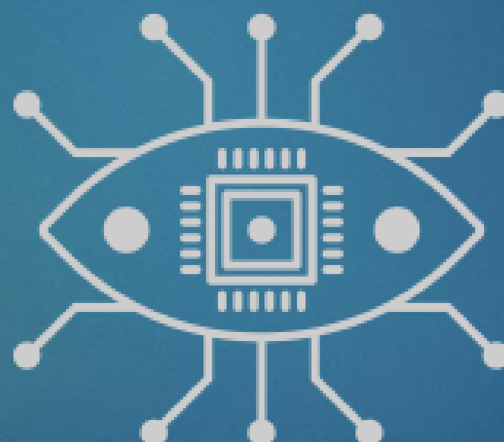
alitourani

Deep Learning from Scratch

Session #7: Convolutional Neural Networks



by: Ali Tourani – Summer 2021

Agenda

- Computer Vision
- Feature Extraction
- Classification vs. Regression
- Convolution
- Convolutional Neural Networks
- CNNs Layers

Computer Vision

- A very popular field of AI and an interdisciplinary scientific field
- Enabling computers to understand from digital images or videos
  - Taking action based on the information obtained from visual inputs
- Data are captured from cameras or generated in computers
- Modelling **image processing** using ML techniques
  - Applying ML to recognize patterns from images/frames
- What actions to do?
  - Distinguishing between objects, classifying them, etc.

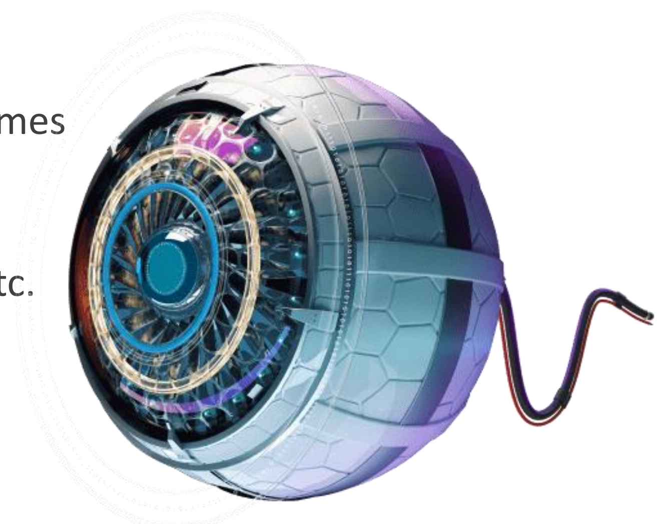


Image source: https://www.softeq.com

Computer Vision

By utilizing Computer Vision algorithms we can identify objects

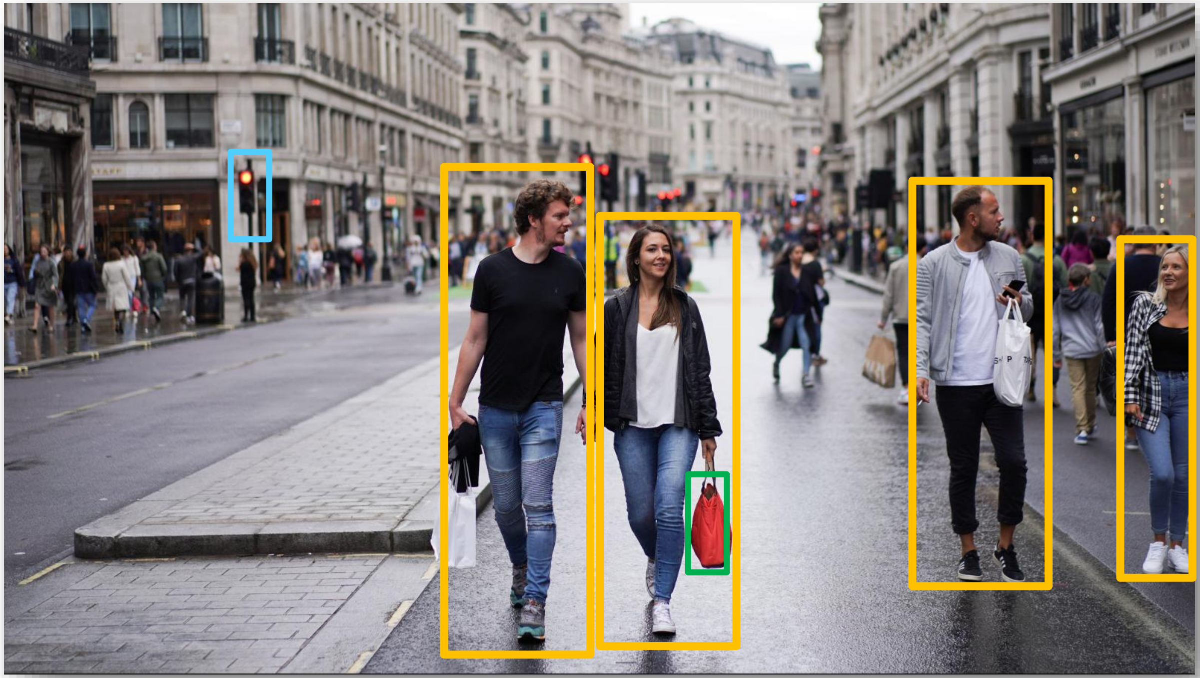


Image source: https://www.qz.com

Computer Vision

By utilizing Computer Vision algorithms we can predict future actions/events

Image source: https://www.qz.com

Computer Vision

By utilizing Computer Vision algorithms we can extract info and discover patterns

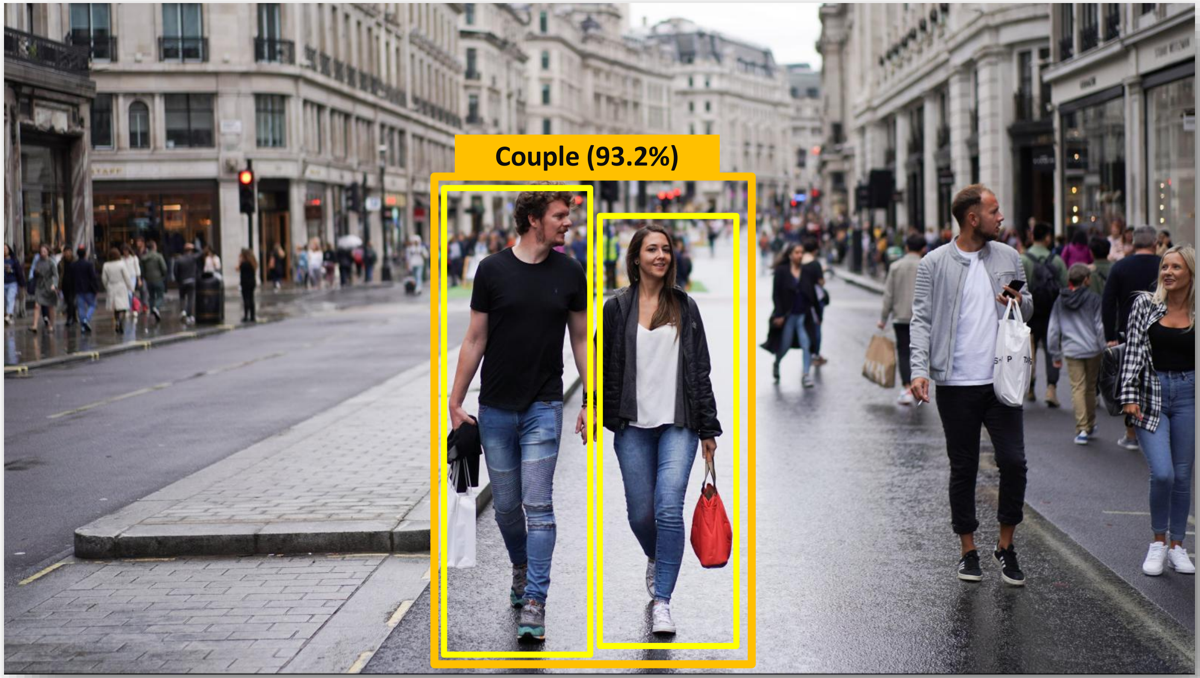


Image source: https://www.qz.com

Computer Vision

Object Tracking

Tracking an object after detecting it in an image

Image Classification

Predicting which class does an image belong

Content-based Image Retrieval

Searching and retrieving images from data sources

Object Detection

Detecting whether an object exists in an image

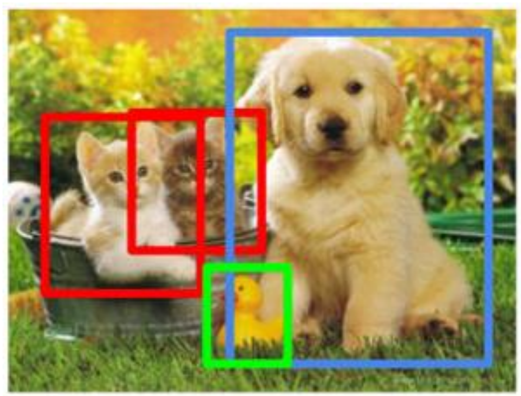
Computer Vision

Image Classification

Computer Vision

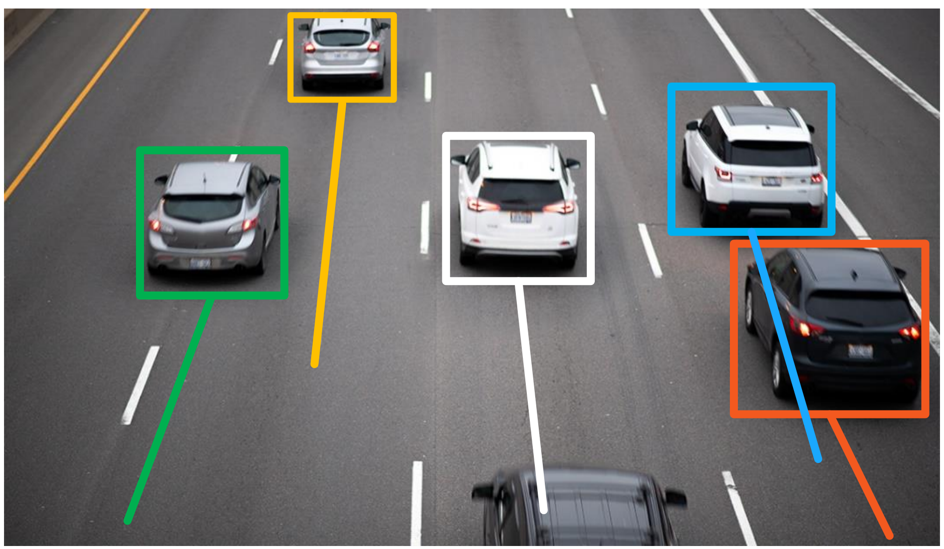
Object Detection

Color	Class
●	Person
●	Car
●	Cat
●	Dog
●	Duck



Computer Vision

Object Tracking



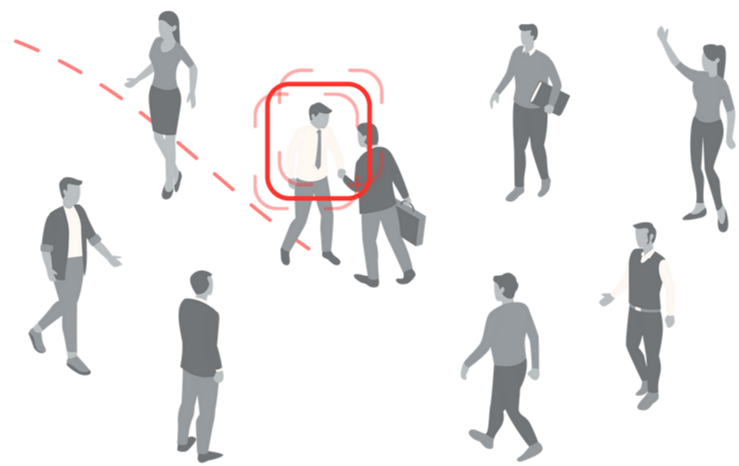


Image source: https://www.aarp.org and https://www.scylla.ai

Computer Vision

Image Retrieval

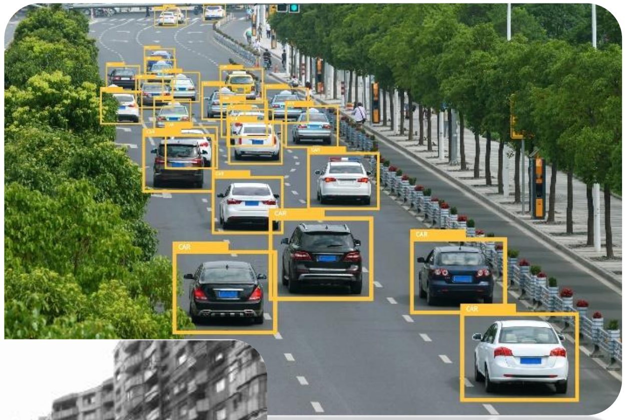
Query

Answer

Computer Vision

Applications of CV – Autonomous Vehicles

- Vehicle Detection
- Pedestrian Detection
- Traffic Sign Recognition
- Lane Markers Detection



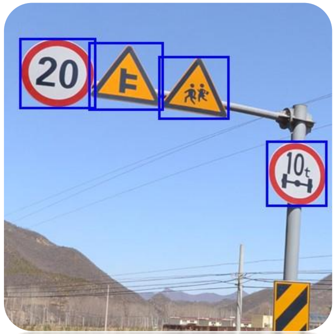
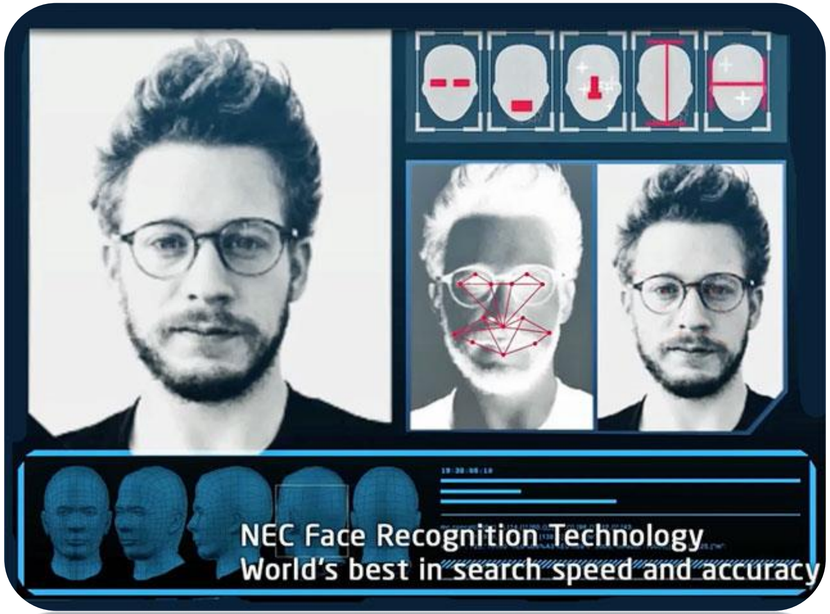


Image source: https://www.mouser.lu

Computer Vision

Applications of CV – Face Recognition



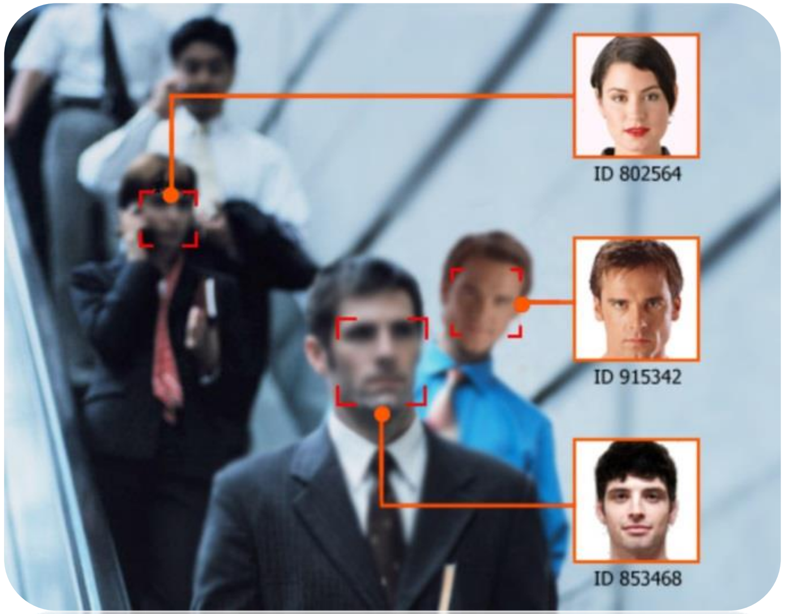


Image source: https://www.protonss.com and https://www.americansecuritytoday.com

Computer Vision

Applications of CV – Robotics

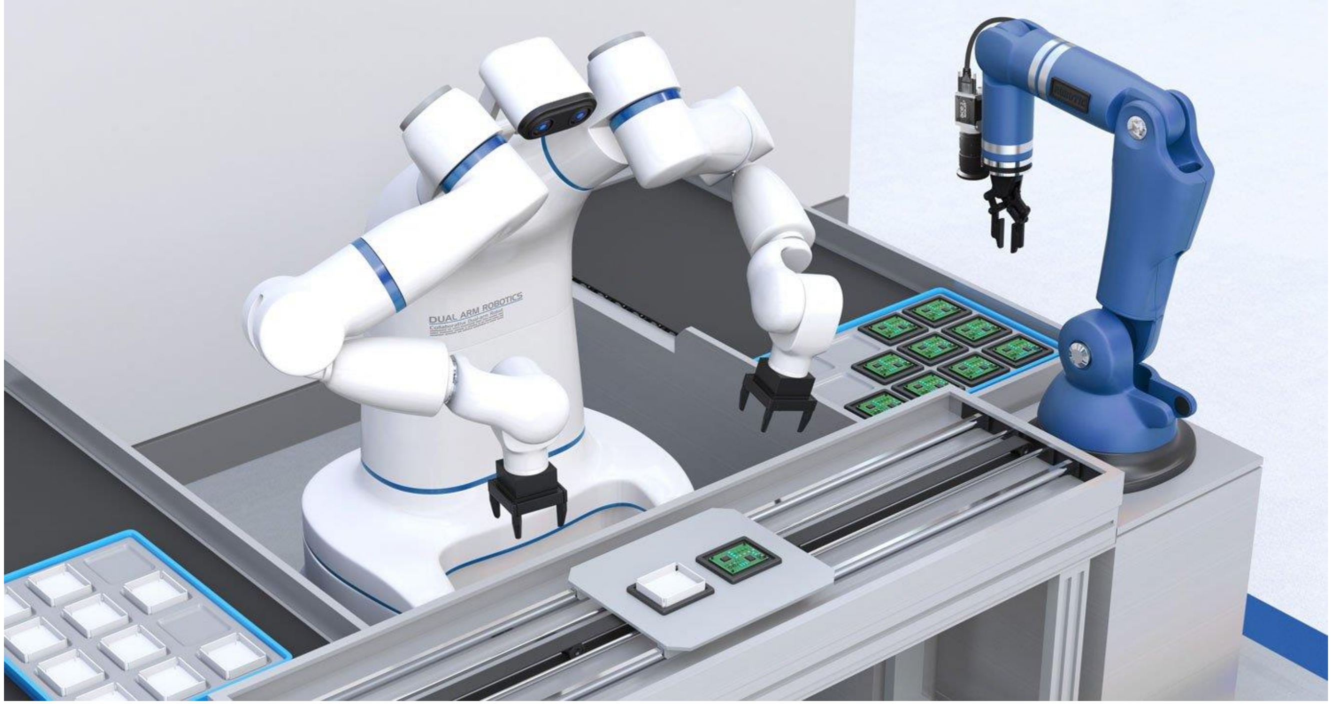


Image source: https://www.manometcurrent.com

Computer Vision

Applications of CV – Google Translate

Image source: https://www.mentalfloss.com

Computer Vision

Applications of CV – Mobile Computing

Image source: https://www.adobe.com and https://www.analyticsvidhya.com

Computer Vision

Applications of CV – Abandoned Object Detection

Image source: https://www.researchgate.org

Computer Vision

Applications of CV – Medicine

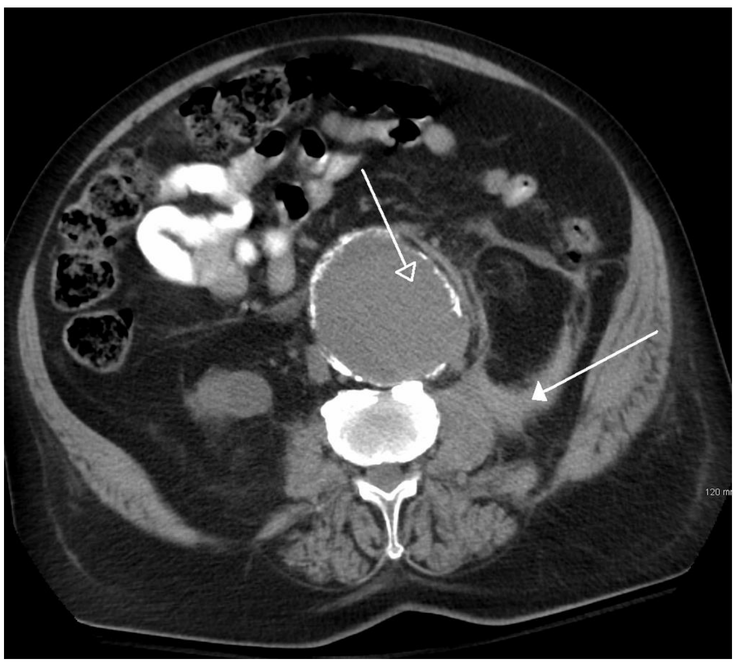


Image source: https://www.exxactcorp.com and https://blog.restb.ai

Feature Extraction

How do computers understand digital images?

- Digital images
  - Representations of real image as a set of numbers
  - Divided into small areas, AKA pixels
  - Pixels hold a small set of numbers for:
    - Color
    - Brightness
    - Intensity of light

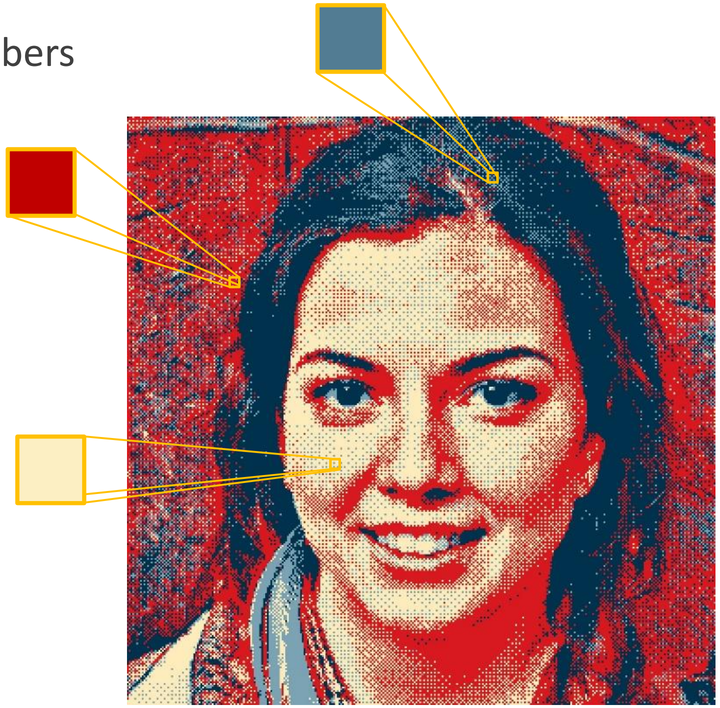


Image source: https://www.fiverr.com

Feature Extraction

How do computers understand digital images?

- Types of Digital Images

Binary

Grayscale

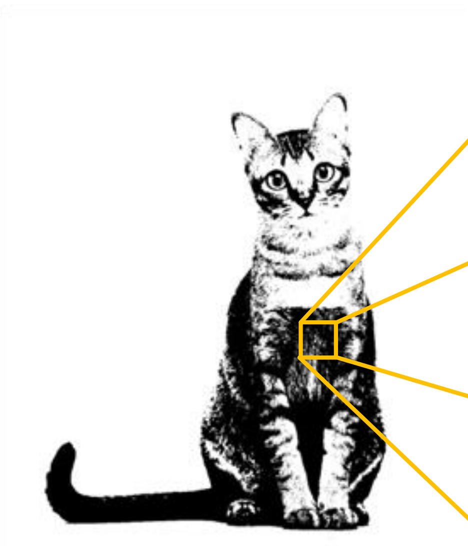
Color

Image source: https://www.fiverr.com

Feature Extraction

How do computers understand digital images?

- Types of Digital Images



Binary image

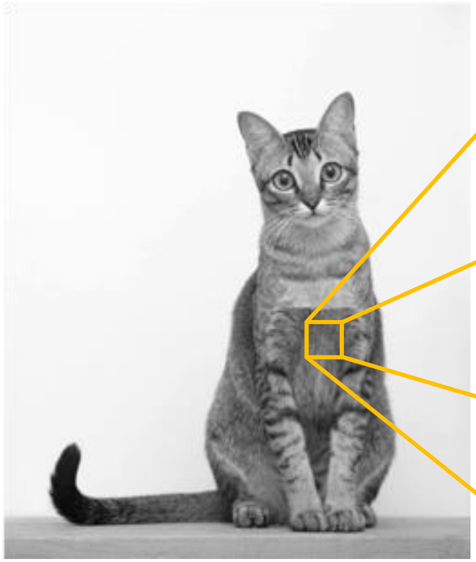
1	1	1	1	1	0	1	1	0	0
1	1	1	0	0	0	0	0	1	1
1	1	1	1	0	0	0	0	1	1
1	1	1	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	1	1
1	1	1	1	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0

Image source: https://www.fiverr.com

Feature Extraction

How do computers understand digital images?

- Types of Digital Images



121	99	78	69	129	194	93	182	81	42
78	92	188	96	95	189	204	219	94	67
105	182	79	94	79	199	219	241	201	95
185	178	98	65	132	14	18	192	182	90
98	182	201	92	94	182	194	182	71	95
84	95	81	95	102	201	29	95	185	206
96	47	173	82	96	195	181	148	103	97
98	57	142	95	84	197	61	184	73	93

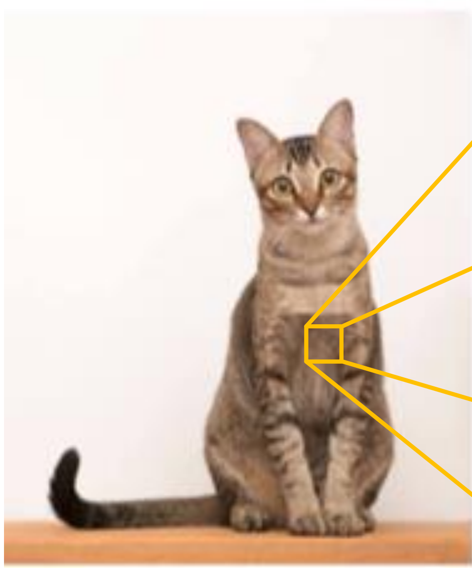
Grayscale image

Image source: https://www.fiverr.com

Feature Extraction

How do computers understand digital images?

- Types of Digital Images



Color image (RGB)

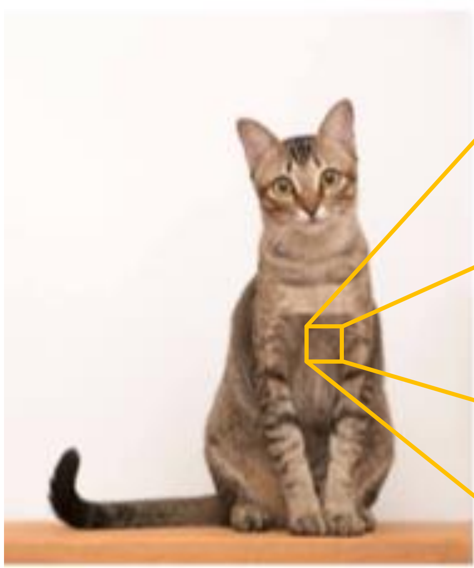
(131, 101, 80)	(131, 101, 80)	(131, 102, 81)	(181, 145, 109)
(174, 139, 91)	(117, 88, 69)	(122, 96, 80)	(162, 130, 99)
(181, 145, 109)	(150, 96, 73)	(163, 138, 117)	(156, 124, 97)
(138, 106, 85)	(97, 73, 55)	(175, 141, 118)	(216, 181, 176)
(99, 78, 61)	(190, 164, 136)	(107, 86, 70)	(97, 73, 55)

Image source: https://www.fiverr.com

Feature Extraction

How do computers understand digital images?

- Types of Digital Images



Color image (RGB)

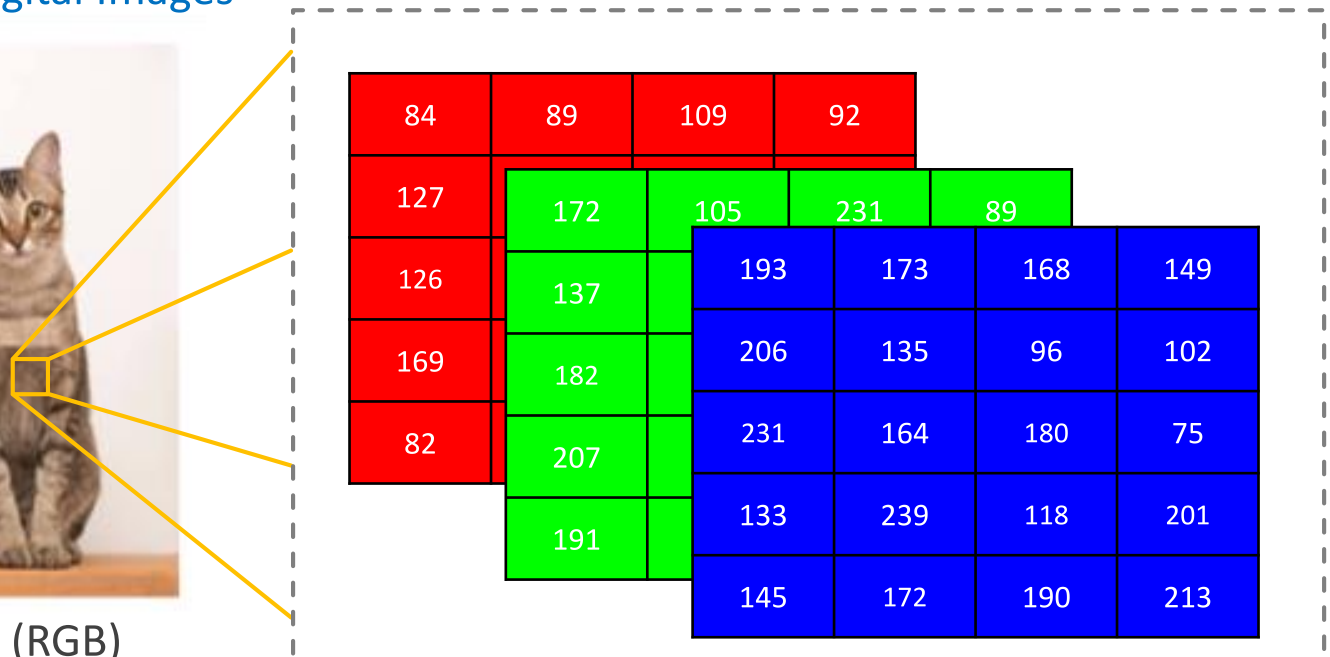


Image source: https://www.fiverr.com

Feature Extraction

What are features of an image?

- Visual parts or patterns in an image
- Useful to identify it and extract information from it

Where to find the features?

- Corners
- Edges
- Regions of Interest (RoI) points
- Etc.

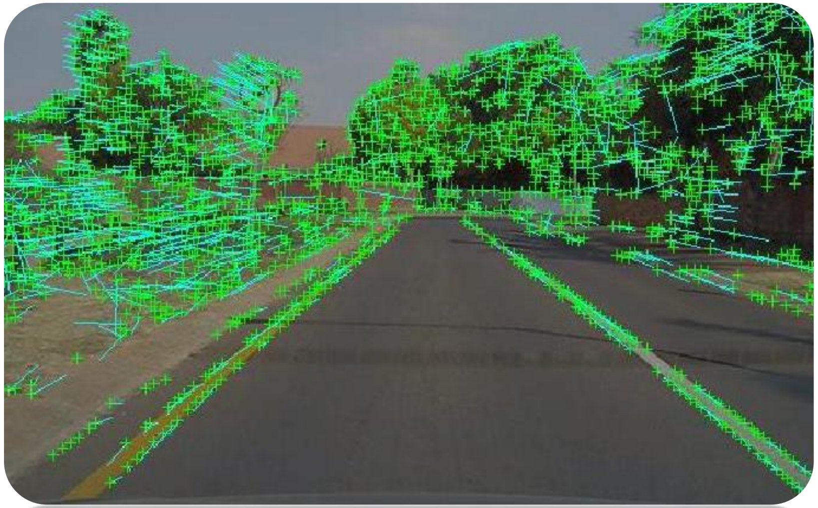


Image source: https://www.semanticscholar.org

Feature Extraction

Where to find these features?

Image source: https://www.github.com

Feature Extraction

Where to find these features?

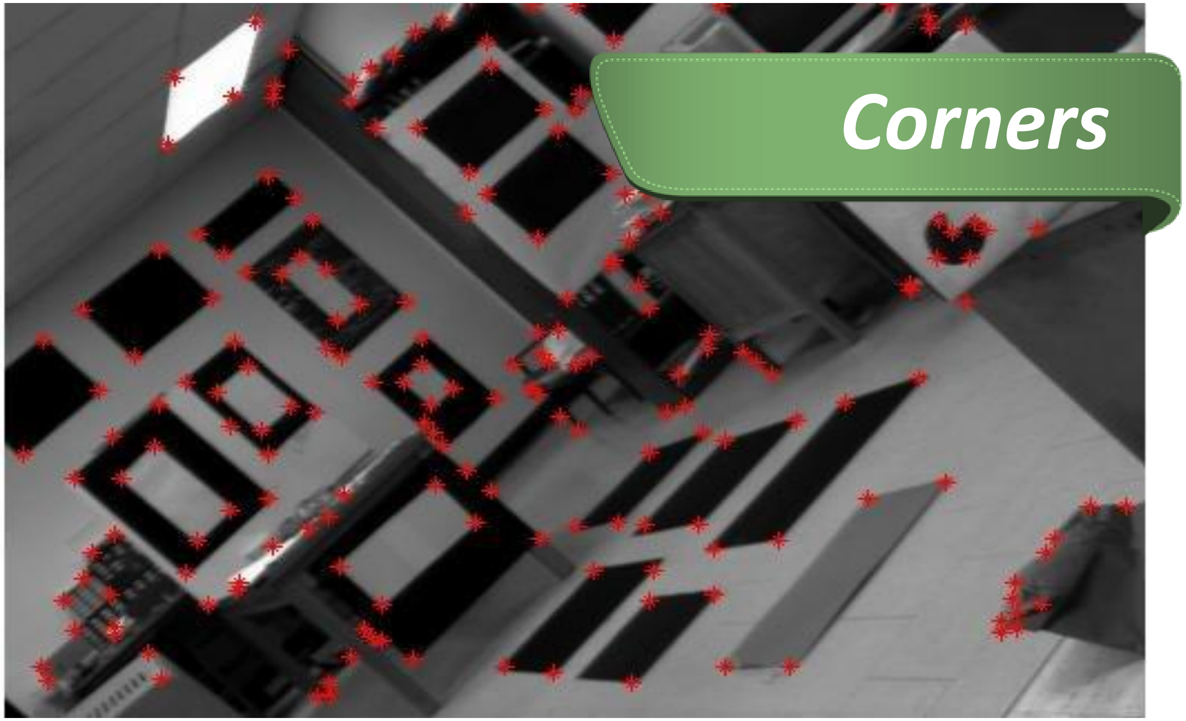


Image source: https://www.researchgate.net

Feature Extraction

Where to find these features?

Image source: https://www.researchgate.net

Feature Extraction

Looking for features using the pixel representation

- Low-level features
  - Capturing local appearance/texture statistics of objects
  - Highly effective in classification tasks
- High-level features
  - Semantic concepts that can be interpreted in a scene
  - The probability of observing an object in an image

Hierarchy of Features

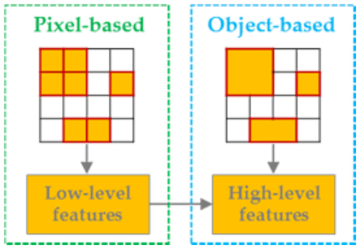
Low-level features
- Algorithms: SIFT, SURF, etc.
- Interest points, edges, etc.

High-level features
- Mainly used in CV
- Birds, leaves

Feature Extraction

Looking for features using the pixel representation

- Low-level and High-level features
  - Recall: Session#1 - Basics



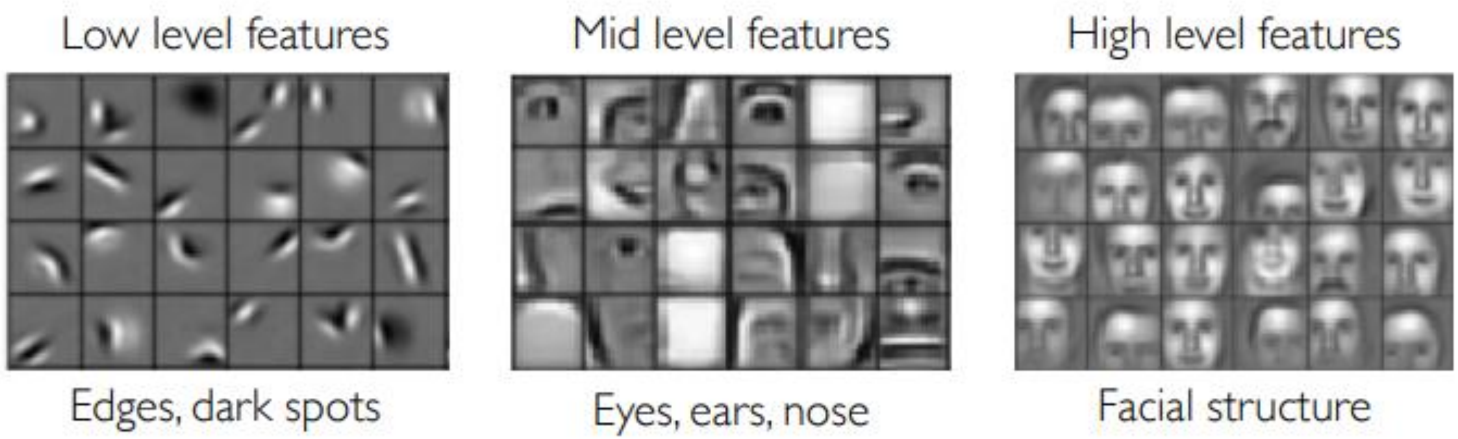
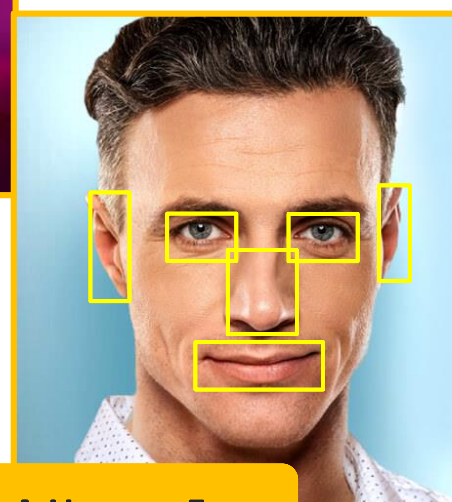


Image source: https://www.medium.com

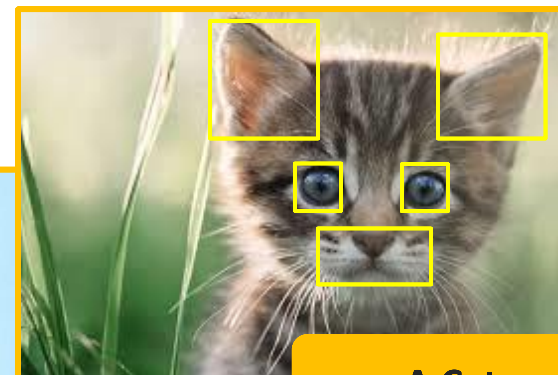
Feature Extraction

What are the high-level features that classify these images?

A Car



A Human Face



A Cat

Feature Extraction

What are the high-level features that classify these images?

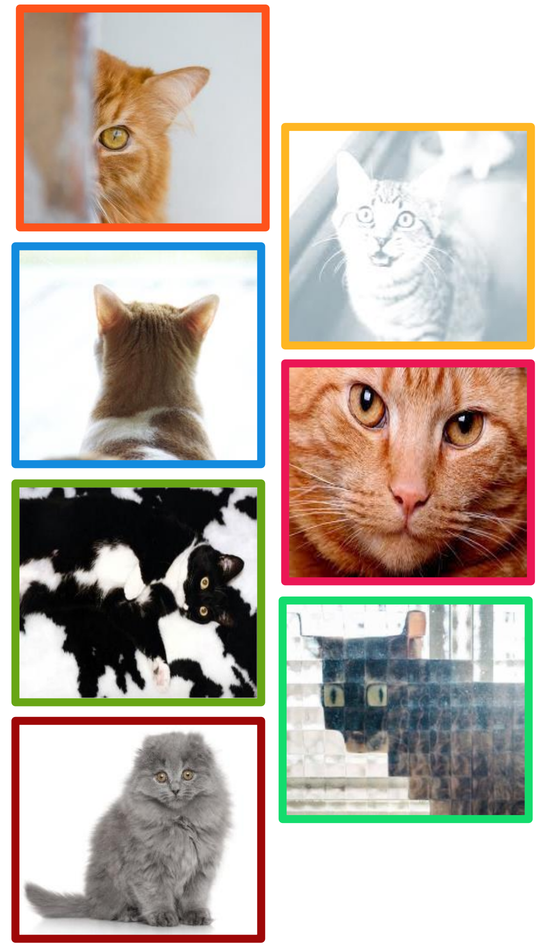
Playing football

Playing basketball

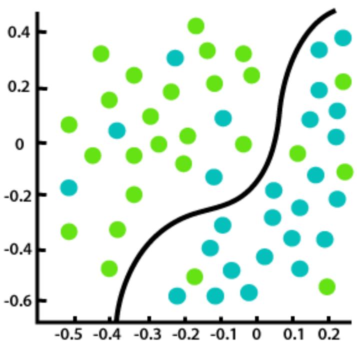
Feature Extraction

What are the challenges?

- Occlusion
- Illumination
- Scale
- Viewpoint
- Background Clutter
- Deformation
- Intra-class Variation



Classification vs. Regression



Classification

- Predicting a label
- Outputs are discrete
- Outputs = labels
- Input values can be continuous

Use case: forecasting the weather (classes: sunny, rainy, cloudy, snowy)

Regression

- Predicting a quantity
- Outputs are continuous (e.g. an integer or floating point value)
- Outputs can be converted into labels to present a classification problem

Use case: predicting the value of a car two years later

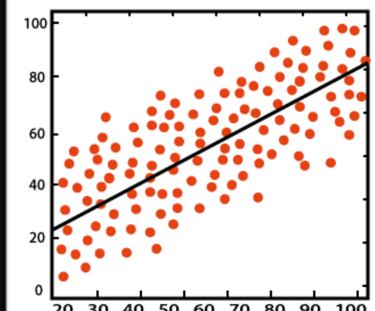


Image source: https://www.javapoint.com

Classification vs. Regression

Classification in Computer Vision

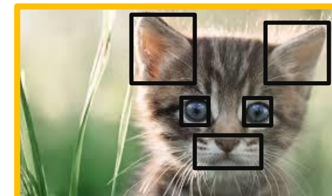
Input image

Pixel representation

121	99	78	69	129	194	93
78	92	188	96	95	189	204
105	182	79	94	79	199	219
185	178	98	65	132	14	18
98	182	201	92	94	182	194
84	95	81	95	102	201	29
96	47	173	82	96	195	181
98	57	142	95	84	197	61

...

....



Unique features of a cat

Class	Probability
Cat	0.85 ✔
Dog	0.09 ✘
Mouse	0.06 ✘

Classification

Image source: https://www.fiverr.com

Classification vs. Regression

Regression in Computer Vision

Input image

Pixel representation

121	99	78	69	129	194	93
78	92	188	96	95	189	204
105	182	79	94	79	199	219
185	178	98	65	132	14	18
98	182	201	92	94	182	194
84	95	81	95	102	201	29
96	47	173	82	96	195	181
98	57	142	95	84	197	61

...

....

Regression

Age?

99+

75

50

25

0

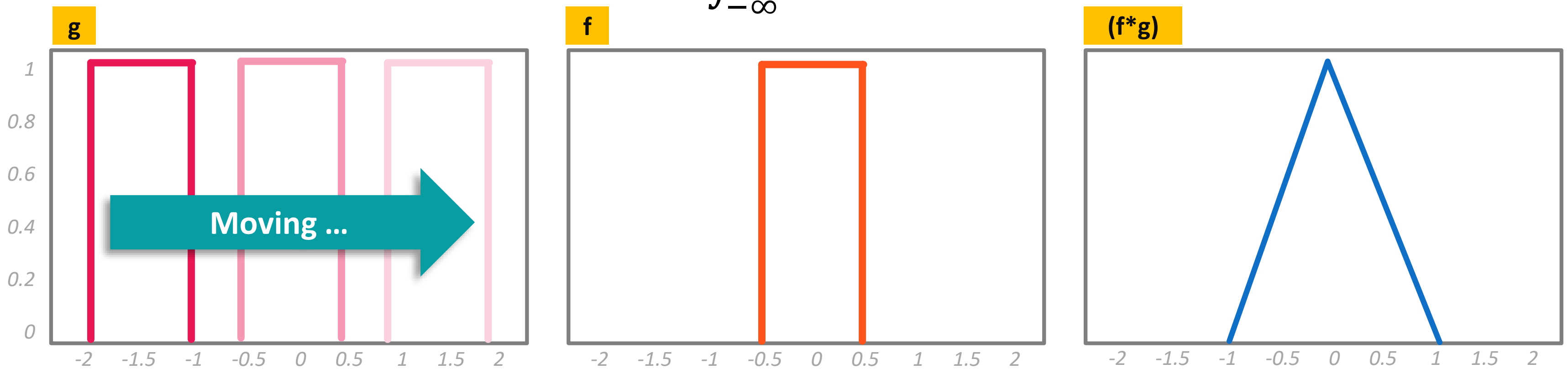
44 (89.72%)

Image source: https://www.today.in-24.com

Convolution

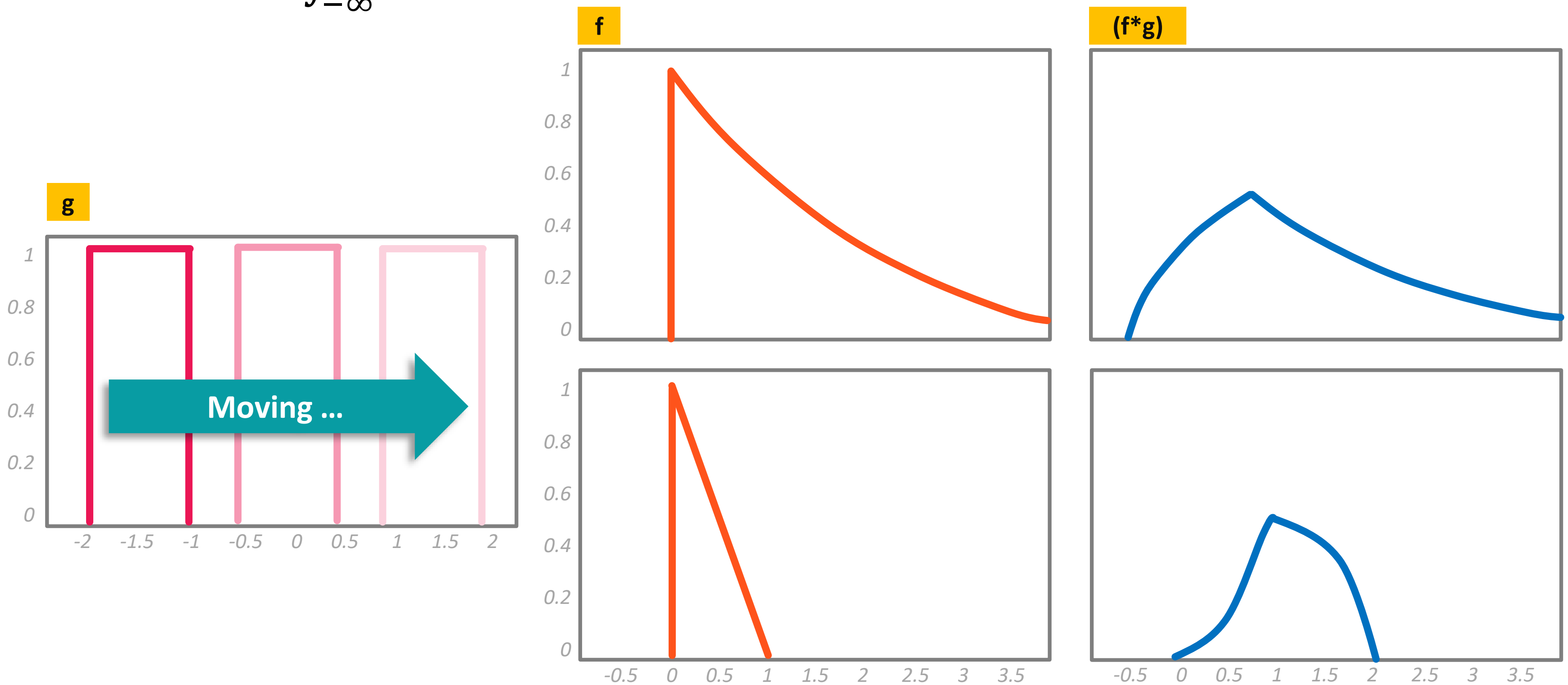
- A mathematical operation on two functions (*f* and *g*) to produce a third one
- **Result:** Identifying how ***f*** can modify the shape of ***g***
- **Definition**: an integral that expresses the amount of overlap of ***f*** as it is shifted over **g**

$$[f * g](t) \equiv \int_{-\infty}^{+\infty} f(\tau) g(t-\tau) . d\tau$$



Convolution

$$[f * g](t) \equiv \int_{-\infty}^{+\infty} f(\tau)g(t-\tau).\,d\tau$$



Convolution

What about applying convolution on images?

Input image

I(0,0)	I(1,0)	I(2,0)	I(3,0)	I(4,0)	I(5,0)	I(6,0)
I(0,1)	I(1,1)	I(2,1)	I(3,1)	I(4,1)	I(5,1)	I(6,1)
I(0,2)	I(1,2)	I(2,2)	I(3,2)	I(4,2)	I(5,2)	I(6,2)
I(0,3)	I(1,3)	I(2,3)	I(3,3)	I(4,3)	I(5,3)	I(6,3)
I(0,4)	I(1,4)	I(2,4)	I(3,4)	I(4,4)	I(5,4)	I(6,4)
I(0,5)	I(1,5)	I(2,5)	I(3,5)	I(4,5)	I(5,5)	I(6,5)
I(0,6)	I(1,6)	I(2,6)	I(3,6)	I(4,6)	I(5,6)	I(6,6)

*

Filter/Kernel/Mask

H(0,0)	H(1,0)	H(2,0)
H(0,1)	H(1,1)	H(2,1)
H(0,2)	H(1,2)	H(2,2)

=

Output image (feature map)

O(0,0)				

Convolution

What about applying convolution on images?

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

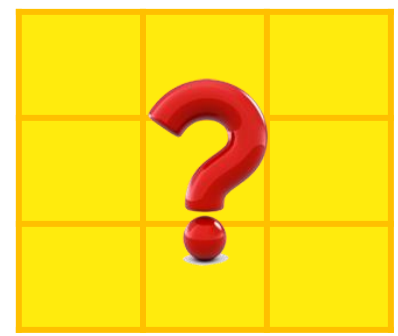
Input image (binary)

*

1	0	1
0	1	1
1	0	1

Filter/Kernel/Mask

=



Output image
(feature map)

Convolution

What about applying convolution on images?

$$
\begin{bmatrix}
1_{\times 1} & 0_{\times 0} & 1_{\times 1} & 0 & 1 \\
1_{\times 0} & 1_{\times 1} & 1_{\times 1} & 0 & 0 \\
1_{\times 1} & 0_{\times 0} & 1_{\times 1} & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{bmatrix}
*
\begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1
\end{bmatrix}
=
\begin{bmatrix}
6 & & \\
 & & \\
 & &
\end{bmatrix}
$$

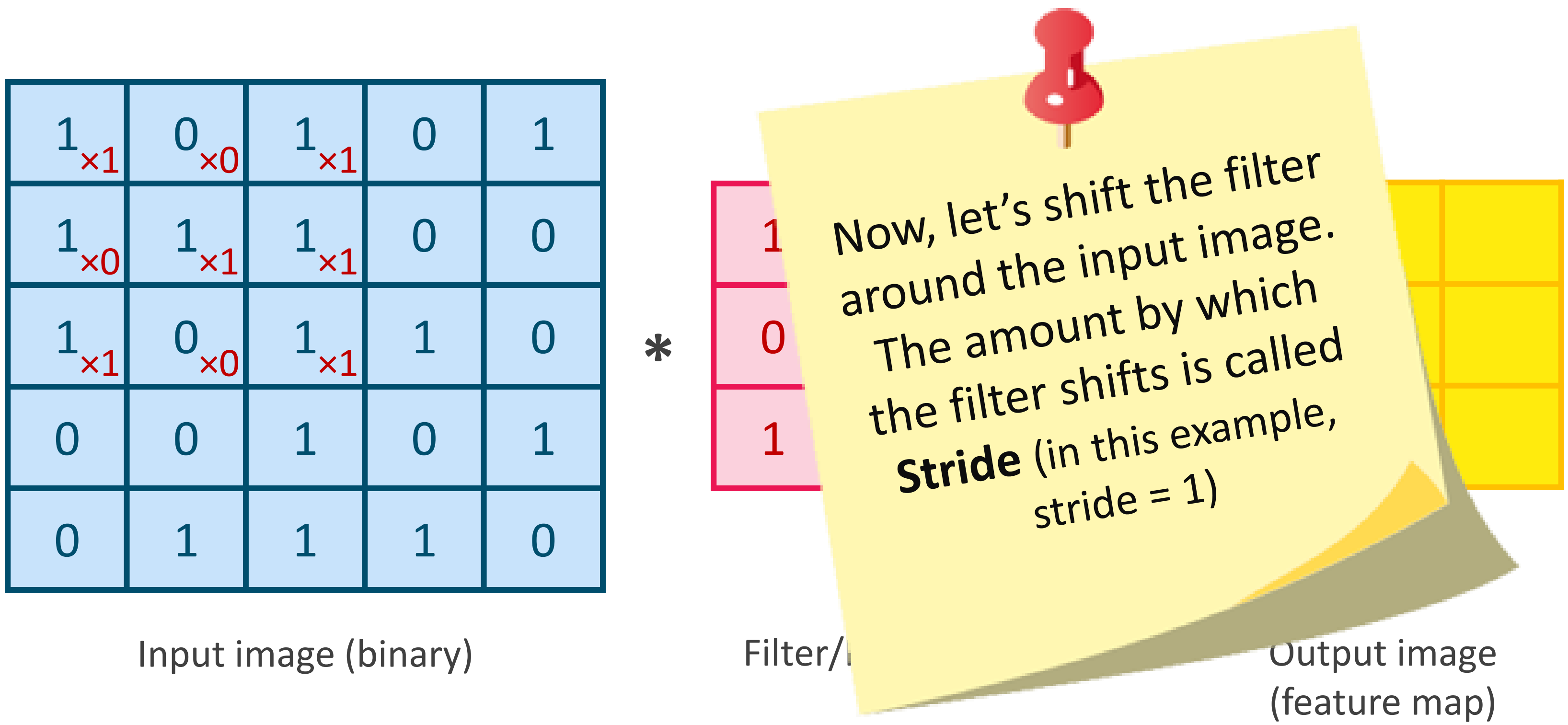
Input image (binary)

Filter/Kernel/Mask

Output image (feature map)

Convolution

What about applying convolution on images?



Convolution

What about applying convolution on images?

$$\begin{bmatrix} 1 & 0_{\times 1} & 1_{\times 0} & 0_{\times 1} & 1 \\ 1 & 1_{\times 0} & 1_{\times 1} & 0_{\times 1} & 0 \\ 1 & 0_{\times 1} & 1_{\times 0} & 1_{\times 1} & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 2 & \\ & & \\ & & \end{bmatrix}$$

Input image (binary)

Filter/Kernel/Mask

Output image (feature map)

Convolution

What about applying convolution on images?

$$
\begin{bmatrix}
1 & 0 & 1_{\times 1} & 0_{\times 0} & 1_{\times 1} \\
1 & 1 & 1_{\times 0} & 0_{\times 1} & 0_{\times 1} \\
1 & 0 & 1_{\times 1} & 1_{\times 0} & 0_{\times 1} \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0
\end{bmatrix}
*
\begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1
\end{bmatrix}
=
\begin{bmatrix}
6 & 2 & 3 \\
 & & \\
 & &
\end{bmatrix}
$$

Input image (binary) | Filter/Kernel/Mask | Output image (feature map)

Convolution

What about applying convolution on images?

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 2 & 3 \\ 4 & 3 & 4 \\ 4 & 4 & 3 \end{bmatrix}$$

Input image (binary)

Filter/Kernel/Mask

Output image (feature map)

Convolution

What about applying convolution on images?

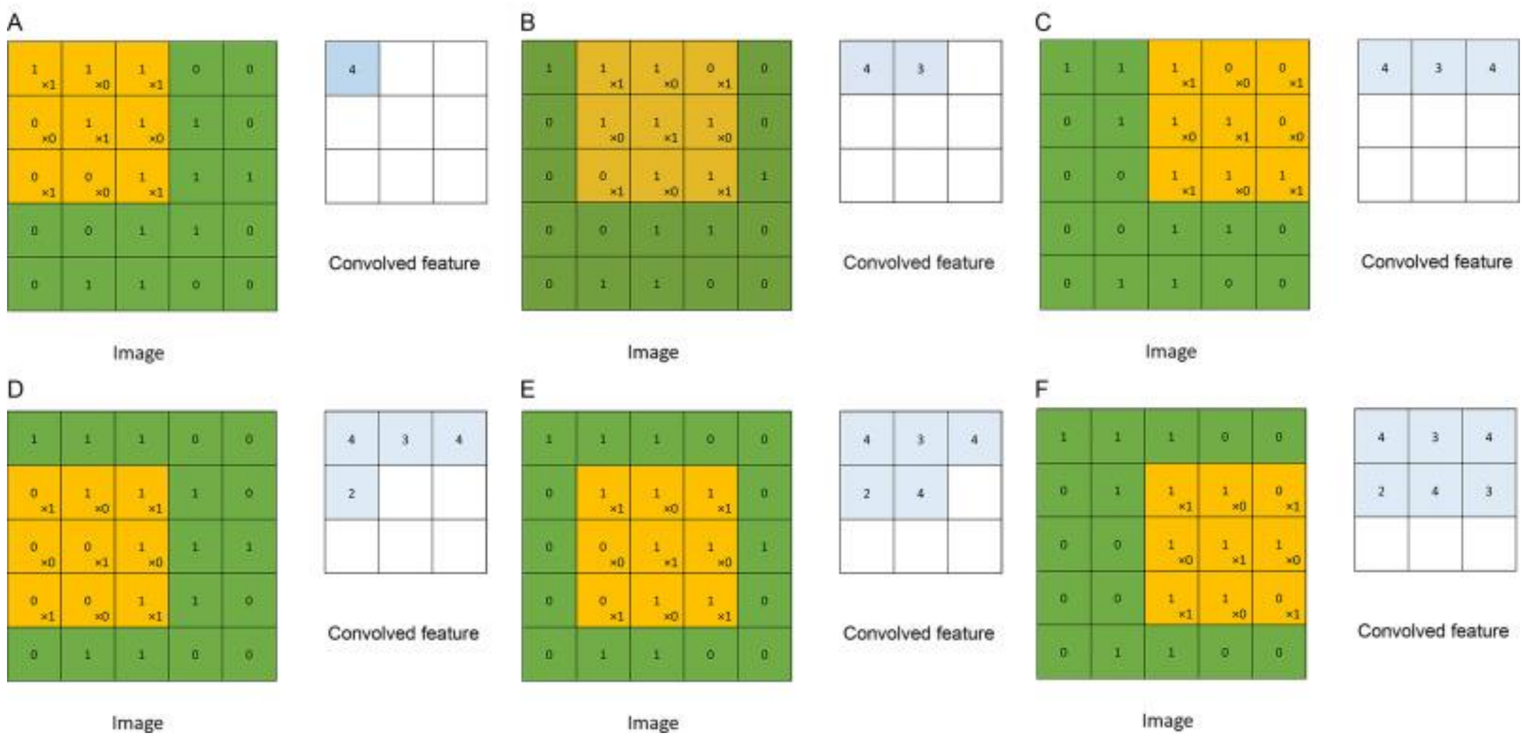


Image source: https://www.sciencedirect.com

Convolution

What about applying convolution on images?

Input image

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Simple box blur

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal lines

Image source: https://www.aishack.in/tutorials

Convolution

What about applying convolution on images?

Input image

-1	-1	-1
-1	8	-1
-1	-1	-1

Edge detection

0	-1	0
-1	4	-1
0	-1	0

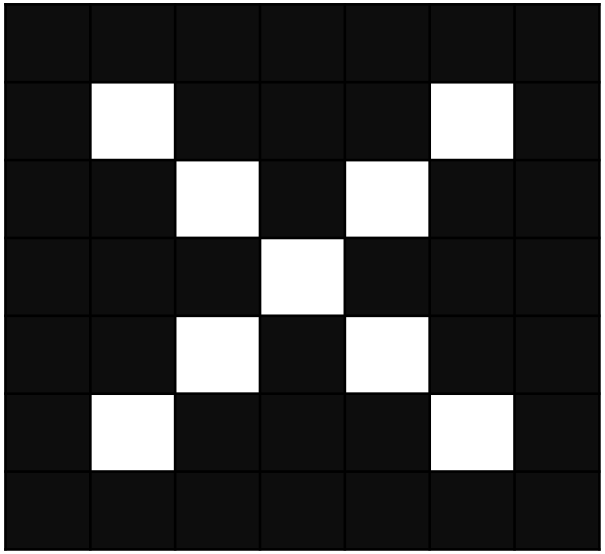
The Laplacian operator

Image source: https://www.aishack.in/tutorials

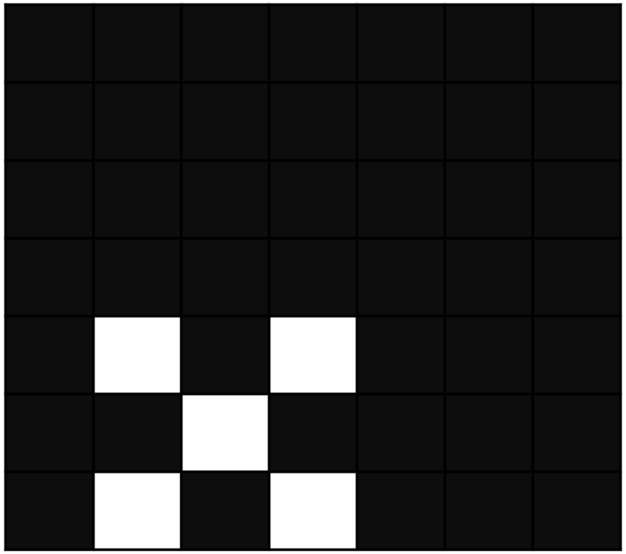
Convolution

Why is it important to use Convolution?

- A simple case study: classifying the letter “X” even if it is rotated/shifted



The letter “X” in a binary image

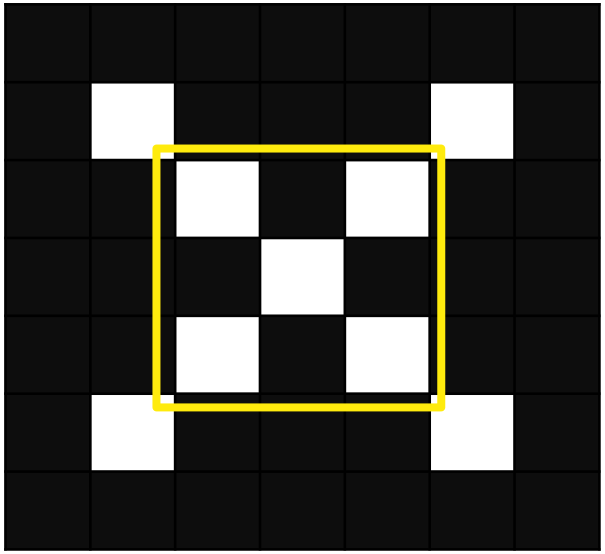


The scaled version

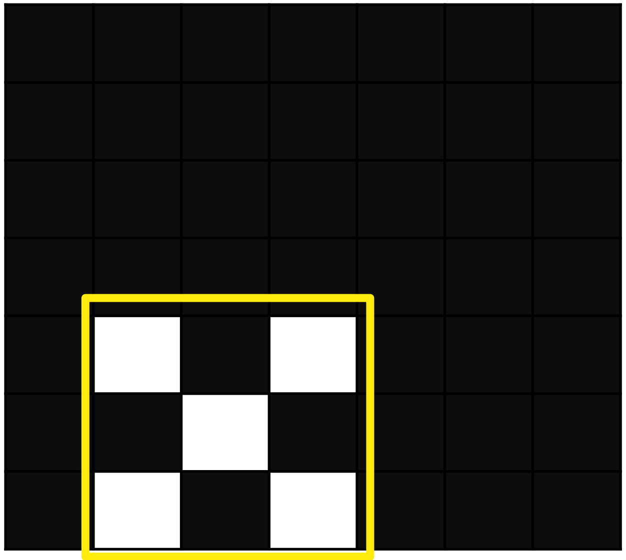
Convolution

Why is it important to use Convolution?

- A simple case study: classifying the letter “X” even if it is rotated/shifted



The letter “X” in a binary image



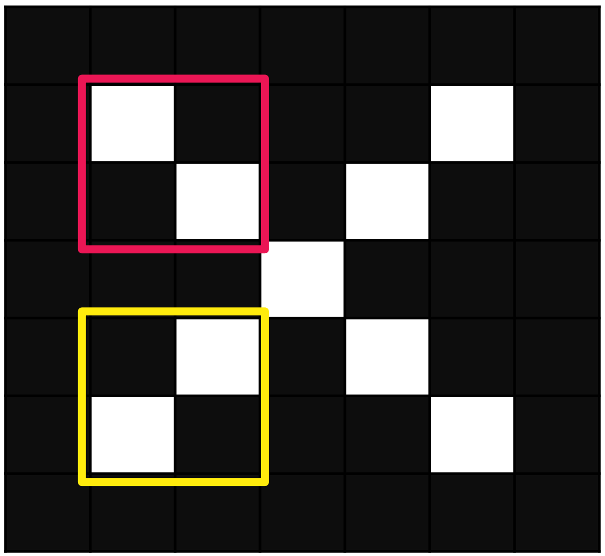
The scaled version

The same features!

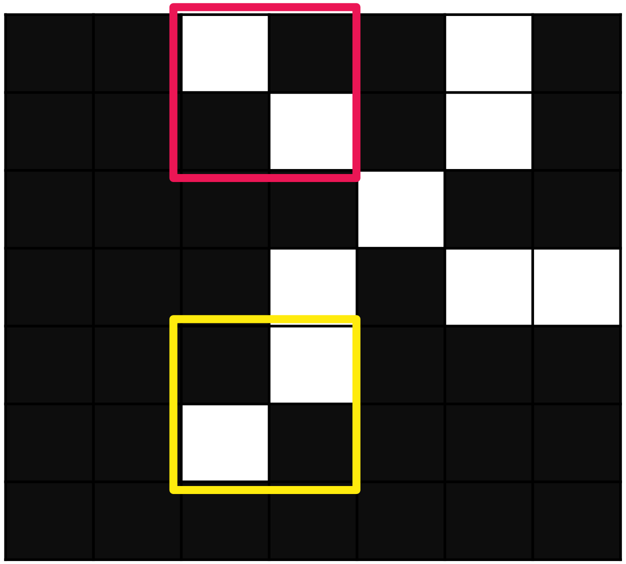
Convolution

Why is it important to use Convolution?

- A simple case study: classifying the letter “X” even if it is rotated/shifted



The letter “X” in a binary image



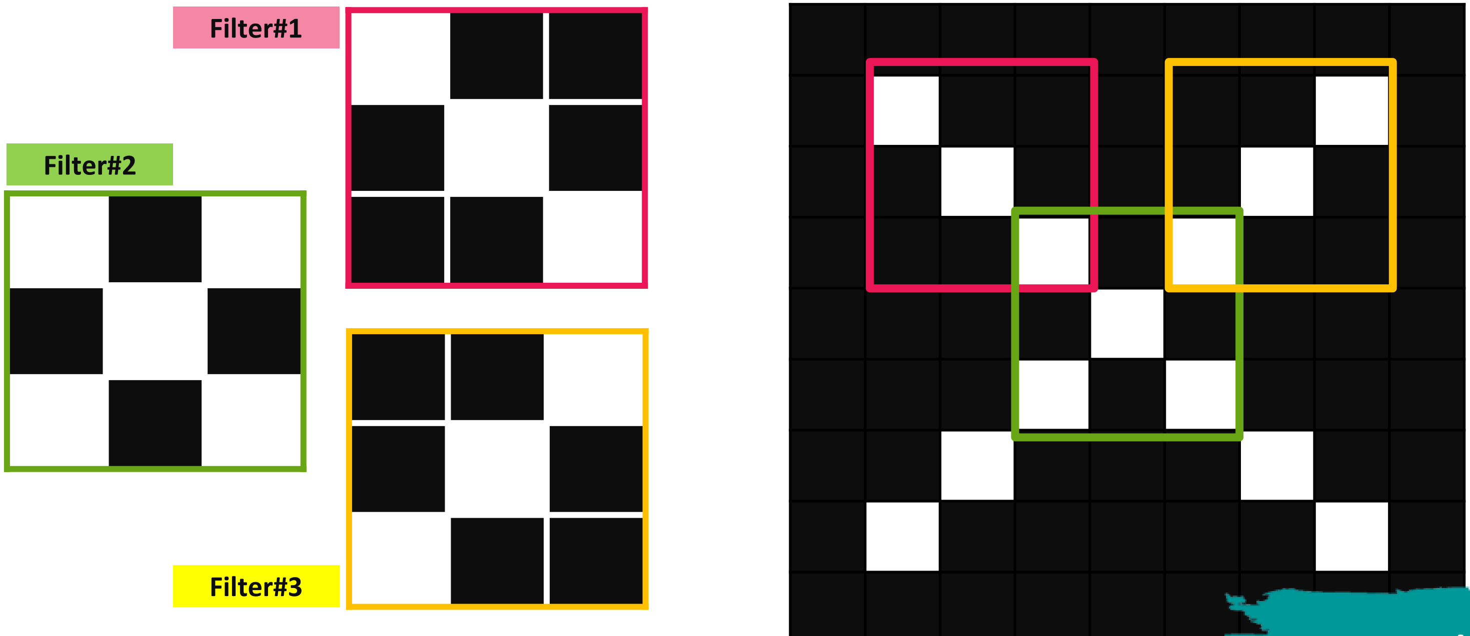
The scaled version

The same features!

Convolution

Why is it important to use Convolution?

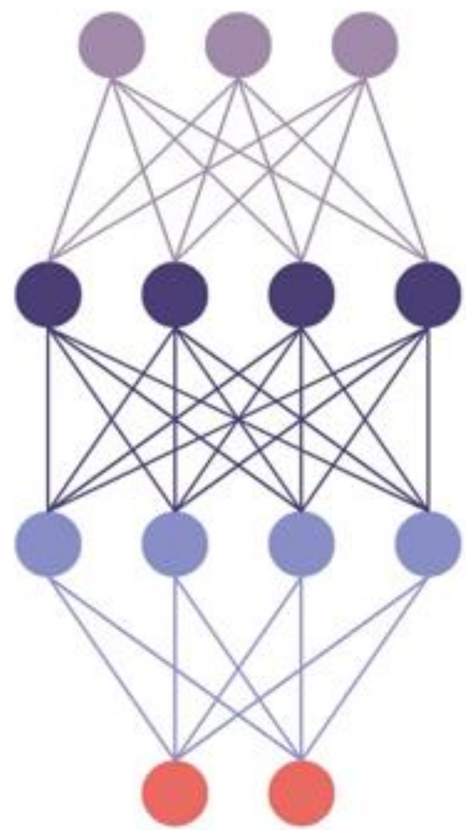
- A simple case study: classifying the letter “X” even if it is rotated/shifted



Sample filters

Convolutional Neural Networks

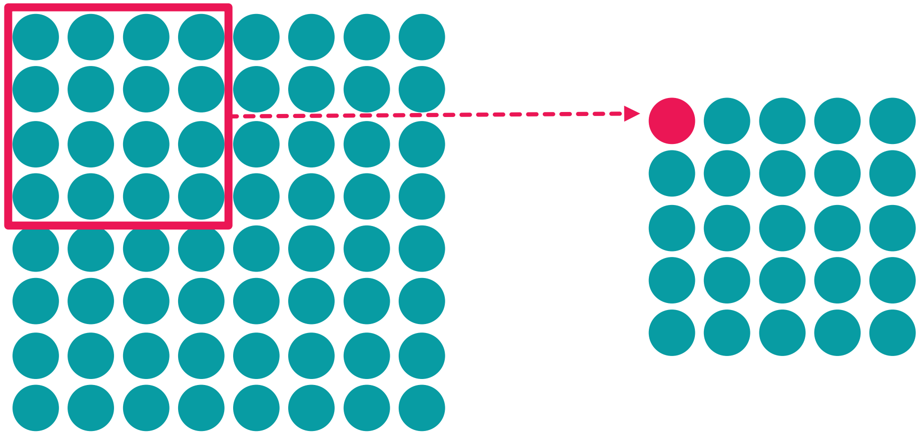
- Fully-connected Neural Networks (Dense NNs)
  - Recall: Session#1 - Basics
- Can we use them for processing mage inputs?
  - Regular NNs do not scale well to full images
  - We need many calculations and parameters!
  - We will lose the spatial information
    - **Reason:** flattening the 2D image into 1D vector
    - **Example:** the nearness of pixels to others



Convolutional Neural Networks

How to preserve the spatial structure?

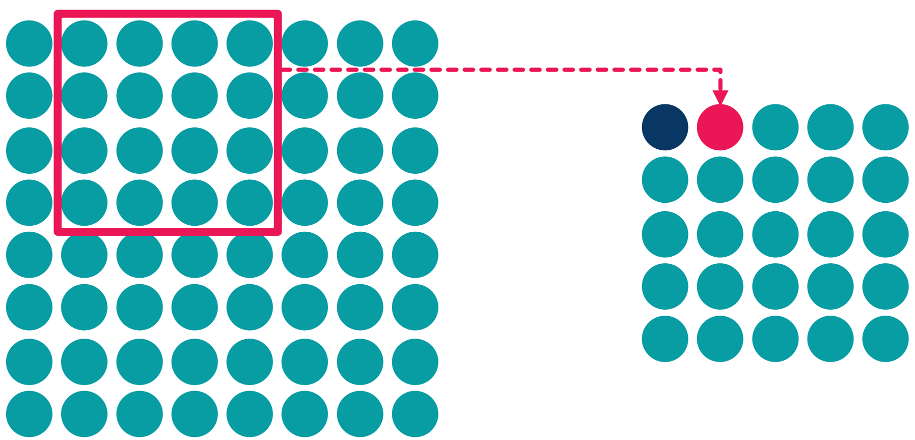
- Connecting slices of the input, instead of connecting all input values to neurons in the hidden layer
  - Each patch (slice) to a single neuron in the next layer
- This approach preserves the spatial information + visual features



Convolutional Neural Networks

How to preserve the spatial structure?

- Connecting slices of the input, instead of connecting all input values to neurons in the hidden layer
  - Each patch (slice) to a single neuron in the next layer
- This approach preserves the spatial information + visual features

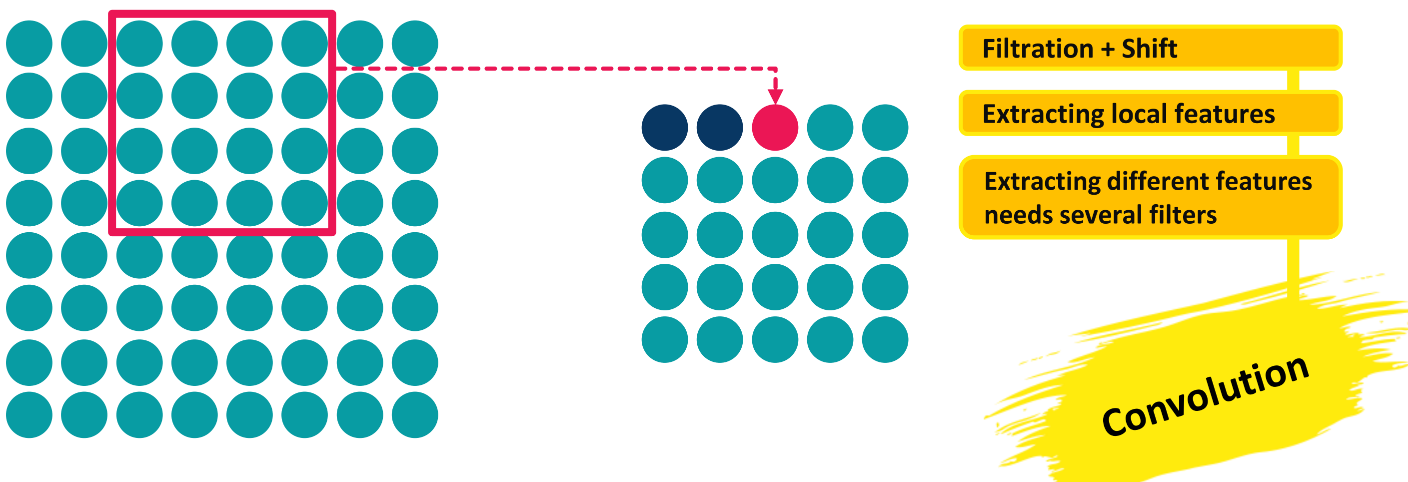


Simply, a sliding window can do the rest!

Convolutional Neural Networks

How to preserve the spatial structure?

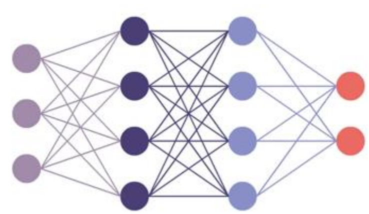
- Connecting slices of the input, instead of connecting all input values to neurons in the hidden layer
  - Each patch (slice) to a single neuron in the next layer
- This approach preserves the spatial information + visual features



Convolutional Neural Networks

- CNNs (*ConvNets*) utilize the mathematical foundation of convolution for computer vision tasks
- Inspired by the organization of the Visual Cortex
- Much lower pre-processing comparing to other classification algorithms
  - CNNs can automatically learn the filters and characteristics
- The overall process in a CNN:

Input image



Learnable weights and biases

Differentiation of objects in the image

Convolutional Neural Networks

- In contrast with regular NNs, the layers of a CNN have neurons arranged in three dimensions: **width, height, depth**
  - **Note:** “depth” is the third dimension of an activation volume
  - **Note:** each layer converts a 3D input to a 3D output of neurons’ AFs

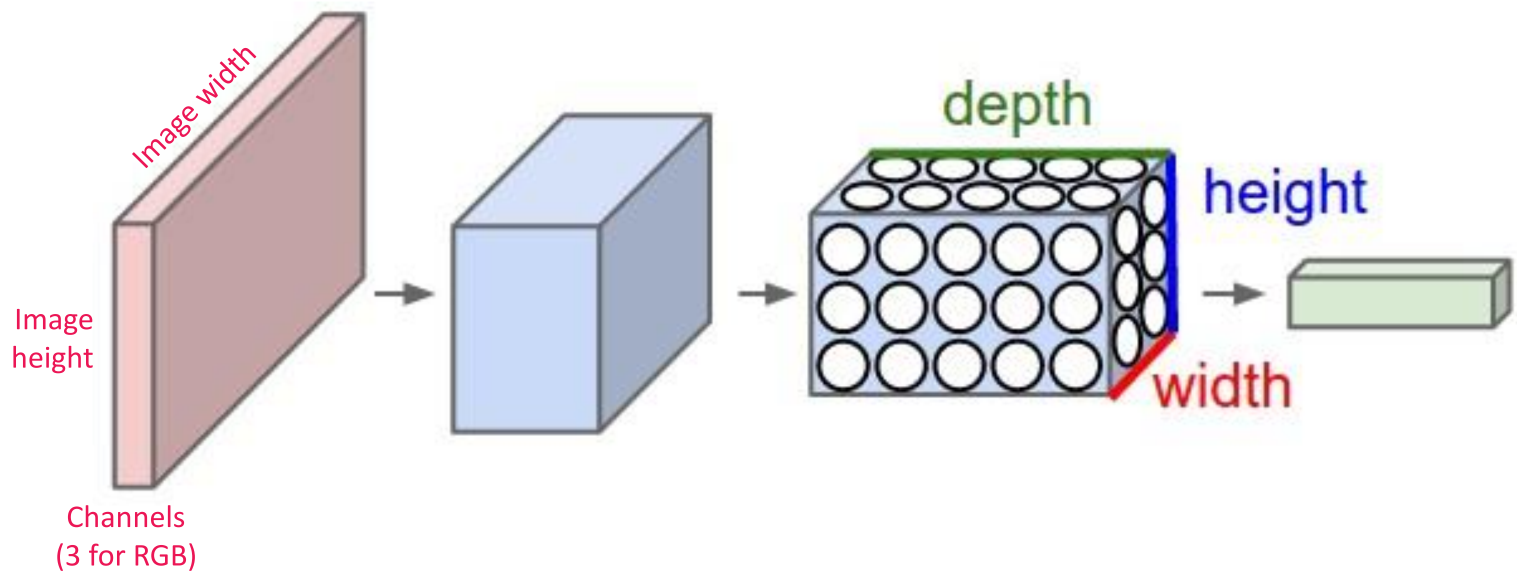


Image source: https://cs231n.github.io/

Convolutional Neural Networks

Roles of CNNs

- Assuming that the inputs are images → more efficient network
- Reducing images into a form that is easier for processing
- Preserving the critical features of images for prediction
- Reducing the complexity of the image classification task

Popular Architectures of CNNs (Click to see the papers)

AlexNet

GoogLeNet

VGGNet

LeNet

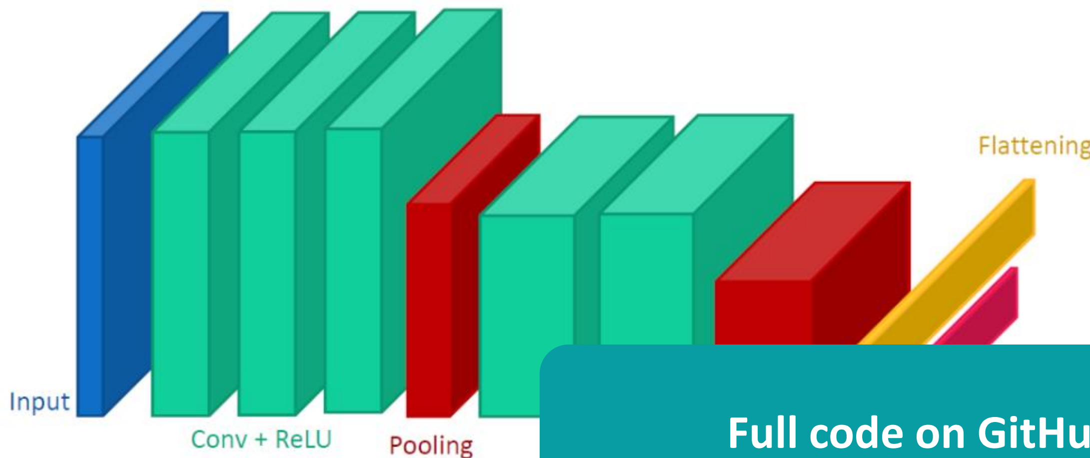
ResNet

Convolutional Neural Networks

21 lines (16 sloc) | 1.1 KB   Raw   Blame

Convolutional Neural Networks (CNNs)

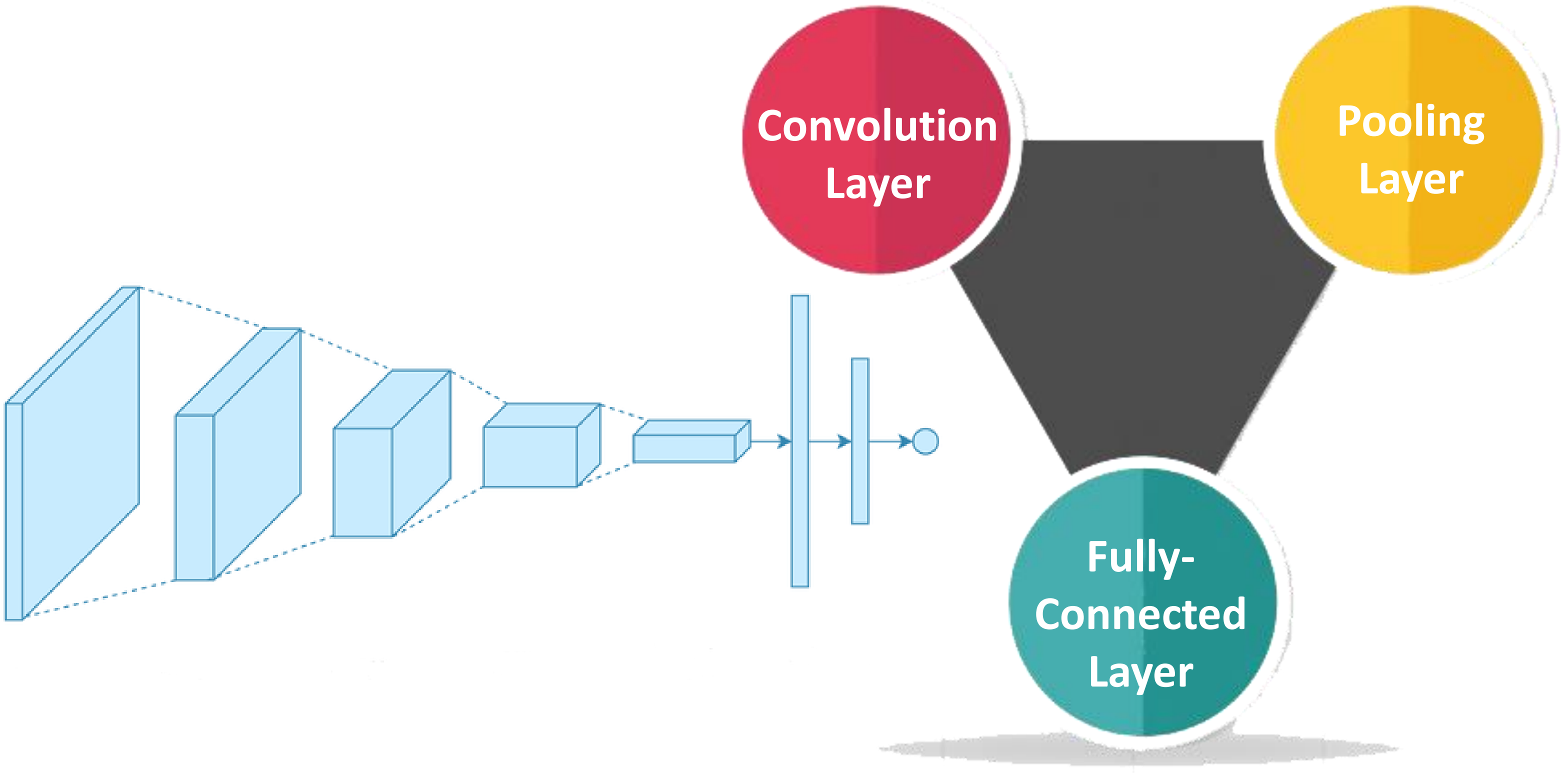
CNNs (ConvNets) utilize the mathematical foundation of convolution for computer vision tasks. Inspired by the organization of the Visual Cortex, CNNs can provide much lower pre-processing comparing to other classification algorithms. These networks can automatically learn the filters and characteristics of the input images. In contrast with regular NNs, the layers of a CNN have neurons arranged in three dimensions: width, height, depth.



Codes

Full code on GitHub

CNNs Layers

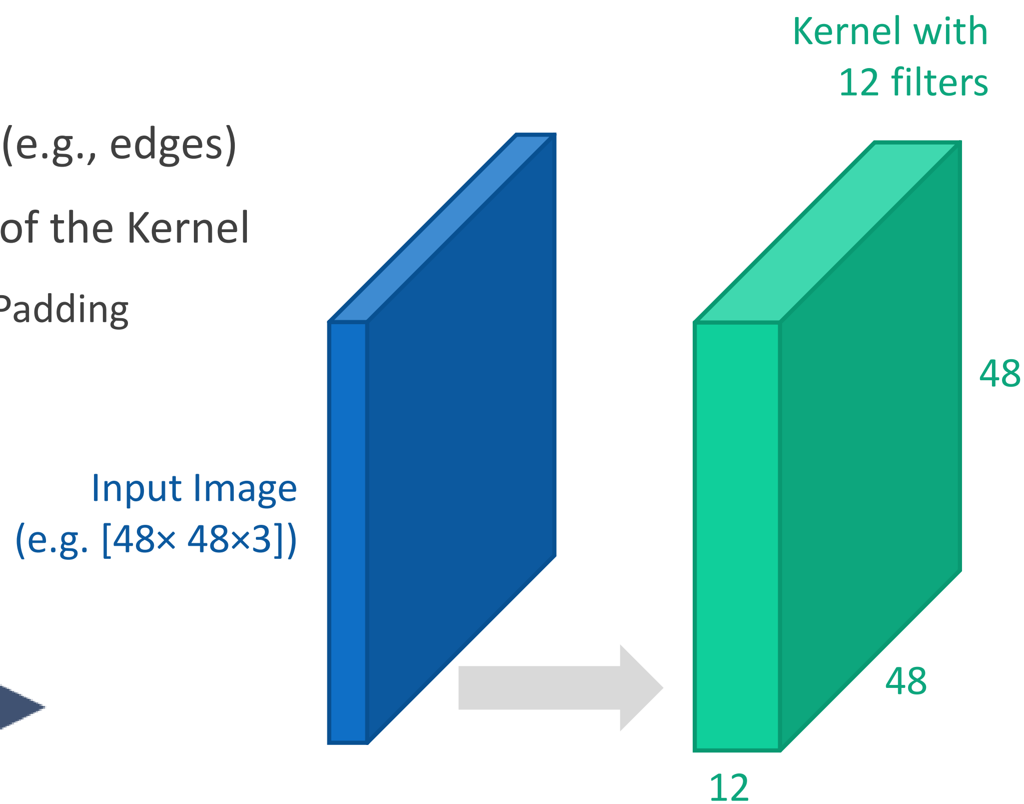


CNNs Layers

Convolution Layer (Kernel)

- **Output:** Feature maps
- Extracting high-level features (e.g., edges)
- Specifying the characteristics of the Kernel
  - Including Kernel size, Stride, Padding

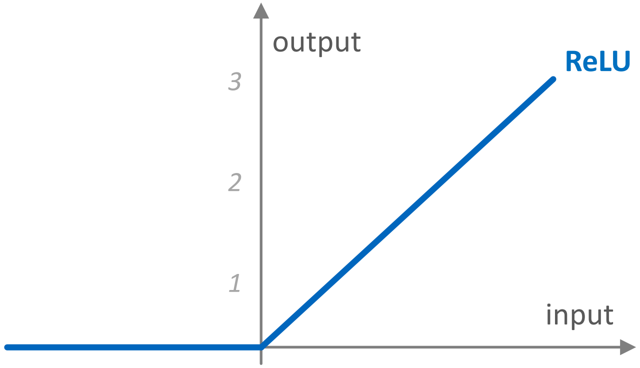
Applying the kernel, ensuring it is traversed the entire image



CNNs Layers

Convolution Layer (Kernel)

- Then, adding a supplementary step to the convolution operation
  - **Activation Function**: often **ReLU** (replacing negative values by zero)
  - **Note**: The Rectified Linear Unit is not a separate component of the convolutional neural networks' process



CNNs Layers

Pooling Layer (Down sampling)

- **Output:** Pooled feature maps
- With a Convolution layer, forms a layer in a CNN
- Reducing the spatial size of the Convolved Feature
- Decreasing the dimensionality and computation
- Two-types of Pooling:
  - **Max-Pooling:** returns the maximum value
    - Recommended due to de-noising advantage
  - **Average-Pooling:** returns the average of all the values

89	112	36	97
19	234	75	38
196	64	39	2
203	88	102	72

234	97
203	102

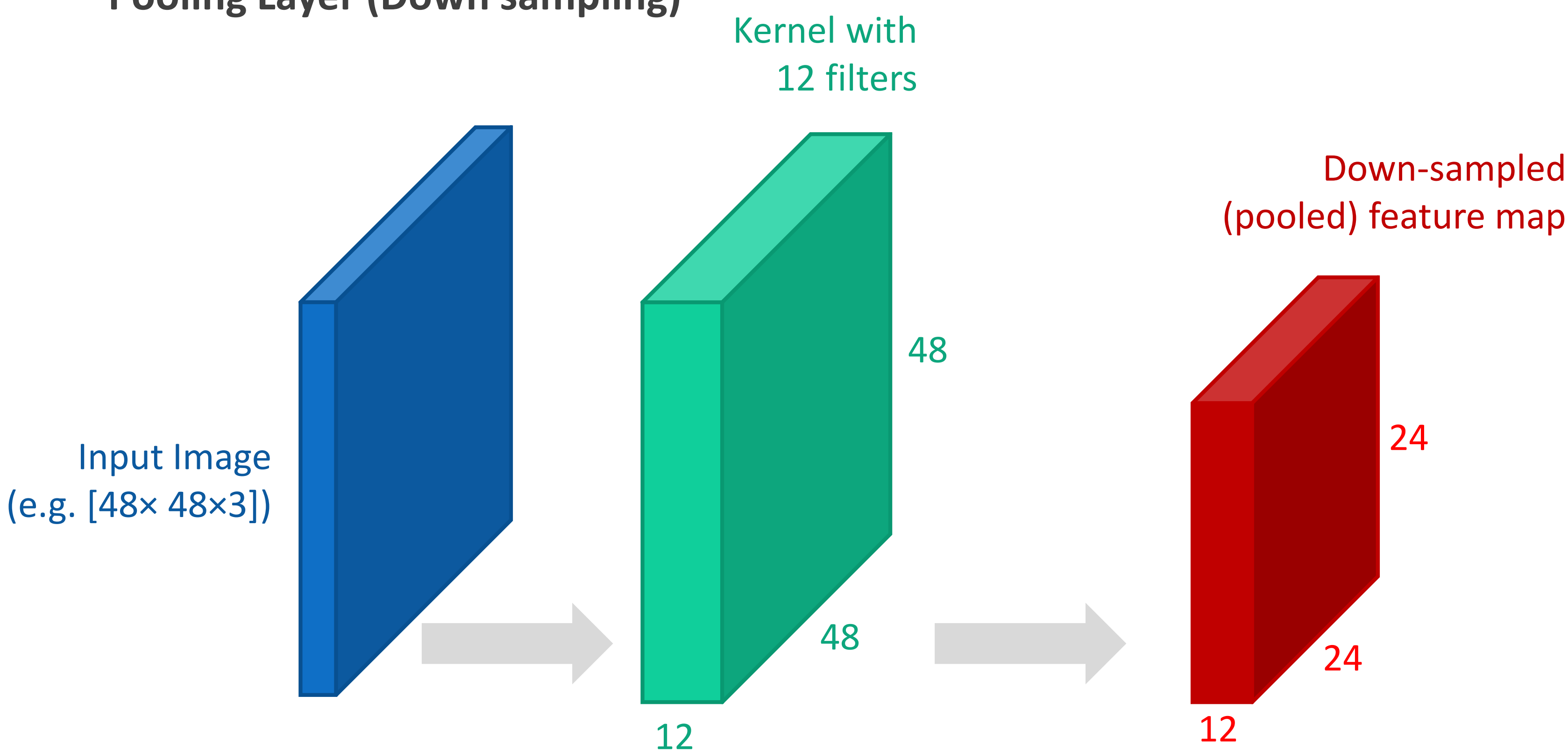
Max-Pooling

113	61
137	54

Avg-Pooling

CNNs Layers

Pooling Layer (Down sampling)



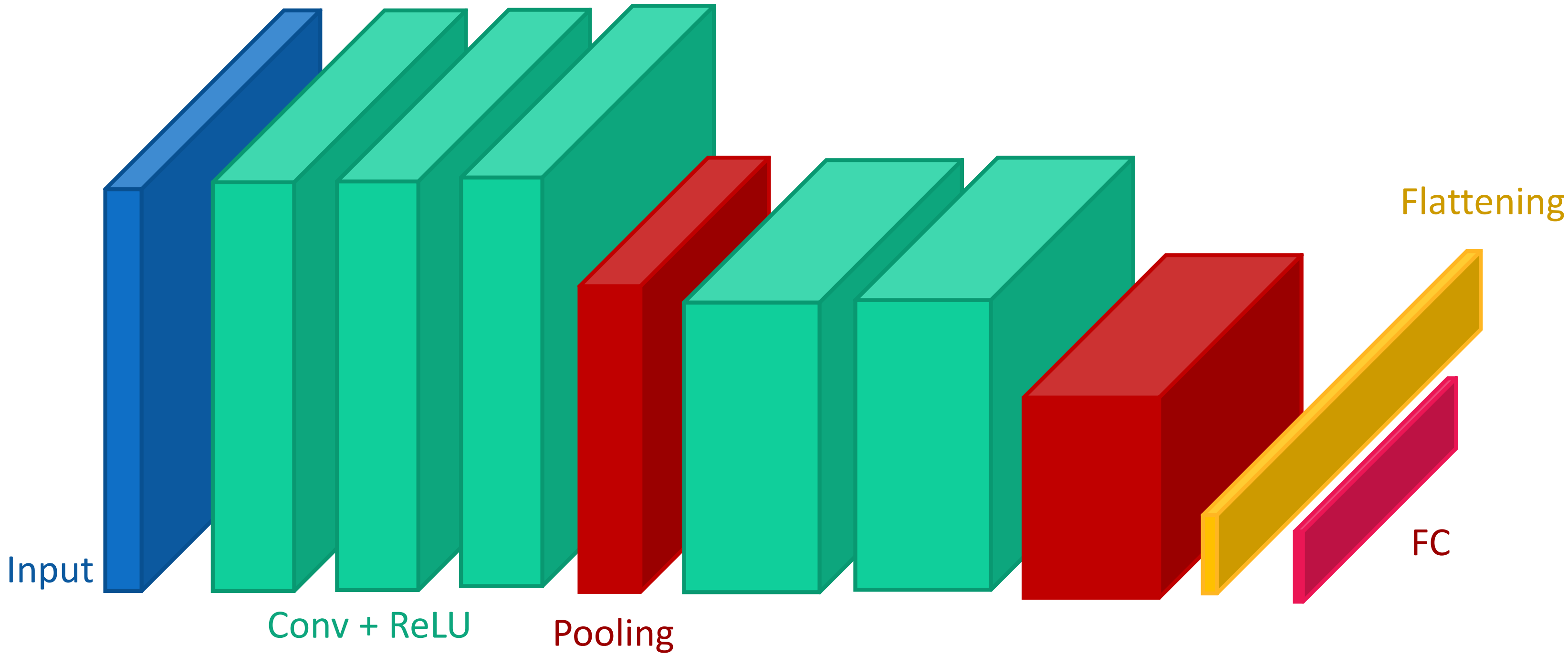
CNNs Layers

Fully Connected (FC) Layer

- **Output:** Mapped class scores
- Optimizing and computing class scores
  - **Note:** the input of this layer should be flattened
  - **Note:** the output can be normalized using SoftMax algorithm
- Each neuron is connected to all nodes in the previous step
- Generally found at the final layers of CNNs
  - At some points, they can be used along with ReLU layers
  - The last FC layer is the output layer

CNNs Layers

Let's put them all together!



CNNs Layers

$$Inp \rightarrow \left((Conv \rightarrow ReLU) * n \rightarrow Pool? \right) * m \rightarrow (FC \rightarrow ReLU) * k \rightarrow FC$$

The magic pattern!

CNNs Layers

A CNN for image classification

Depth shows the number of filters

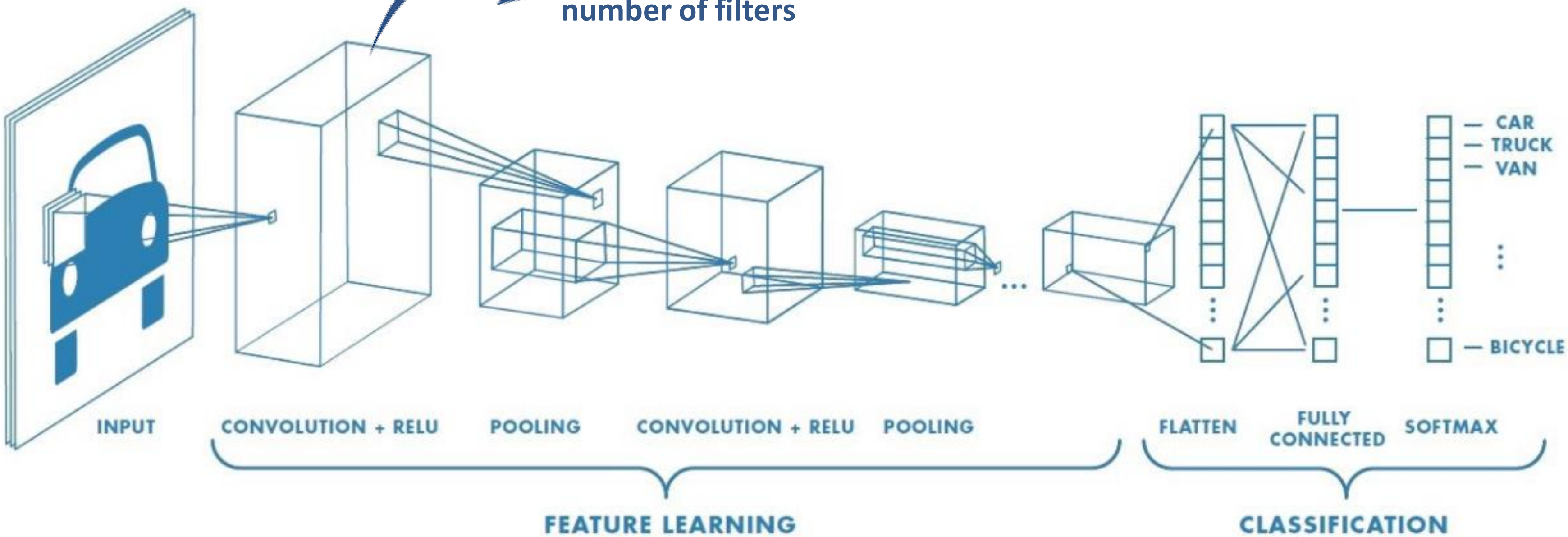


Image source: https://www.towardsdatascience.com

CNNs Layers

A CNN for image classification

Flattening the image into
a column vector

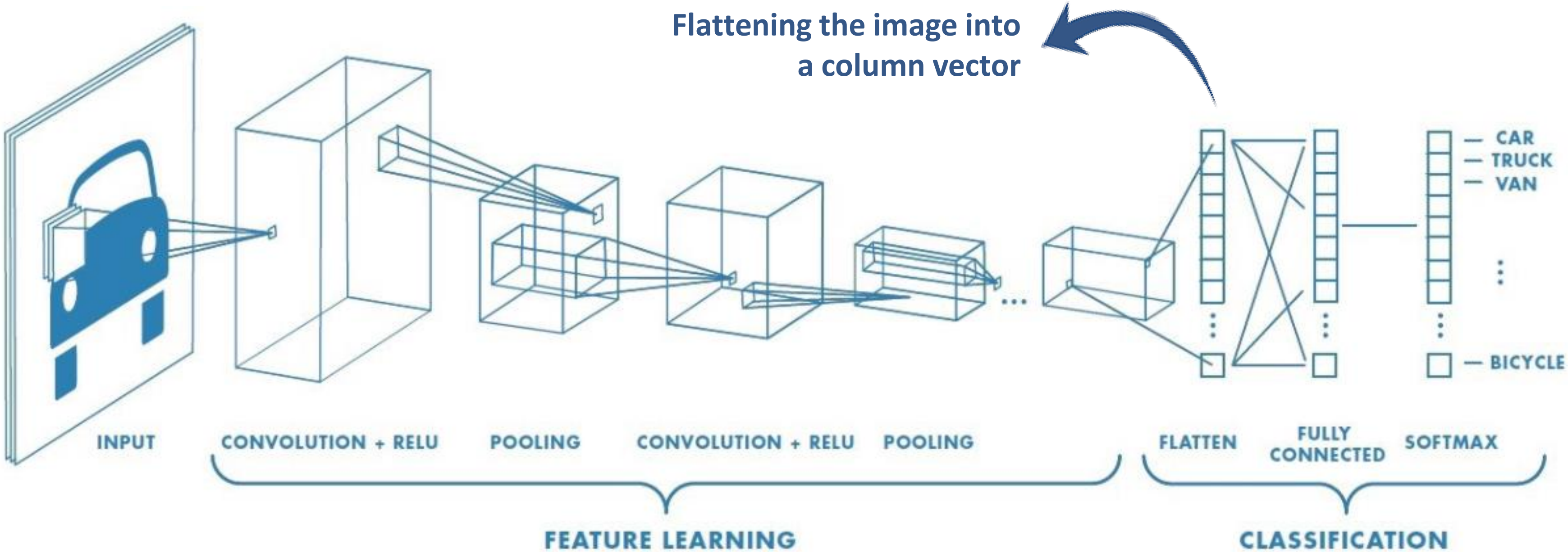


Image source: https://www.towardsdatascience.com

CNNs Layers

A CNN for image classification

Squashes class scores and calculates the probabilities for each class

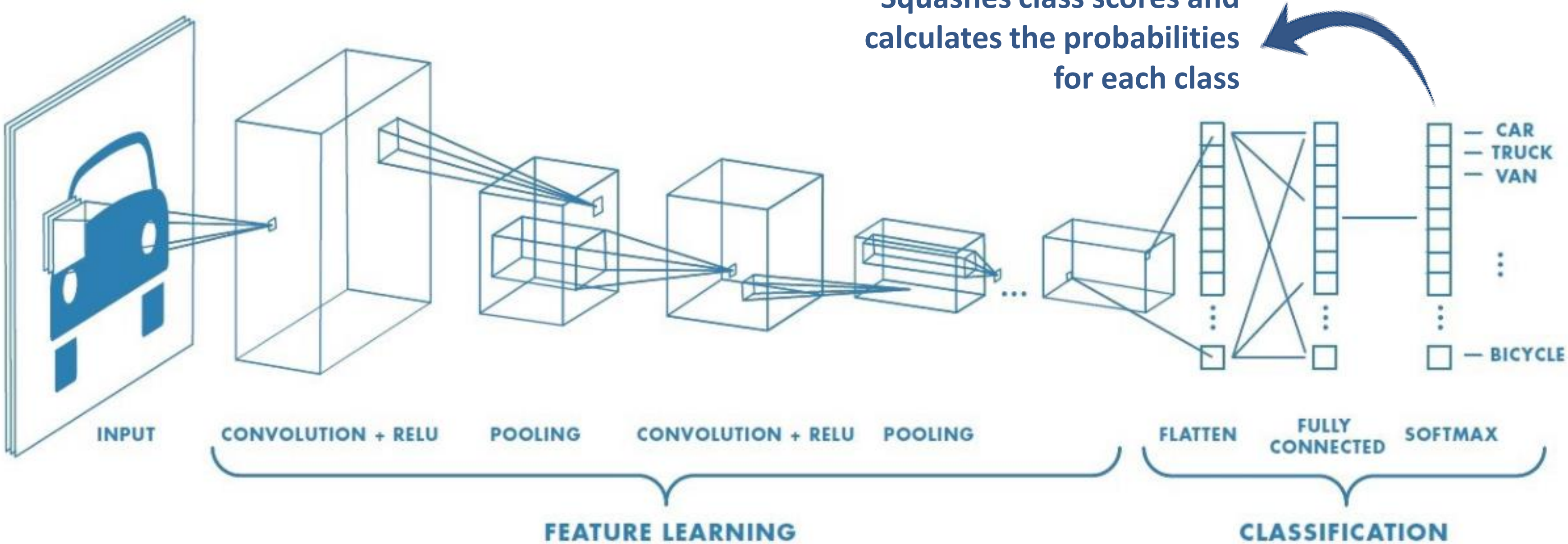
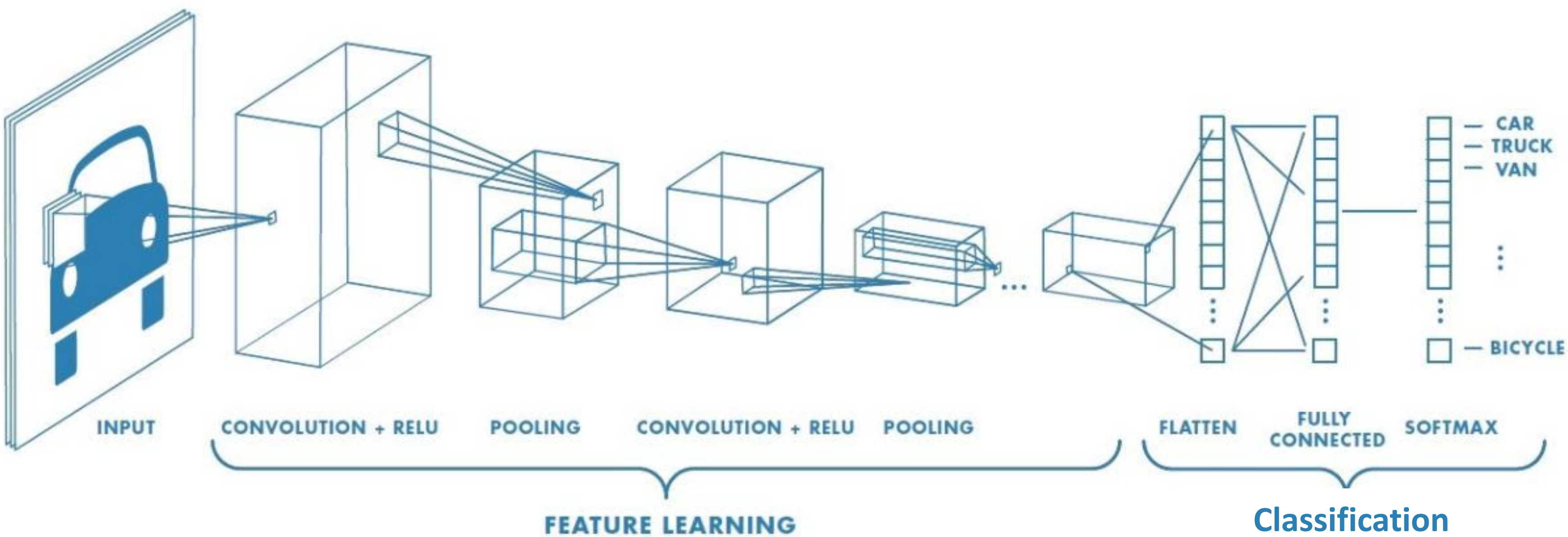


Image source: https://www.towardsdatascience.com

CNNs Layers

A CNN for image classification



Classification
Object Detection
Segmentation

Image source: https://www.towardsdatascience.com

CNNs Layers

A CNN for image classification – Vehicle Classification

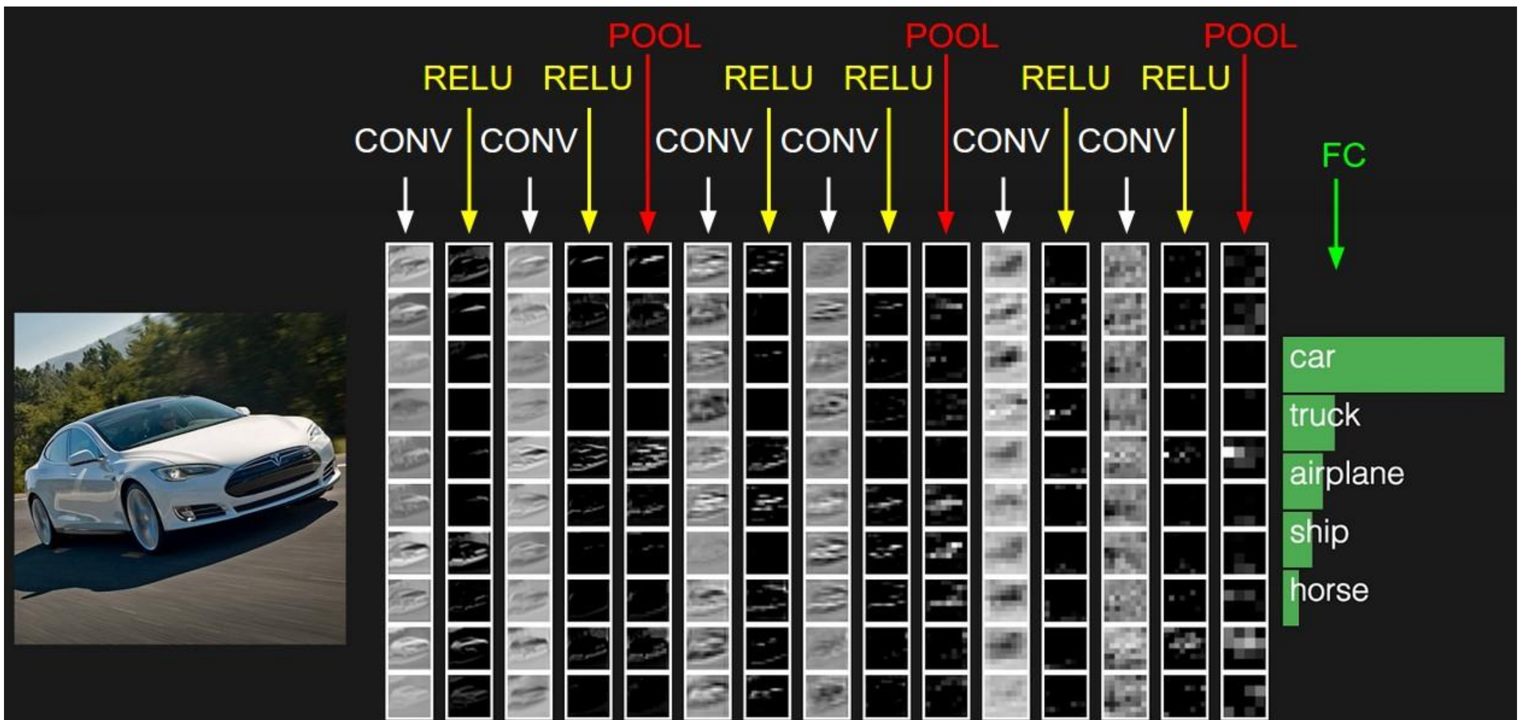


Image source: https://cs231n.github.io/

CNNs Layers

A CNN for image classification – Handwritten digits classification

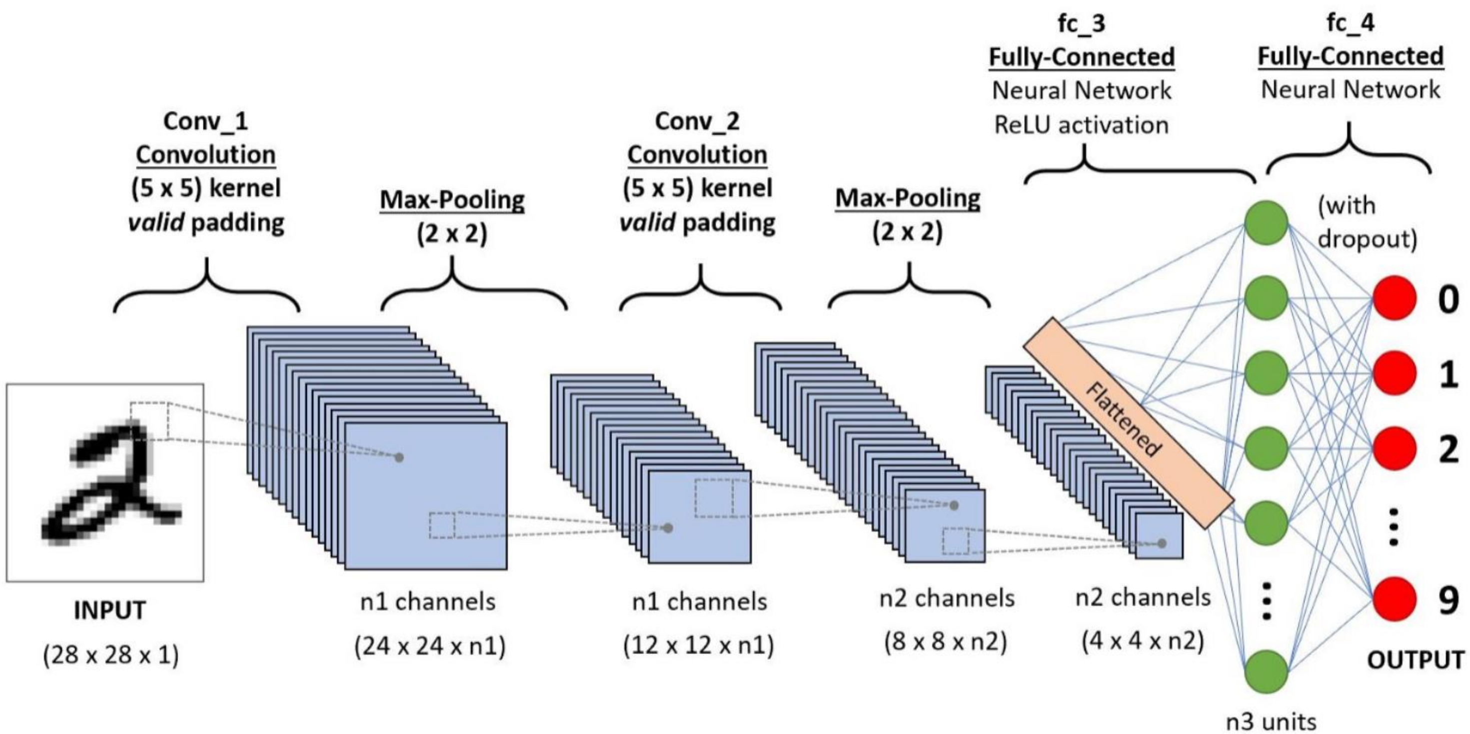


Image source: https://www.towardsdatascience.com

CNNs Layers

A CNN for image classification – Image classification

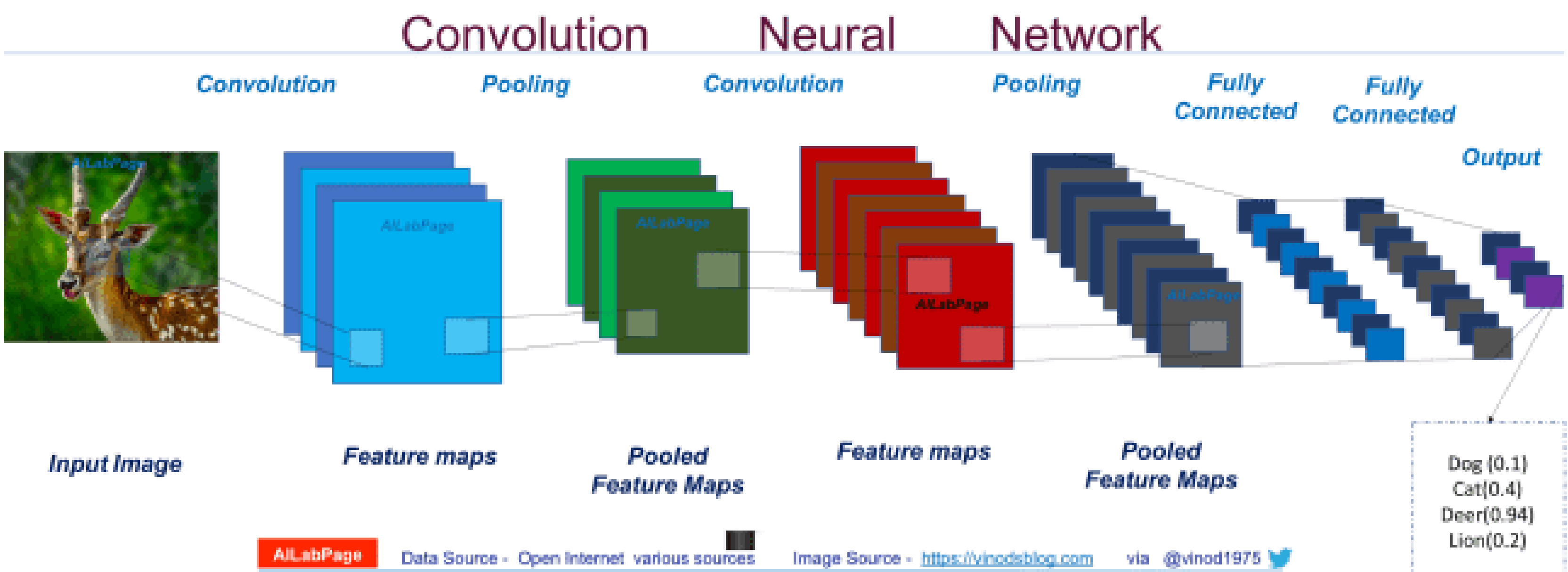


Image source: https://www.vinodsblog.com

References

- http://introtodeeplearning.com/
- https://towardsdatascience.com/image-feature-extraction-traditional-and-deep-learning-techniques-ccc059195d04
- https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/digital-images
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://machinelearningmastery.com/classification-versus-regression-in-machine-learning
- https://mathworld.wolfram.com/Convolution.html
- https://aishack.in/tutorials/image-convolution-examples/
- https://cs231n.github.io/convolutional-networks/

Questions?