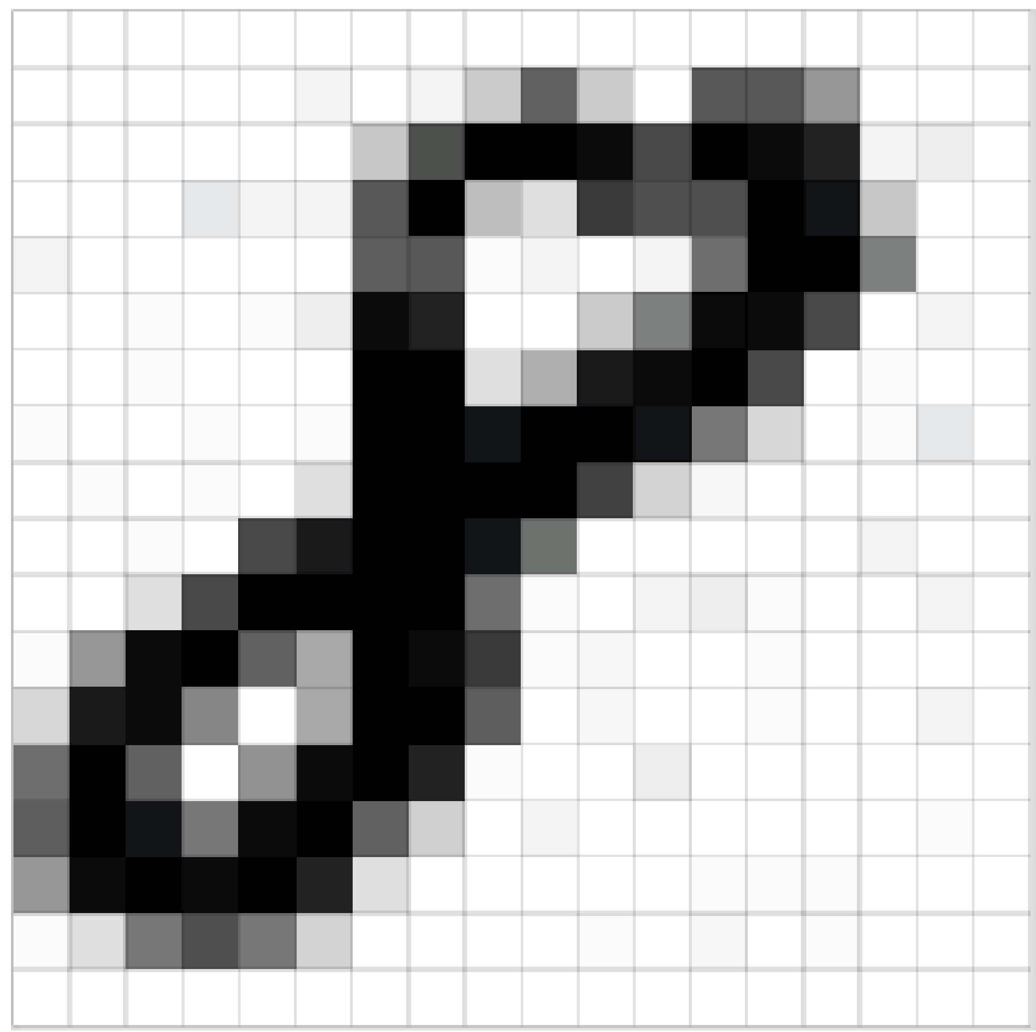
- Input layer,
- Convolution layer,
- Activation layer,
- Pooling layer,
- Fully Connected layer.

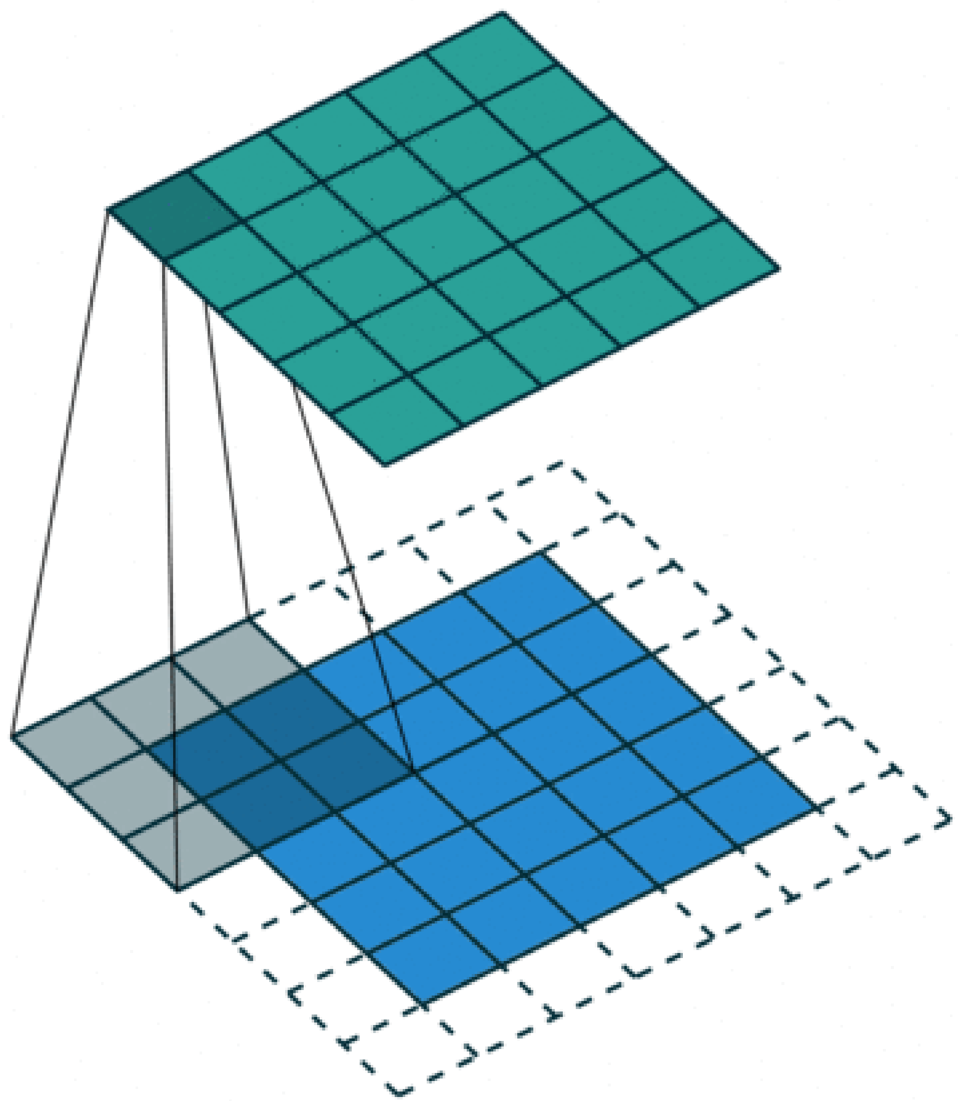
FUNDAMENTAL CONCEPT





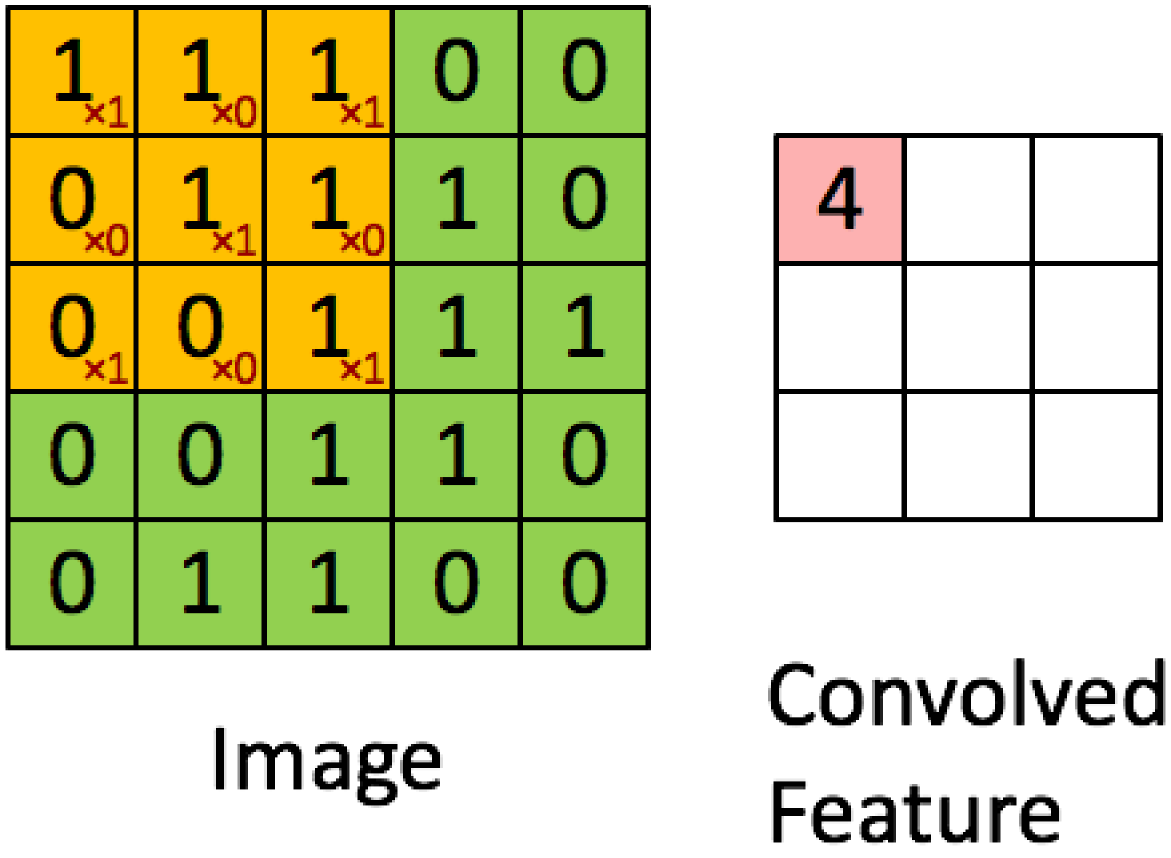
WHAT IS OUR INPUT?

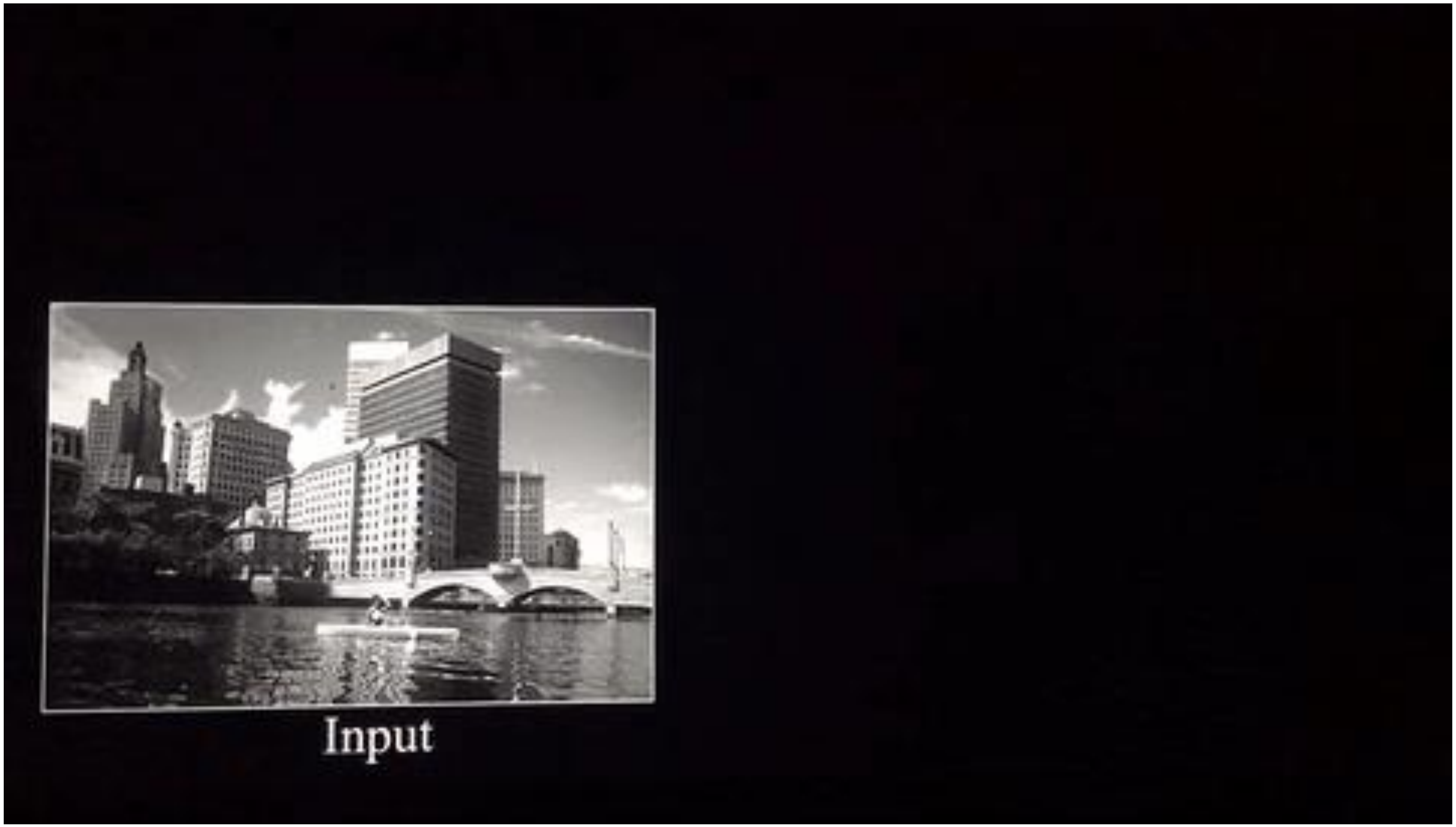




CONVOLUTION

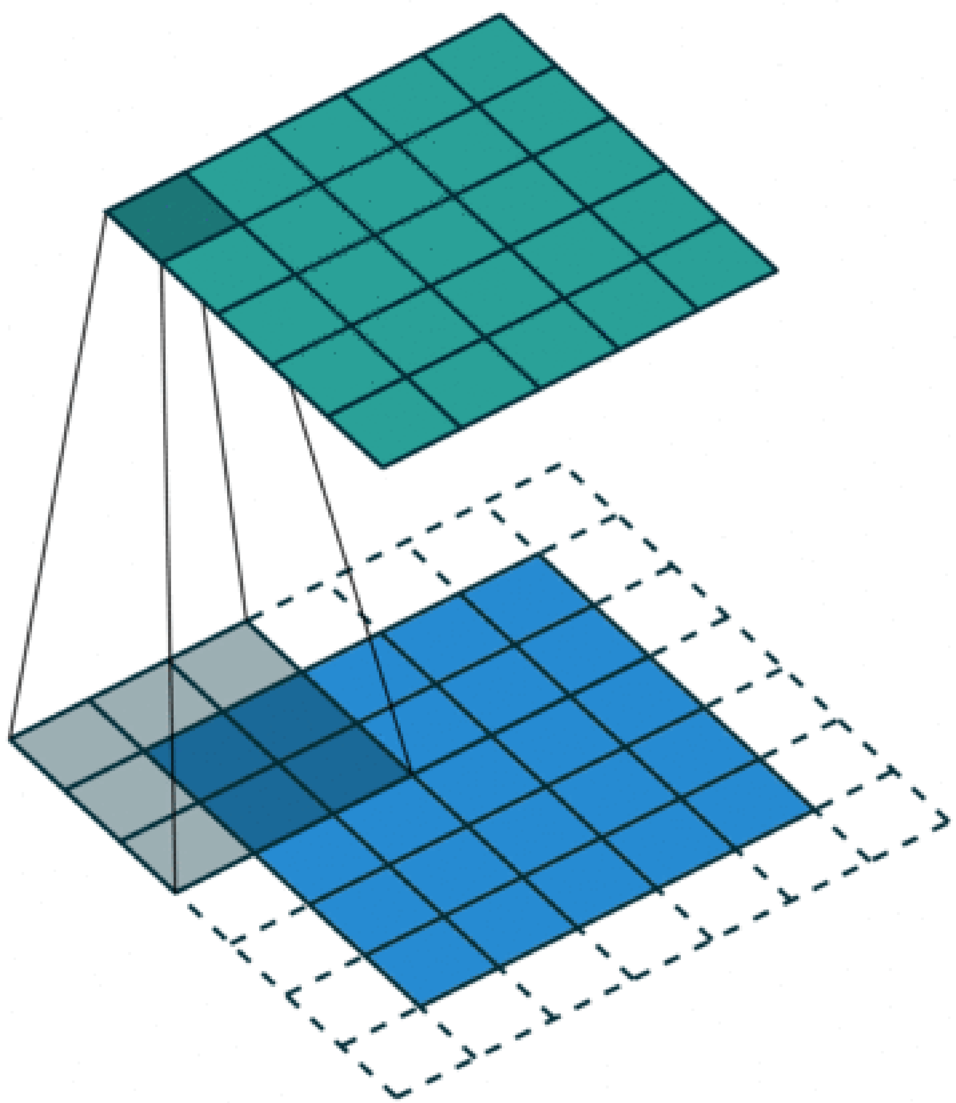
- Creates "**feature maps**",
- Apply **filters** on the image,
- Move such filter over the image and calculate **feature**,
- Follow the **stride** (how many fields it should "jump"),
- Is defined by **kernel size** (filter size),
- Can use **padding** for bigger receptive field.


1 ×1
1 ×0
1 ×1
0
0
0 ×0
1 ×1
1 ×0
1
0
0 ×1
0 ×0
1 ×1
1
1
0
0
1
1
0
0
1
1
0
0
Image
4
Convolved
Feature

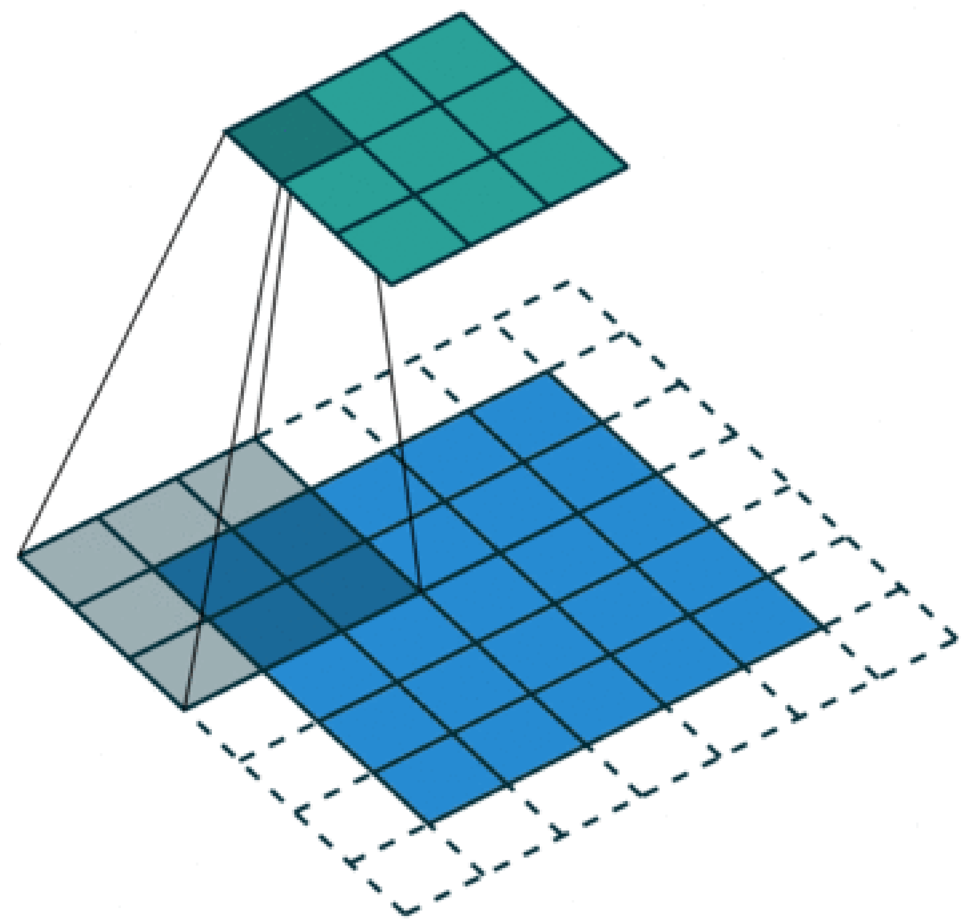

Input

STRIDE

Stride = 1



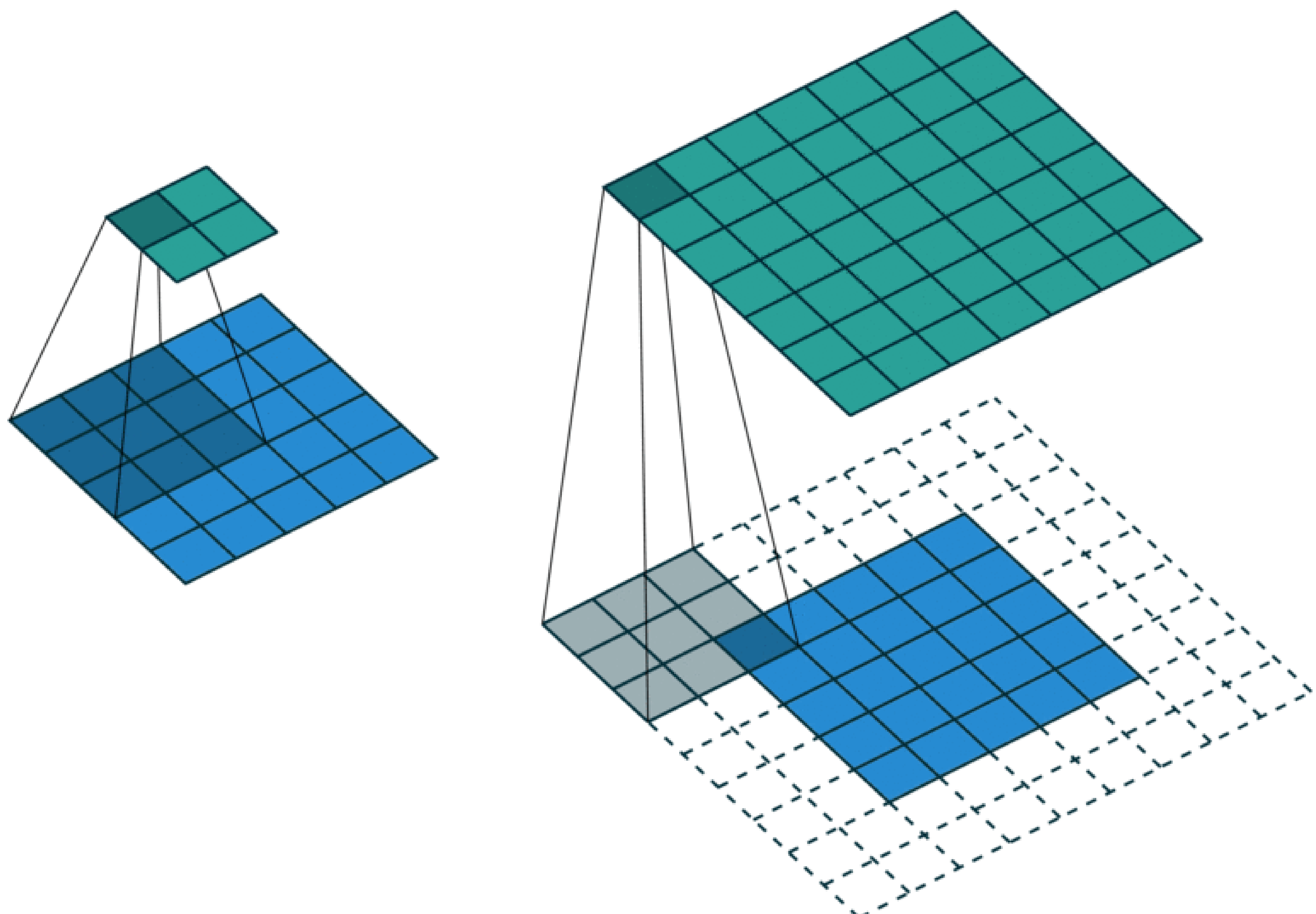
Stride = 2



PADDING

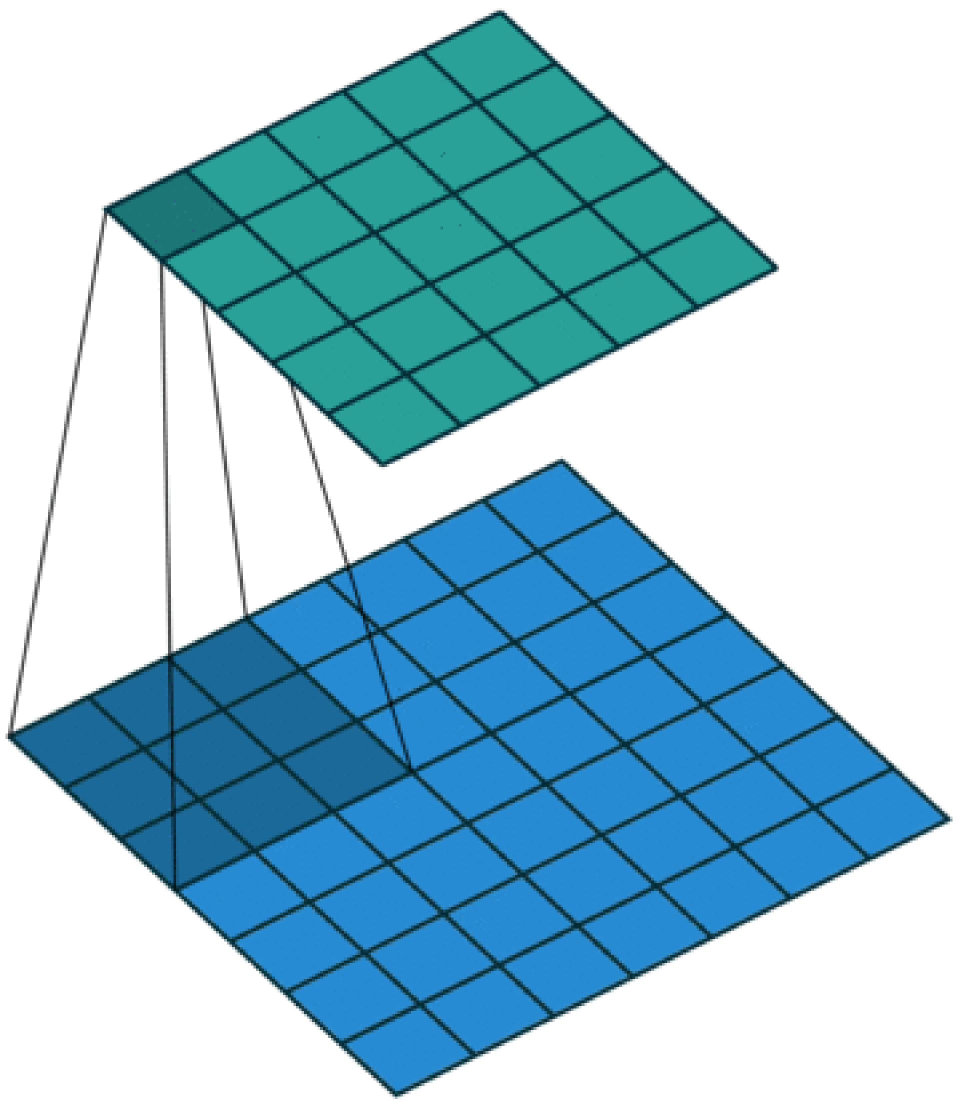
Padding = 0

Padding = 2

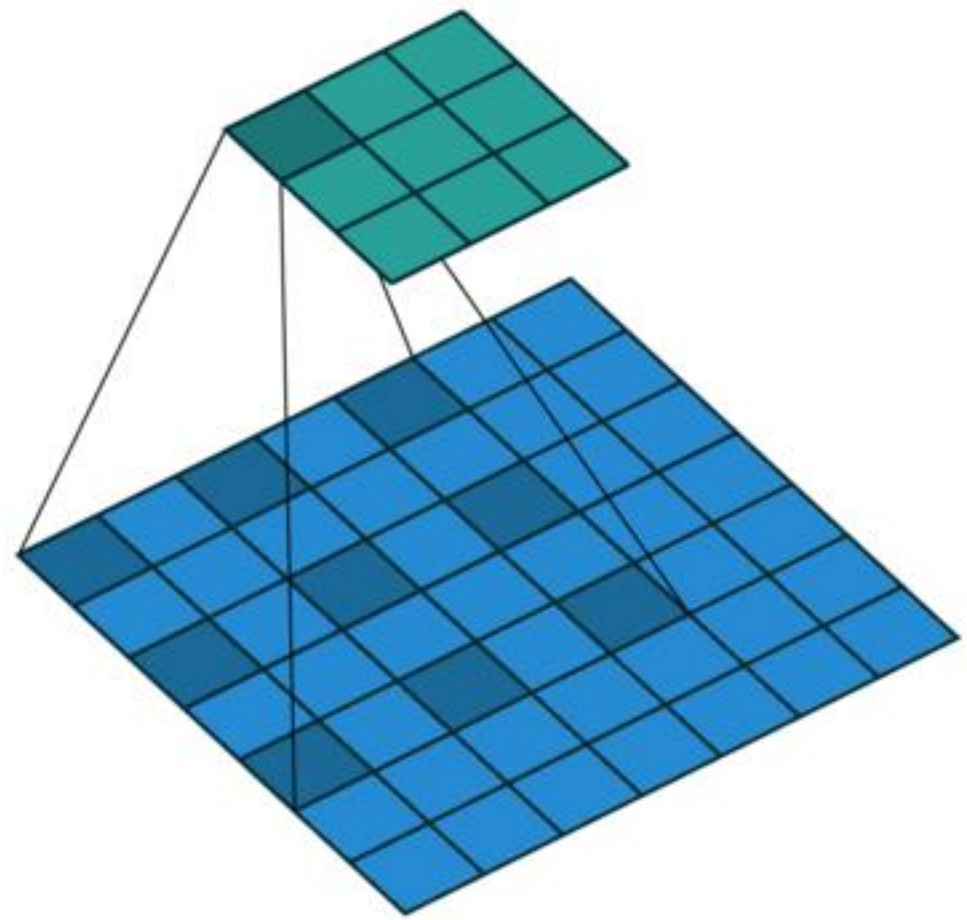


DILATION

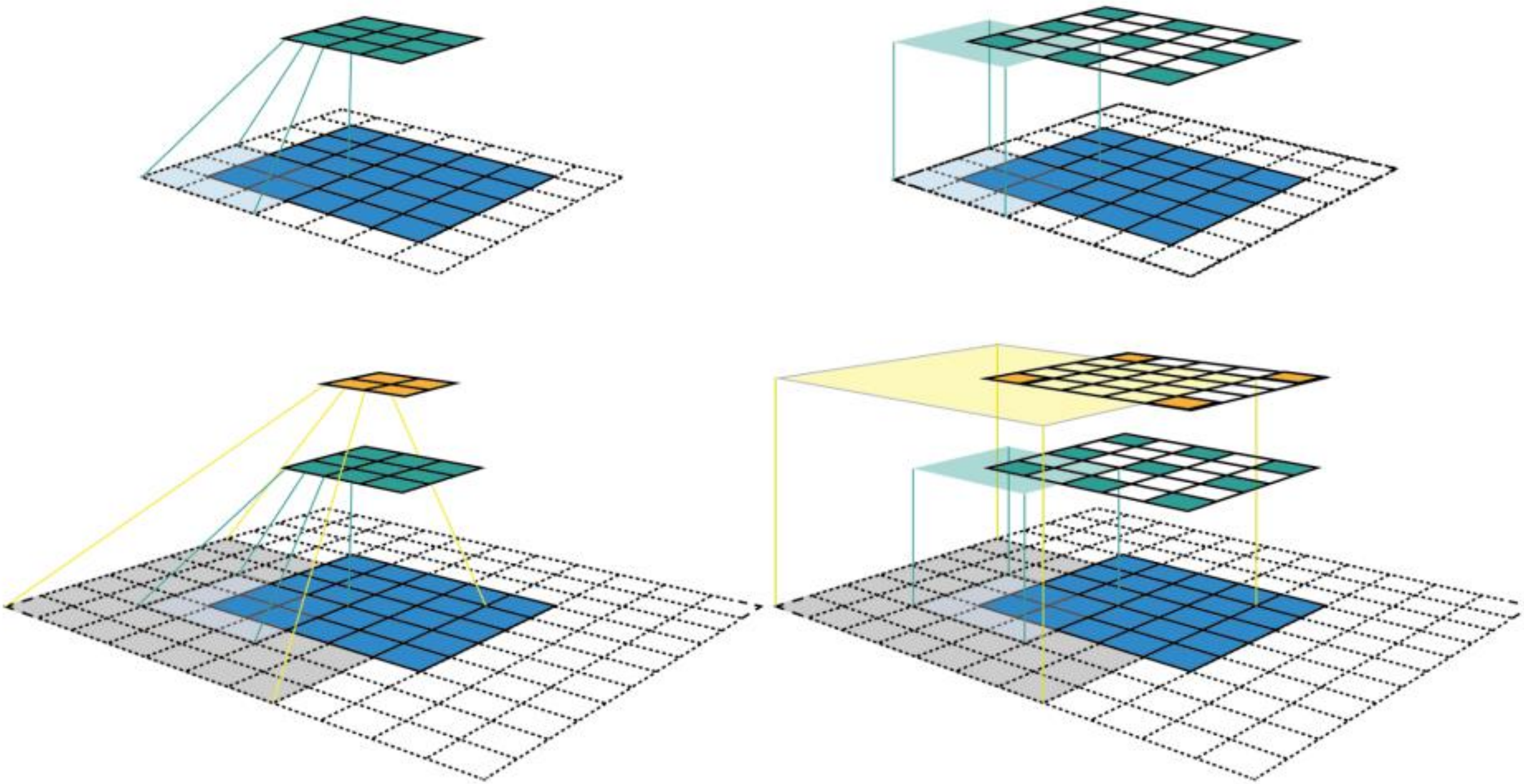
Dilation = 1



Dilation = 2



RECEPTIVE FIELD



EQUATION FOR NUMBER OF OUTPUT FEATURES

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features

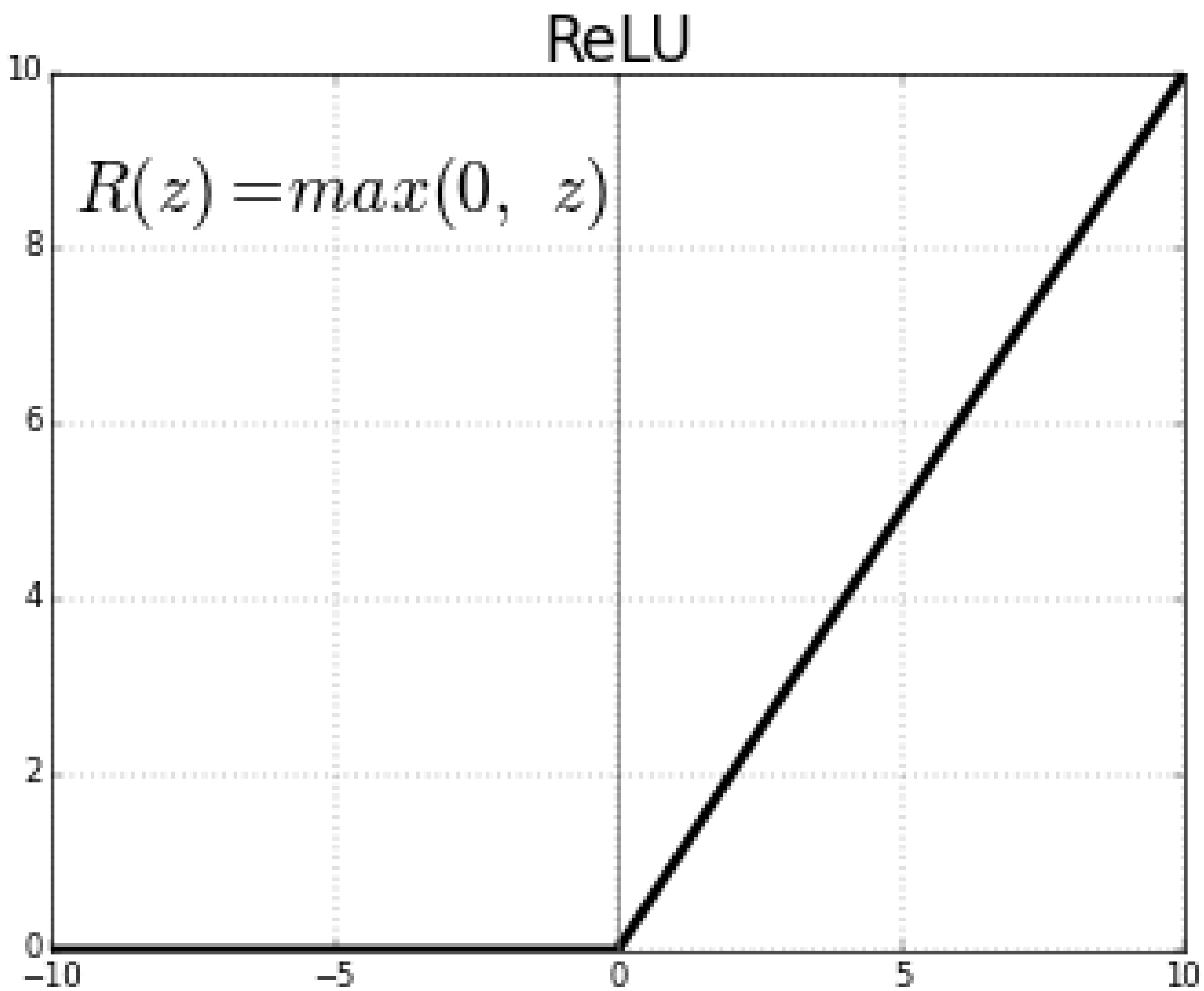
n_{out}: number of output features

k: convolution kernel size

p: convolution padding size

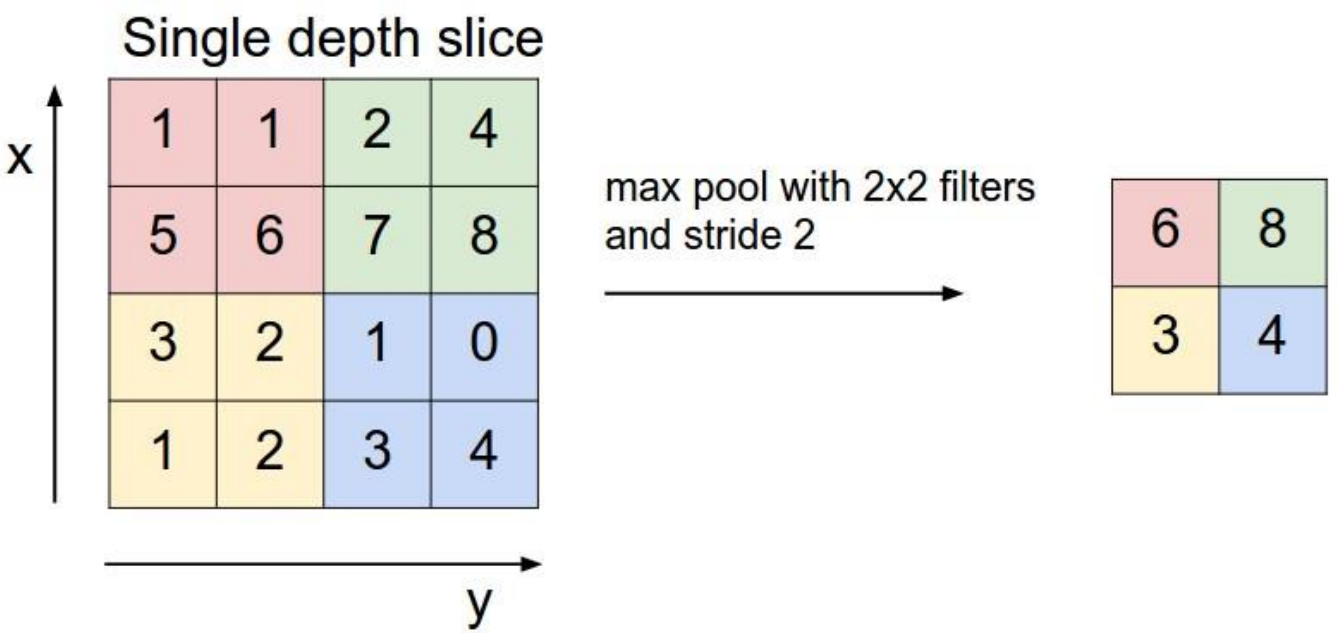
s: convolution stride size

ACTIVATION



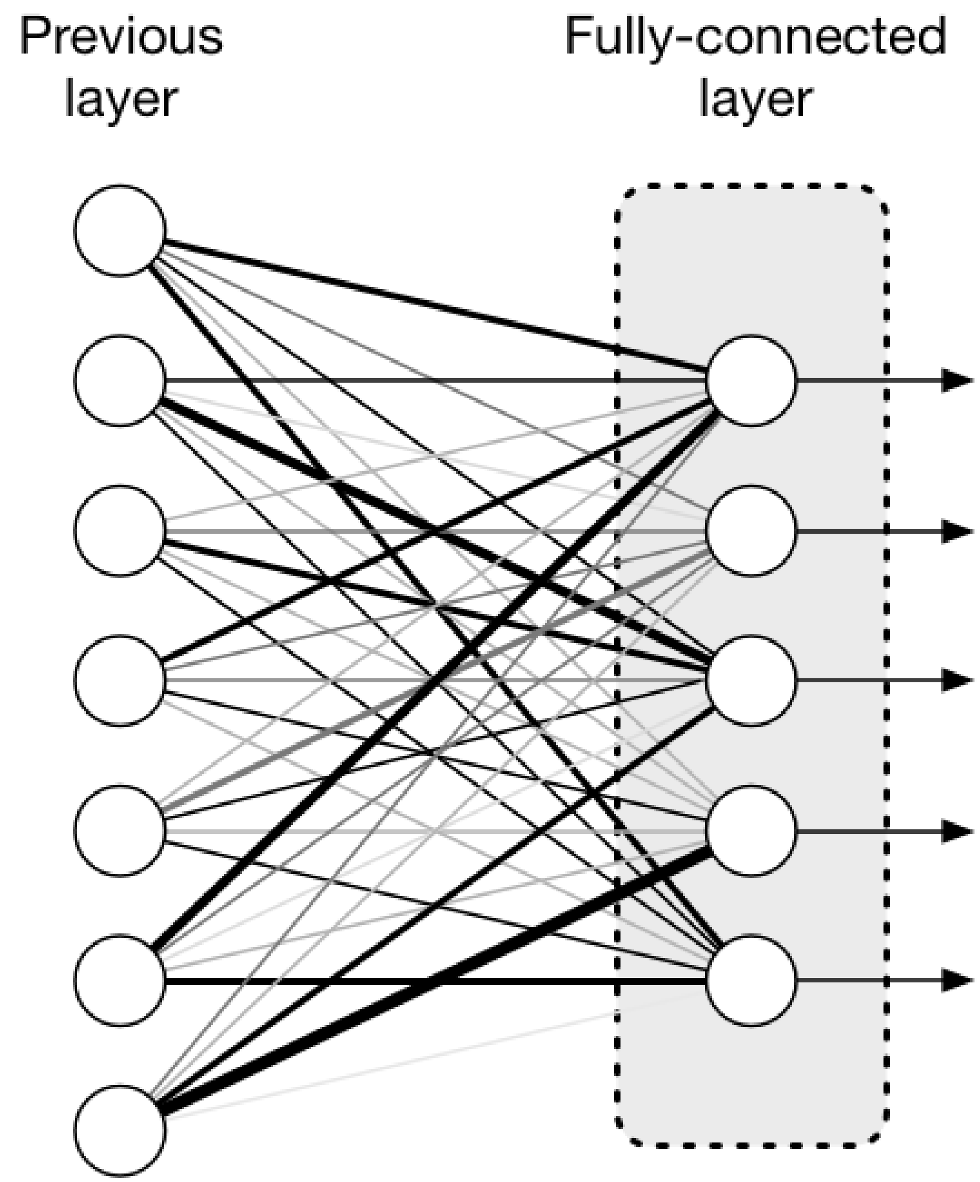
- **ReLU** is a default way to go,
- Some similar layers like:
    - Leaky ReLU,
    - Maxout,
- There are many other layers but have many problems (vanishing gradients, etc.).

POOLING

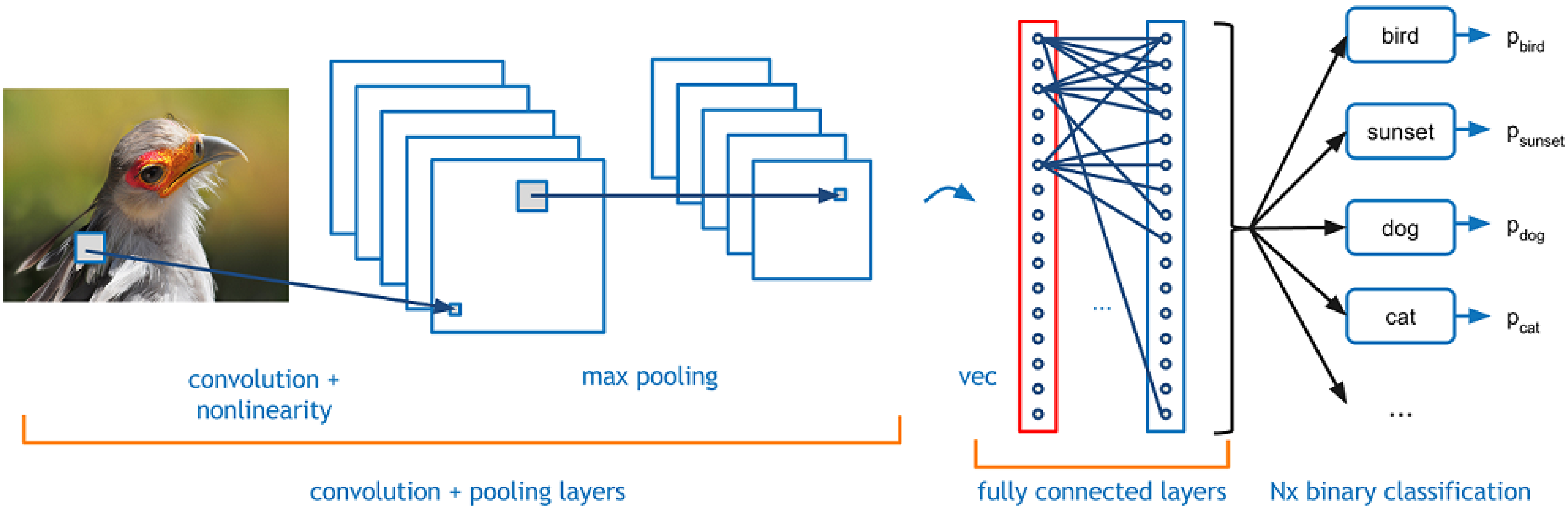


- Pooling **reduces** spatial space,
- **Reduces** amount of parameters,
- Reduces overfitting,
- A simple **routing** (during back propagation),
- Most common: MaxPooling,
- Also: AvgPooling, ...

FULLY CONNECTED LAYER



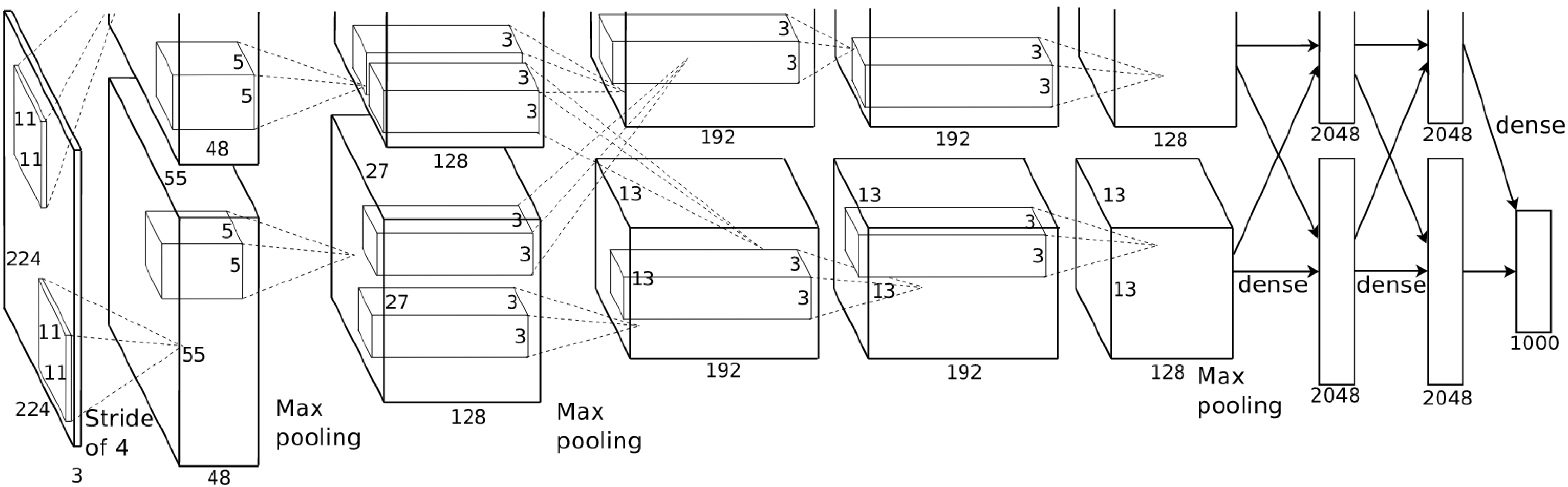
FULLY CONNECTED LAYER



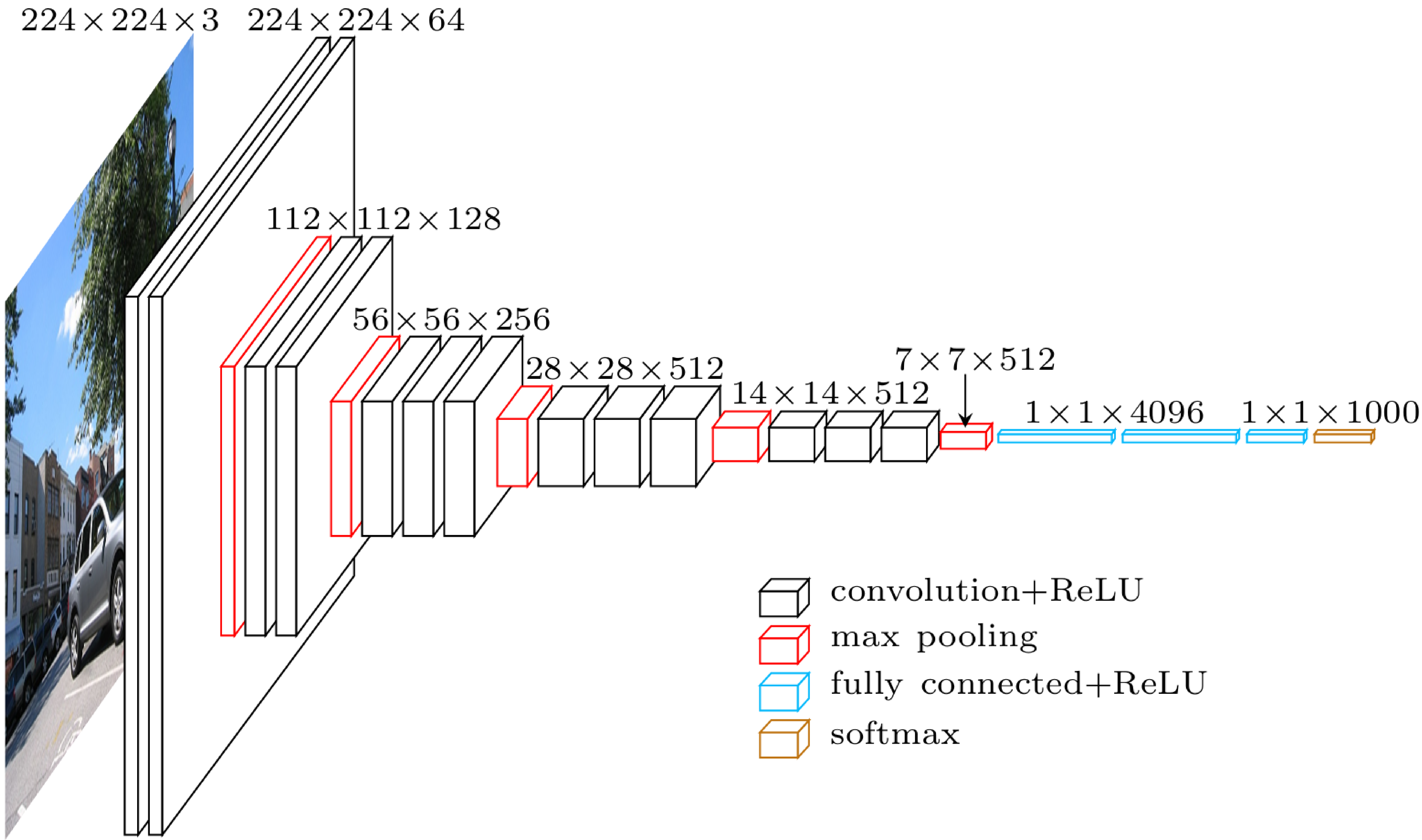
POPULAR ARCHITECTURES

…that might be helpful!

ALEXNET - 2012

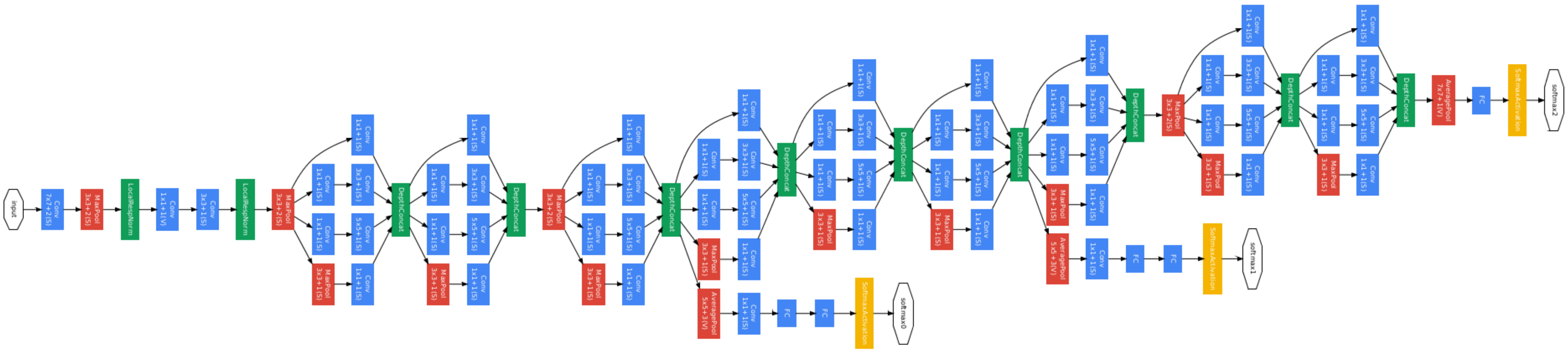


VGG - 2014

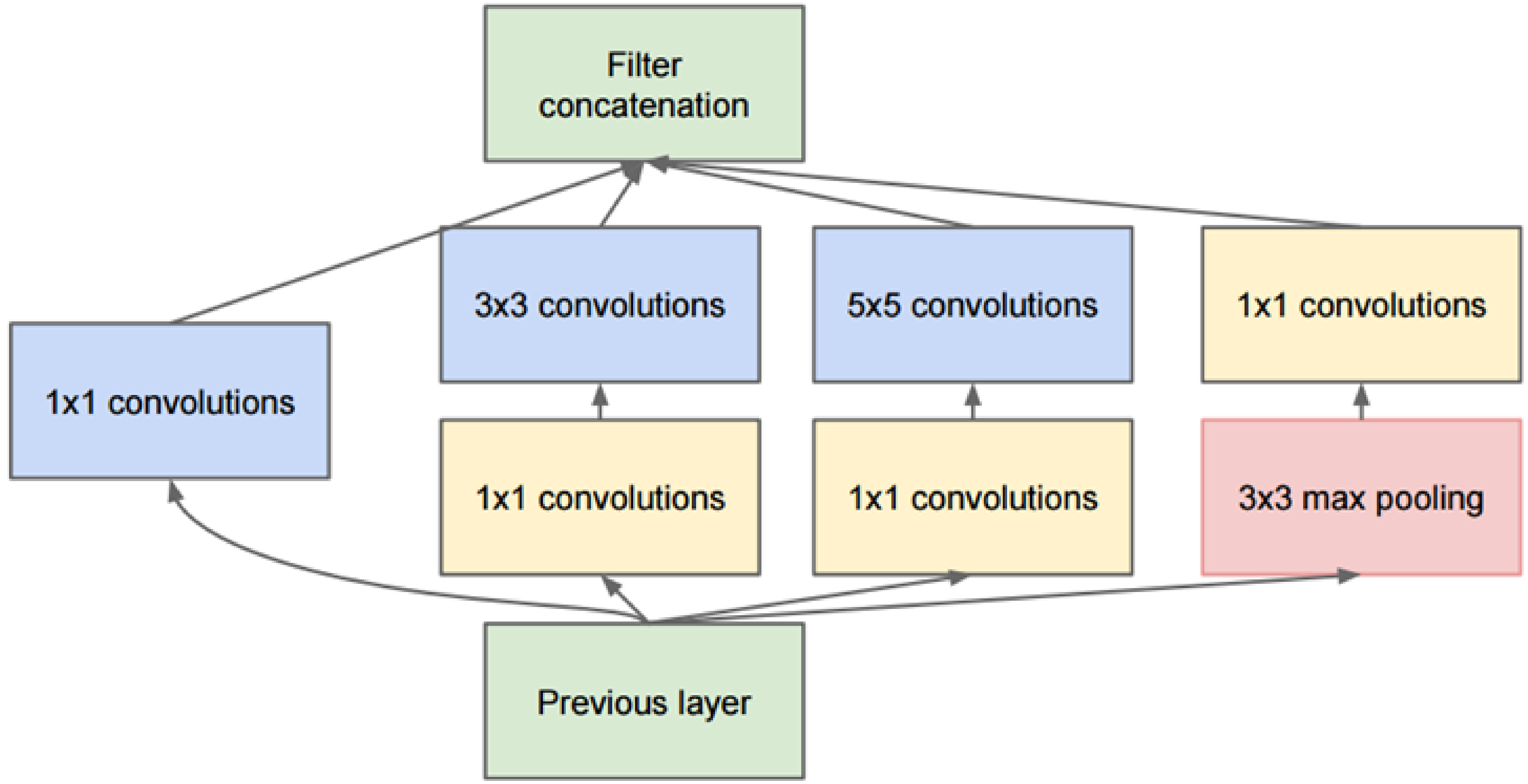


224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

GOOGLENET - 2014

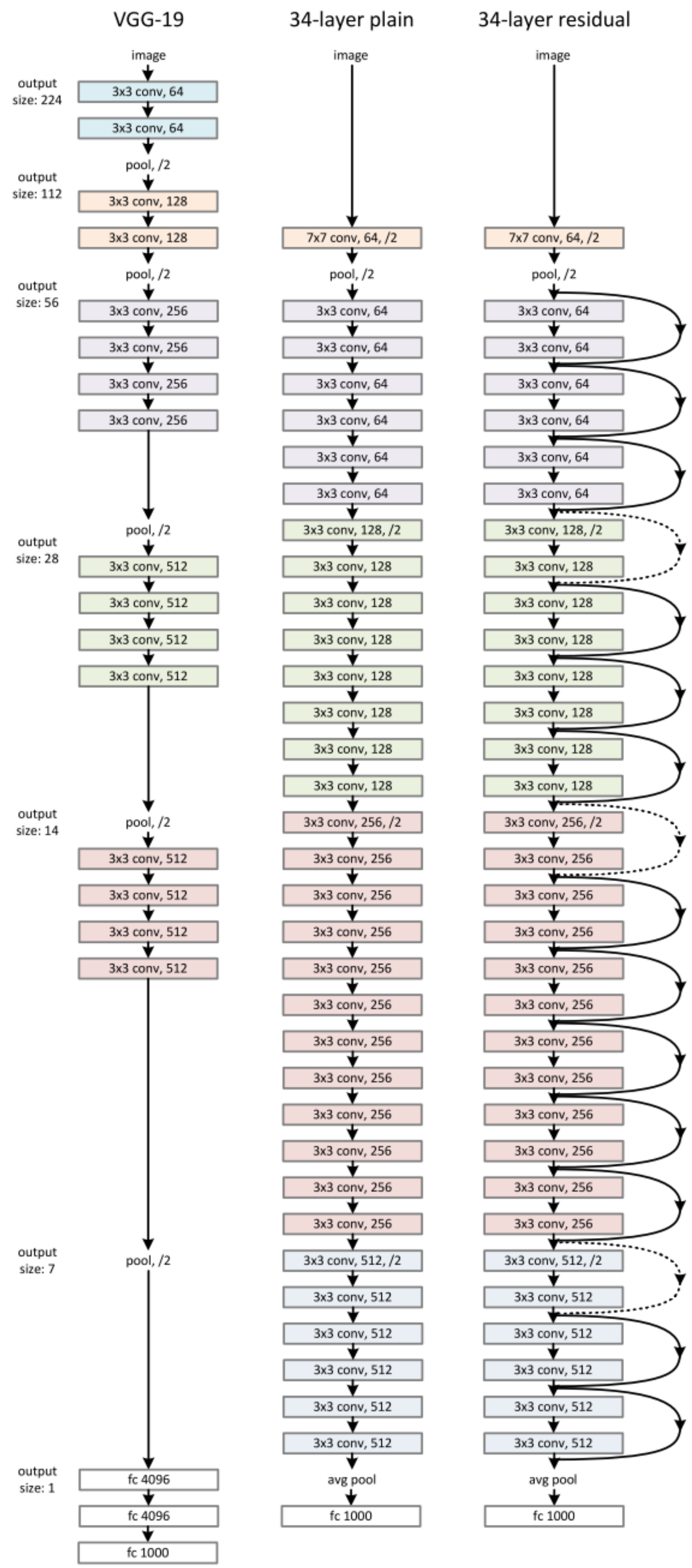


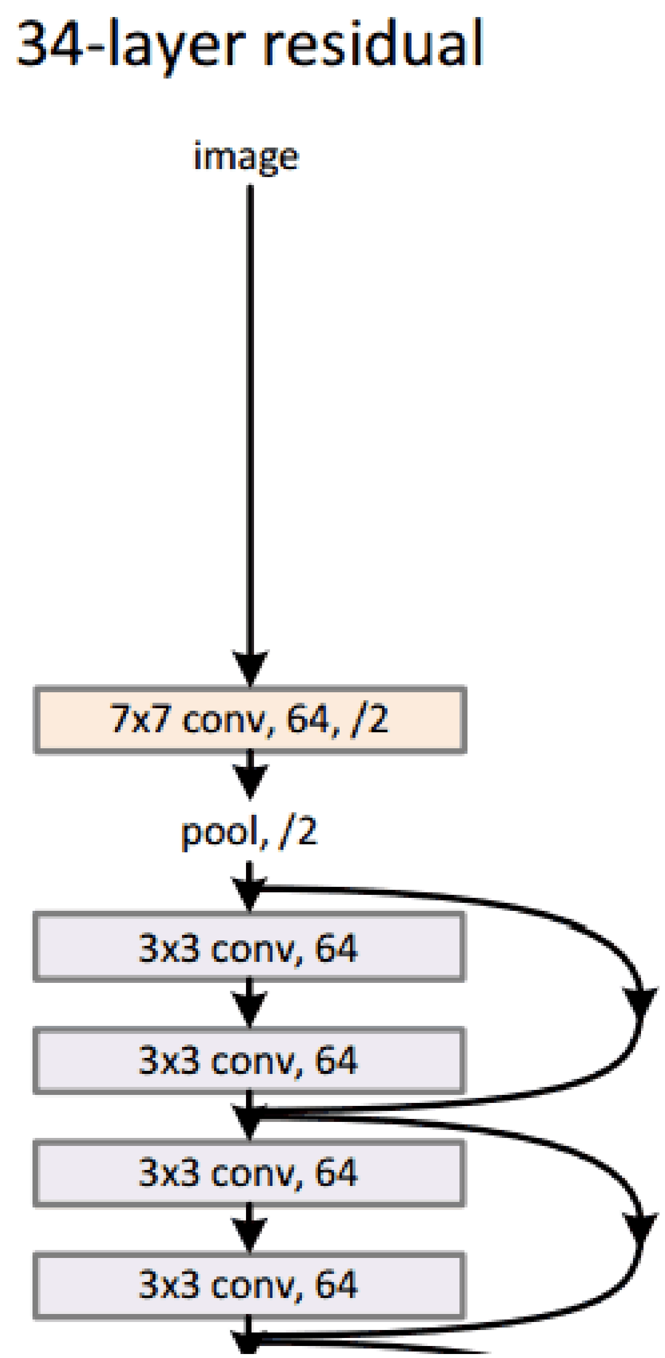
GOOGLENET



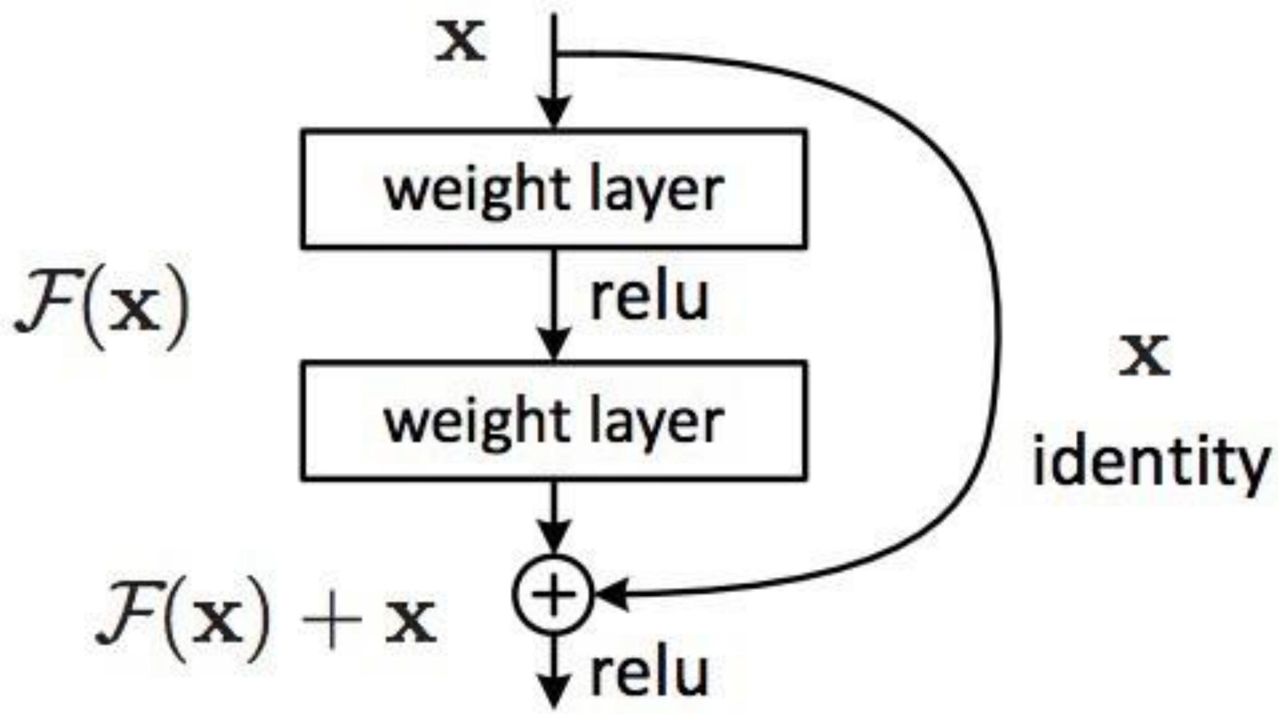
Full Inception module

RESNET


VGG-19
34-layer plain
34-layer residual
image
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64

RESNET

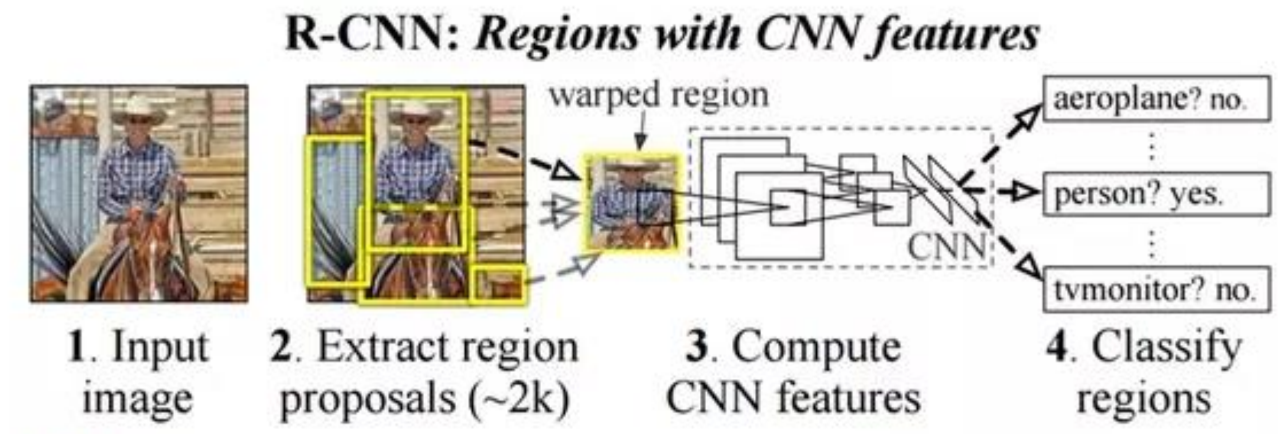


$\mathbf{x}$
weight layer
$\mathcal{F}(\mathbf{x})$
relu
$\mathbf{x}$
weight layer
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

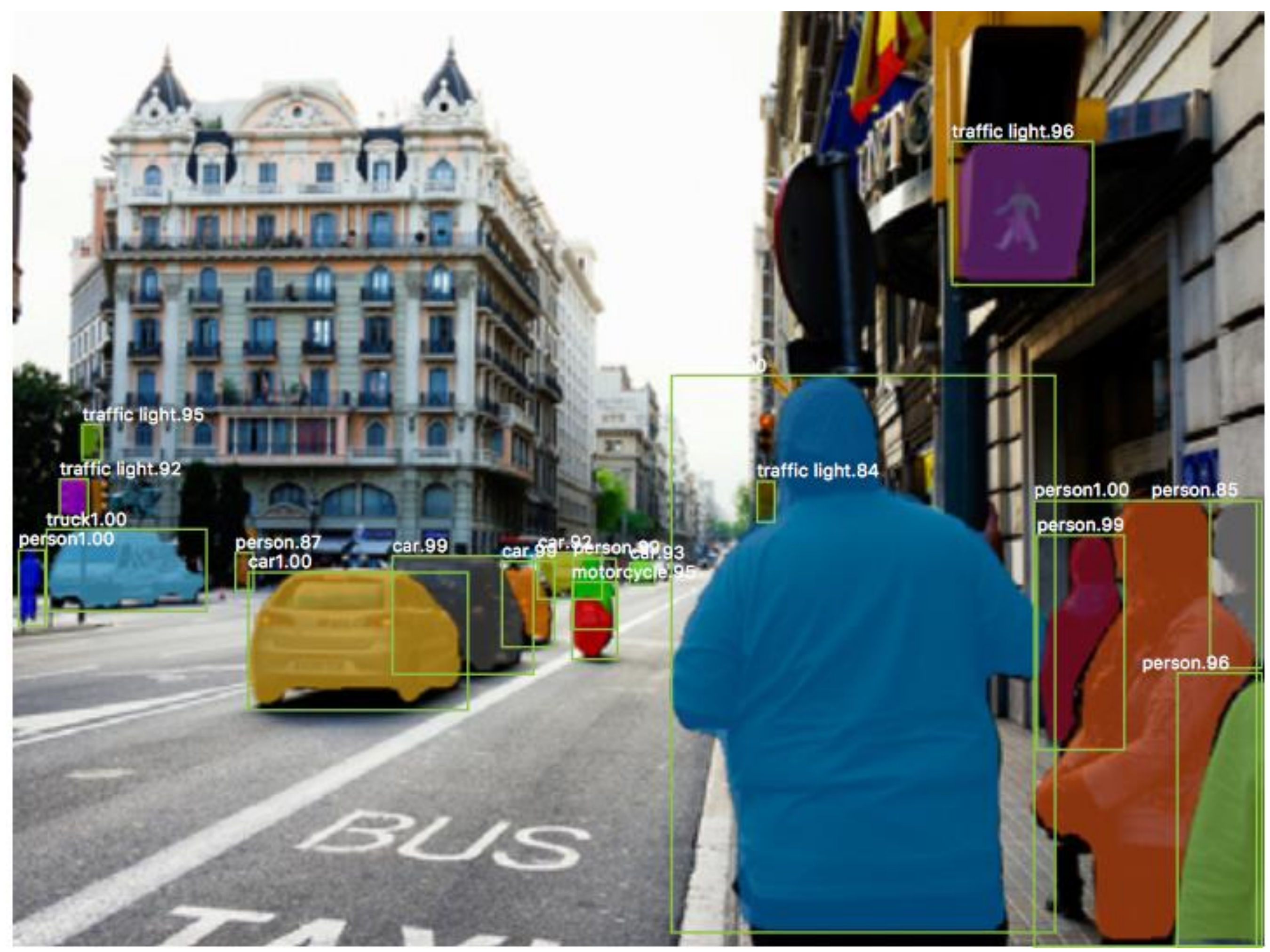
JOURNEY OF CNN IN 3 YEARS

1. R-CNN
2. Fast R-CNN
3. Faster R-CNN
4. Mask R-CNN

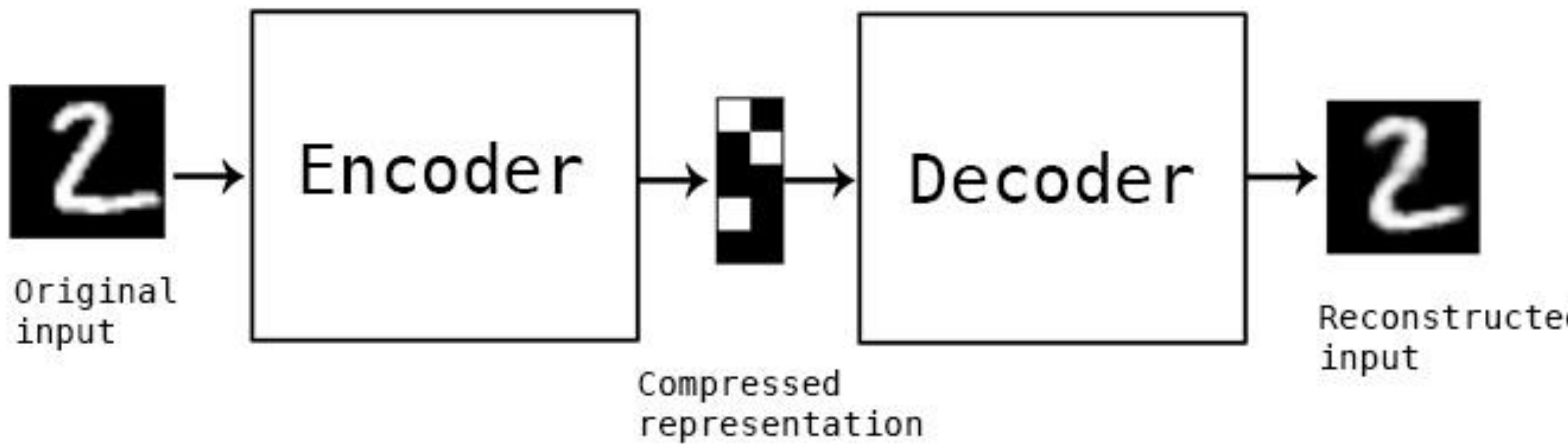
Worth reading more!



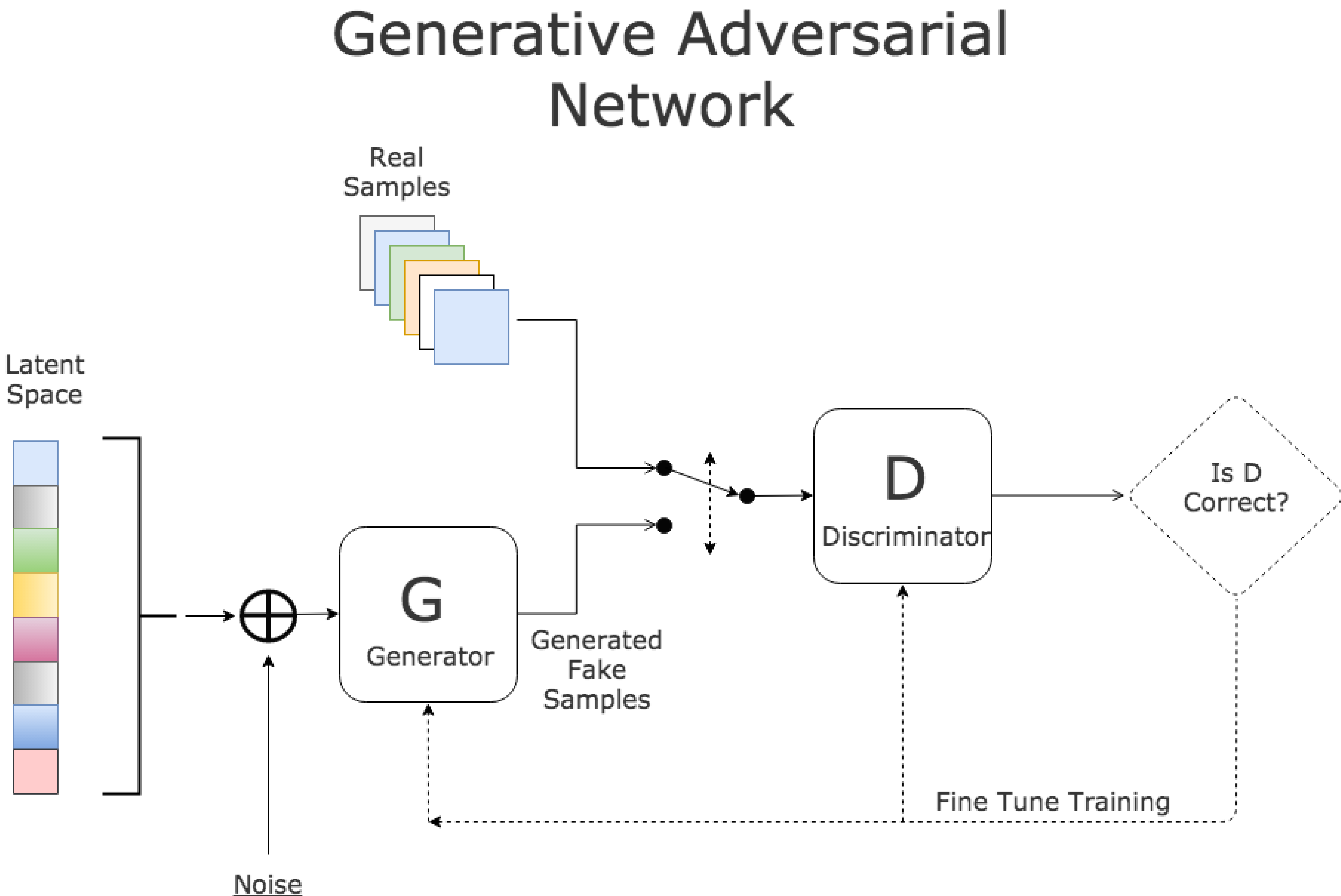
R-CNN workflow



AUTOENCODERS



GAN



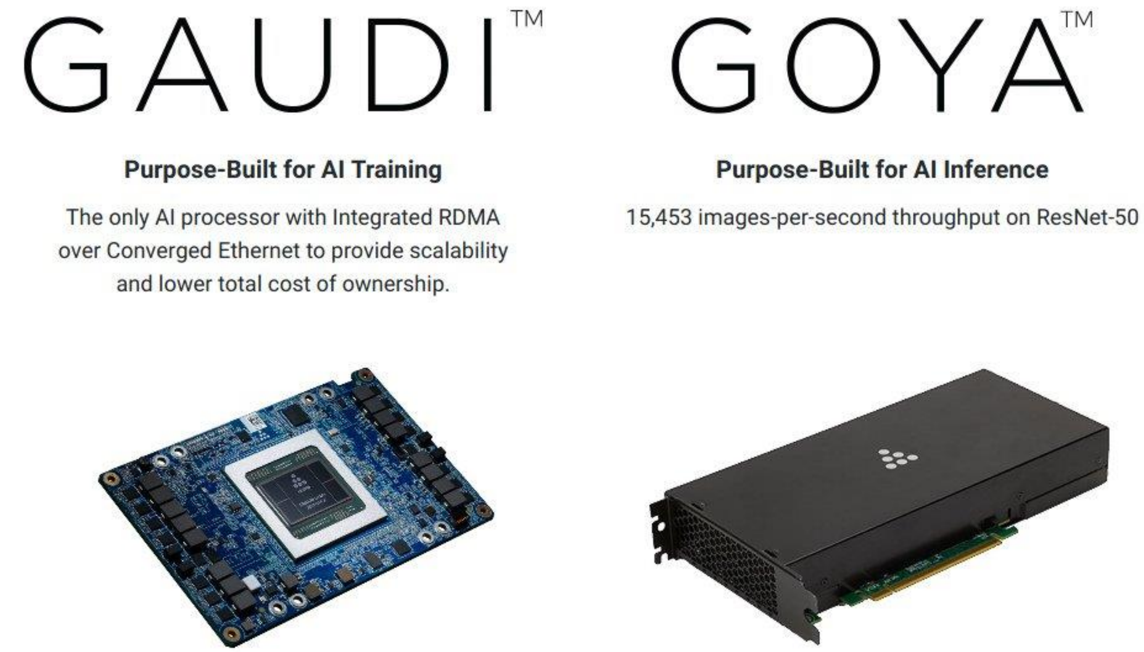
Generative Adversarial
Network
Real
Samples
Latent
Space
G
Generator
Generated
Fake
Samples
D
Discriminator
Is D
Correct?
Fine Tune Training
Noise

RESNET 152 – 60 MILLION PARAMETERS, 152 LAYERS

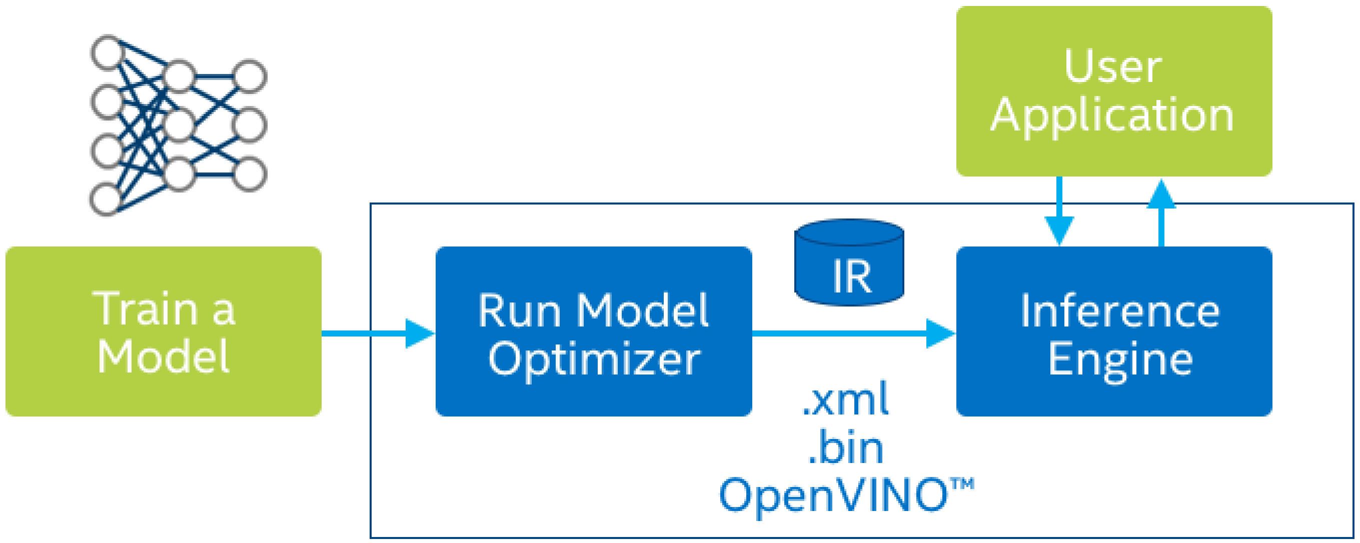
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
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

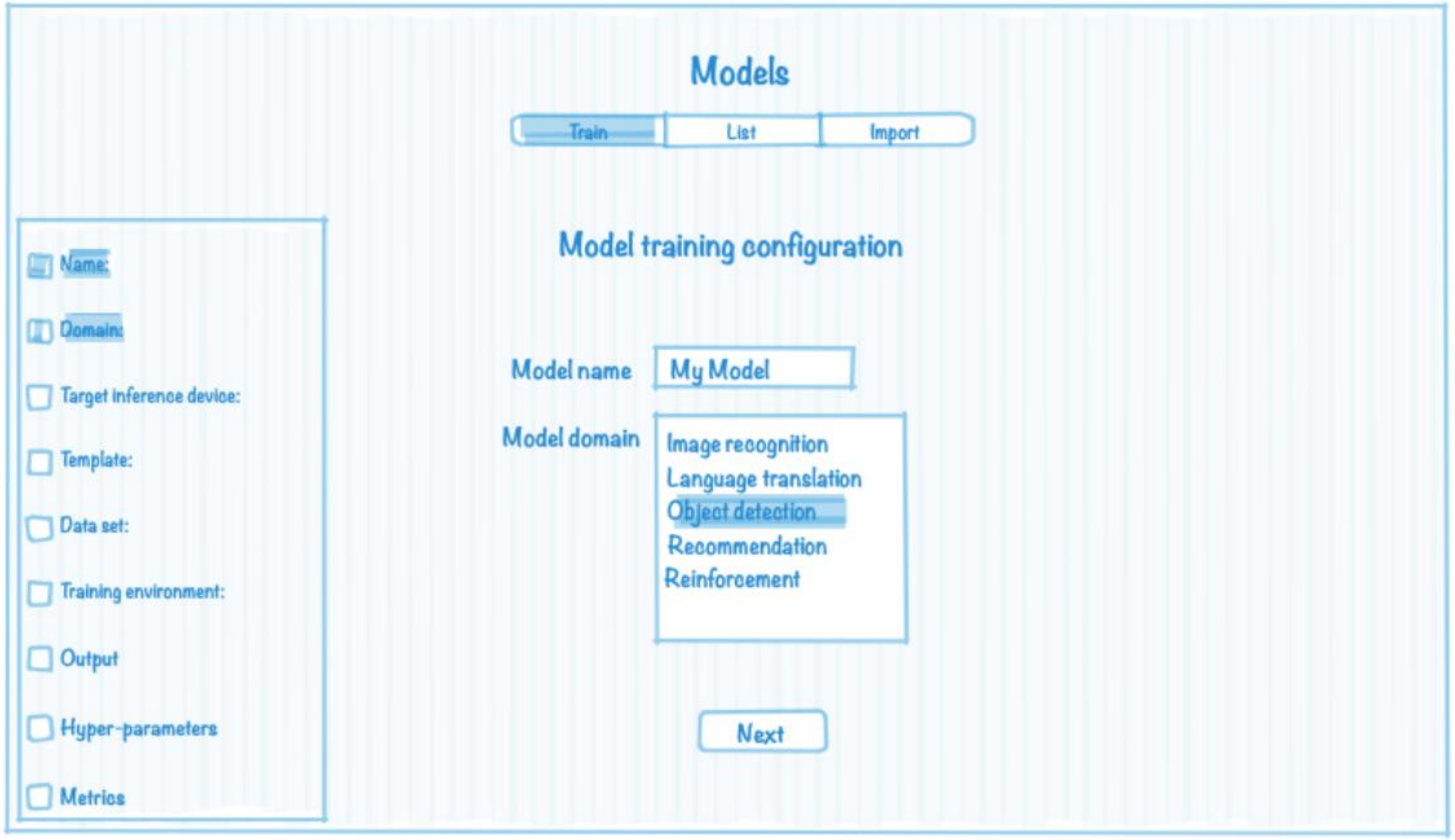
HOW TO DEAL WITH ALL OF THIS?

AI OPTIMIZED HARDWARE – TRAINING AND INFERENCE



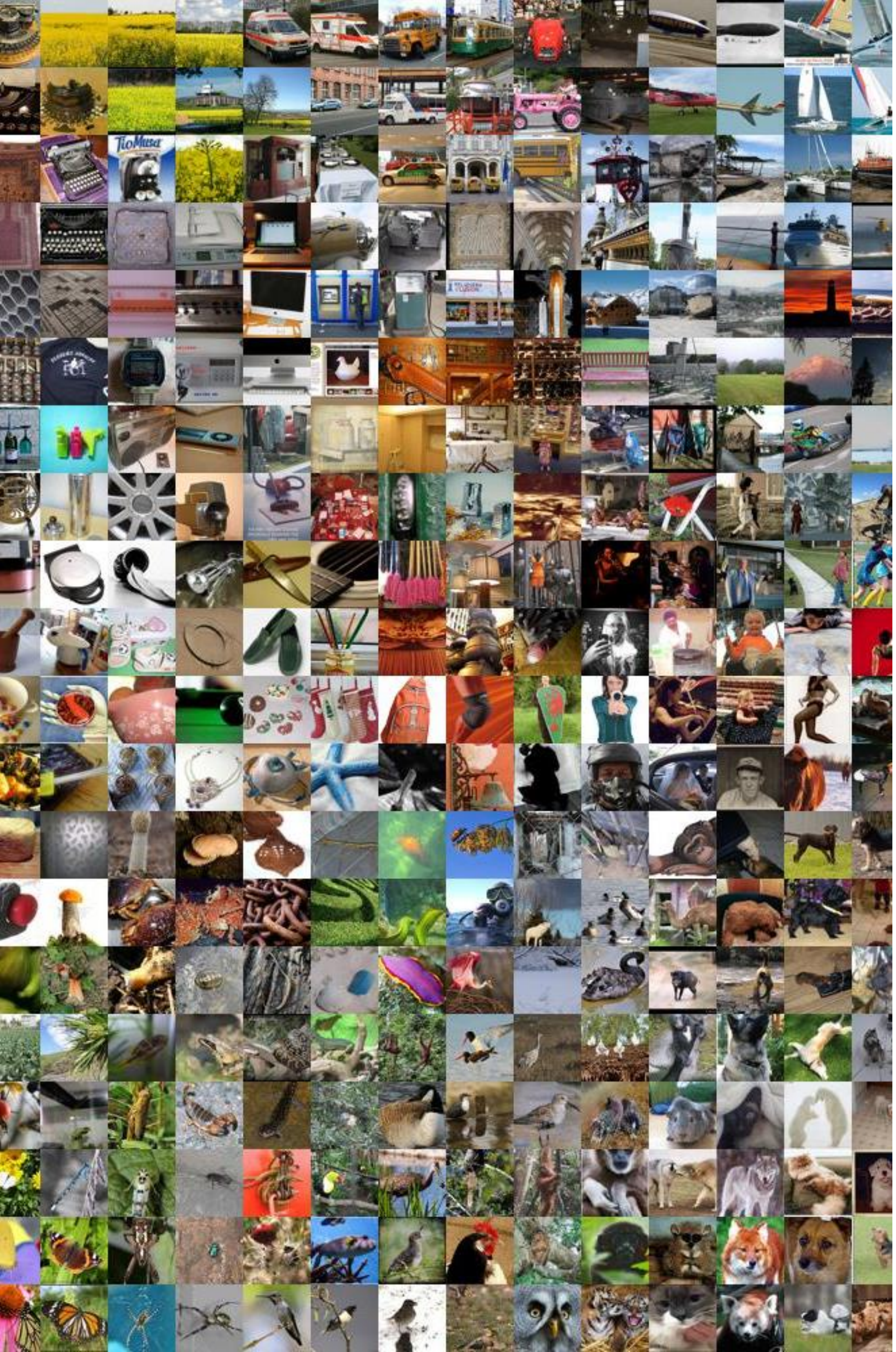
OPEN VINO AND MODEL PREPARATION PLATFORM





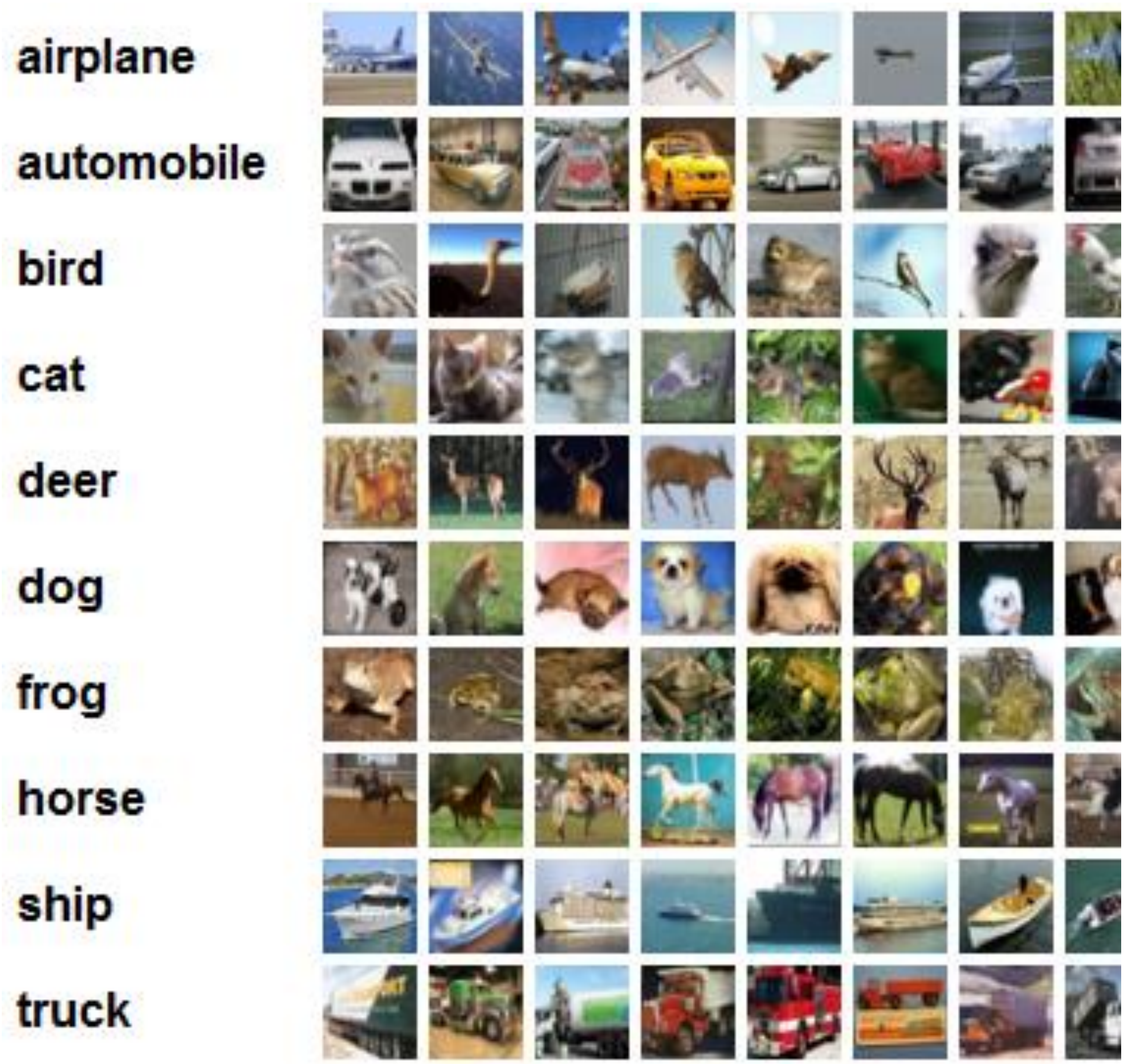
POPULAR DATASETS

…to train your CNNs!



IMAGENET

- 10 millions hand-labelled images,
- 1 million with bounding boxes,
- Labels based on WordNet (hierarchical dictionary).



CIFAR10

- 60.000 images for training,
- 6000 images for testing,
- 10 classes,
- Also: Version with 100 classes (CIFAR100).

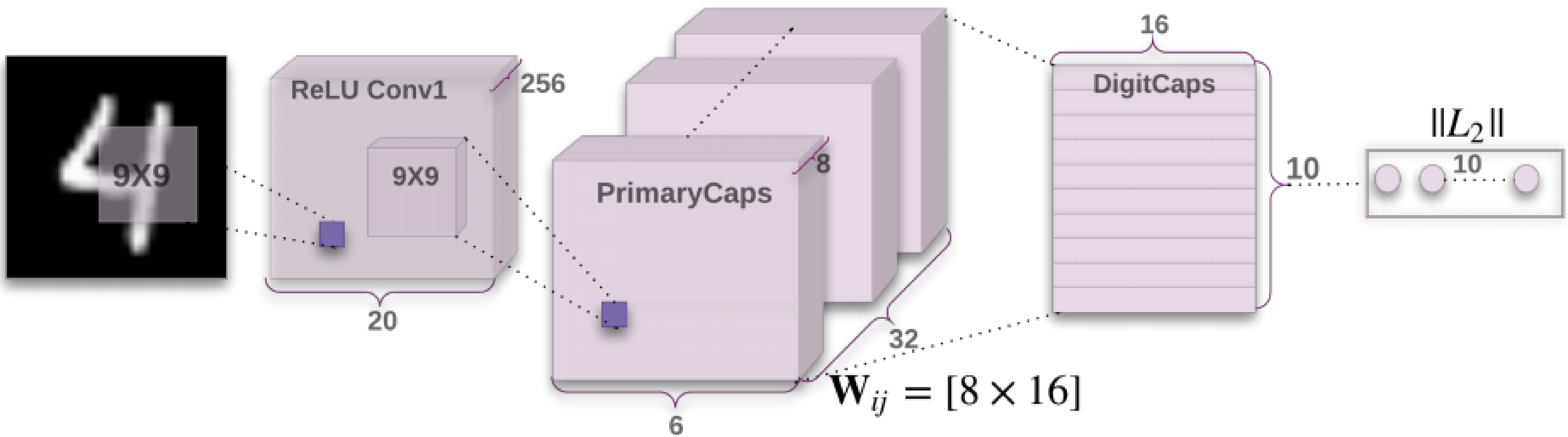
KAGGLE

- Competitions,
- 100+ Datasets,
- Community,
- Many code examples,
- Many CNNs challenges!

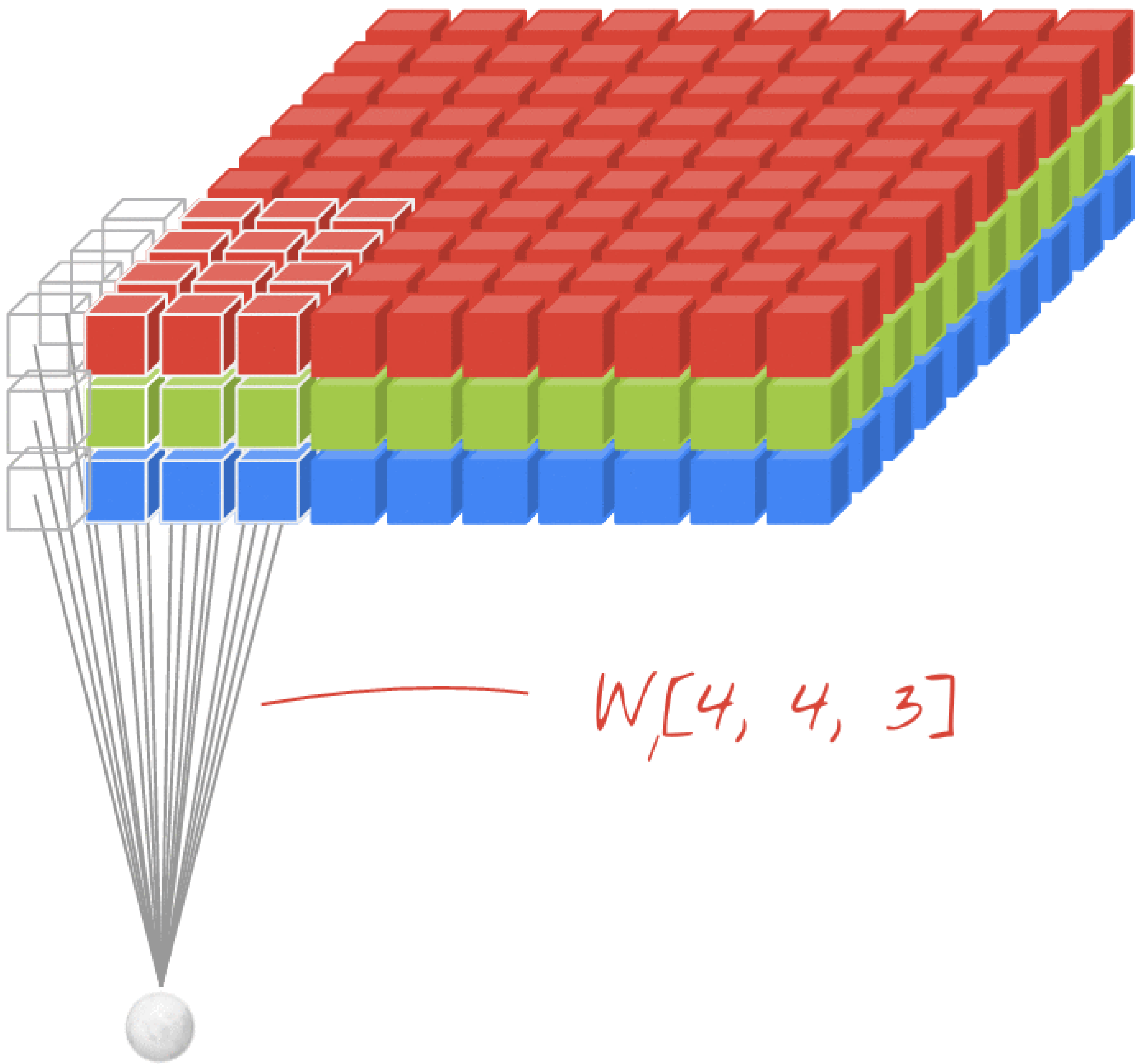
WHAT'S NEXT?

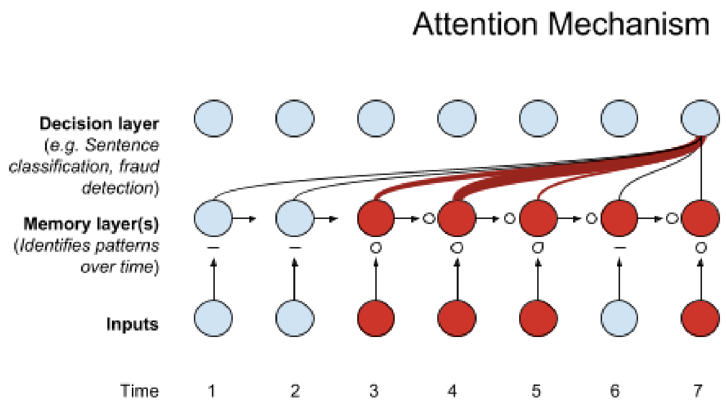
Future of CNNs...

CAPSNET


9X9
ReLU Conv1
9X9
256
20
PrimaryCaps
8
32
6
$\mathbf{W}_{ij} = [8 \times 16]$
16
DigitCaps
10
$\|L_2\|$
10

3D CONVOLUTIONS





THANKS FOR ATTENTION!

Q&A
+
Let's code!