Workshop of Processing and analyzing micrographs with artificial intelligence

Instructor : M.Sc. Setareh Medgahlchi

Place:Nanomechanical testing in Materials Research and Development VIII, Split, Croatia

Date: 02.Oct.2022

NHR4CES NHR for Computational Engineering Science

Institut für Metallkunde und Materialphysik

RWTH AACHEN UNIVERSITY

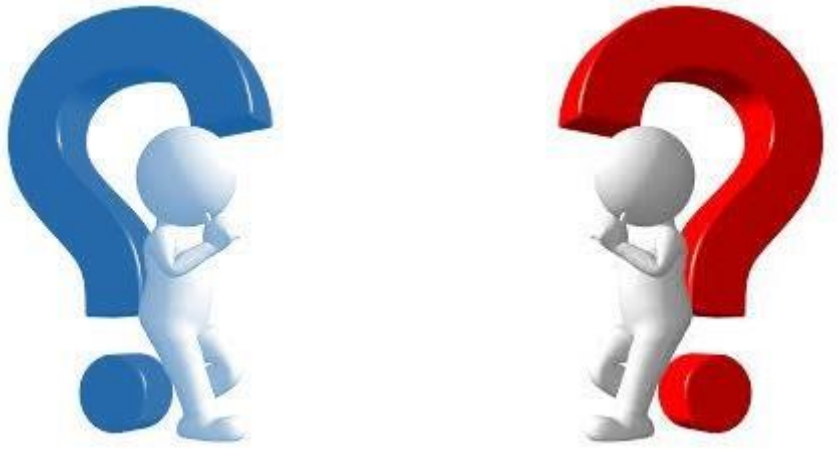
Outlook

Introduction and application to some image analysis tools	Notebook
Introduction to the application of machine learning	Slides
Application of machine learning techniques on data • Supervised learning methods • Unsupervised learning methods	Notebooks

GitHub - SetarehMed/Micrograph_Process_AI_ECI_2022_Binder

SetarehMed/Micrograph_Process_AI_ECI_2022_Binder (github.com)

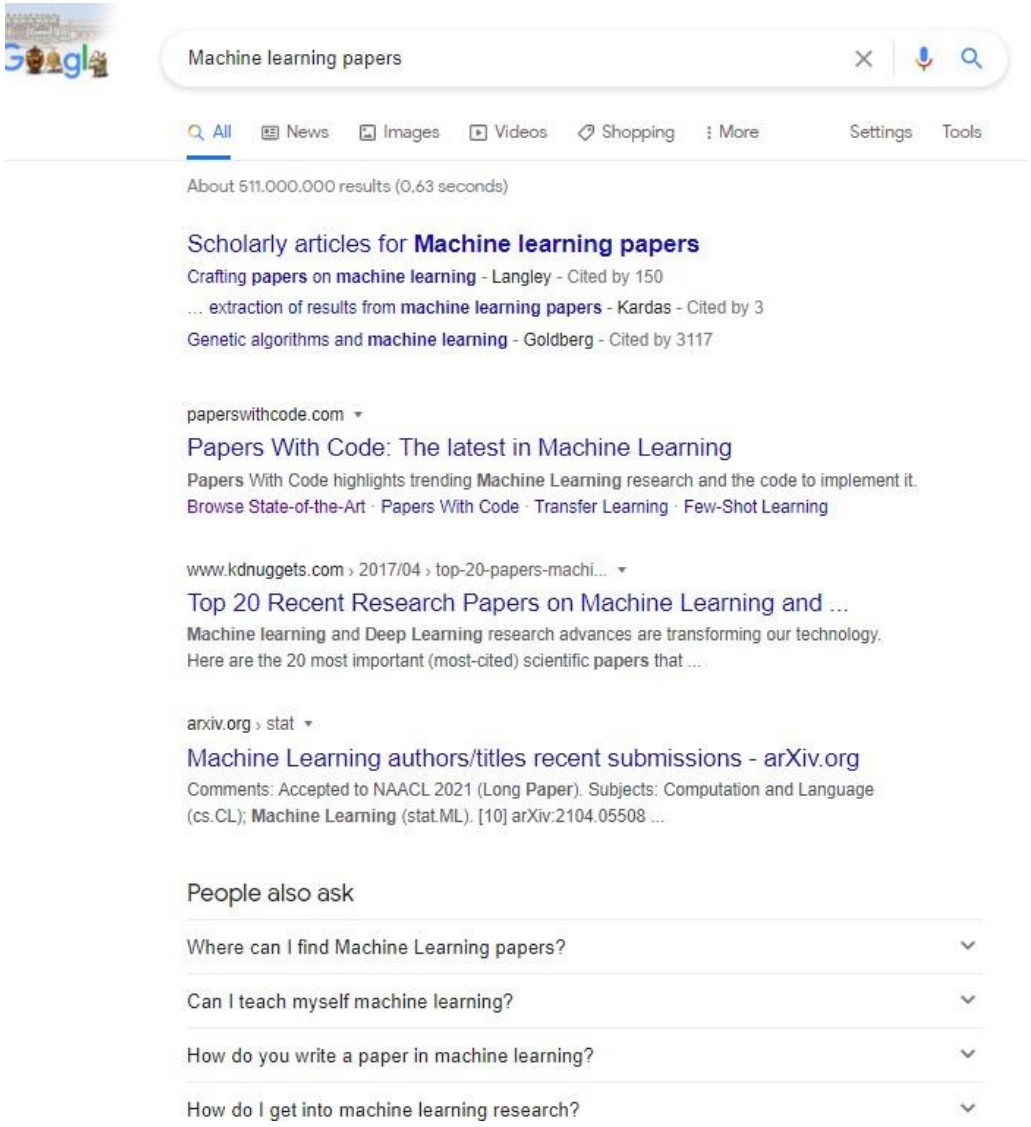
Machine Learning ?? Machine ?! Learning ?!



- Do you know what can be Machine Learning ? What comes to your mind ?
- Can you think of an application ? Do you use machine learning systems already ?
- How deep has ML penetrated in our daily life ?

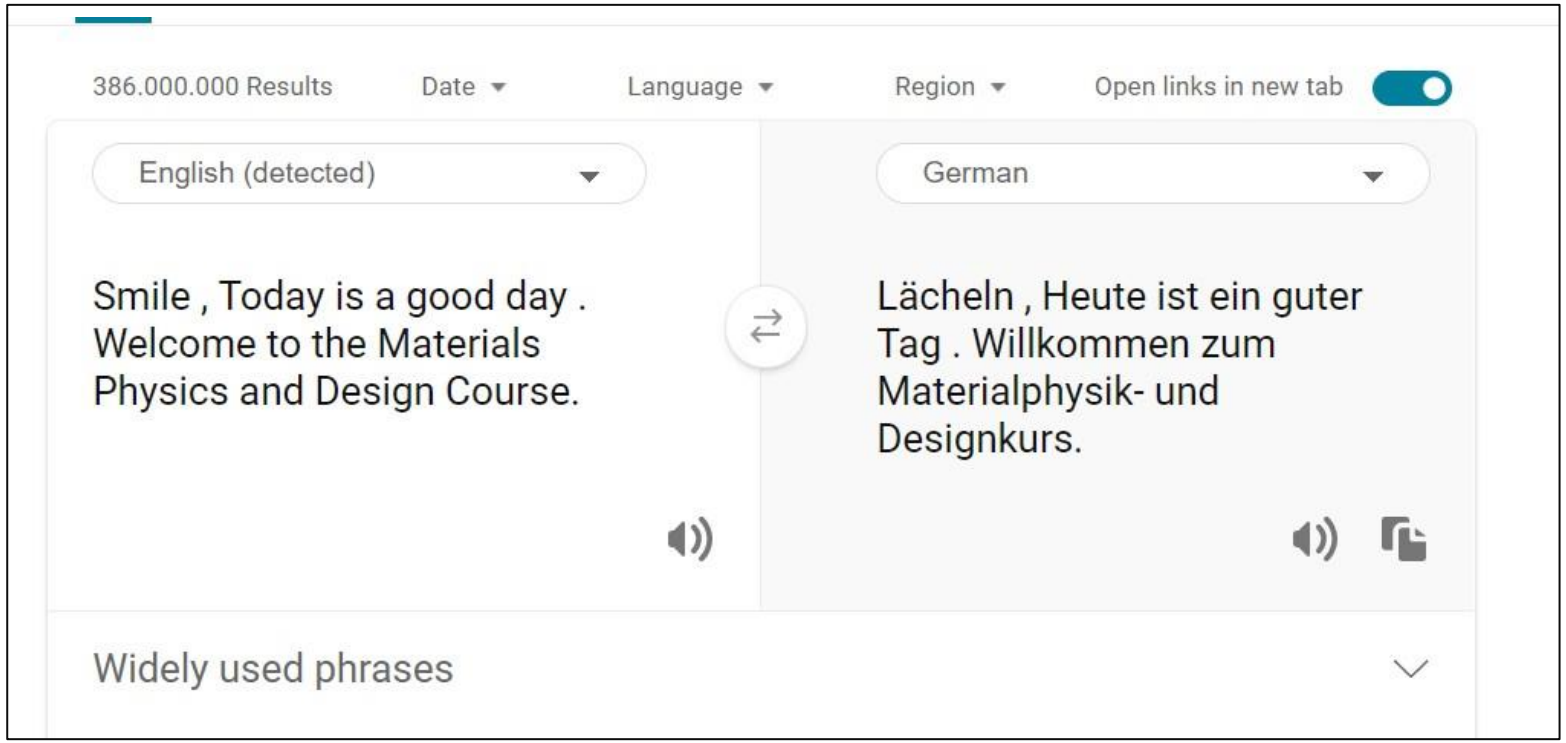
Applications in our daily life

Let‘s Google ... !
Infomation retrieval



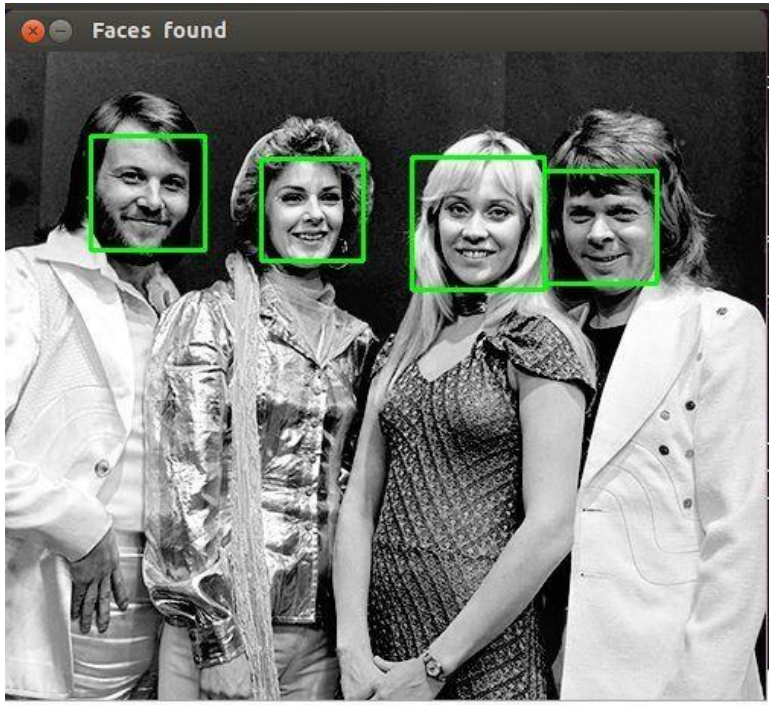
Applications in our daily life

- Machine translation
- Language Translation



Applications in our daily life

Face Detection



Applications in our daily life

Text Filtering, Spam filtering:

Applications in our daily life

Autonomous Driving Cars

When did it start ?

- **The first definition**
- Arthur Samuel (1959): Machine Learning is the field of study that gives the computer the ability to learn without being explicitly programmed.

- Tom Mitchell (1998): by machine learning a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Machine Learning- A slightly more formal definition

Goal of machine learning:

- ***Machines that learn to perform a task from experience***

- We can formalize this as

  y is called output variable,
  x the input variable and
  w the model parameters (typically learned)

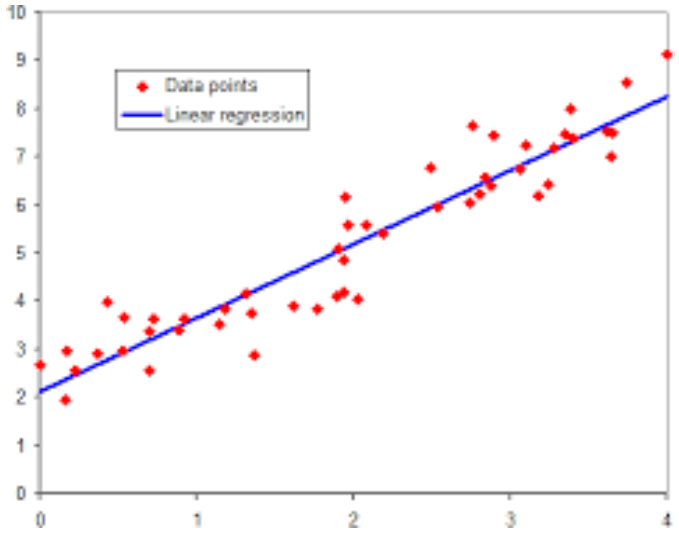
$$y = f(x, w)$$

Tools

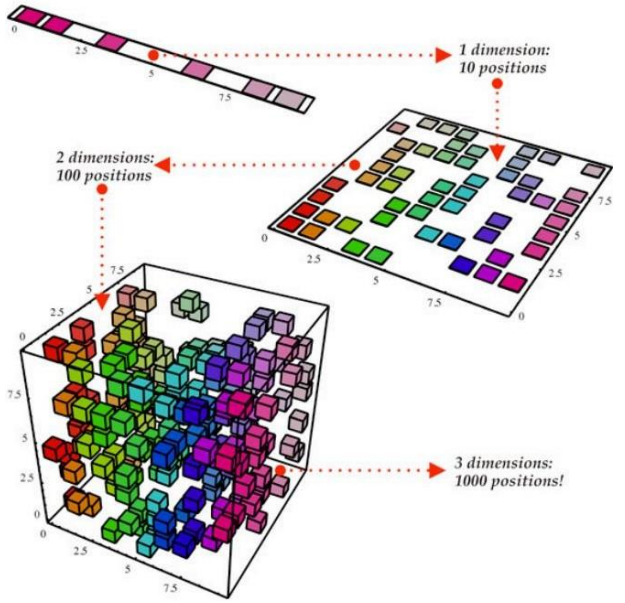
- Statistics
- Probability theory
- Decision theory
- Information theory
- Optimization theory

Approaches

Regression

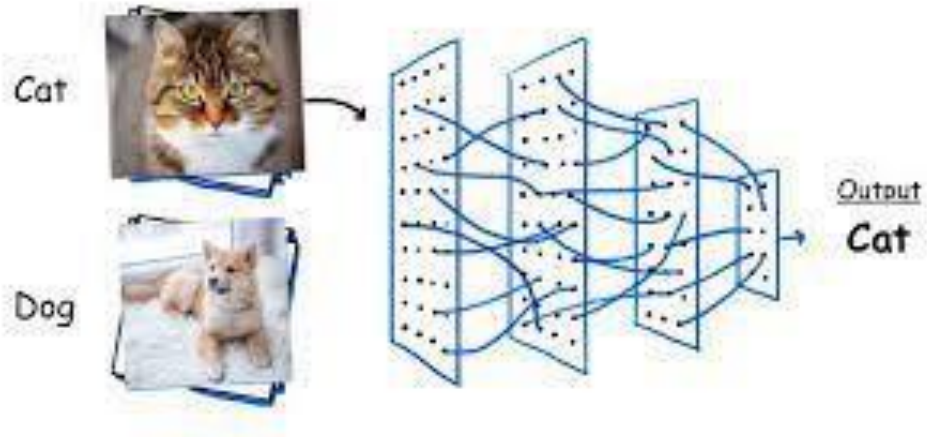


Dimensionality reduction



Clustering

Classification



General Classification of Machine Learning Approaches

Supervised Learning: Data with labels

- Input data with known labels are given .

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

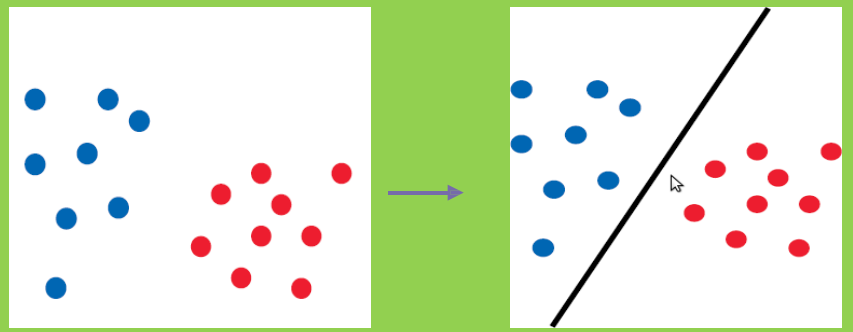
- Task : Learn the relationship beween the input and output.

$$y = f(x, w)$$

- Goal: To genralize to unseen data, in such a way that the network can predict the label for an unseen data.

Examples:

- Linear classifier:



Common Models:

Linear Classifier
Multilayer Perceptron (Backpropagation)
(Deep) Convolutional Neural Networks (Backpropagation)
Support Vector Machine (SVM)
Linear Regression, Logistic Regression
Boosting
Graphical models , . . .

General Classification of Machine Learning Approaches

Unsupervised Learning: Data without labels

- Some input data without labels are given

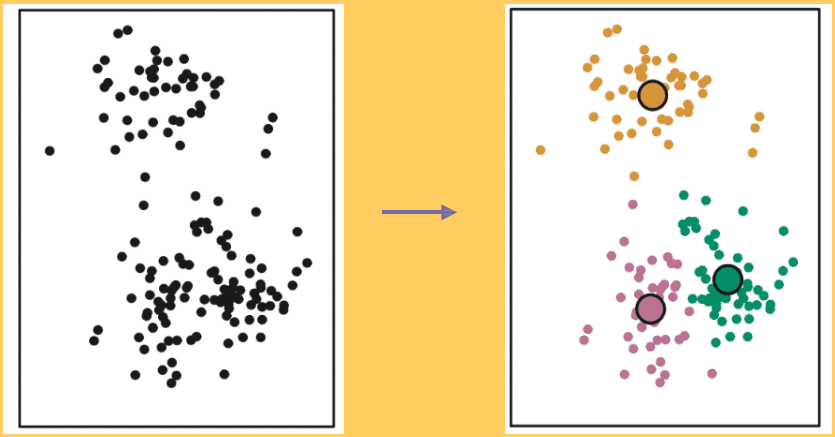
$D = \{x_1, x_2, \ldots, x_n\}$

- Tasks : Determine the data distribution p(x) → density estimation
- Visualize the data by projections → dimensionality reduction

- Goal: Find groupings of the data → clustering

Examples:

- Clustering



Common Models:

Mixture Models
K-Means clustering
Kernel Density Estimation
Neural Networks, e.g. Auto-Encoder Networks
Principal Component Analysis (PCA) , …

General Classification of Machine Learning Approaches

Semi-supervised Learning: Some data with and some without labels.

- a dataset of l labeled examples are given as in supervised learning

$$D_l = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$$

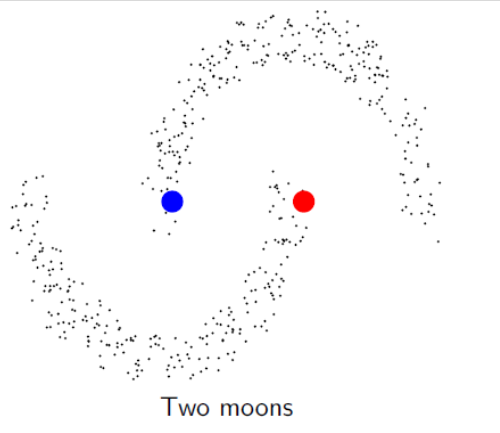
- additionaly a set of unlabeled examples as in unsupervised learning

$$D_u = \{x_{l+1}, x_2, \dots, x_{l+u}\}$$

- Goal :

$$y = f(x, w)$$

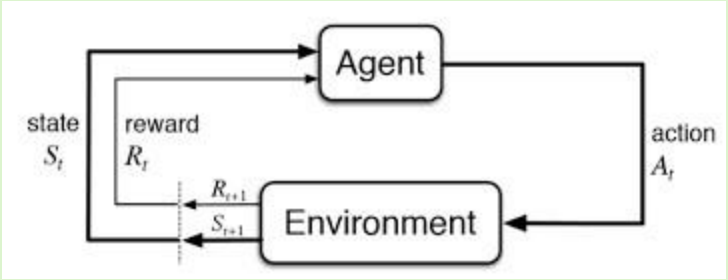
Example :
Red and blue poitns are labeled and the rest of black Points are unlabled.


Two moons

General Classification of Machine Learning Approaches

Reinforcement Learning: Feedback/rewards

- Problems involving an agent interacting with an enviroment, which provides numeric reward signals.
- Feedback:
  - how well we are doing
  - we do not get the feedback what the best action would be (indirect taching)
- Feedback as reward:
  - each action yields reward
  - a reward is given at the end



Example :

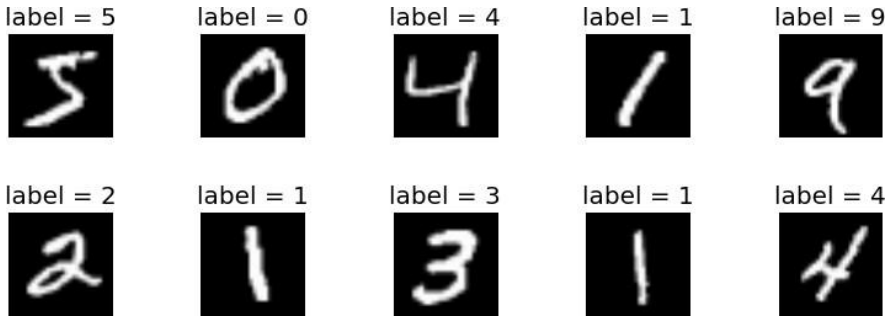
robot has found his goal, computer has won game in Backgammon.

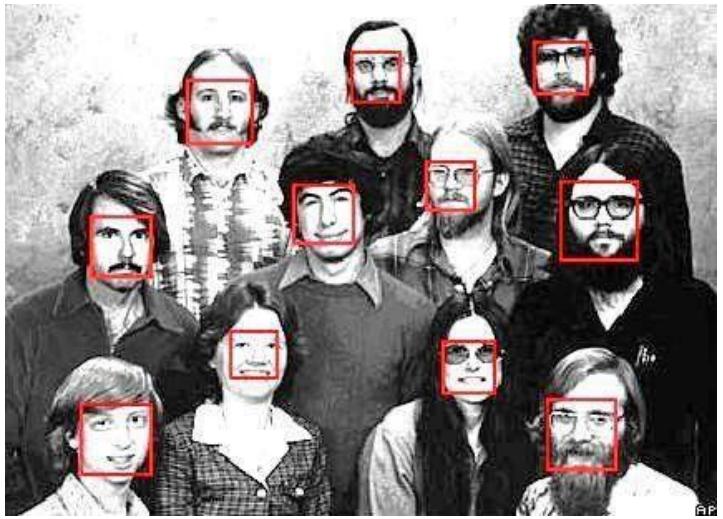
Exploration: try out new actions
Exploitation: use known actions that yield high rewards
Find a good trade-off between exploration and exploitation

Dig into some examples

Check your Knowledge: To which class belong the following examples ?

Person – Object – Speech recognition / identification / classifcation	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	



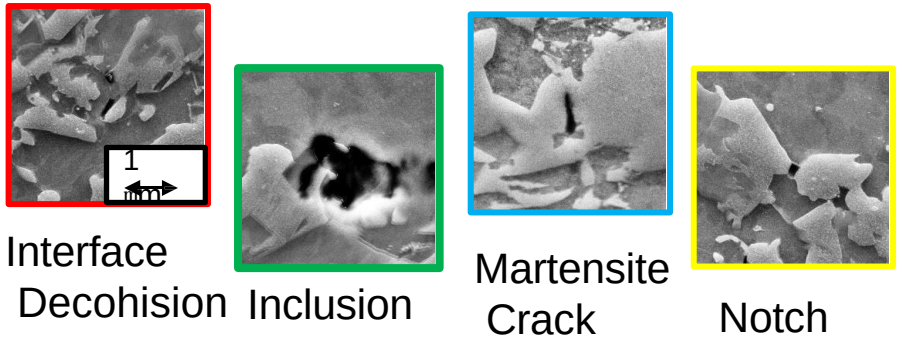


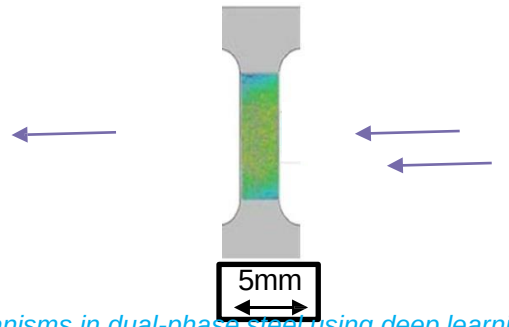
Dig into some examples

Check your Knowledge: To which class belong the following examples ?

Person – Object – Speech recognition / identification / classifcation	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	

Damage classification in the Dual phase steels:





C. Kusche et al. Large-area, high-resolution characterisation and classification of damage mechanisms in dual-phase steel using deep learning. PloS one, 2019.
S. Medghalchi et al. Damage Analysis in Dual-Phase Steel Using Deep Learning: Transfer from Uniaxial to Biaxial Straining Conditions by Image Data Augmentation, JOM, 2020

Dig into some examples

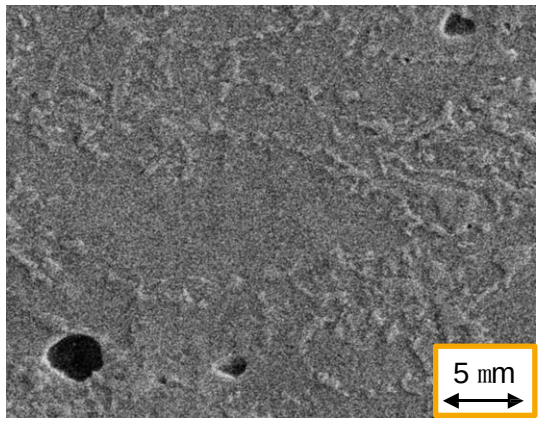
Check your Knowledge: To which class belong the following examples ?

Person – Object – Speech recognition / identification / classifcation	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	

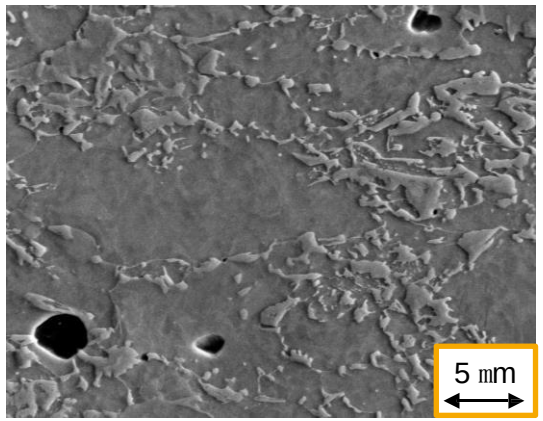
Dig into some examples

Check your Knowledge: To which class belong the following examples ?

Person – Object – Speech recognition / identification / classifcation	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	



Scan speed = 100 ns/ frame



Scan speed = 10 ms/ frame

Dig into some examples

Check your Knowledge: To which class belong the following examples ?

Person – Object – Speech recognition / identification / classifcation	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	

Dig into some examples

Check your Knowledge: To which class belong the following examples ?

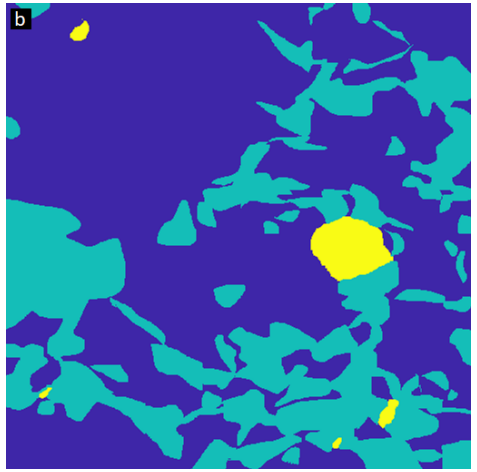
Person – Object – Speech recognition / identification / classifcation	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	

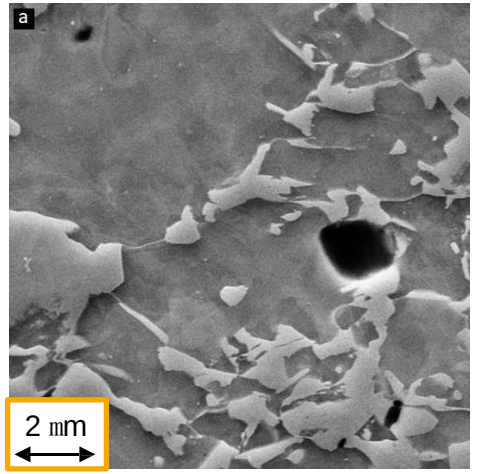
Dig into some examples

Check your Knowledge: To which class belong the following examples ?

Person – Object – Speech recognition / identification / classifcation	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	

Microstructural segmentation





Ferrite

Martensite

Damage

Setareh Medghalchi ,
Dario Plüsch,
Ehsan Karimi

Dig into some examples

Person – Object – Speech recognition / identification	SL
Credit card fraud detection	SL
Image denoising	USL
Image inpainting	USL
Prediction survival rate of a patient	SL
Clustering scientific publications according to topics	USL
Semantic Image Segmentation	SL

Few examples of state of art application of AI and in other aspects of metallurgy and materials science

Metallurgy and Process Sciences for Application of AI in Steel Manufacturing.

- AI system connects every part of complex steel manufacturing process to find an optimum solution that is integrated at a level that was previousely thought impossible.
- Every point of steelmaking process from the furnace to shipping bay can be analyzed and optimized for maximum quality and productivity.
- By using predictive data capabilities, steelmaking can be at the highest levels of accuracy and efficincy.
-

Prof. Paul Monks
University of Leichester

Few examples of state of art application of AI and in other aspects of metallurgy and materials science

Empirical learning in materials science.

- Learnning about the physical and mechanical properties of the materials by collecting the data.e.g. Predicting the Fatigue behaviour of single crystal Nickle super alloys used in turbine blades.

Prof. Harry Bhadeshia
University of Cambridge

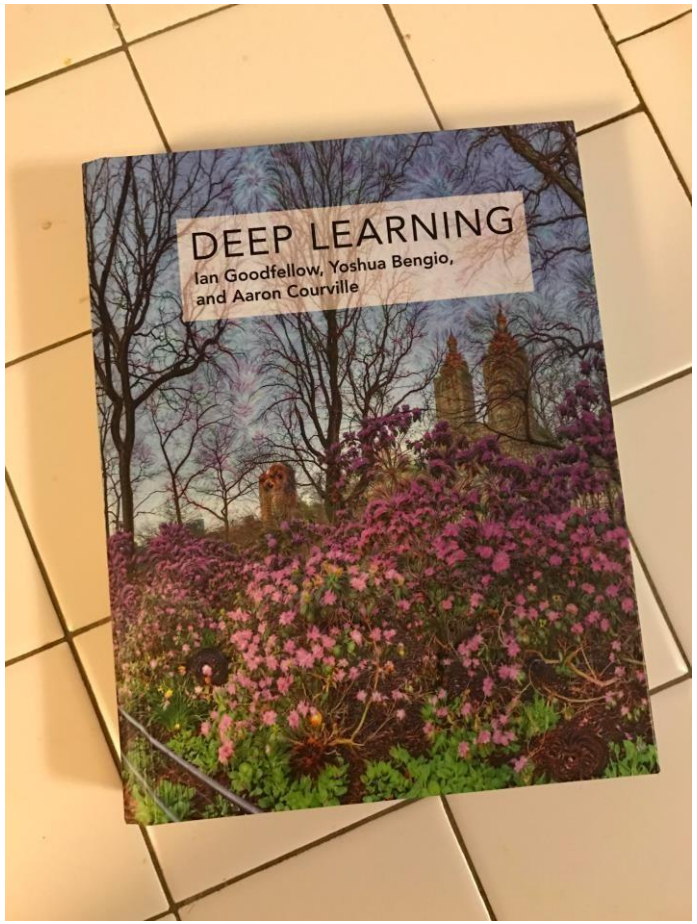
Image Classification using Convolutional Neural Networks (CNNs)

NHR4CES NHR for Computational Engineering Science

Institut für Metallkunde und Materialphysik | RWTH AACHEN UNIVERSITY

Optional book

Deep Learning by Goodfellow, Bengio, and Courville (Free online)



Introduction

Image classification is the task of taking an input image and outputting a class or a **probability of classes** that best describes the image

- For humans, this task is one of the first skills we learn and it comes **naturally** and **effortlessly** as adults

Being able to quickly **recognize patterns**, generalize from **prior knowledge**, and adapt to **different image environments** are difficult tasks for machines

Agenda

Introduction
- What we see vs. What computers see (MNIST and CIFAR Datasets)
- Hand-Crafted Features for Image Classification

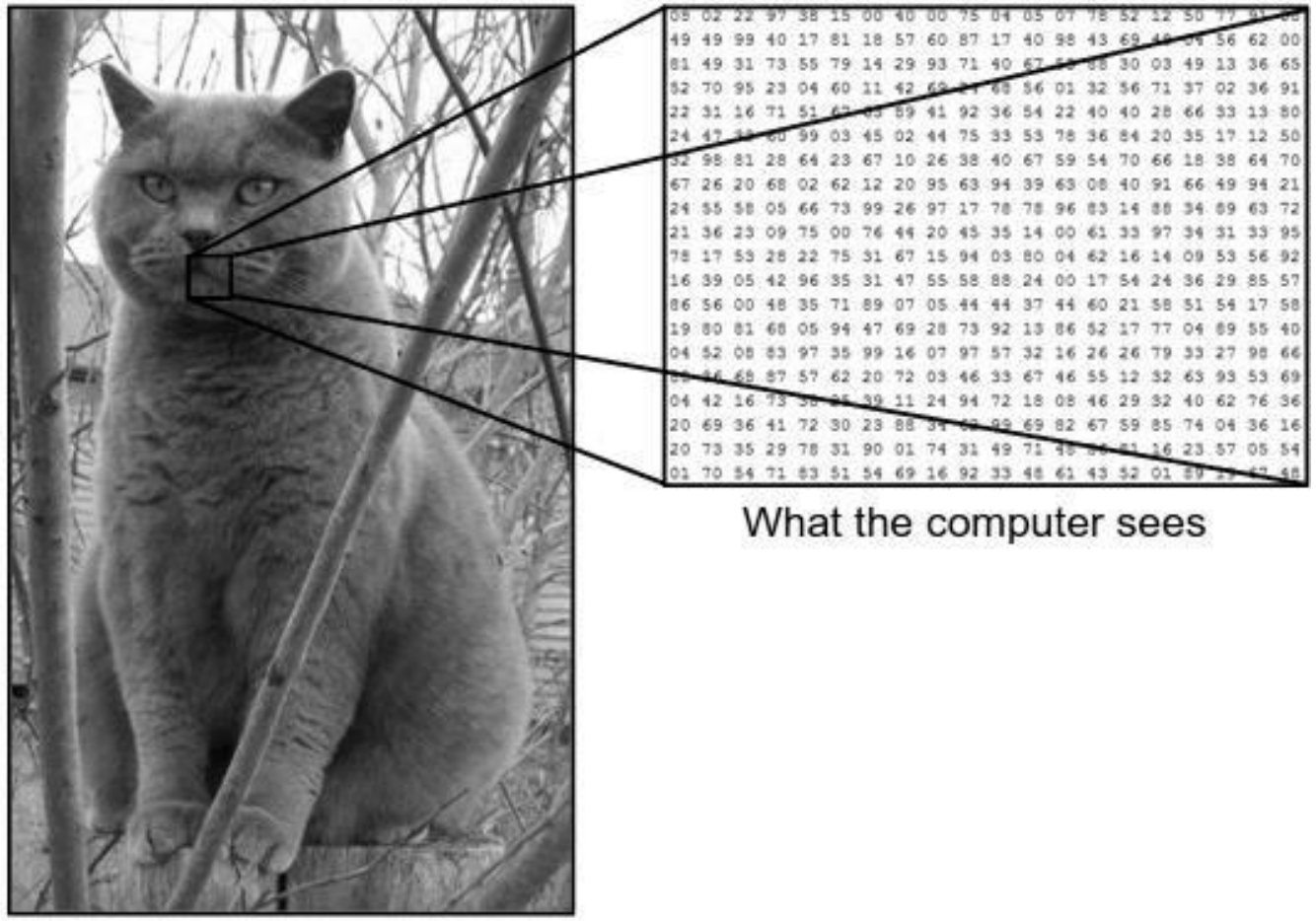
Deep Learning
- Convolutional Neural Networks (CNNs)
- Architecture (Convolutional, Pooling, and Fully Connected Layers)
- Successful CNN Architectures

Training
- Backpropagation
- Overfitting, Regularization and Dropout

Experiments

Introduction

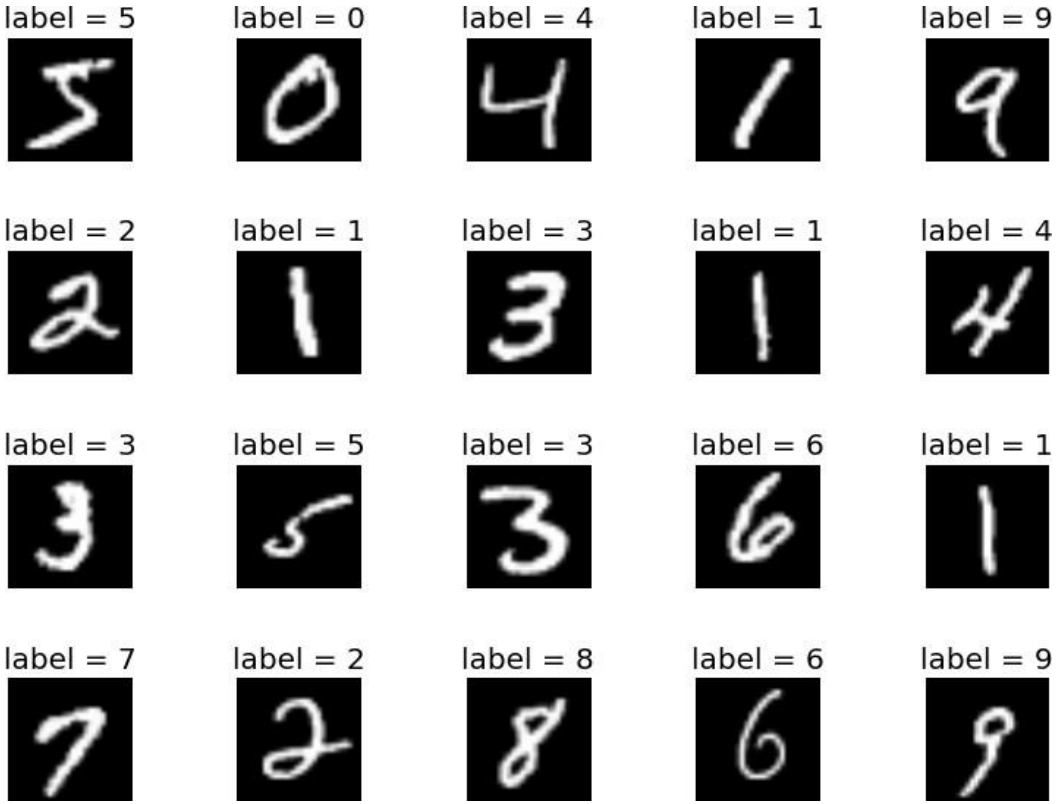
What we see vs. What computers see



Introduction

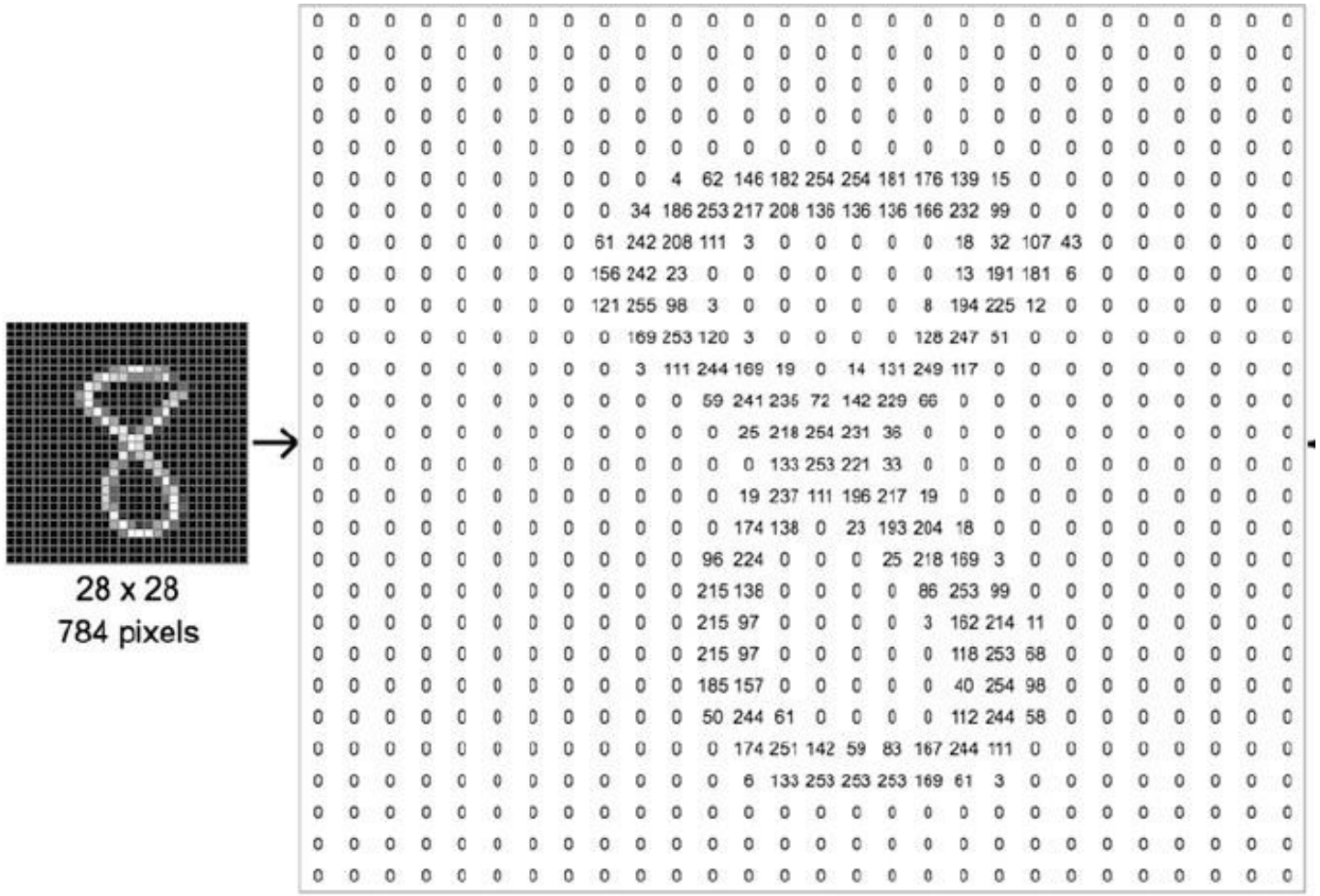
MNIST Dataset (***http://yann.lecun.com/exdb/mnist/***)

- 60,000 training examples
- 10,000 test examples
- Rank of best Classifiersand Errors



Introduction

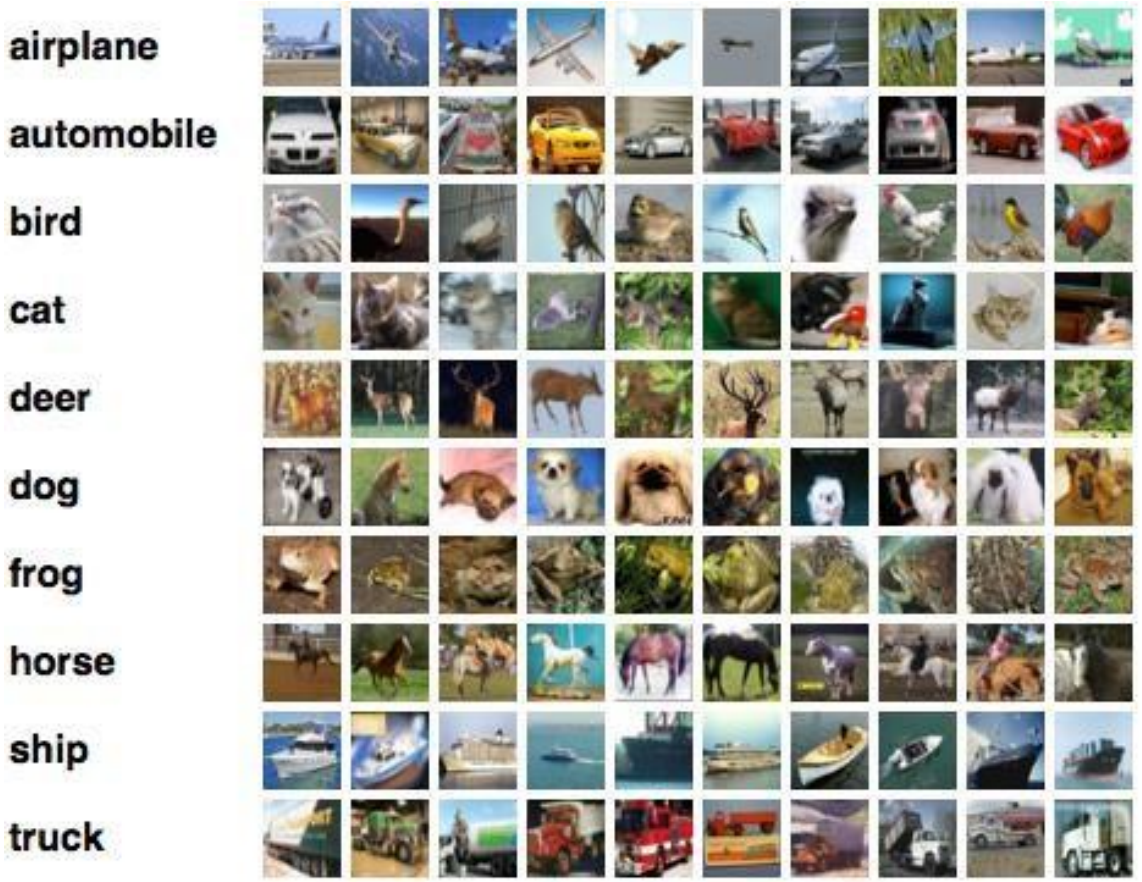
MNIST datasete



Introduction

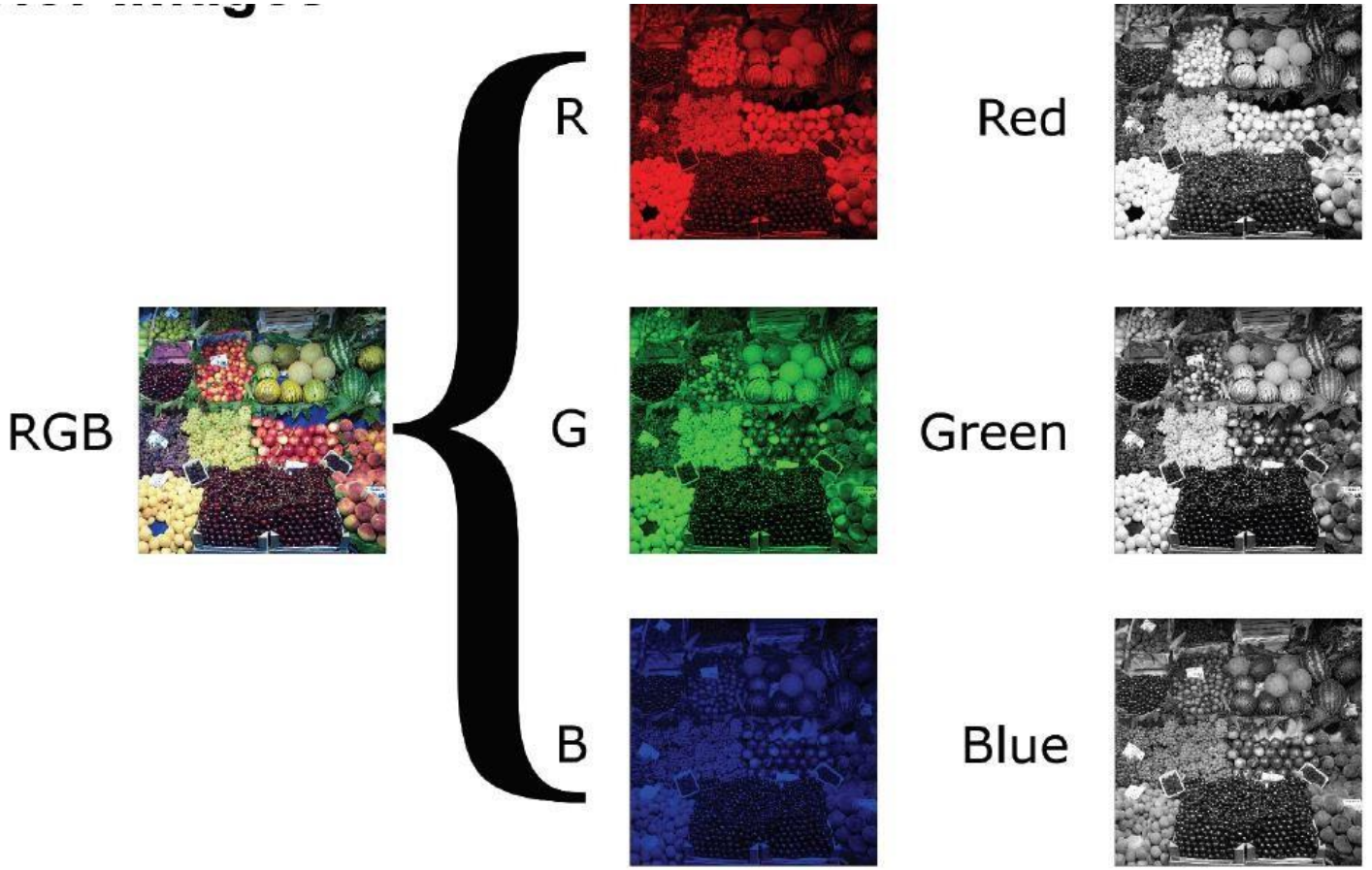
CIFAR datasete : (*https://www.cs.toronto.edu/~kriz/cifar.html*)

•Consists of 60,000 32x32 **color** images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.



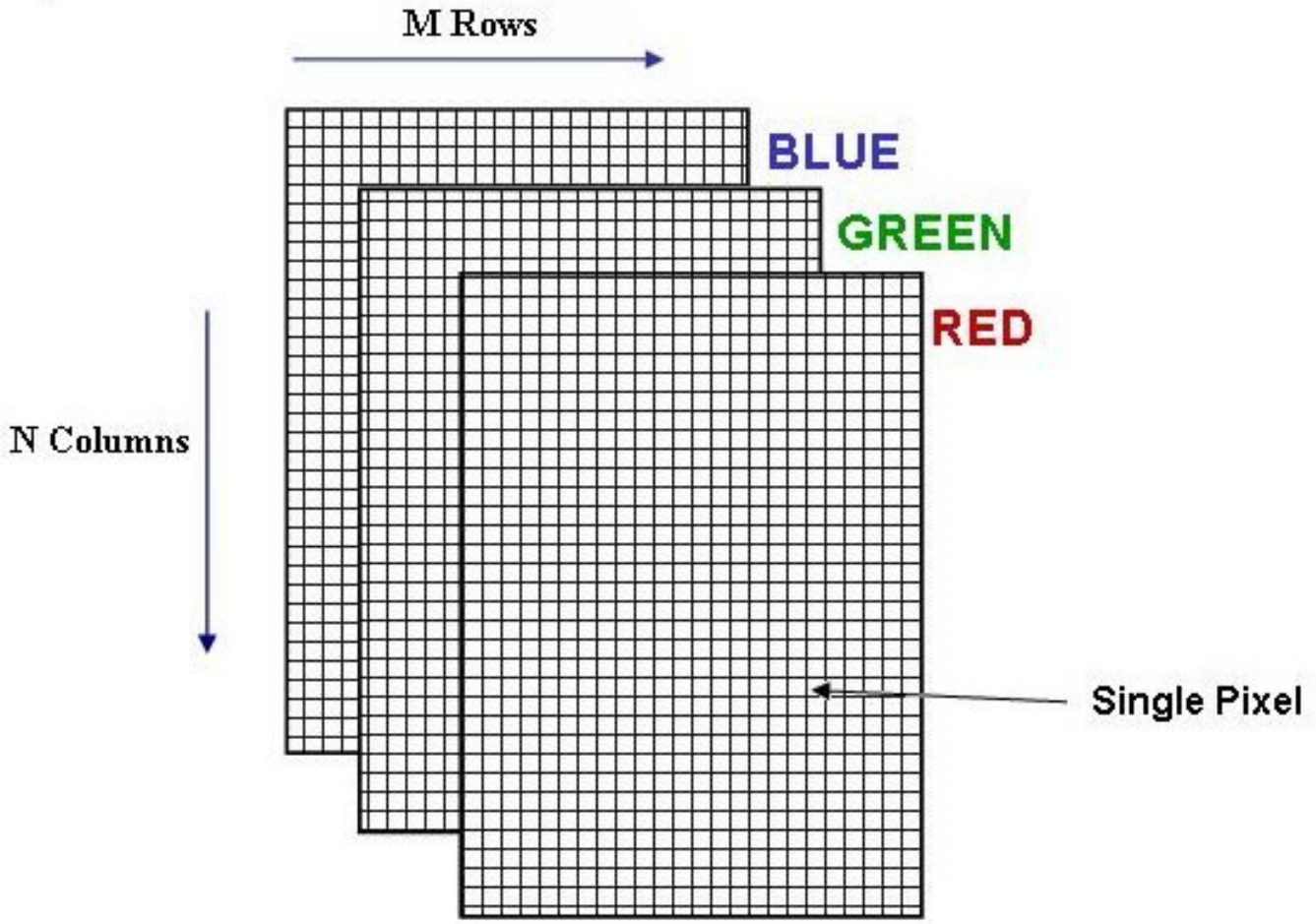
Introduction

Color images



Introduction

Color images



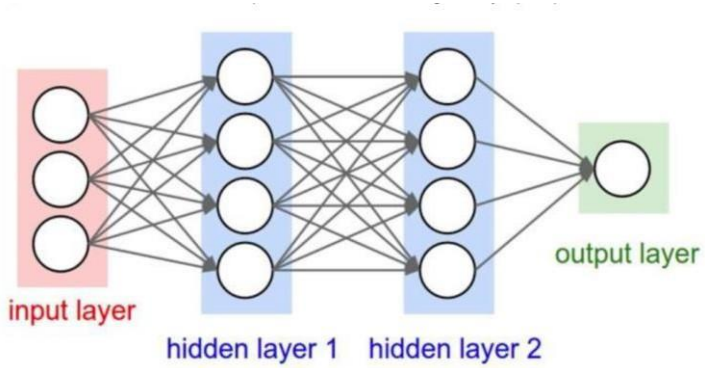
Deep Learning

"Deep Learning is a new area of Machine Learning, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence." http://deeplearning.net/

- Key Concepts of Deep Neural Networks

- Deep-learning networks are distinguished from the more common single-hiddenlayer-neural networks by their **depth**

- More than **three** layers (including input and output) qualifies as "deep" learning



- In deep-learning networks, each layer of nodes trains on a distinct set of features based on the **previous** layer's output
- The further you advance into the neural net, the more **complex** the features your nodes can recognize, since they aggregate and recombine features from the previous layer

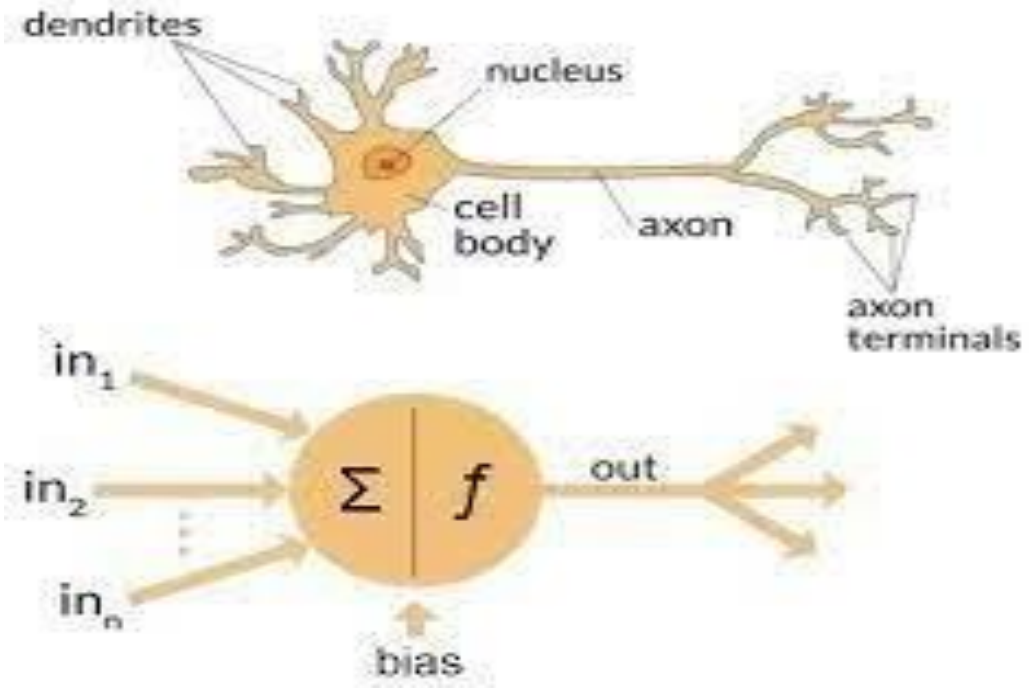
Convolutional Neural Networks (CNNs)

- CNNs take a **biological** inspiration from the visual cortex

- The visual cortex has small regions of cells that are sensitive to **specific regions** of the visual field

- For example, some neurons fired when exposed to **vertical** edges and some when shown **horizontal** or **diagonal** edges

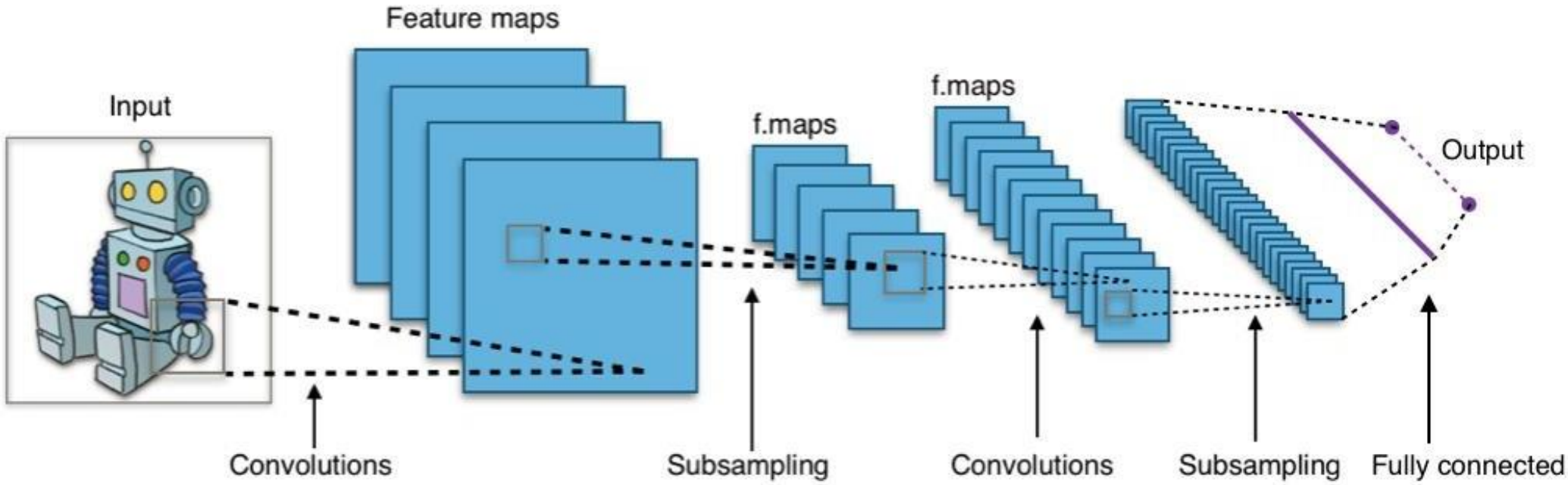
•Having the neuronal cells in the visual cortex looking for **specific characteristics** is the basis behind CNNs



Convolutional Neural Networks (CNNs)

Network Architecture

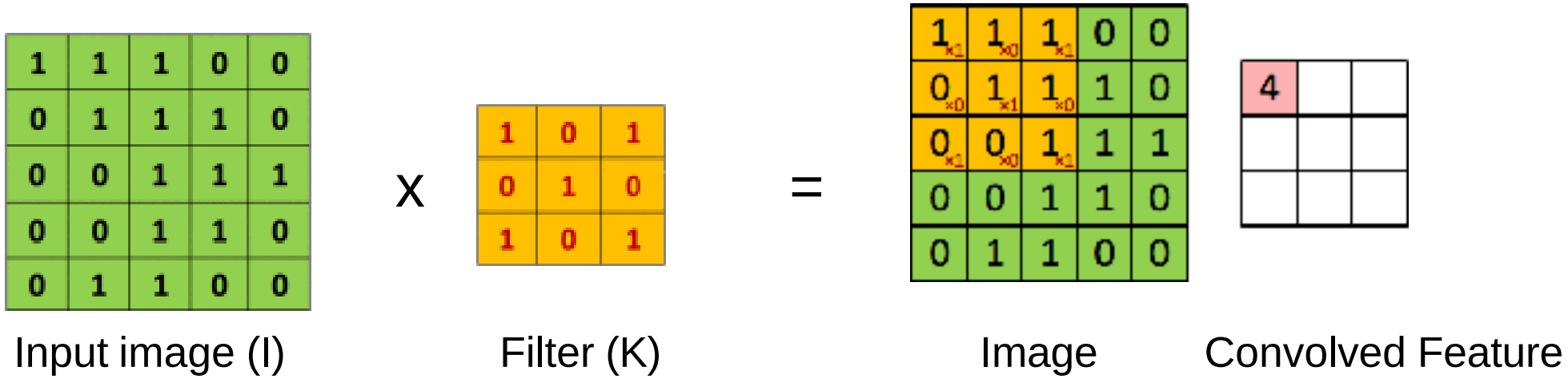
- Convolutional Layer, Pooling Layer, Fully Connected Layer



Convolutional Neural Networks (CNNs)

Convolution Operator

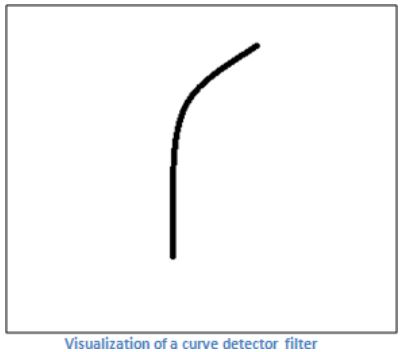
$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y-j-1}$$



Input image (I) Filter (K) Image Convolved Feature

•The 3×3 matrix (*K*) is called a '**filter**' or '**kernel**' or '**feature detector**' and the matrix formed by sliding the filter over the image and computing the dot product is called the '**Convolved Feature**' or '**Activation Map**' or the '**Feature Map**'.

Convolutional Neural Networks (CNNs)

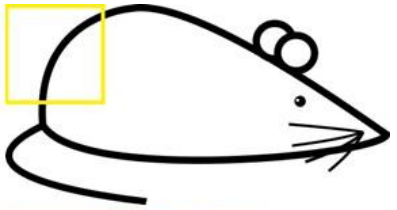


Visualization of a curve detector filter

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Original image



Visualization of the filter on the image

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

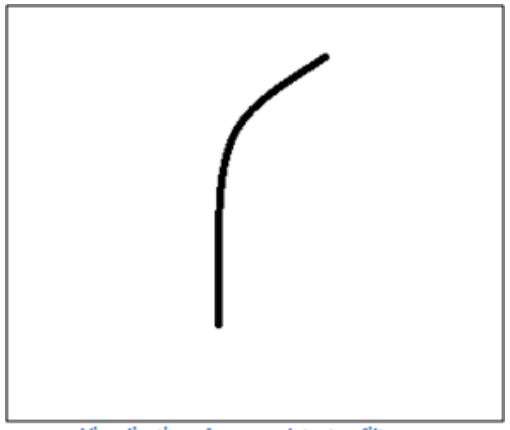
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Convolutional Neural Networks (CNNs)



Visualization of a curve detector filter

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

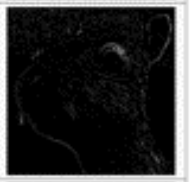
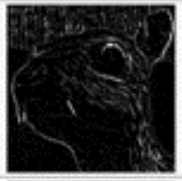
Multiplication and Summation = 0

Convolutional Neural Networks (CNNs)

Convolution Operator

- Different filters will produce different **Feature Maps** for the same input image. For example:

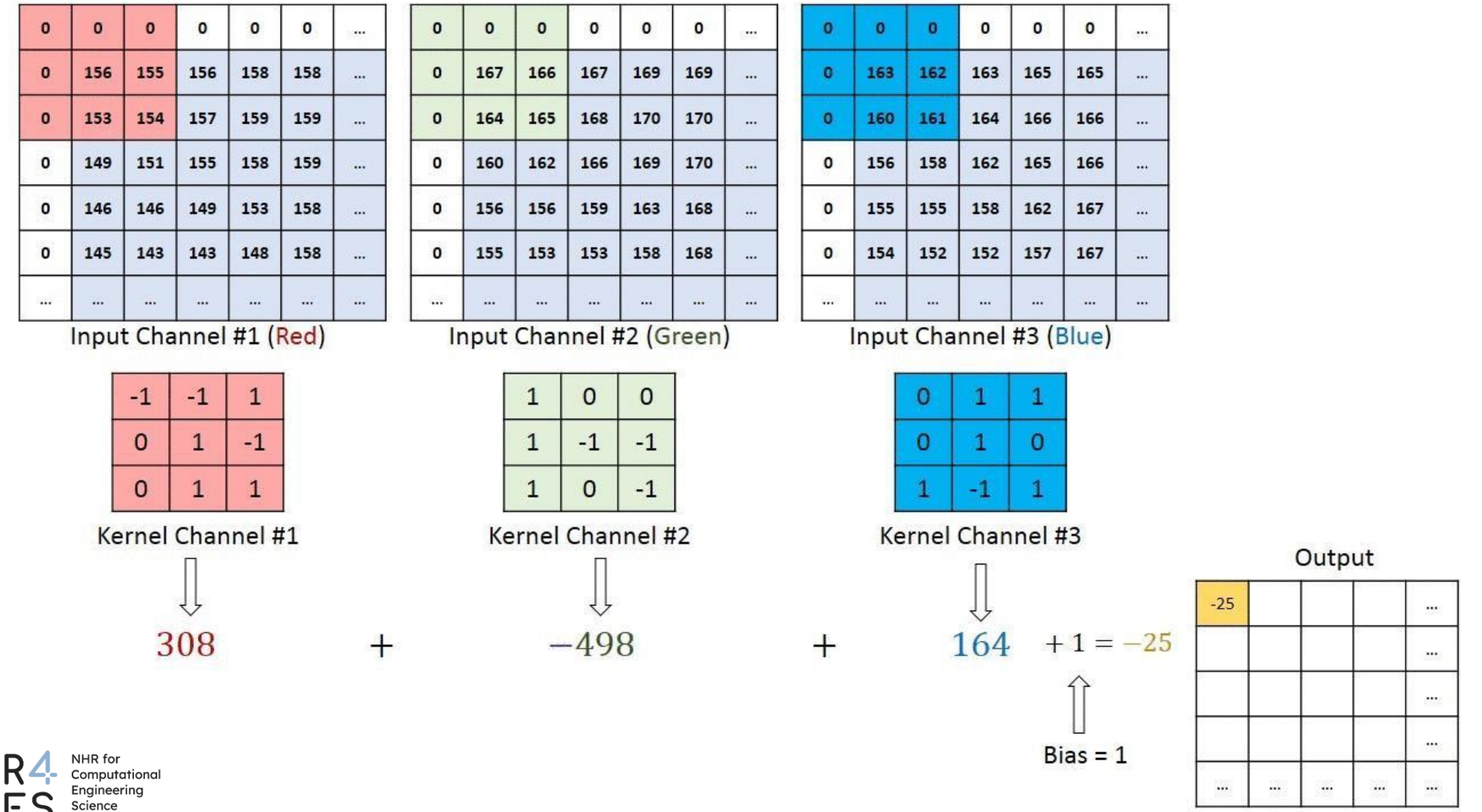
Input Image

Operation	Filter	Convolved Image
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	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Convolutional Neural Networks (CNNs)

CNN for color images:

$$\mathrm{conv}(I,K)_{xy} = \sigma\left(b + \sum_{i=1}^{h}\sum_{j=1}^{w}\sum_{k=1}^{d} K_{ijk}\cdot I_{x+i-1,y+j-1,k}\right)$$



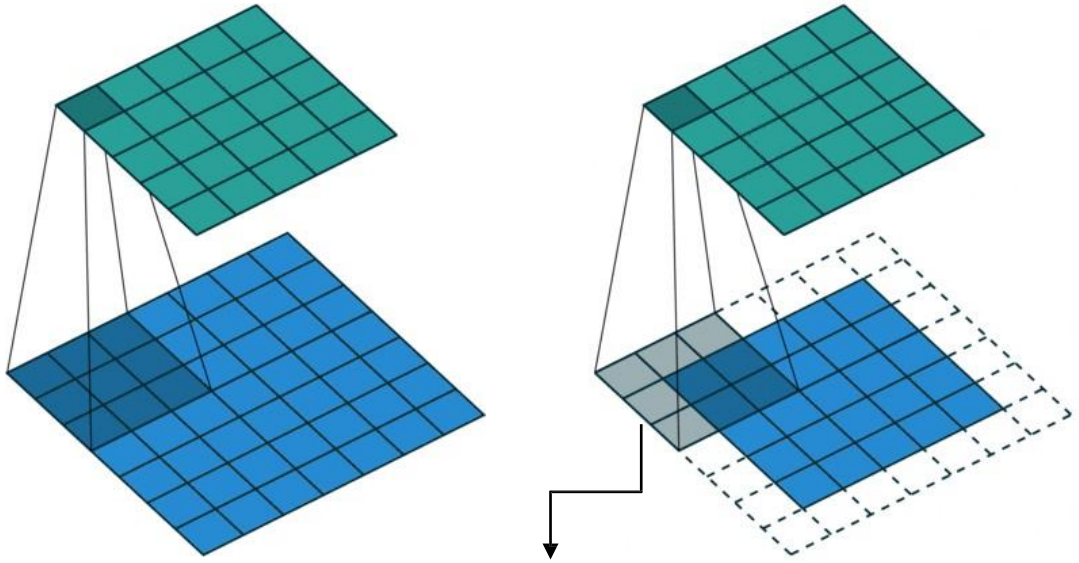
Convolutional Neural Networks (CNNs)

Convolutional Layer

• In practice, a CNN learns the values of these **filters** on its own during the **training process**

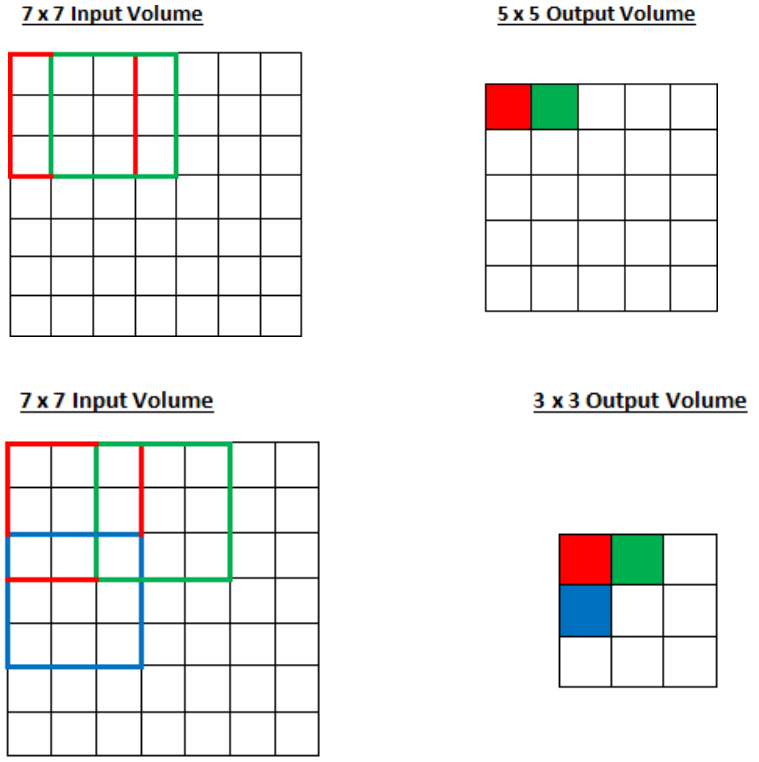
•Although we still need to specify parameters such as **number of filters**, **filter size**, **padding,** and **stride** before the training process

- **Padding：**



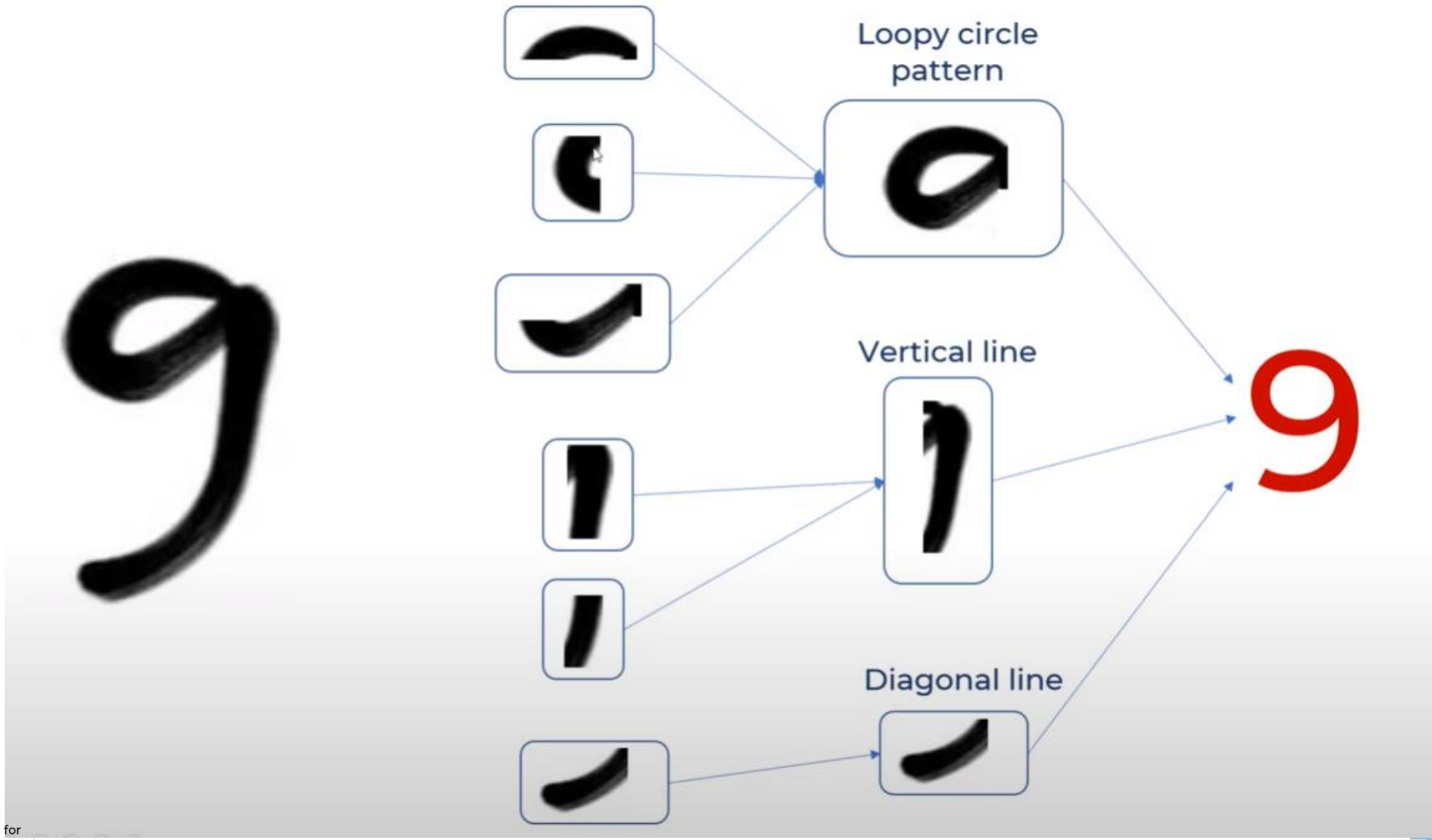
- Padding is the method for preserving image data by avoiding shrinking the image and loosing the data on the edges of the image. Here , we have padding = 1

- **Stride：**



- Stride is the size of filter window jumping over the imput image.

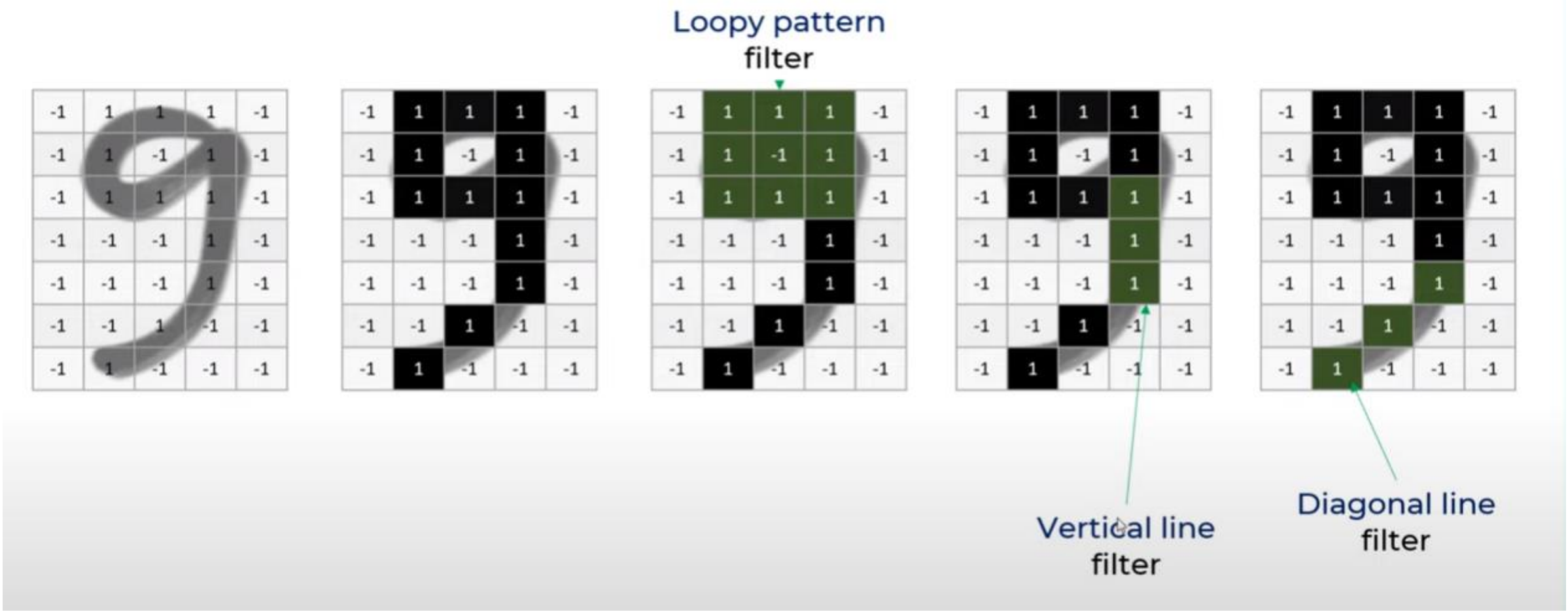
Example Case



https://www.youtube.com/watch?v=zfiSAzpy9NM

Convolutional Neural Networks (CNNs)

Different Feature Maps



Convolutional Neural Networks (CNNs)

-1+1+1-1-1-1-1+1+1 = -1 → -1/9 = -0.11

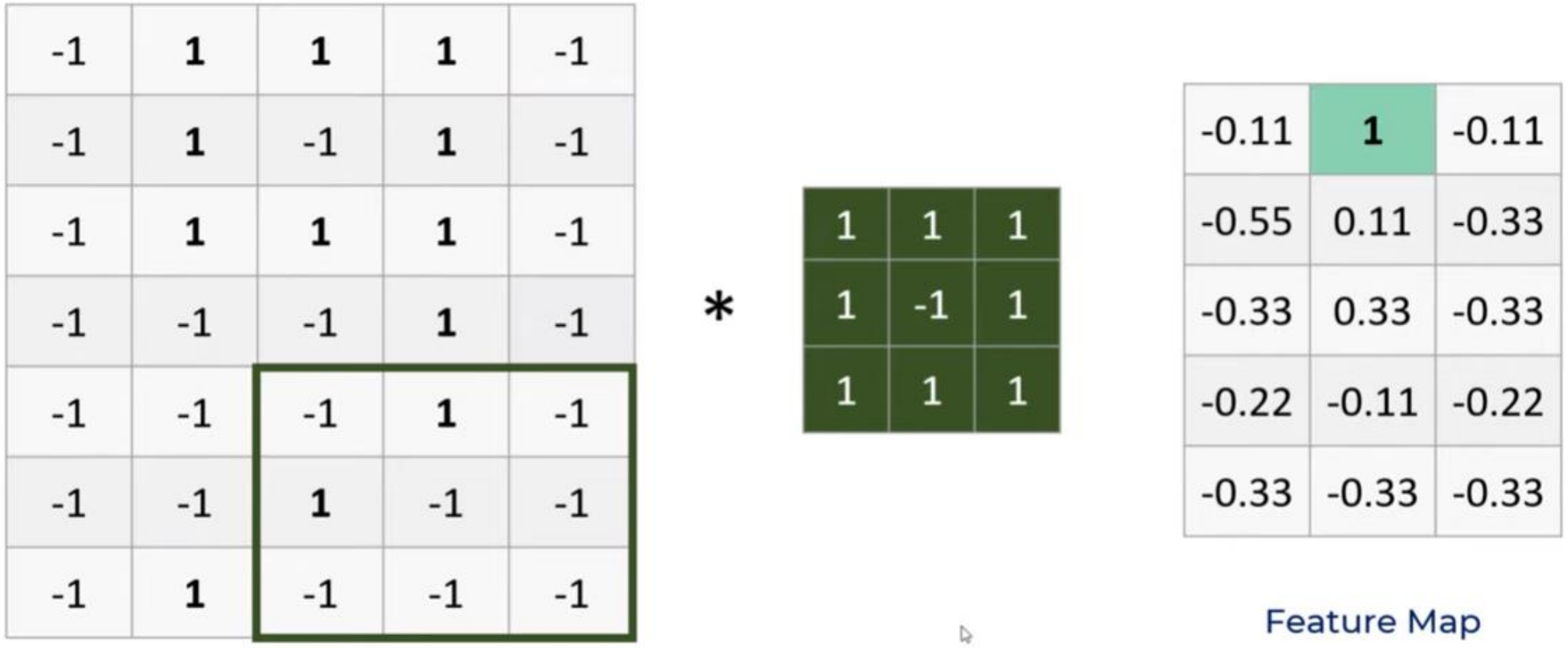
-1	**1**	**1**	**1**	-1
-1	**1**	-1	**1**	-1
-1	**1**	**1**	**1**	-1
-1	-1	-1	**1**	-1
-1	-1	-1	**1**	-1
-1	-1	**1**	-1	-1
-1	**1**	-1	-1	-1

*

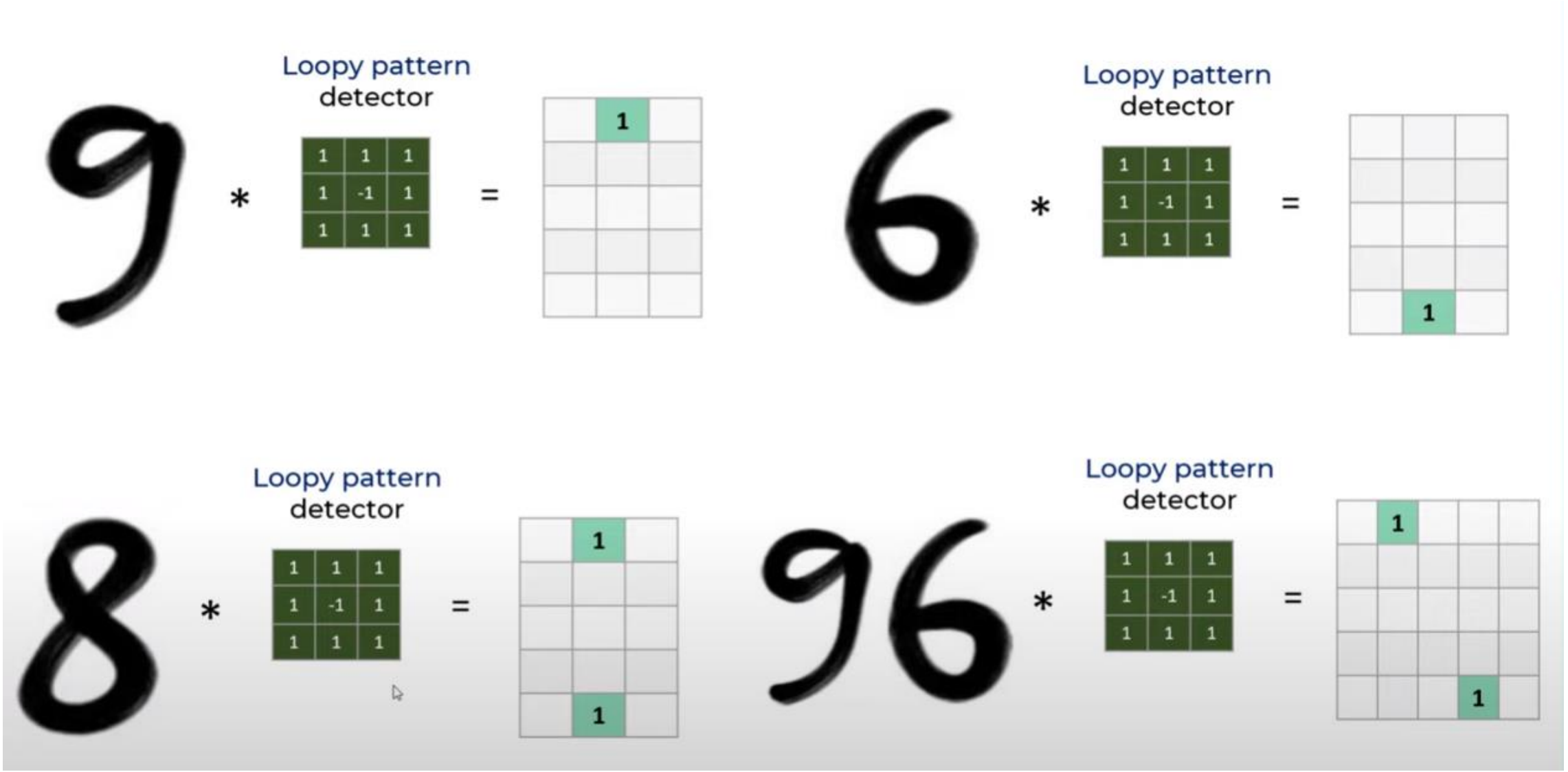
1	1	1
1	-1	1
1	1	1

-0.11		

Convolutional Neural Networks (CNNs)

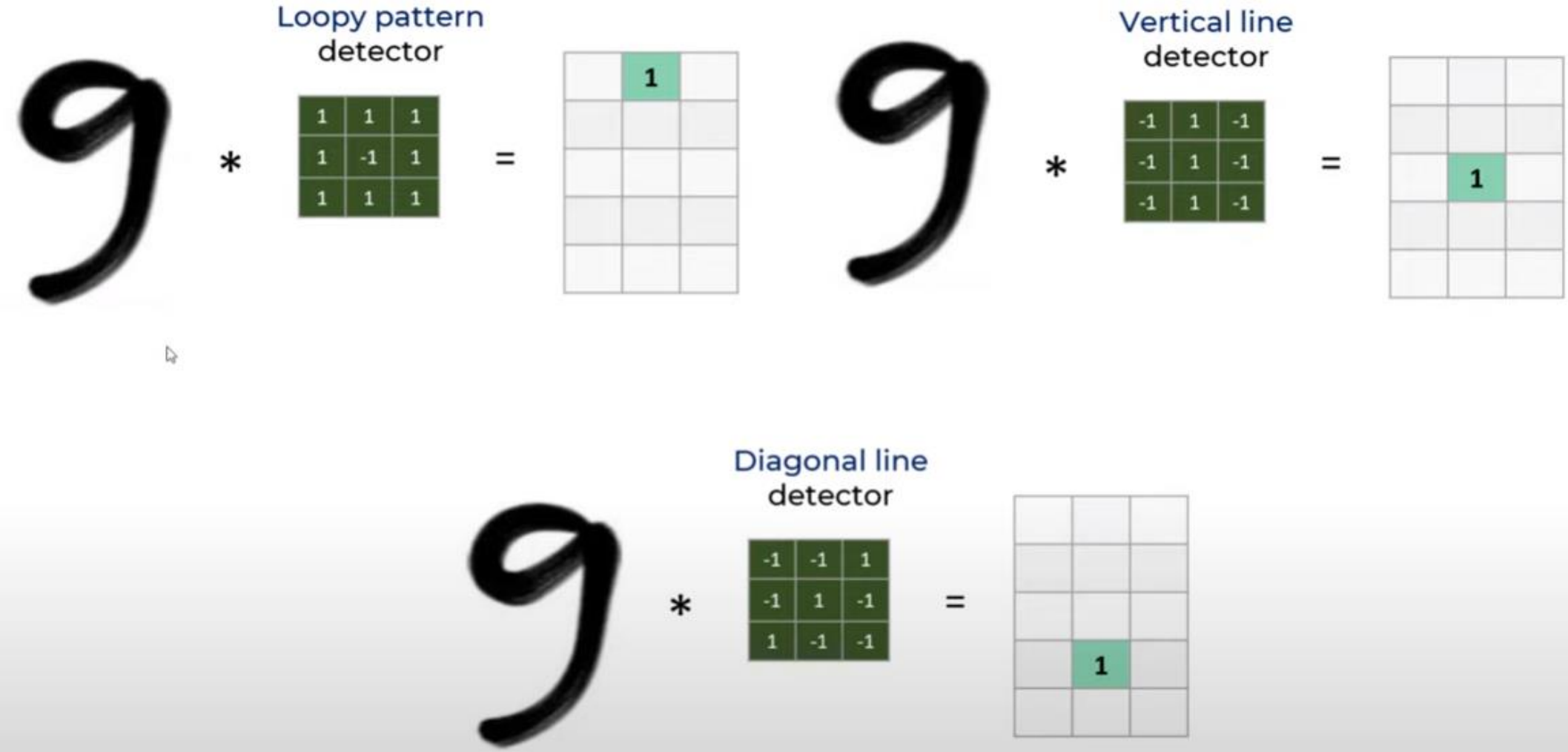


Convolutional Neural Networks (CNNs)



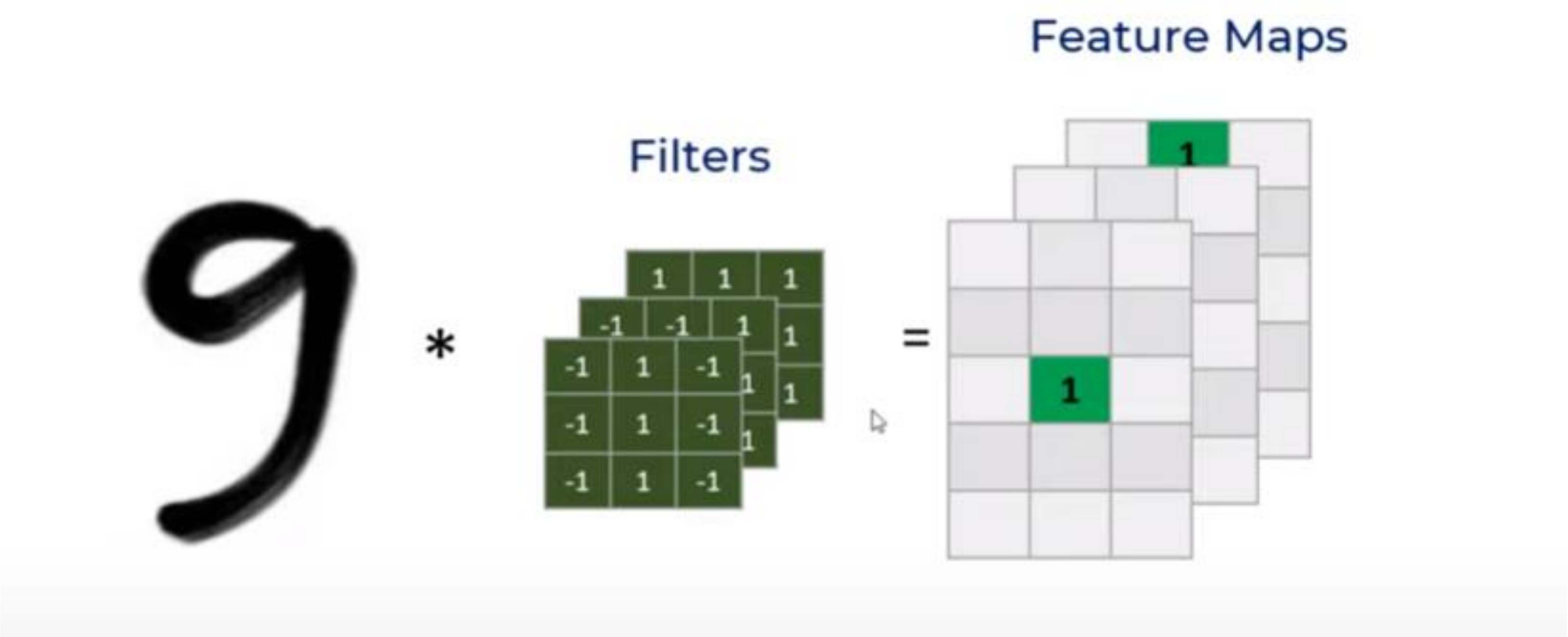
Convolutional Neural Networks (CNNs)

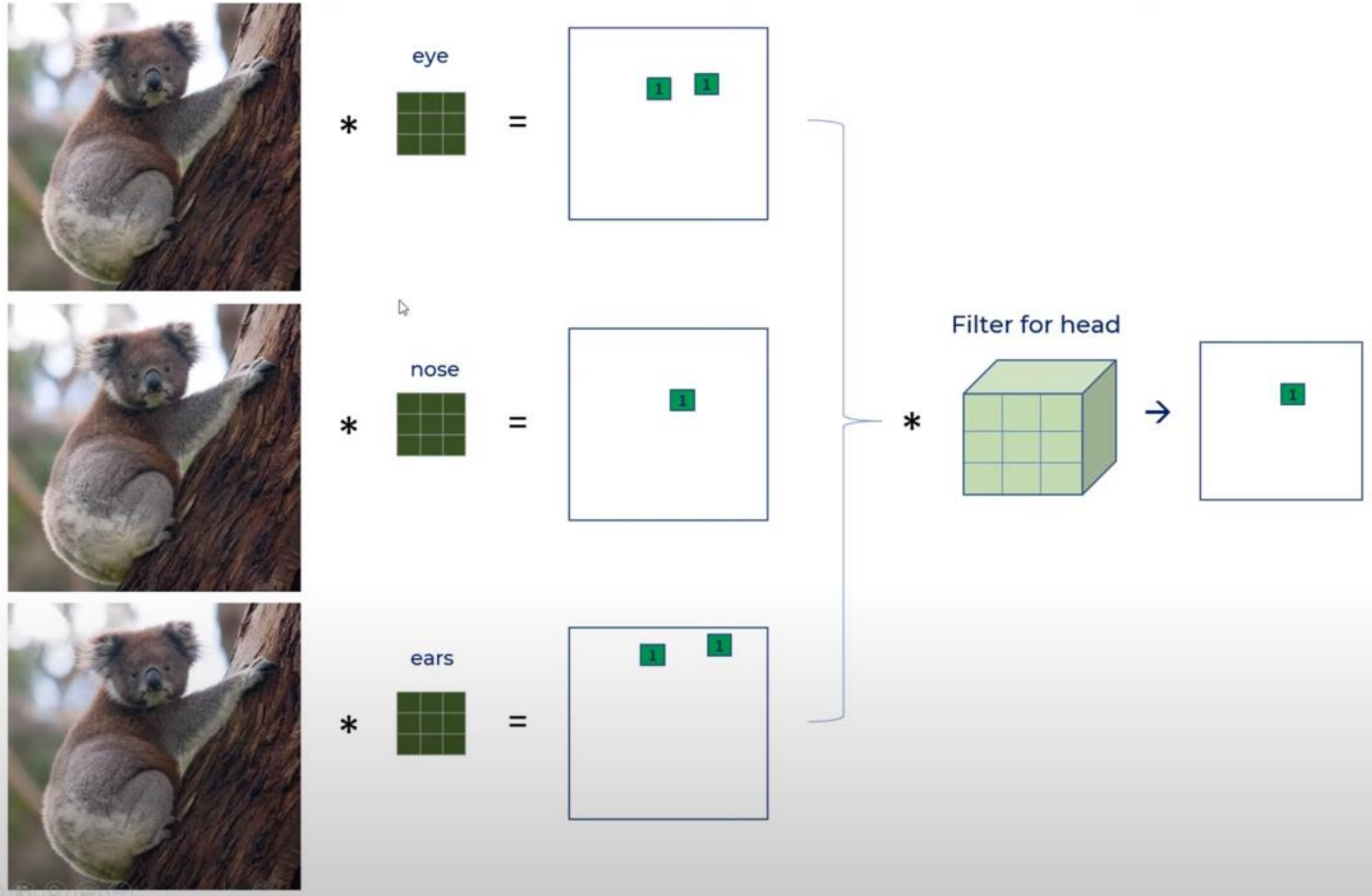
Different Features Detector



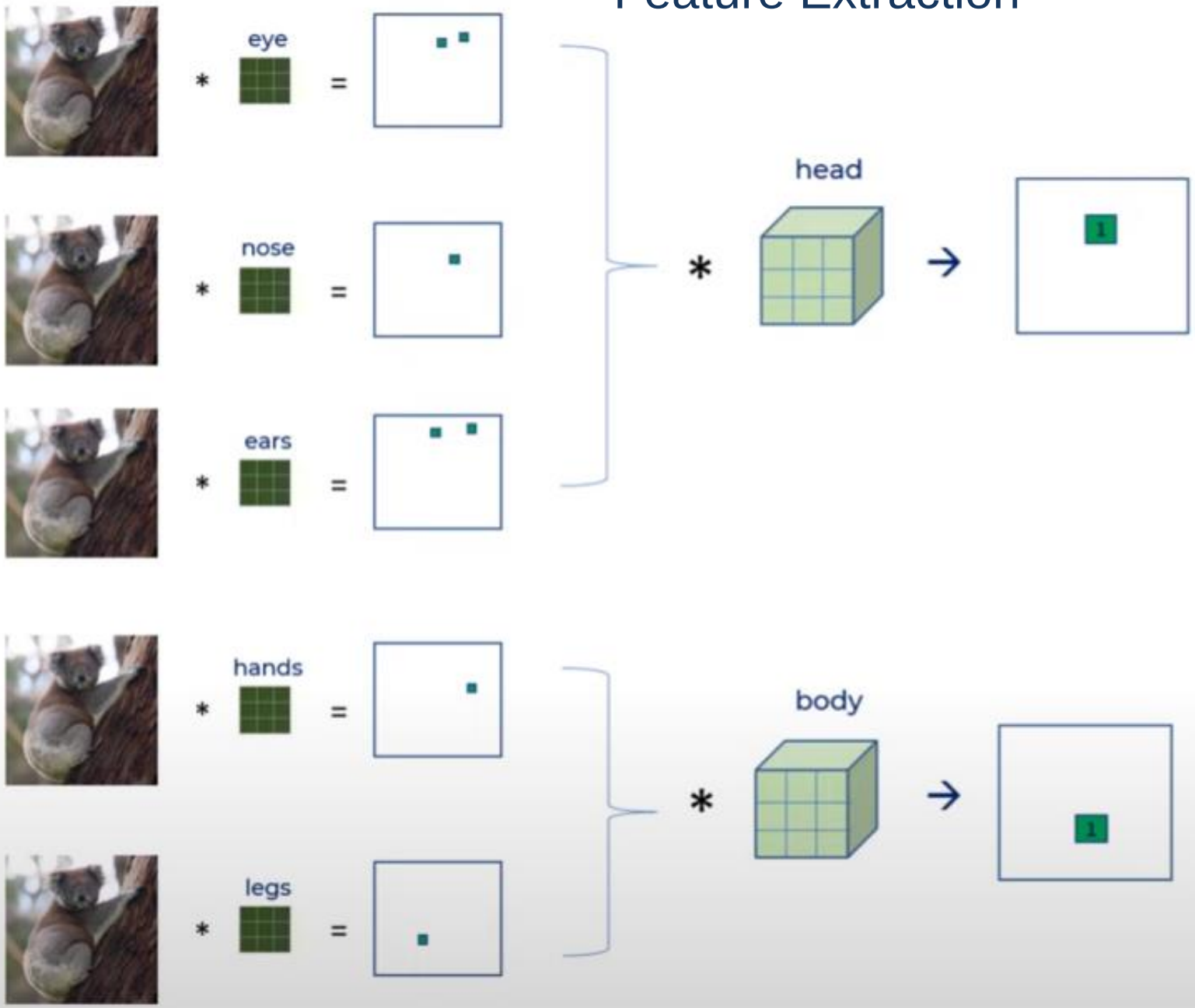
Convolutional Neural Networks (CNNs)

Aggrgation of the feature maps

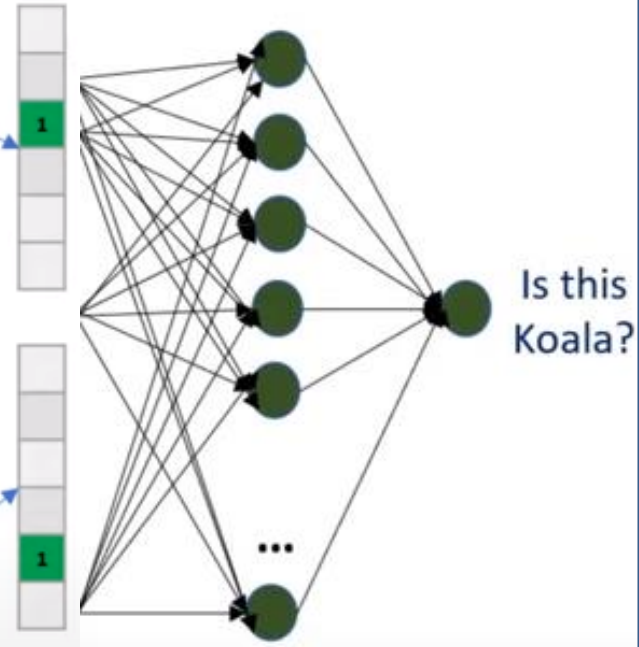



eye
nose
ears
Filter for head
1
1
1
1
1
1

Feature Extraction

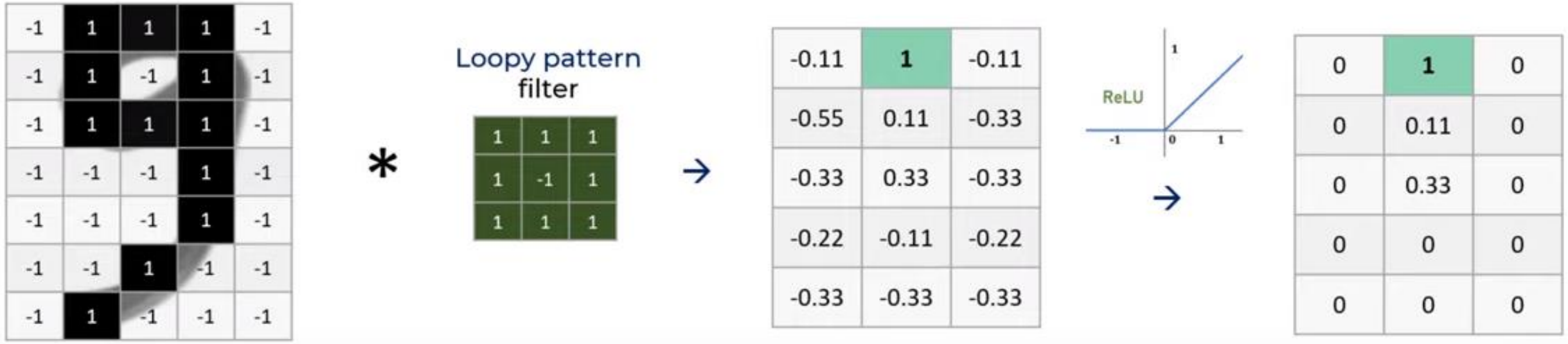


Classification



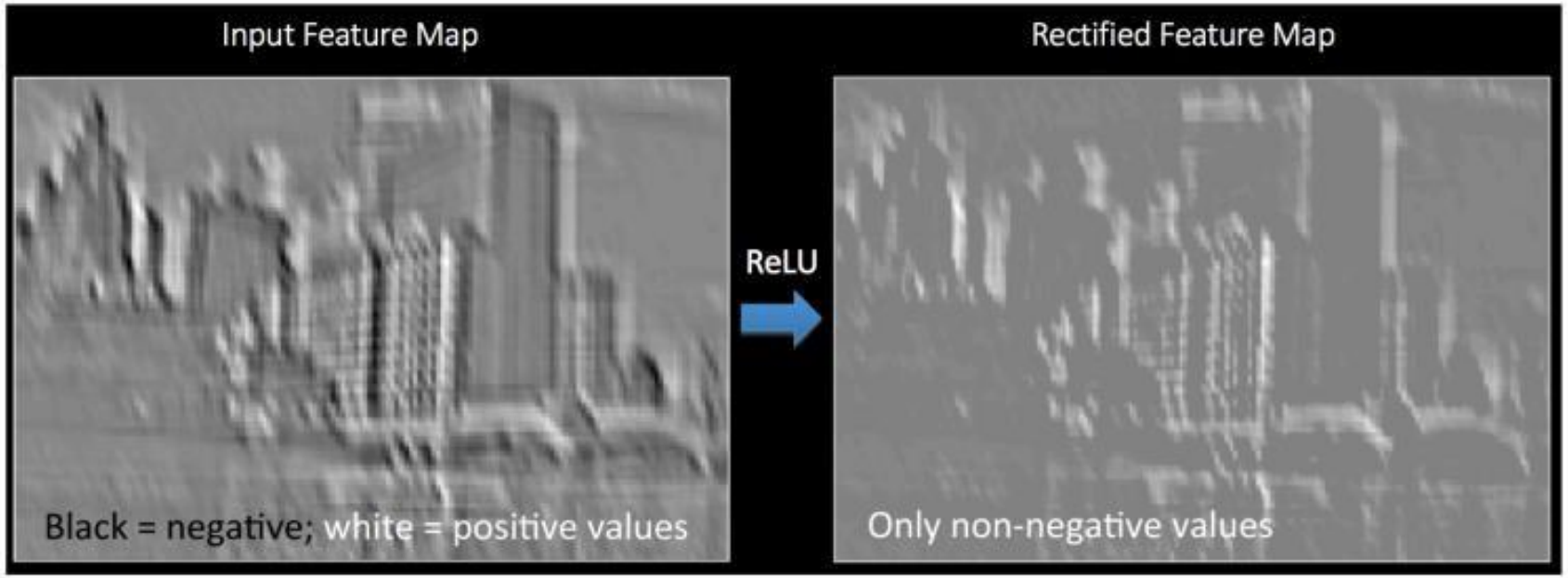
Convolutional Neural Networks (CNNs)

Activation Functions



Convolutional Neural Networks (CNNs)

Activation Layer (ReLU)



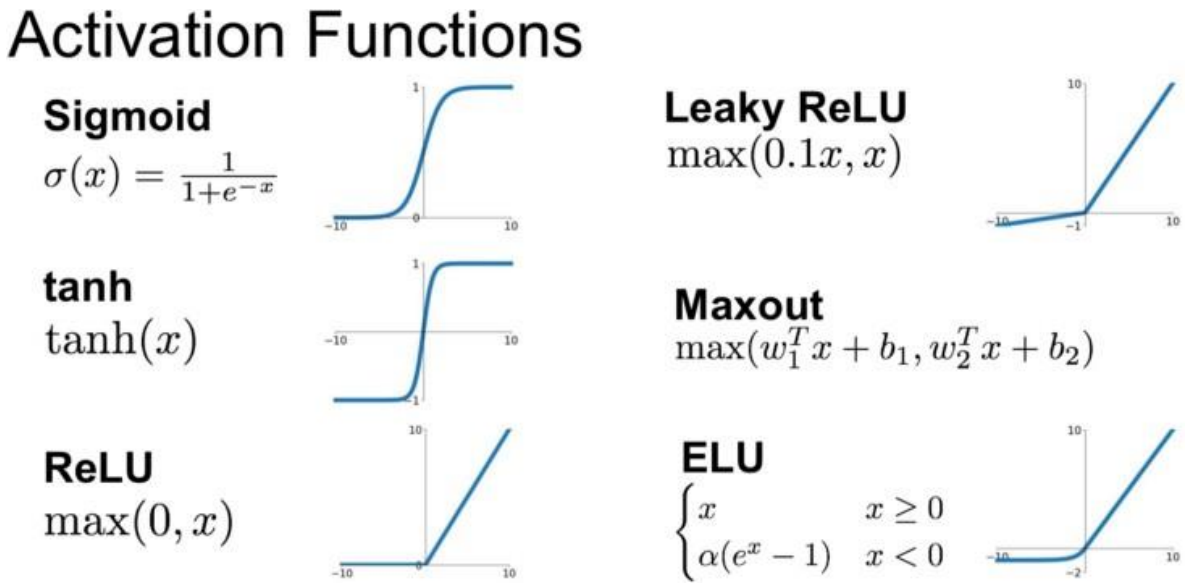
•Other non linear functions such as ***tanh*** or ***sigmoid*** can also be used instead of ReLU, but ReLU has been found to perform better in most situations.

Convolutional Neural Networks (CNNs)

Activation Function

The activation function is a node that is put at the end of or in between Neural Networks. They help to decide if the
neuron would fire or not.

"The activation function is the non linear transformation that we do over the input signal. This transformed output is then sent to the next layer of neurons as input." — Analytics Vidhya



Activation Layer (ReLU)

It is one of the most common activation functions.

- An additional operation called Rectified Linear Unit (ReLU) has been used after every Convolution operation
- Basically, ReLU is an element wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero
- The purpose of ReLU is to introduce non-linearity to the network





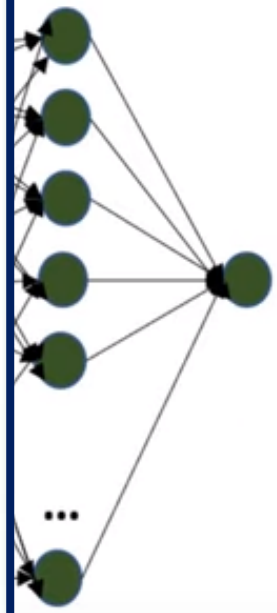
1920 x 1080

Image size = 1920 x 1080 X 3

First layer neurons = 1920 x 1080 X 3 ~ 6 million

Hidden layer neurons = Let's say you keep it ~ 4 million

Weights between input and hidden layer = 6 mil * 4 mil
= 24 million



Is this Koala?

5	1	3	4
8	2	9	2
1	3	0	1
2	2	2	0

8	**9**
3	2

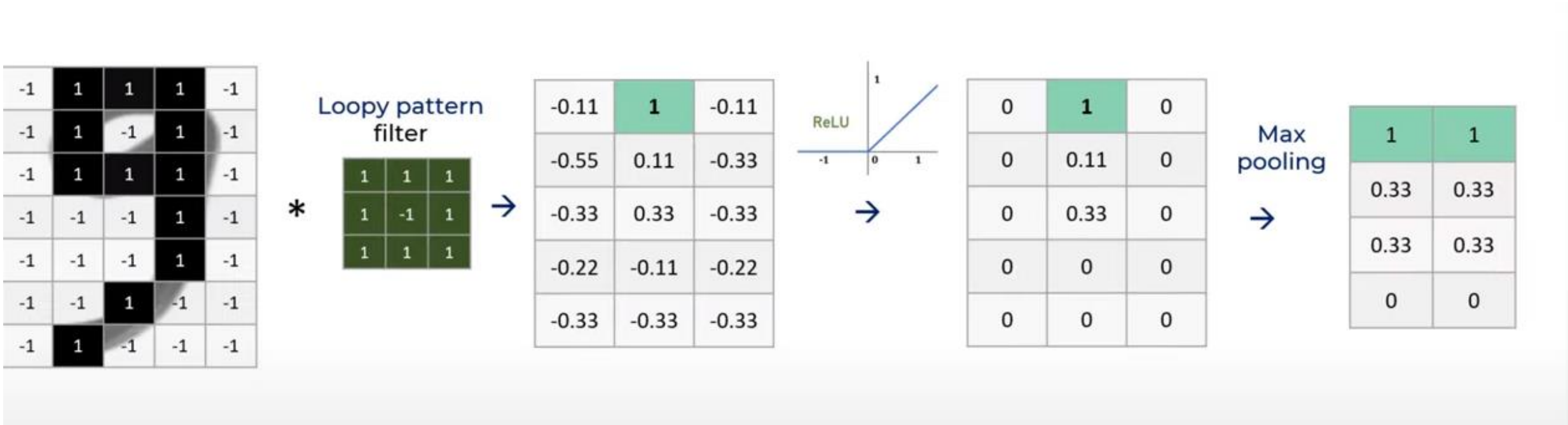
2 by 2 filter with stride = 2

Pooling on the digit 9

0	**1**	0
0	0.11	0
0	0.33	0
0	0	0
0	0	0

1	**1**
0.33	0.33
0.33	0.33
0	0

2 by 2 filter with stride = 1


-1 1 1 1 -1
-1 1 -1 1 -1
-1 1 1 1 -1
-1 -1 -1 1 -1
-1 -1 -1 1 -1
-1 -1 1 -1 -1
-1 1 -1 -1 -1
*
Loopy pattern filter
1 1 1
1 -1 1
1 1 1
-0.11 1 -0.11
-0.55 0.11 -0.33
-0.33 0.33 -0.33
-0.22 -0.11 -0.22
-0.33 -0.33 -0.33
ReLU
1
-1 0 1
0 1 0
0 0.11 0
0 0.33 0
0 0 0
0 0 0
Max pooling
1 1
0.33 0.33
0.33 0.33
0 0

Shifted 9 at different position

1	1	1	-1	-1
1	-1	1	-1	-1
1	1	1	-1	-1
-1	-1	1	-1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1
1	-1	-1	-1	-1

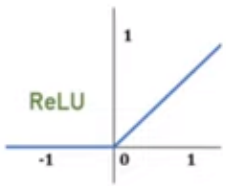
*

Loopy pattern filter

1	1	1
1	-1	1
1	1	1

→

1	-0.11	-0.11
0.11	-0.33	0.33
0.33	-0.33	-0.33
-0.11	-0.55	-0.33
-0.55	-0.33	-0.55



→

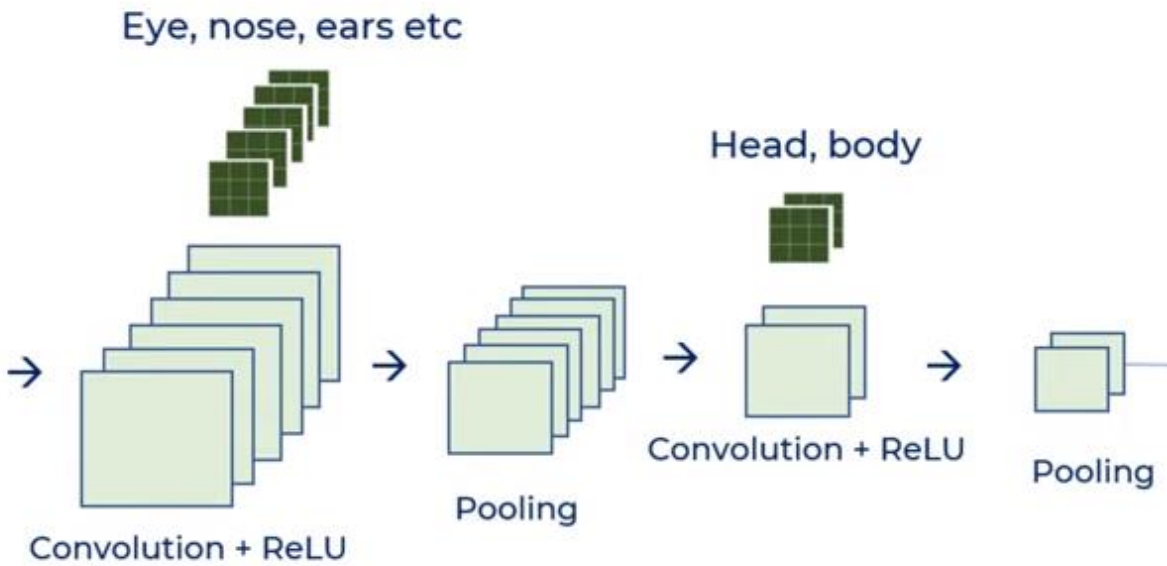
1	0	0
0.11	0	0.33
0.33	0	0
0	0	0
0	0	0

Max pooling

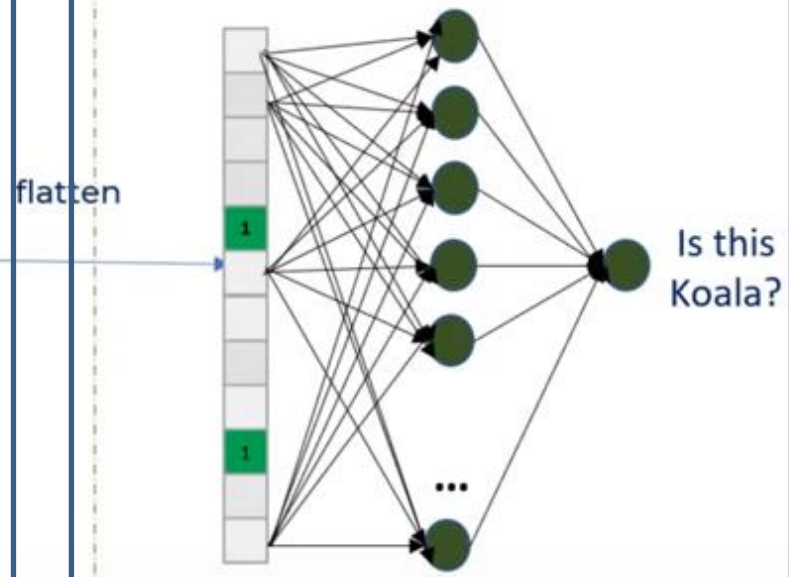
→

1	0.33
0.33	0.33
0.33	0
0	0

Feature Extraction

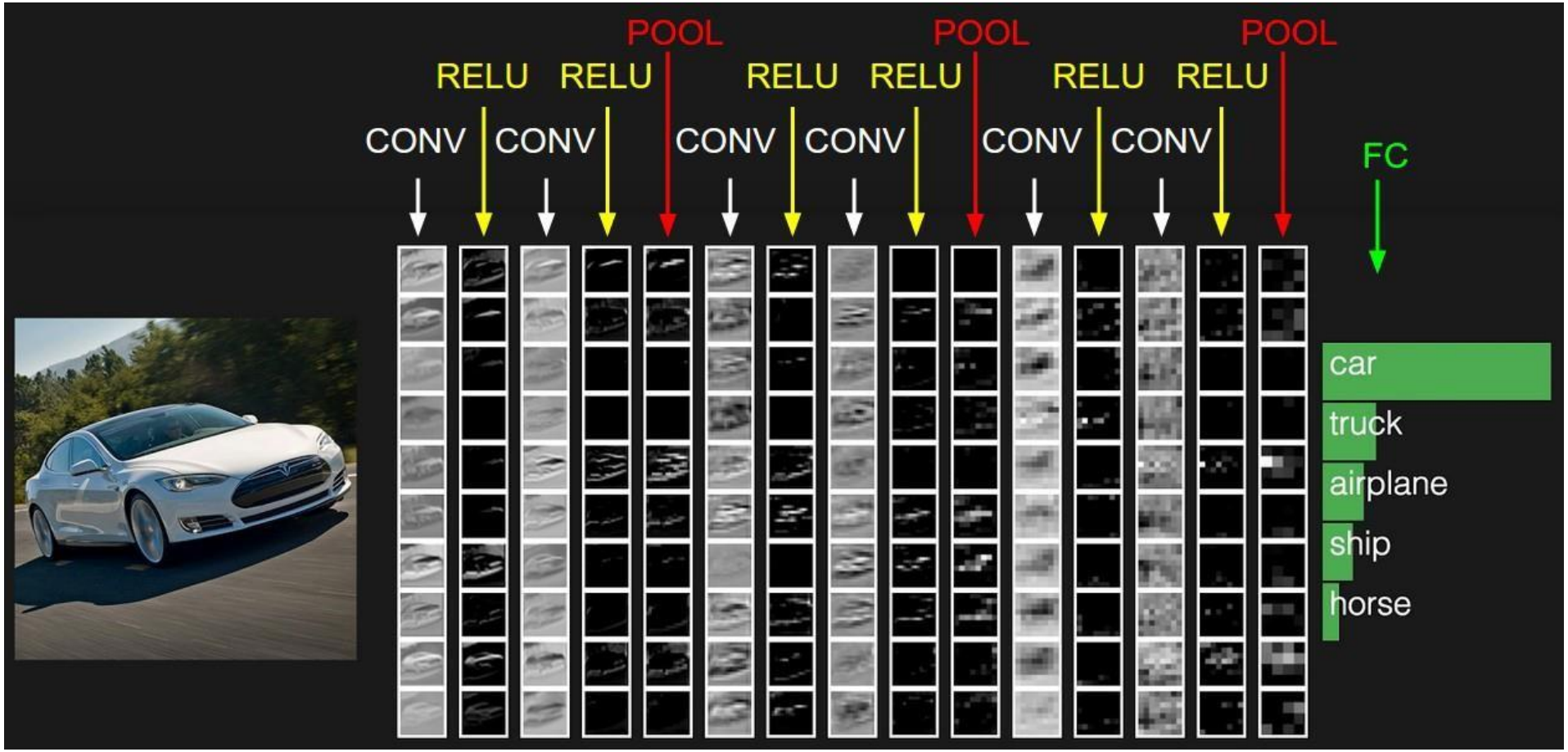


Classification



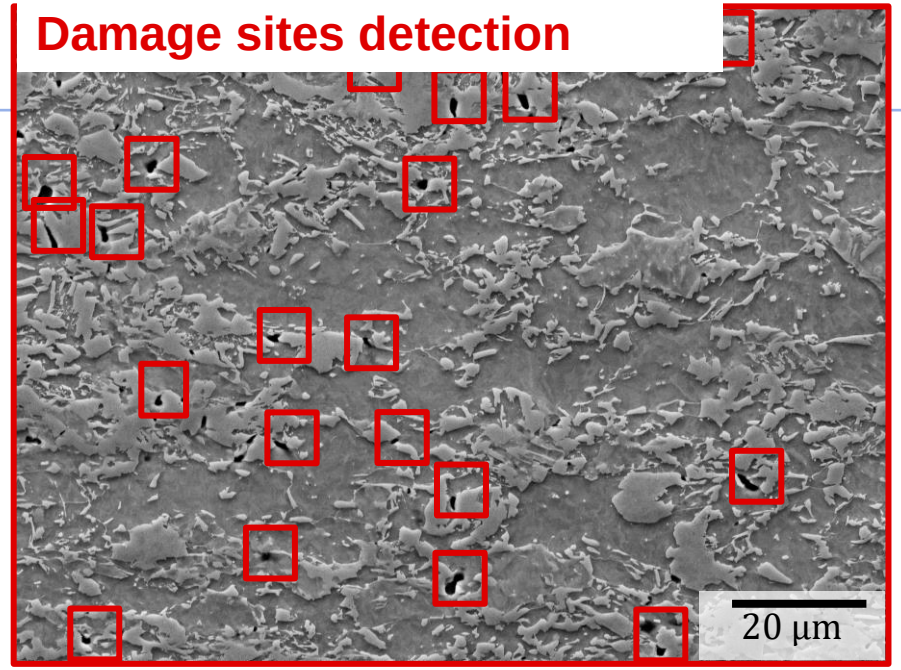
Convolutional Neural Networks (CNNs)

Example: Input >> [[Conv >> ReLU] * 2 >> Pool] * 3 >> FC



http://cs231n.stanford.edu/

Damage sites detection

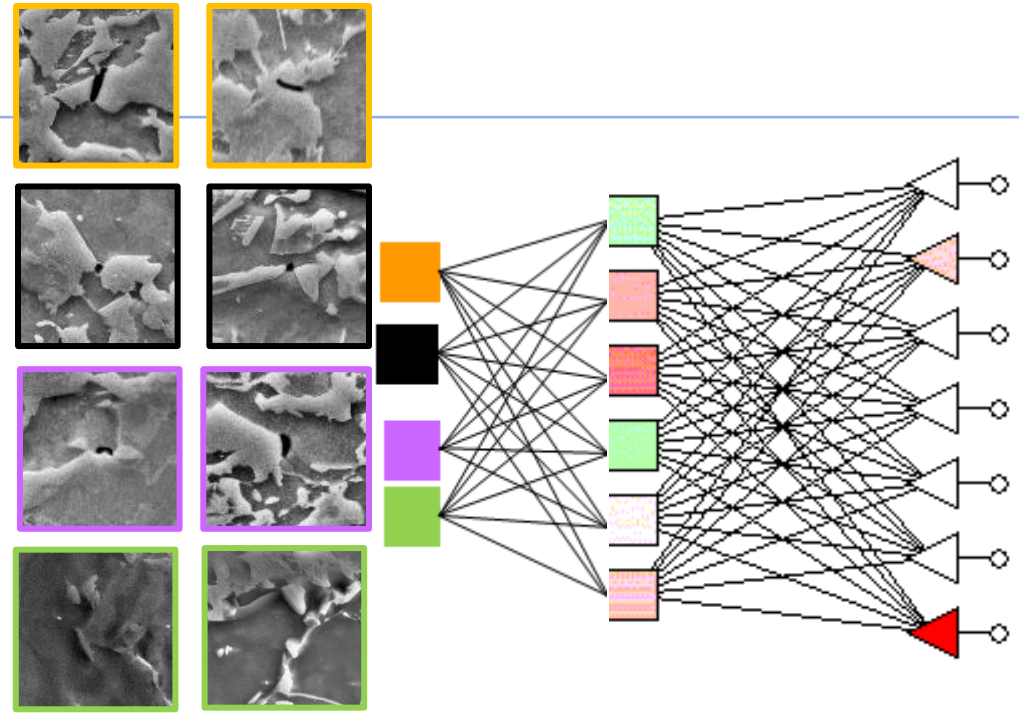


Martensite Crack

Notch

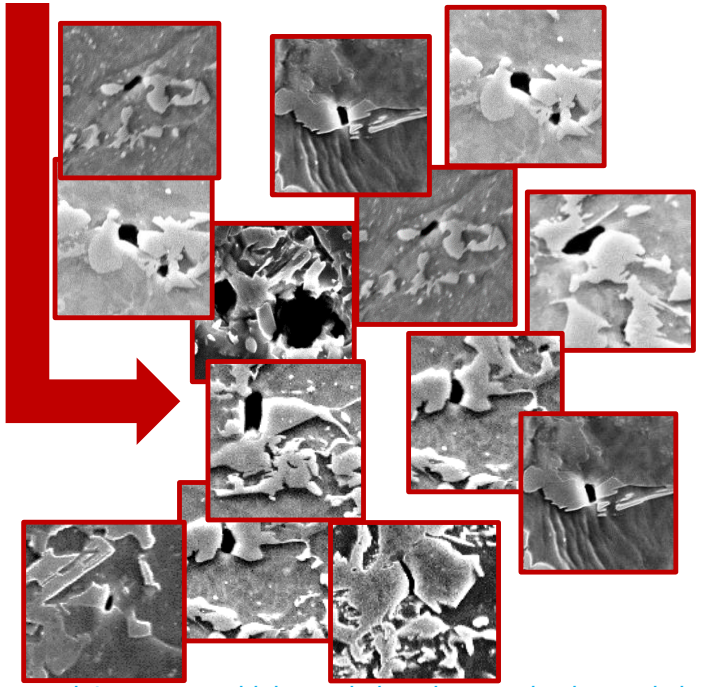
Interface Deco.

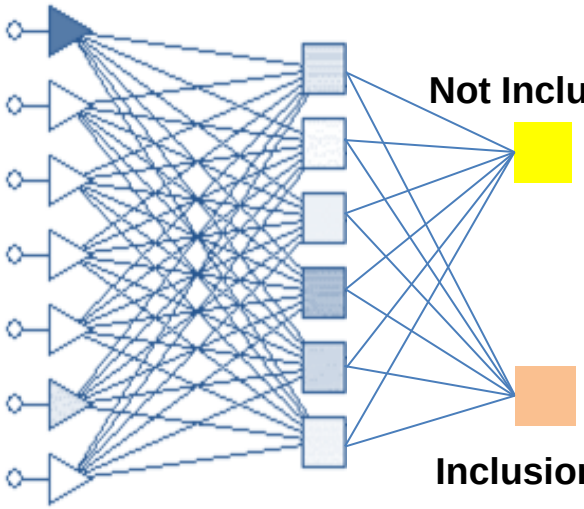
Shadowing



2nd CNN

DBSCAN Clustering algorithm

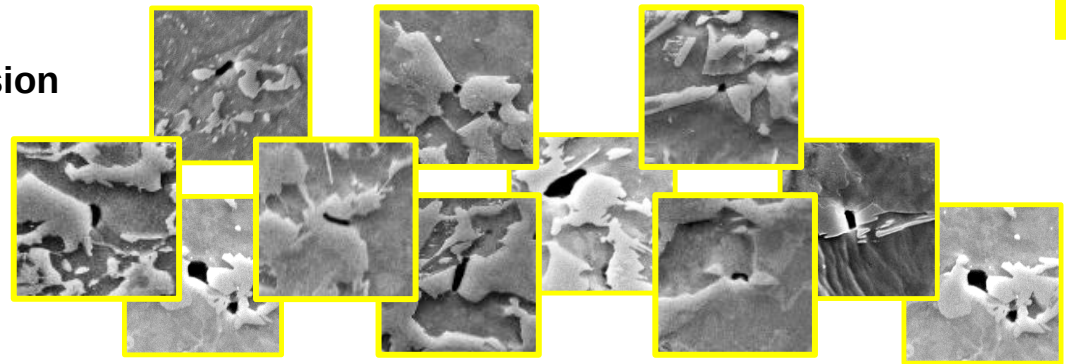




Not Inclusion

Inclusion

1st CNN



C. Kusche et al. Large-area, high-resolution characterisation and classification of damage mechanisms in dual-phase steel using deep learning. PloS one, 2019.

S. Medghalchi et al. Damage Analysis in Dual-Phase Steel Using Deep Learning: Transfer from Uniaxial to Biaxial Straining Conditions by Image Data Augmentation, JOM, 2020

iM | RWTH AACHEN UNIVERSITY

Convolutional Neural Networks (CNNs)

Architecture Examples

`INPUT -> [[CONV -> RELU]*N -> POOL?]*M -> [FC -> RELU]*K -> FC`

where the `*` indicates repetition, and the `POOL?` indicates an optional pooling layer. Moreover, `N >= 0` (and usually `N <= 3`), `M >= 0`, `K >= 0` (and usually `K < 3`). For example, here are some common ConvNet architectures you may see that follow this pattern:

- `INPUT -> FC`, implements a linear classifier. Here `N = M = K = 0`.
- `INPUT -> CONV -> RELU -> FC`
- `INPUT -> [CONV -> RELU -> POOL]*2 -> FC -> RELU -> FC`. Here we see that there is a single CONV layer between every POOL layer.
- `INPUT -> [CONV -> RELU -> CONV -> RELU -> POOL]*3 -> [FC -> RELU]*2 -> FC` Here we see two CONV layers stacked before every POOL layer. This is generally a good idea for larger and deeper networks, because multiple stacked CONV layers can develop more complex features of the input volume before the destructive pooling operation.

Successful CNN architectures

LeNet-5

• This architecture is an excellent “first architecture” for a CNN

AlexNet

- Famous for winning the **ImageNet** Large Scale Visual Recognition Challenge (**ILSVRC**) in 2012

- **VGGNet**

Experiments

Go to your notebooks...