



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

Outlook

Introduction to some image analysis tools – Notebooks

Introduction to the application of machine learning

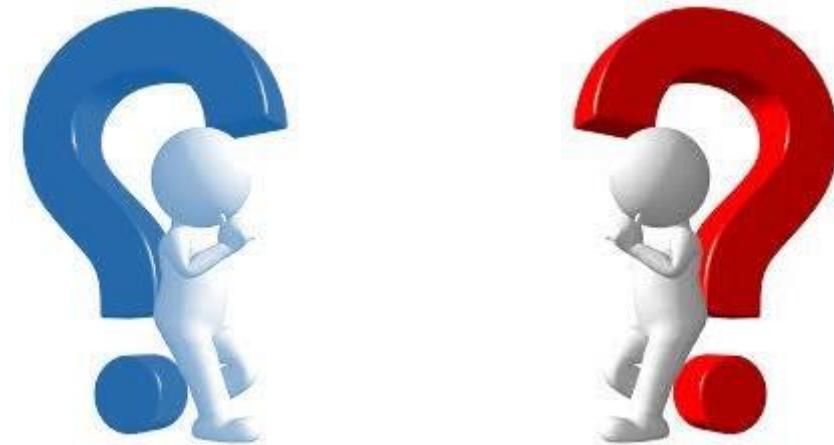
Application of machine learning techniques on data

- Supervised learning methods
- ~~Unsupervised learning methods~~

Introduction to some image analysis tools	Notebooks	45 min
Introduction to the application of machine learning	Slides	25 min
Application of machine learning techniques on data	Notebooks	

[GitHub - SetarehMed/Micrograph_Process_AI_ECI_2022_Binder](#)

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

The screenshot shows a Google search results page. The search bar at the top contains the query "Machine learning papers". Below the search bar, there are navigation links for "All", "News", "Images", "Videos", "Shopping", "More", "Settings", and "Tools". A message indicates "About 511.000.000 results (0,63 seconds)".

The first result is a snippet titled "Scholarly articles for Machine learning papers" with several links to academic papers:

- Crafting papers on machine learning - Langley - Cited by 150
- ... extraction of results from machine learning papers - Kardas - Cited by 3
- Genetic algorithms and machine learning - Goldberg - Cited by 3117

The second result is a link to "paperswithcode.com" with the title "Papers With Code: The latest in Machine Learning". It includes a brief description: "Papers With Code highlights trending Machine Learning research and the code to implement it. Browse State-of-the-Art · Papers With Code · Transfer Learning · Few-Shot Learning".

The third result is a link to "www.kdnuggets.com" with the title "Top 20 Recent Research Papers on Machine Learning and ...". It includes a brief description: "Machine learning and Deep Learning research advances are transforming our technology. Here are the 20 most important (most-cited) scientific papers that ...".

The fourth result is a link to "arxiv.org" with the title "Machine Learning authors/titles recent submissions - arXiv.org". It includes a brief description: "Comments: Accepted to NAACL 2021 (Long Paper). Subjects: Computation and Language (cs.CL); Machine Learning (stat.ML). [10] arXiv:2104.05508 ...".

Below the search results, there is a "People also ask" section with the following questions:

- Where can I find Machine Learning papers?
- Can I teach myself machine learning?
- How do you write a paper in machine learning?
- How do I get into machine learning research?

Applications in our daily life

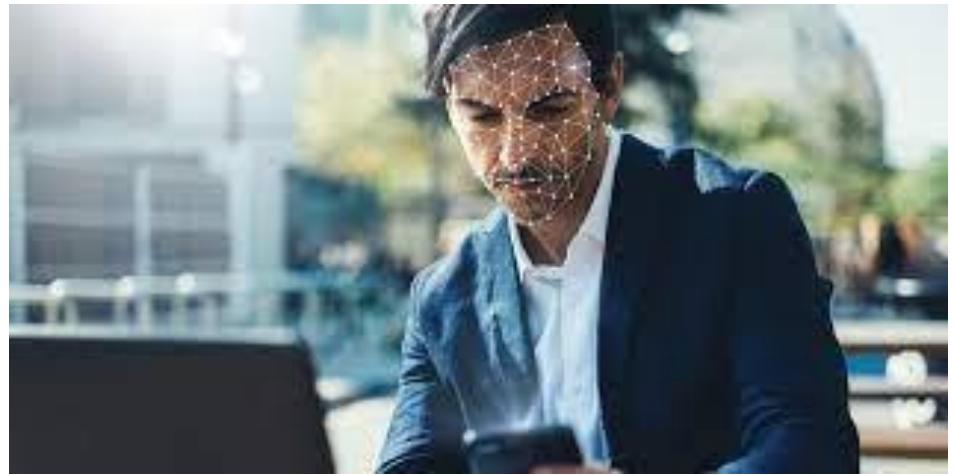
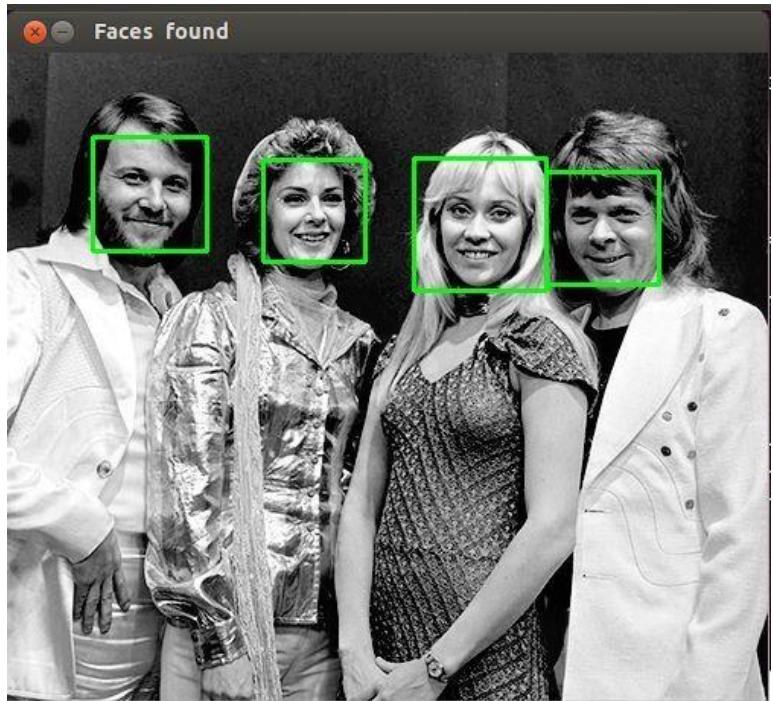
- Machine translation
- Language Translation



A screenshot of the Google Translate interface. At the top, there are search filters: '386.000.000 Results', 'Date', 'Language', 'Region', and a toggle for 'Open links in new tab'. Below these, two language dropdown menus show 'English (detected)' on the left and 'German' on the right. A central area contains a bidirectional arrow icon. On the left, the English input reads: 'Smile , Today is a good day . Welcome to the Materials Physics and Design Course.' On the right, the German output reads: 'Lächeln , Heute ist ein guter Tag . Willkommen zum Materialphysik- und Designkurs.' Below the main input/output area, there are three small icons: a speaker icon, a download icon, and a settings/cog icon. At the bottom, a section titled 'Widely used phrases' is partially visible.

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)

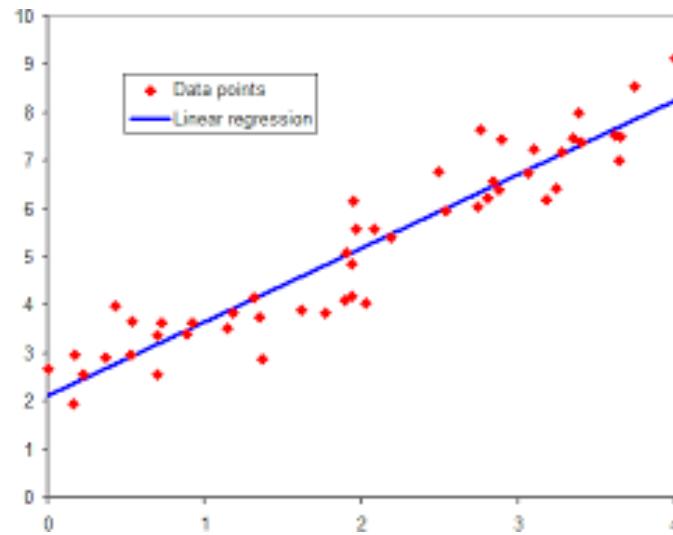
$$y = f(x, w)$$

Tools

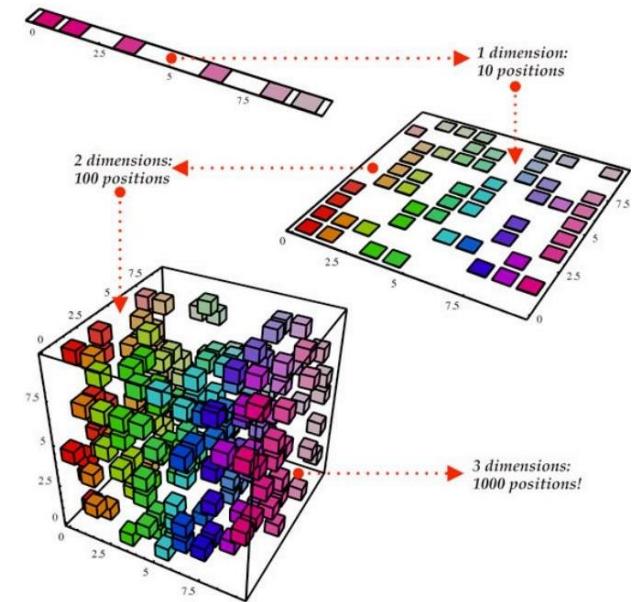
- Statistics
- Probability theory
- Decision theory
- Information theory
- Optimization theory

Approaches

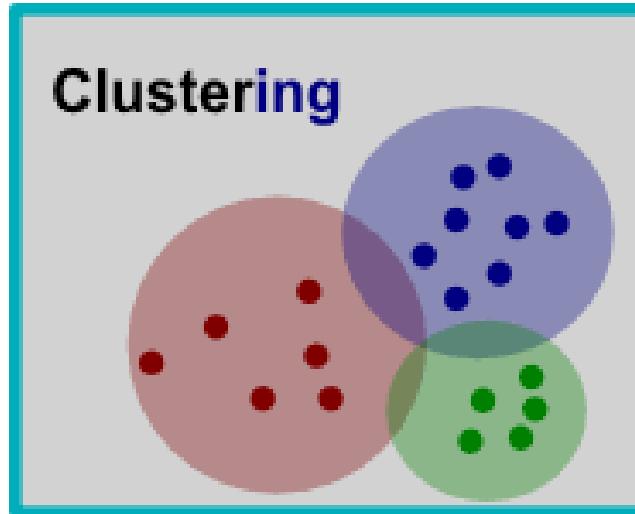
Regression



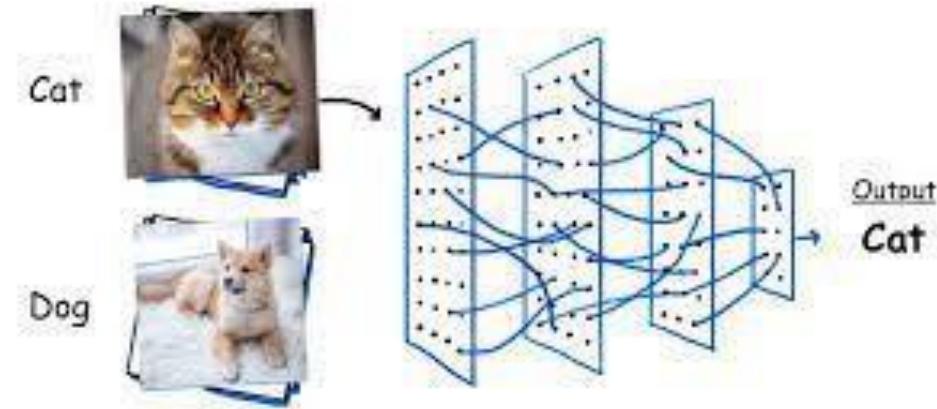
Dimensionality reduction



Clustering



Classification



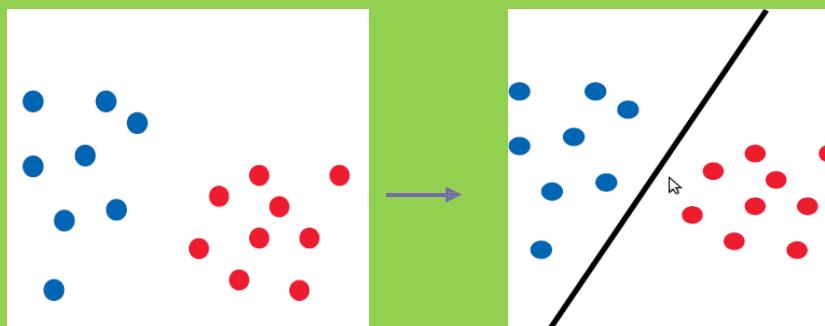
General Classification of Machine Learning Approaches

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 between the input and output.
$$y = f(x, w)$$
- Goal: To generalize 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 , . . .

Unsupervised Learning: Data without labels

- Some input data without labels are given
- Tasks : Determine the data distribution $p(x) \rightarrow$ density estimation
- Visualize the data by projections \rightarrow dimensionality reduction
- Goal: Find groupings of the data \rightarrow clustering

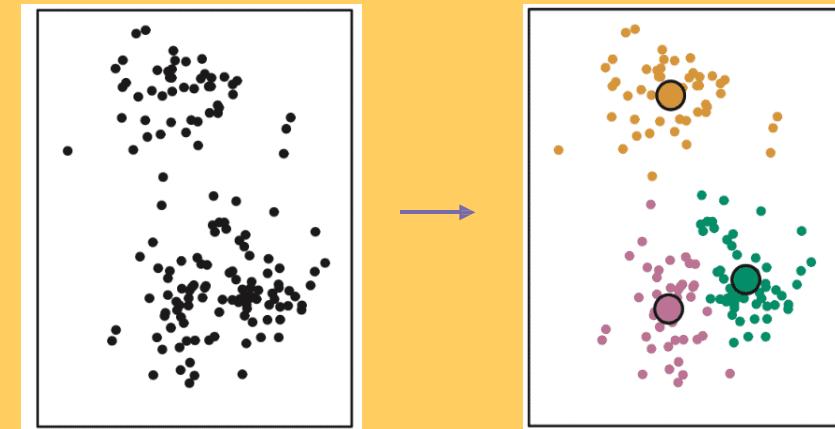
$$D = \{x_1, x_2, \dots, x_n\}$$

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

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

- additionally a set of unlabeled examples as in unsupervised learning

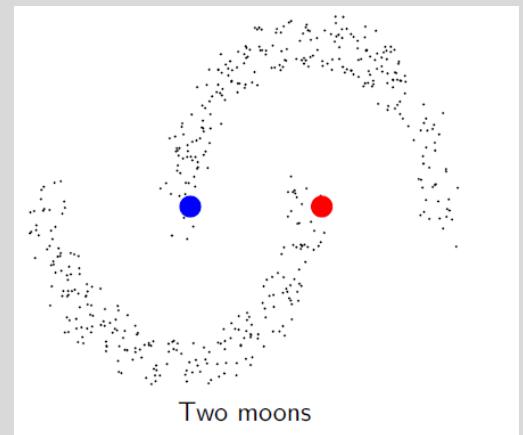
$$D_u = \{x_{l+1}, x_2, \dots, x_{l+u}\}$$

- Goal :

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

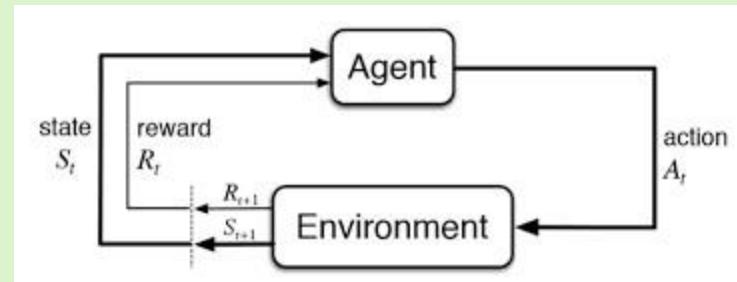
Example :

Red and blue points are labeled and the rest of black Points are unlabeled.



Reinforcement Learning: Feedback/rewards

- Problems involving an agent interacting with an environment, 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

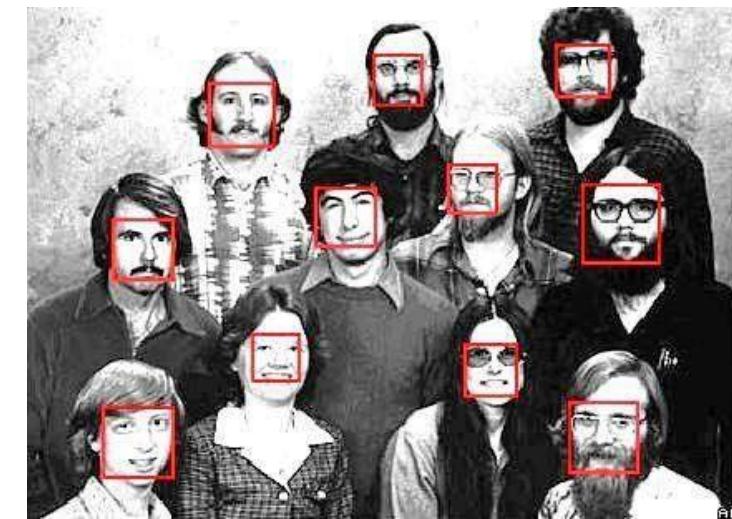
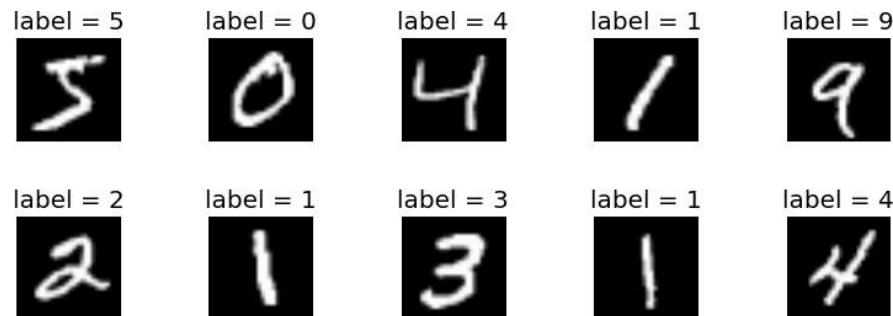
Exploitation: use known actions that yield high rewards

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 / classification	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	

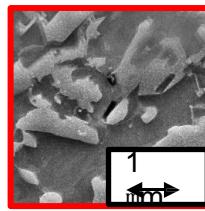


Dig into some examples

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

Person – Object – Speech recognition / identification / classification	
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:



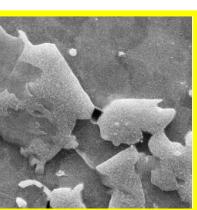
Interface
Decohesion



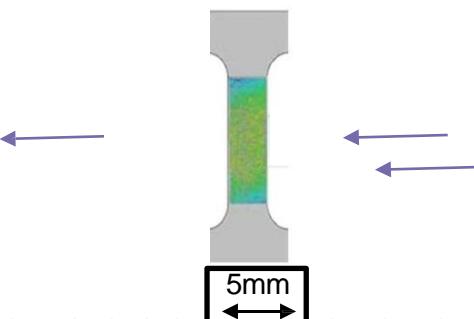
Inclusion



Martensite
Crack



Notch



Dig into some examples

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

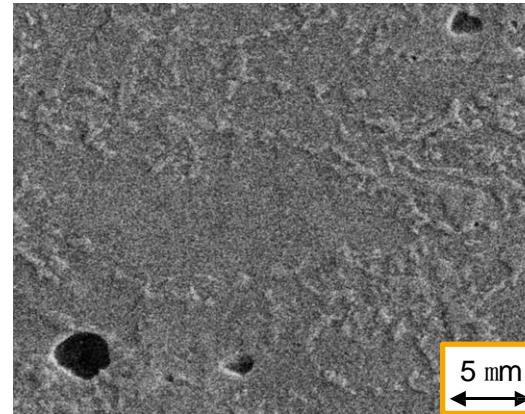
Person – Object – Speech recognition / identification / classification	
Image denoising	
Image inpainting	
Credit card fraud detection	
Prediction survival rate of a patient	
Clustering scientific publications according to topics	
Semantic Image Segmentation	



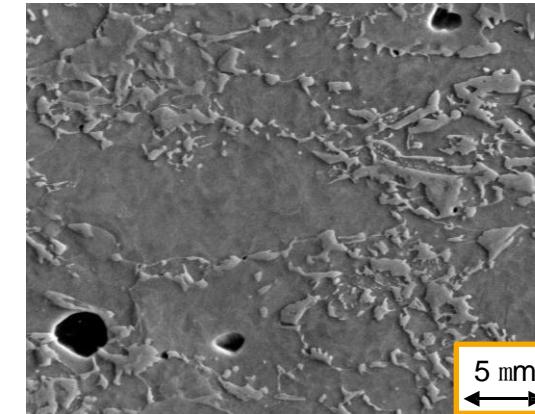
Dig into some examples

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

Person – Object – Speech recognition / identification / classification	
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

Setareh Medghalchi ,
Philipp Schumacher

Dig into some examples

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

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

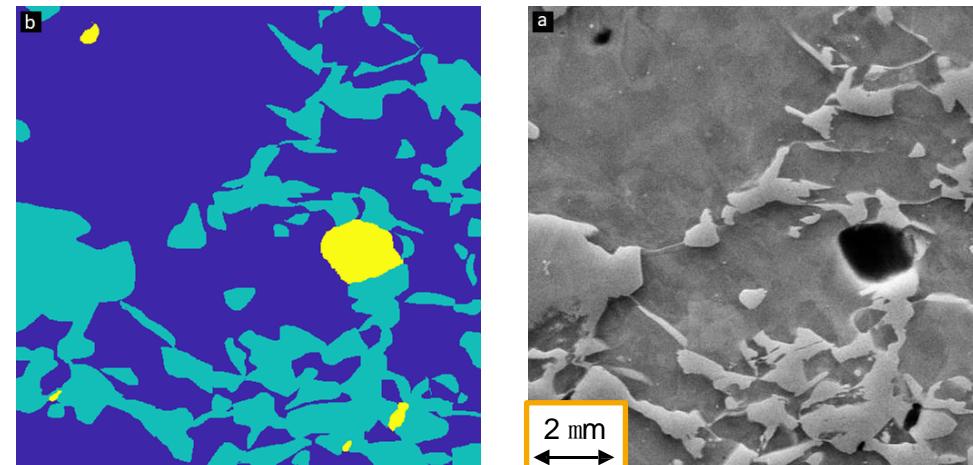


Dig into some examples

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

Person – Object – Speech recognition / identification / classification	
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 Plüscher,
Ehsan Karimi

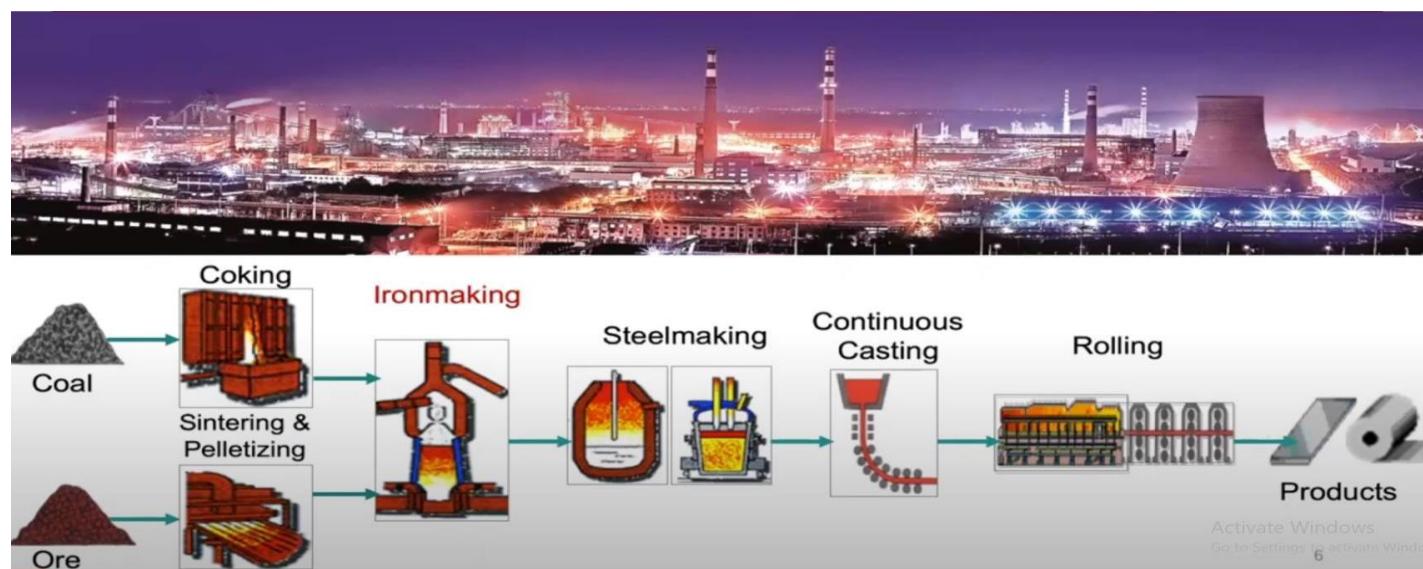
Dig into some examples

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

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 previously 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 efficiency.
-

Prof. Paul Monks
University of Leicester



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

Empirical learning in materials science.

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

Prof. Harry Bhadeshia
University of Cambridge

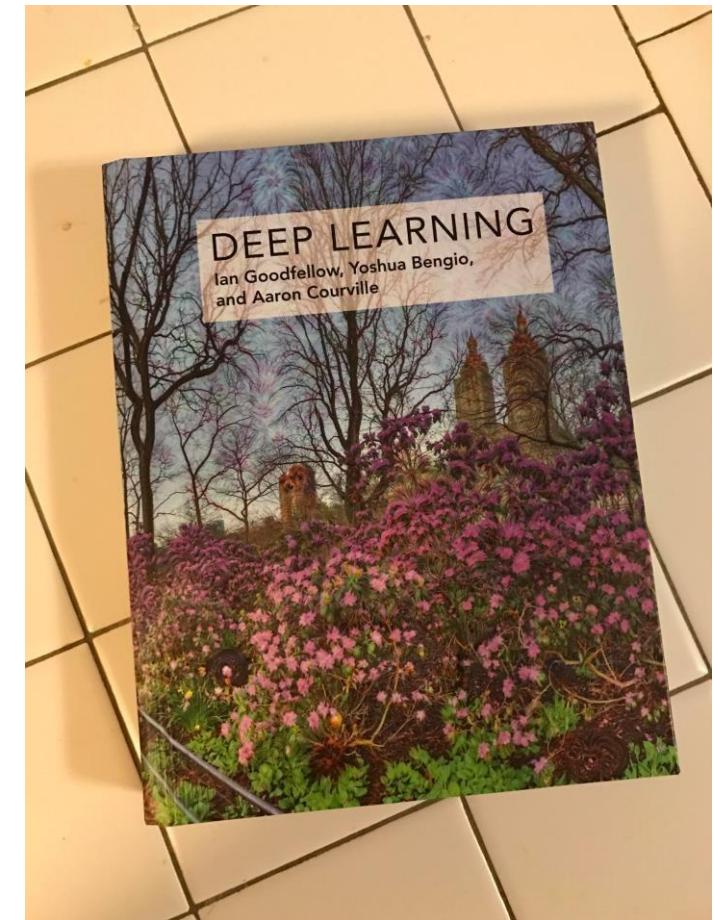




Image Classification using Convolutional Neural Networks (CNNs)

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

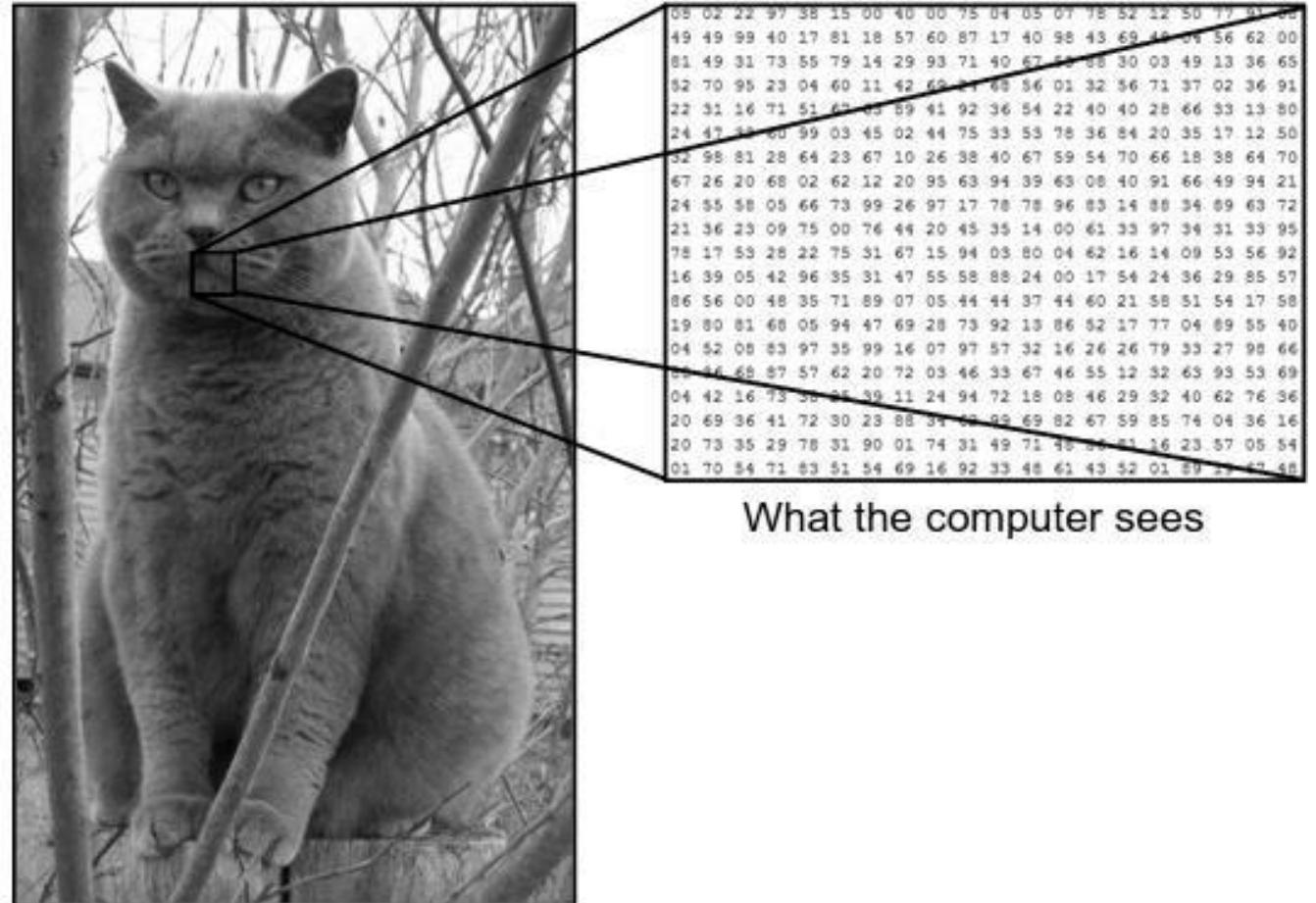
Training

- Backpropagation
- Overfitting, Regularization and Dropout

Experiments

Introduction

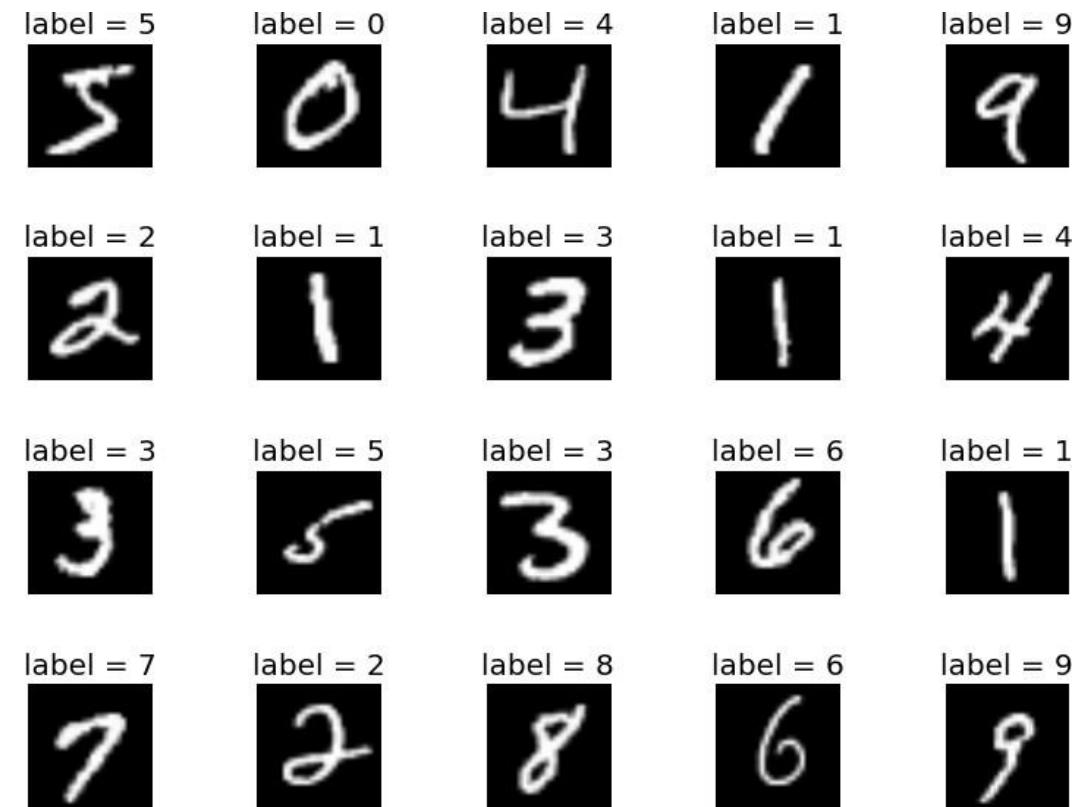
What we see vs. What computers see



Introduction

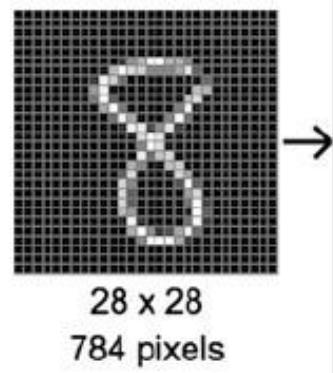
MNIST Dataset (<http://yann.lecun.com/exdb/mnist/>)

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



Introduction

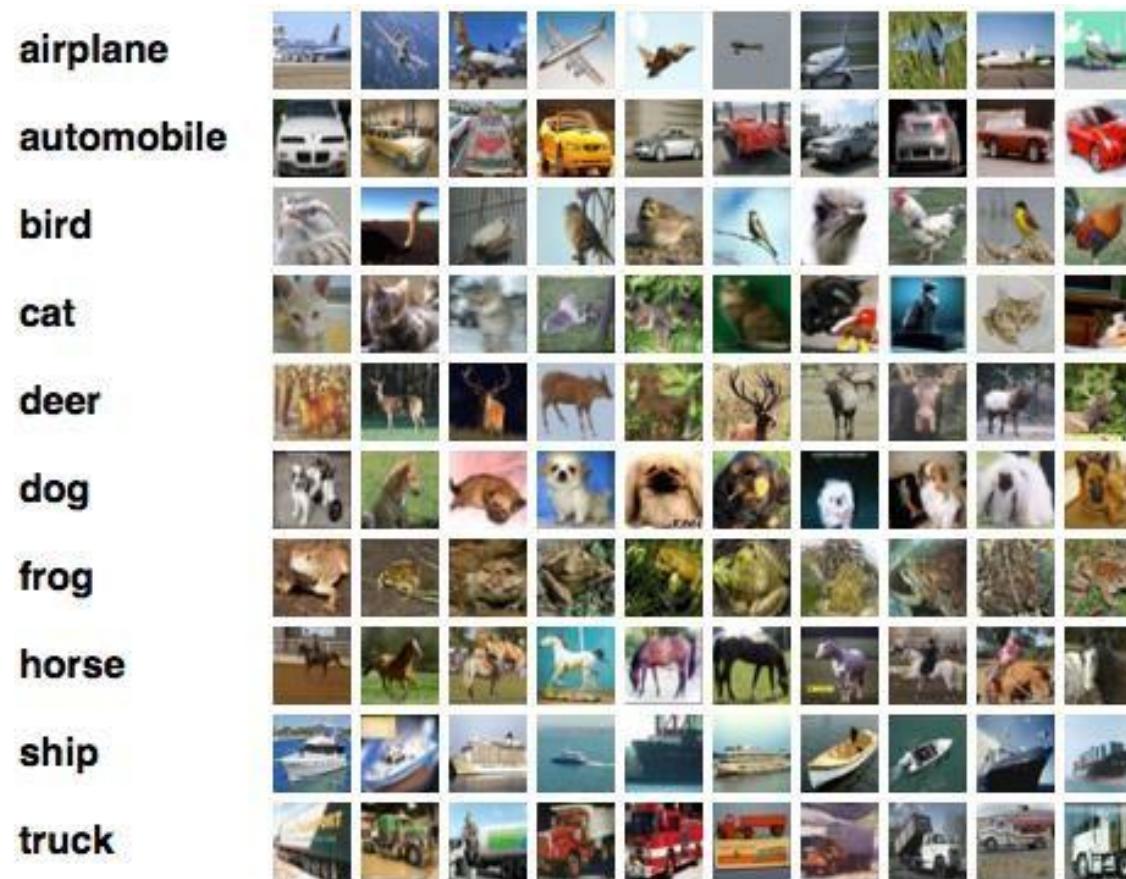
MNIST datasete



Introduction

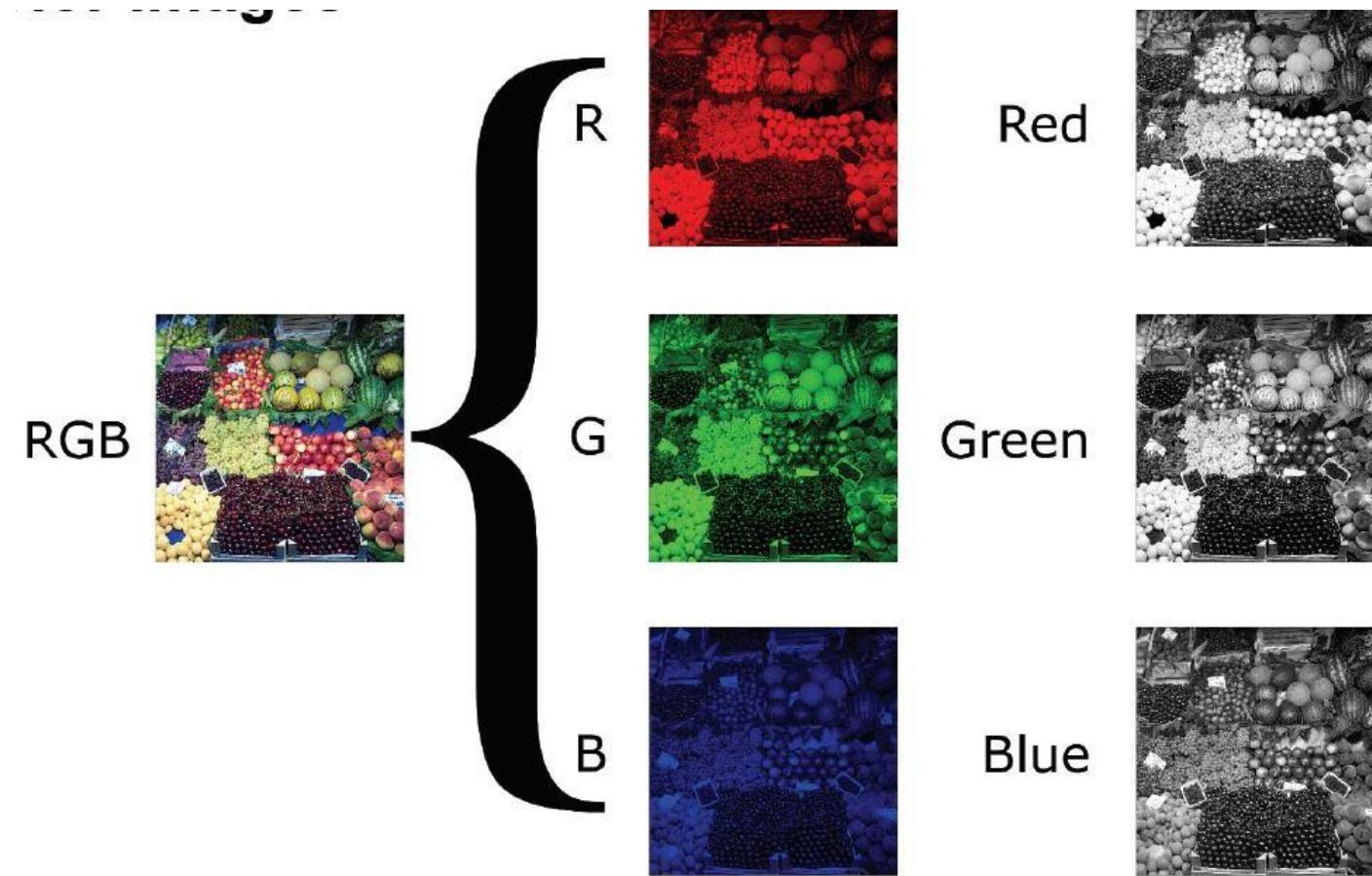
CIFAR datasete : (<https://www.cs.toronto.edu/~kriz/cifar.html>)

- 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.



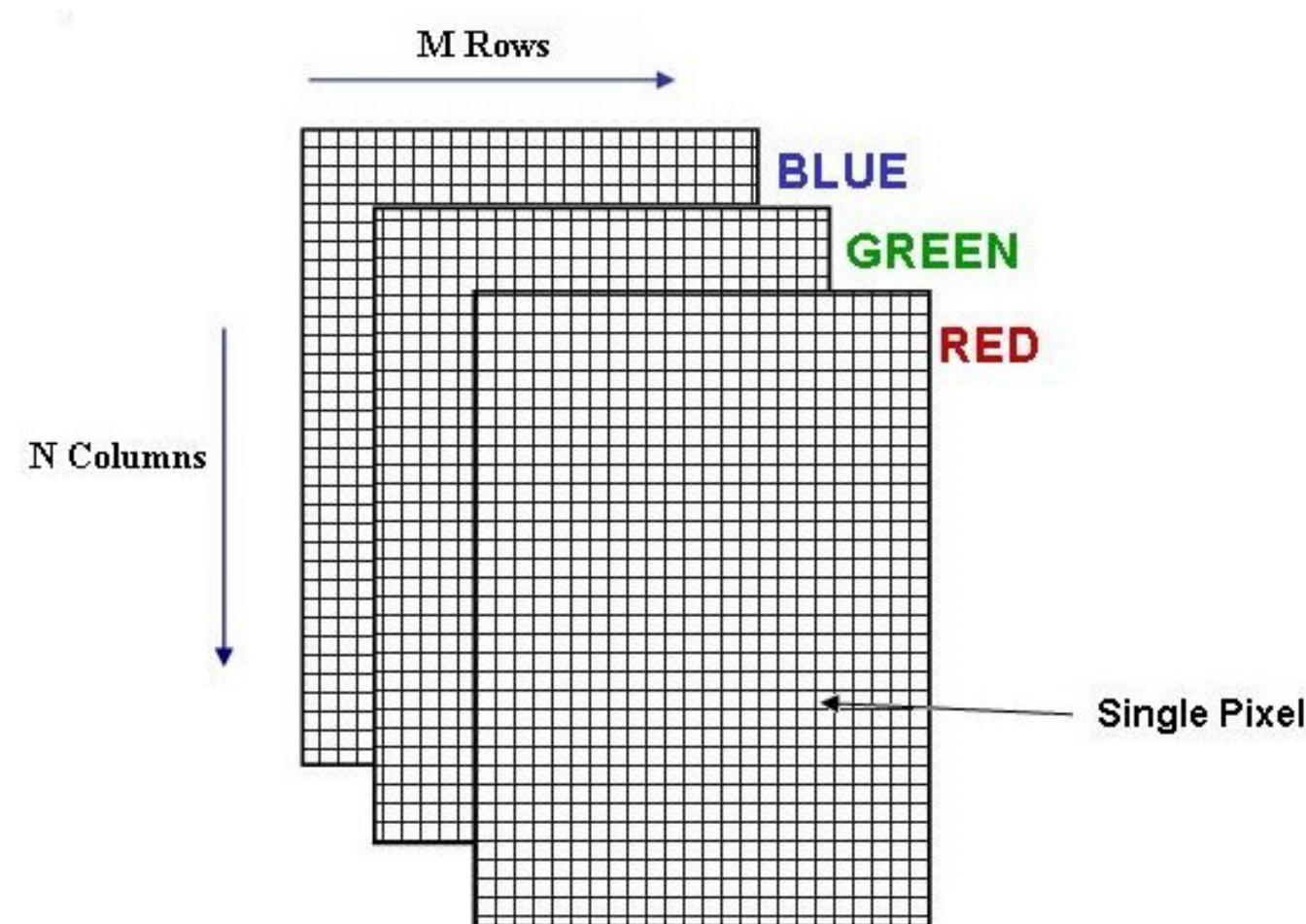
Introduction

Color images



Introduction

Color images

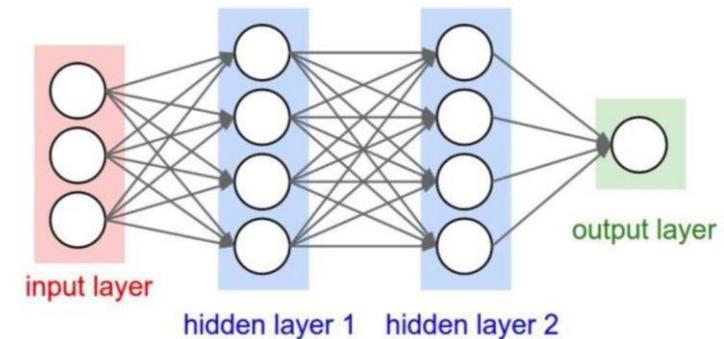


Deep Learning

"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

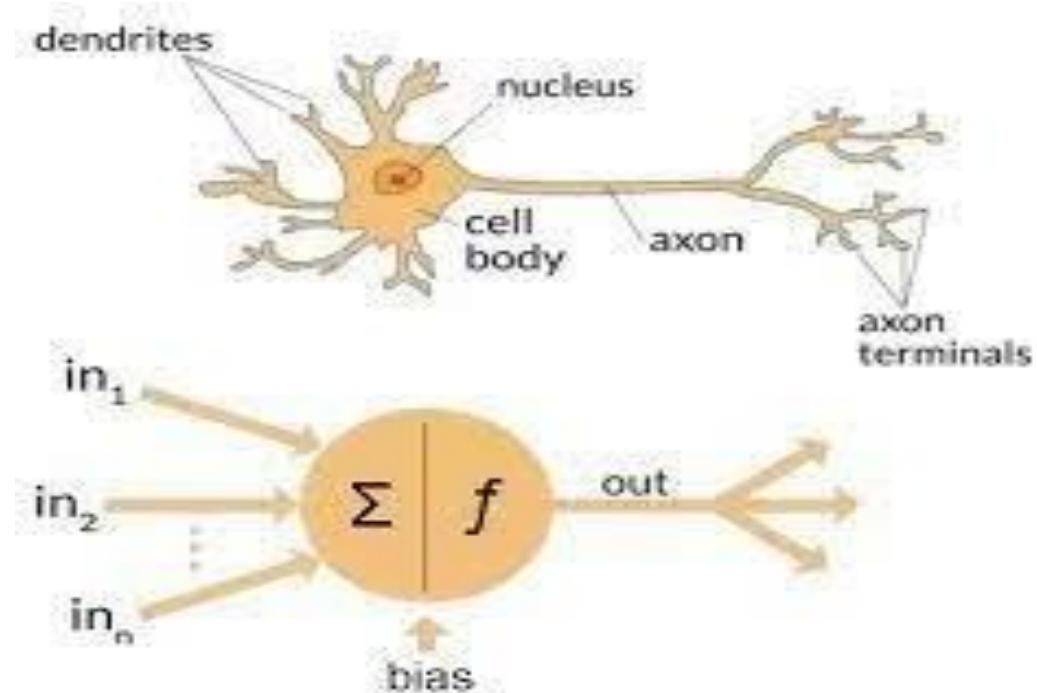
- Deep-learning networks are distinguished from the more common single-hiddenlayer-neural networks by their **depth**
- More than **three** layers (including input and output) qualifies as “deep” learning



- 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)

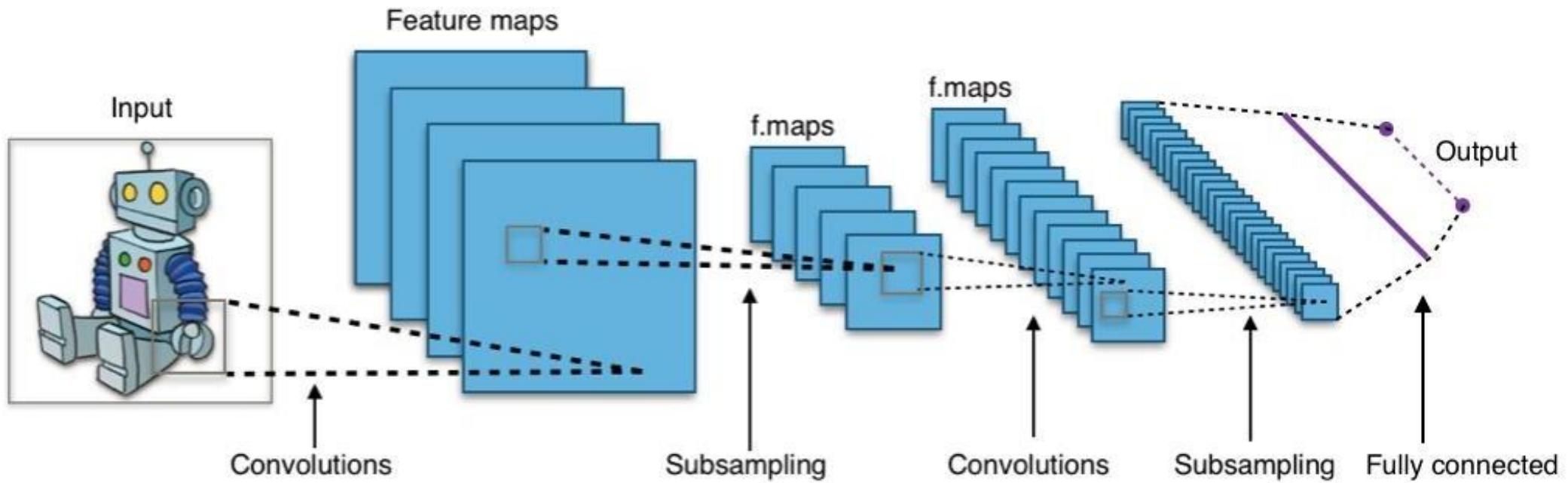
- CNNs take a **biological** inspiration from the visual cortex
- The visual cortex has small regions of cells that are sensitive to **specific regions** of the visual field
- For example, some neurons fired when exposed to **vertical** edges and some when shown **horizontal** or **diagonal** edges
- 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

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

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

X

1	0	1
0	1	0
1	0	1

=

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

4		

Input image (I)

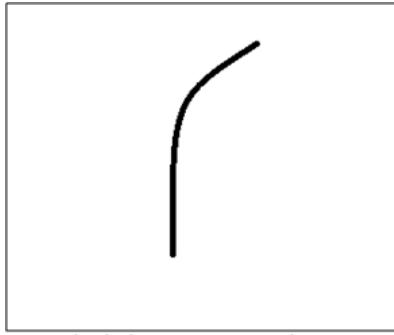
Filter (K)

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)



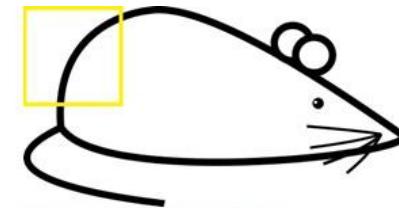
Visualization of a curve detector filter

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

Pixel representation of the receptive field

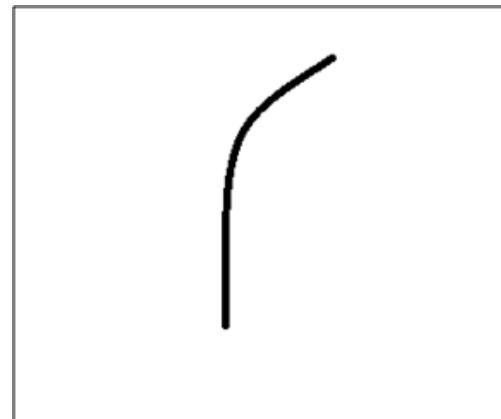
*

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

Pixel representation of filter

$$\text{Multiplication and Summation} = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 \text{ (A large number!)}$$

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

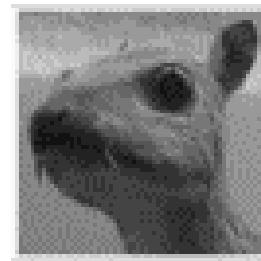
Pixel representation of filter

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	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
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:

0	0	0	0	0	0	0	...
0	156	155	156	158	158	158	...
0	153	154	157	159	159	159	...
0	149	151	155	158	159	159	...
0	146	146	149	153	158	158	...
0	145	143	143	148	158	158	...
...

Input Channel #1 (Red)

0	0	0	0	0	0	0	...
0	167	166	167	169	169	169	...
0	164	165	168	170	170	170	...
0	160	162	166	169	170	170	...
0	156	156	159	163	168	168	...
0	155	153	153	158	168	168	...
...

Input Channel #2 (Green)

0	0	0	0	0	0	0	...
0	163	162	163	165	165	165	...
0	160	161	164	166	166	166	...
0	156	158	162	165	166	166	...
0	155	155	158	162	167	167	...
0	154	152	152	157	167	167	...
...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1



308

+

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2



-498

0	1	1
0	1	0
1	-1	1

Kernel Channel #3



+ 164 + 1 = -25

Bias = 1

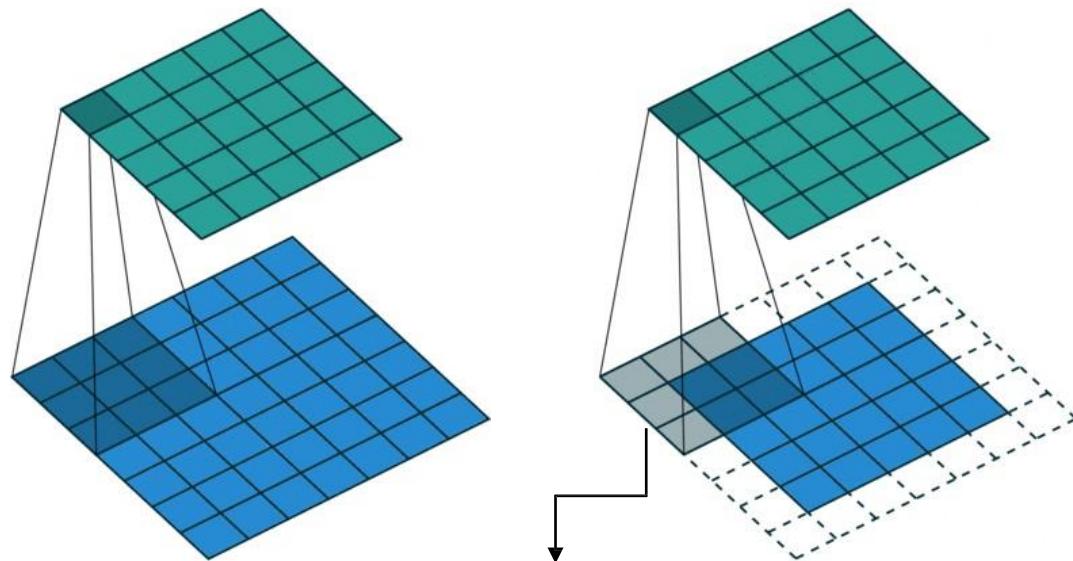
-25			
			...
			...
			...
...

Output

Convolutional Neural Networks (CNNs)

Convolutional Layer

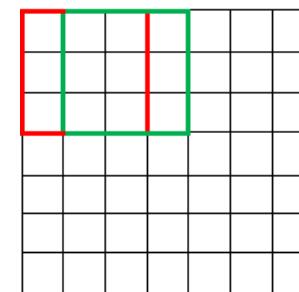
- In practice, a CNN learns the values of these **filters** on its own during the **training process**
- Although we still need to specify parameters such as **number of filters**, **filter size**, **padding**, and **stride** before the training process
- **Padding :**



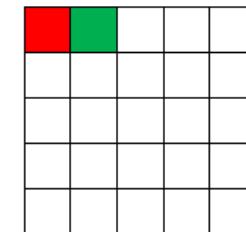
- Padding is the method for preserving image data by avoiding shrinking the image and loosing the data on the edges of the image. Here , we have padding = 1

- **Stride :**

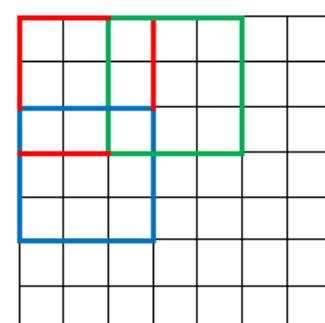
7 x 7 Input Volume



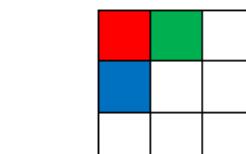
5 x 5 Output Volume



7 x 7 Input Volume



3 x 3 Output Volume

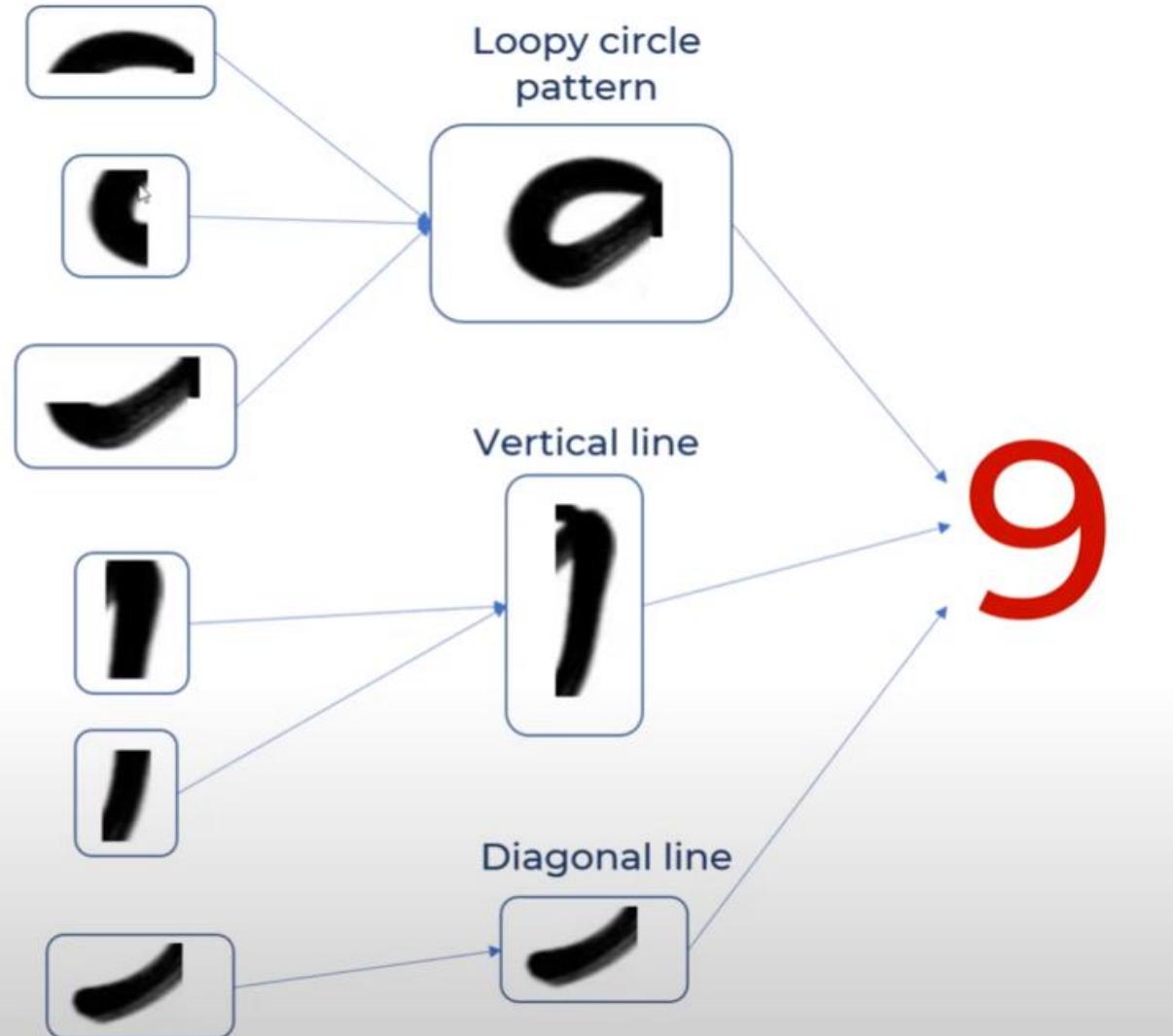


- Stride is the size of filter window jumping over the imput image.

Convolutional Neural Networks (CNNs)

Example Case

g



Convolutional Neural Networks (CNNs)

Different Feature Maps

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

Vertical line
filter

Diagonal line
filter

Convolutional Neural Networks (CNNs)

$$-1+1+1-1-1-1-1+1+1 = -1 \rightarrow -1/9 = -0.11$$

-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)

-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	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

Feature Map



Convolutional Neural Networks (CNNs)

Loopy pattern detector

$$g * \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & \textbf{1} & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array}$$

6 * $\begin{matrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{matrix}$ =

			1

8 * $\begin{matrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{matrix}$ =

Convolutional Neural Networks (CNNs)

Different Features Detector



Loopy pattern
detector

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

=

1		

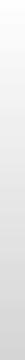


Vertical line
detector

$$\begin{matrix} -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \end{matrix}$$

=

4



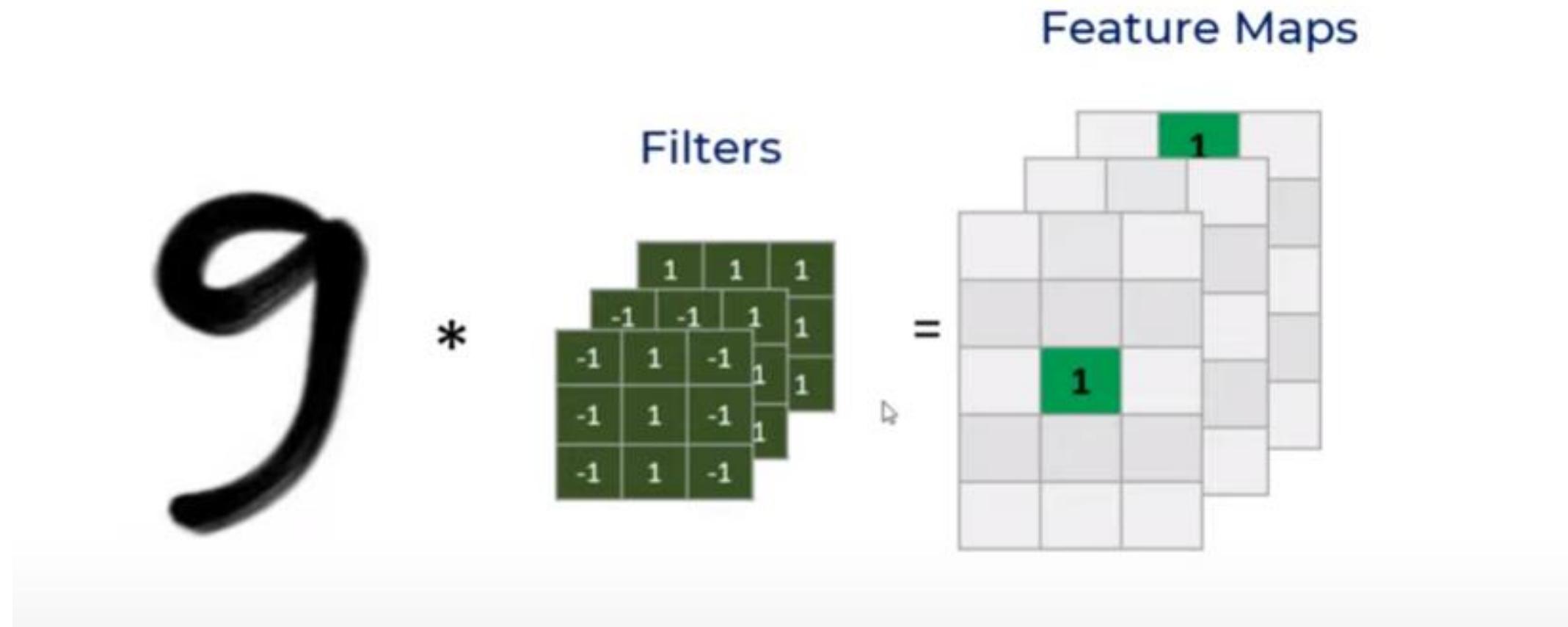
Diagonal line
detector

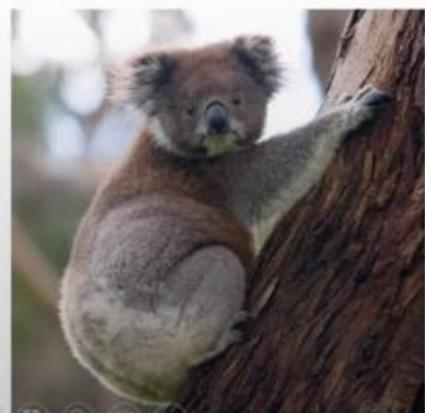
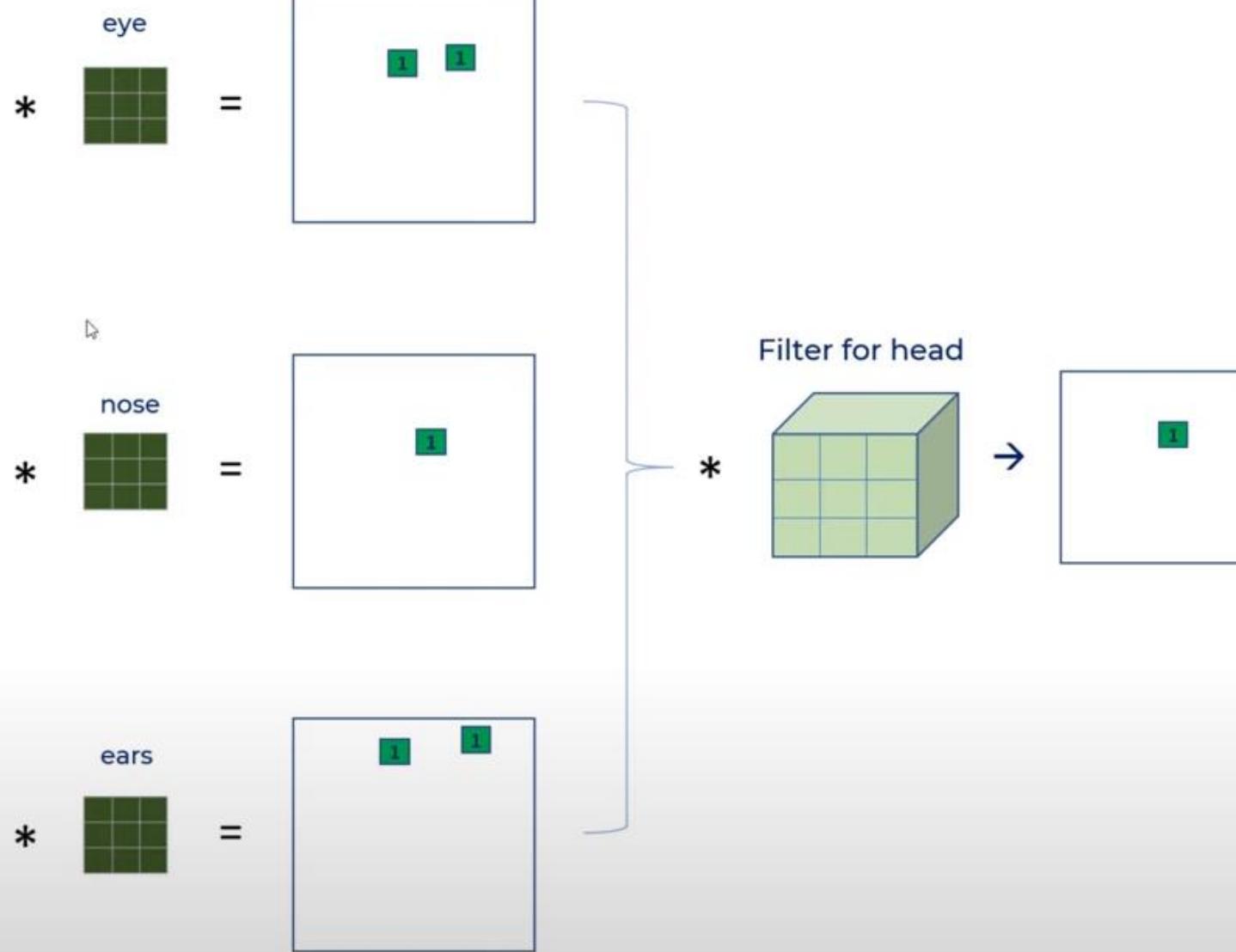
$$\begin{matrix} -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{matrix}$$

=

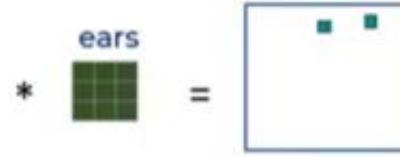
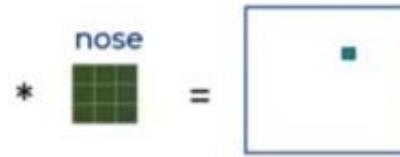
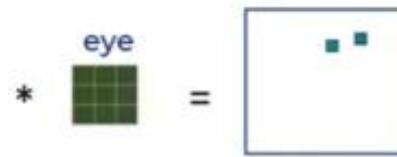
Convolutional Neural Networks (CNNs)

Aggregation of the feature maps

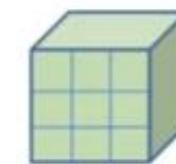




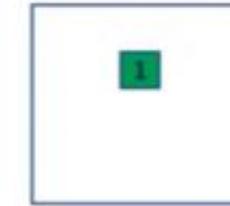
Feature Extraction



head

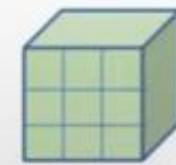


→



flatten

body

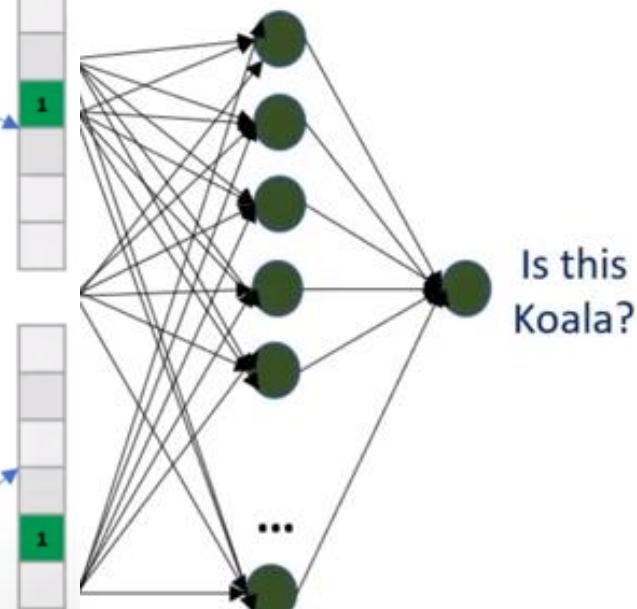


→



flatten

Classification



Convolutional Neural Networks (CNNs)

Activation Functions

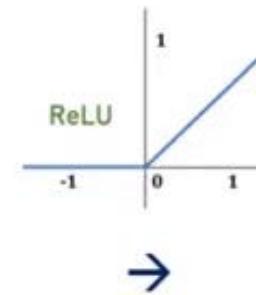
-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

* Loopy pattern filter

→

-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

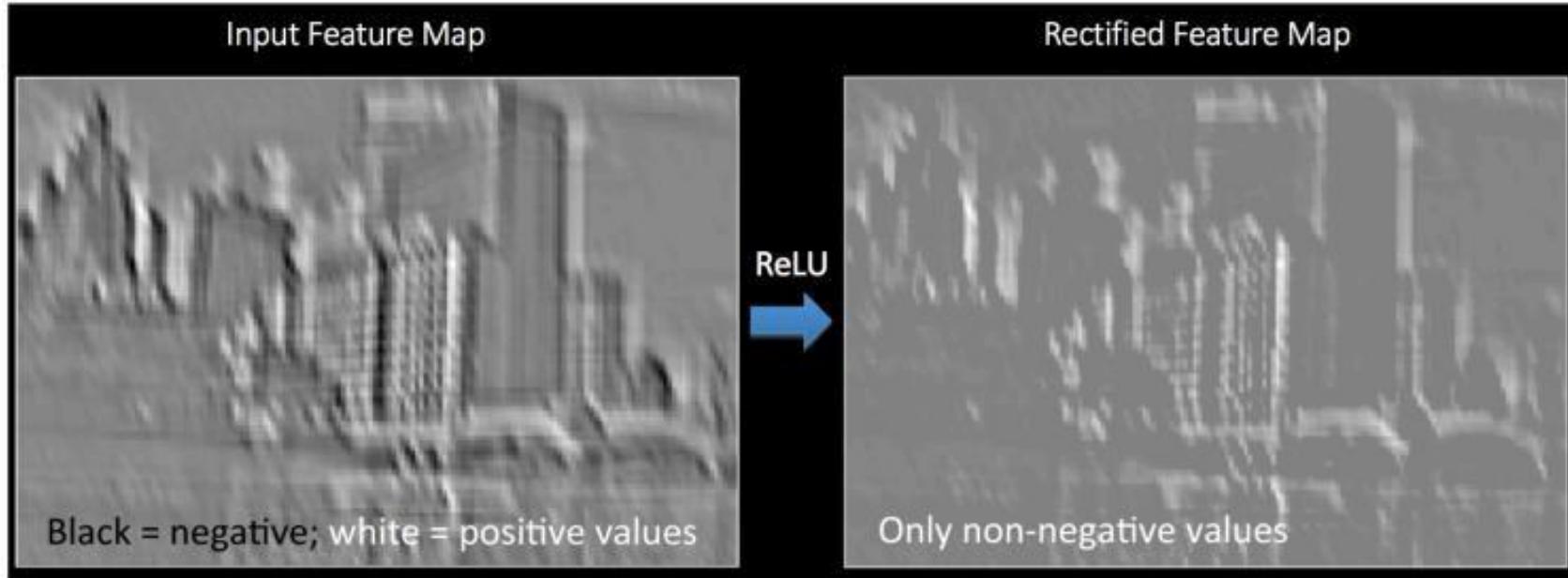
-0.11	1	-0.11
-0.55	0.11	-0.33
-0.33	0.33	-0.33
-0.22	-0.11	-0.22
-0.33	-0.33	-0.33



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

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)

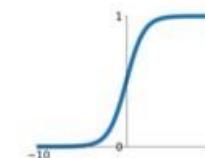
It is one of the most common activation functions.

- 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

Activation Functions

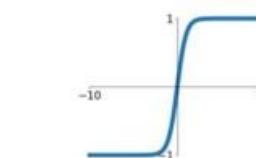
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



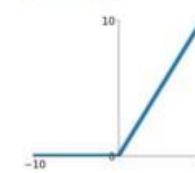
tanh

$$\tanh(x)$$

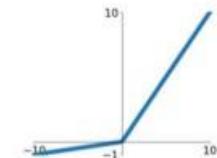


ReLU

$$\max(0, x)$$



Leaky ReLU

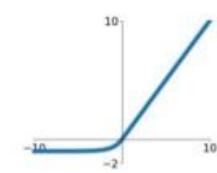
$$\max(0.1x, x)$$


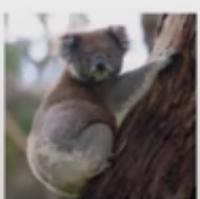
Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$





1920 x 1080

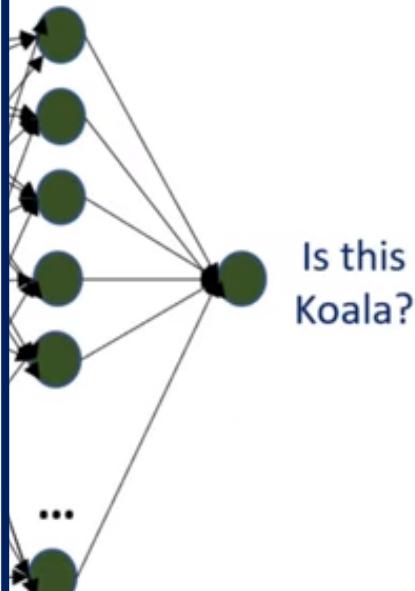
* eye
* nose
* ear
* hand
* leg

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 = $6 \text{ mil} * 4 \text{ mil}$
= 24 million



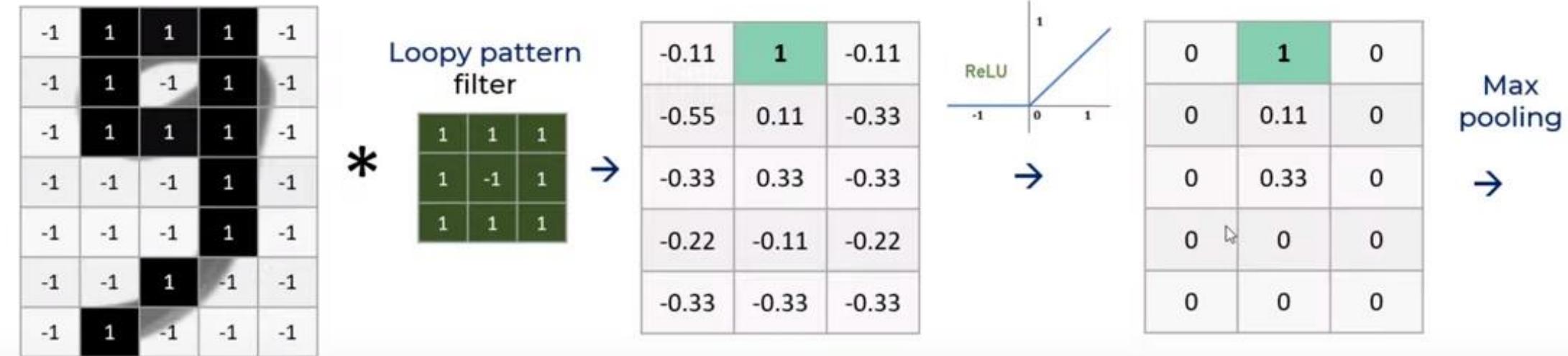
5	1	3	4
8	2	9	2
1	3	0	1
2	2	2	0



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

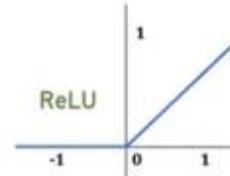
Loopy pattern filter

*

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



→

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

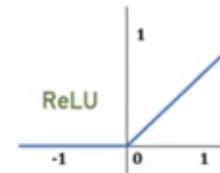
Shifted 9 at different position

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

$$* \quad \begin{matrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{matrix} \rightarrow$$

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

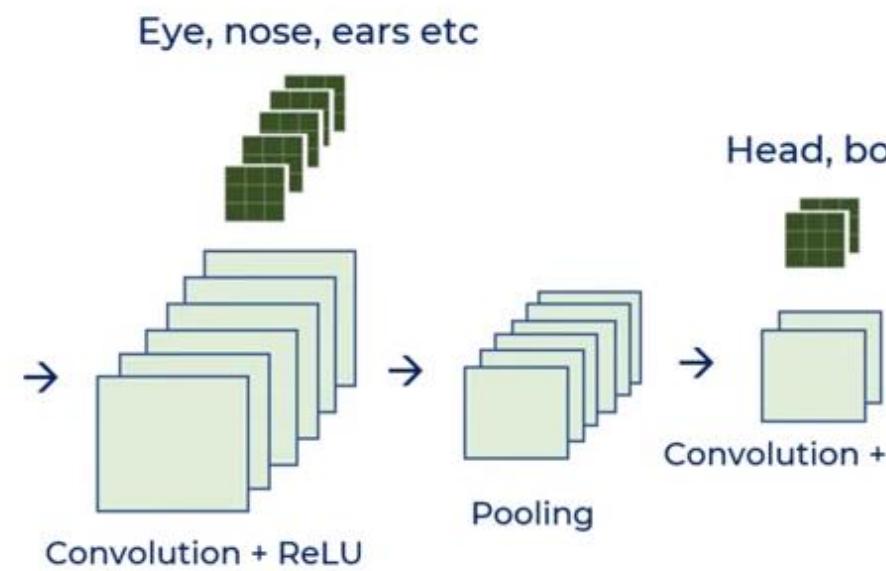


1	0	0
0.11	0	0.33
0.33	0	0
0	0	0
0	0	0

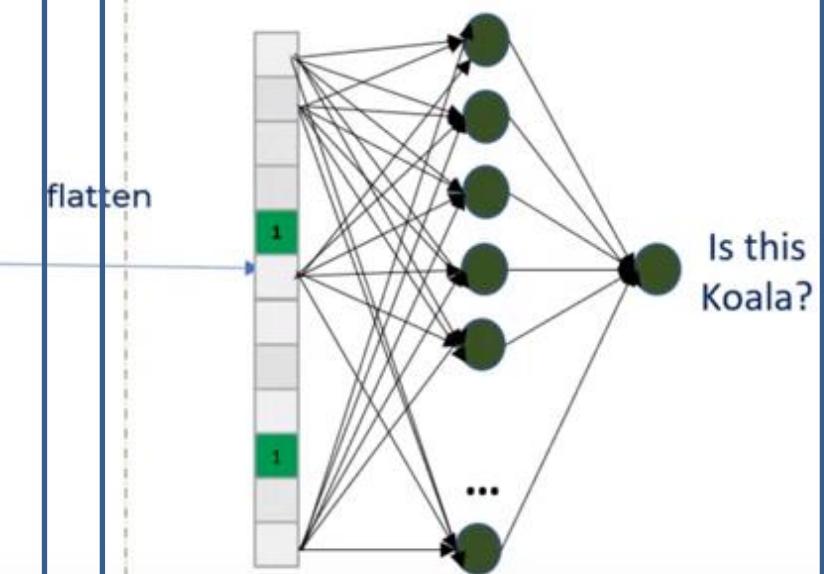
Max pooling
→

1	0.33
0.33	0.33
0.33	0
0	0

Feature Extraction

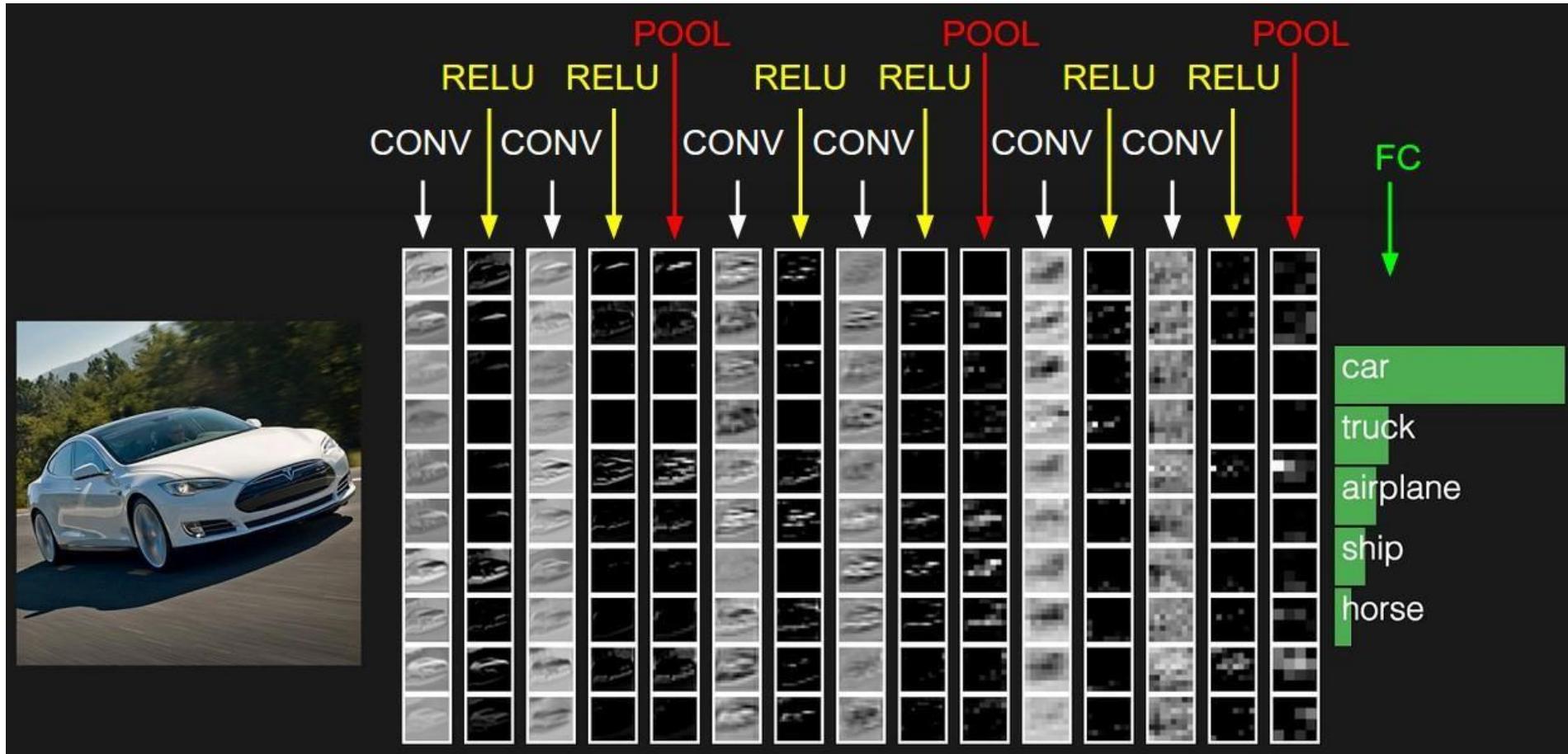


Classification

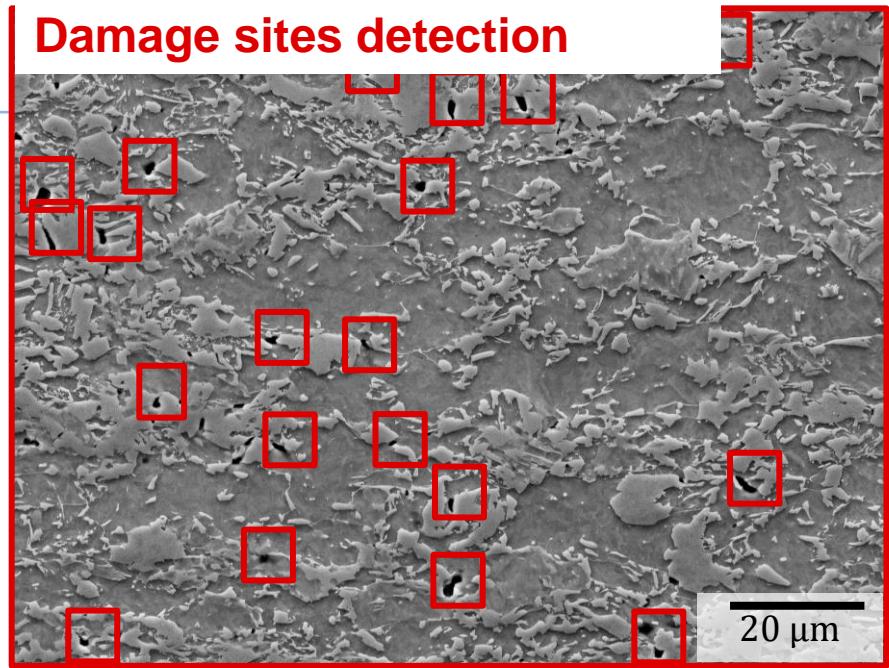


Convolutional Neural Networks (CNNs)

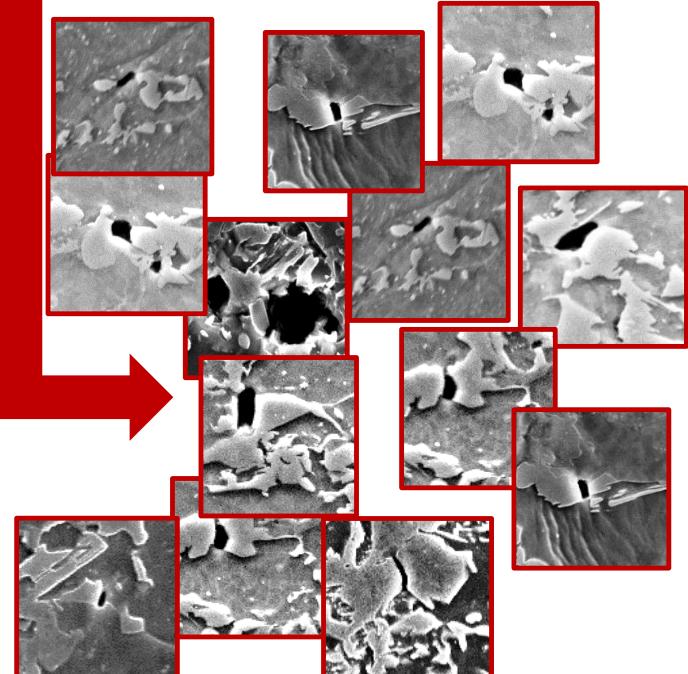
Example: Input >> [[Conv >> ReLU] * 2 >> Pool] * 3 >> FC



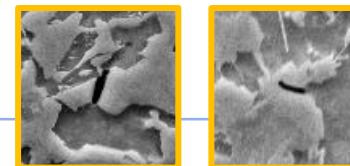
Damage sites detection



DBSCAN Clustering algorithm



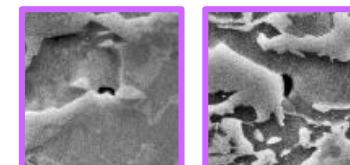
Martensite Crack



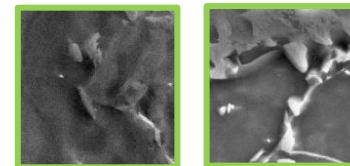
Notch



Interface Deco.

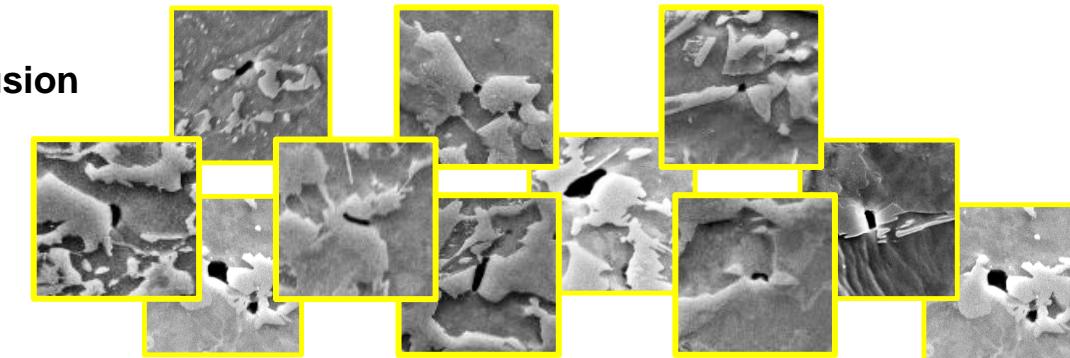


Shadowing



1st CNN

Not Inclusion



Inclusion



2nd CNN



RWTH AACHEN
UNIVERSITY

Convolutional Neural Networks (CNNs)

Architecture Examples

INPUT \rightarrow [[CONV \rightarrow RELU] N \rightarrow POOL?] M \rightarrow [FC \rightarrow RELU] K \rightarrow FC

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

- INPUT \rightarrow FC, implements a linear classifier. Here $N = M = K = 0$.
- INPUT \rightarrow CONV \rightarrow RELU \rightarrow FC
- INPUT \rightarrow [CONV \rightarrow RELU \rightarrow POOL] 2 \rightarrow FC \rightarrow RELU \rightarrow FC. Here we see that there is a single CONV layer between every POOL layer.
- INPUT \rightarrow [CONV \rightarrow RELU \rightarrow CONV \rightarrow RELU \rightarrow POOL] 3 \rightarrow [FC \rightarrow RELU] 2 \rightarrow 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...