University of Windsor

GENG-8900 MACHINE LEARNING

Instructor: Dr. Yasser Alginahi

Convolutional Neural Networks (CNNs)

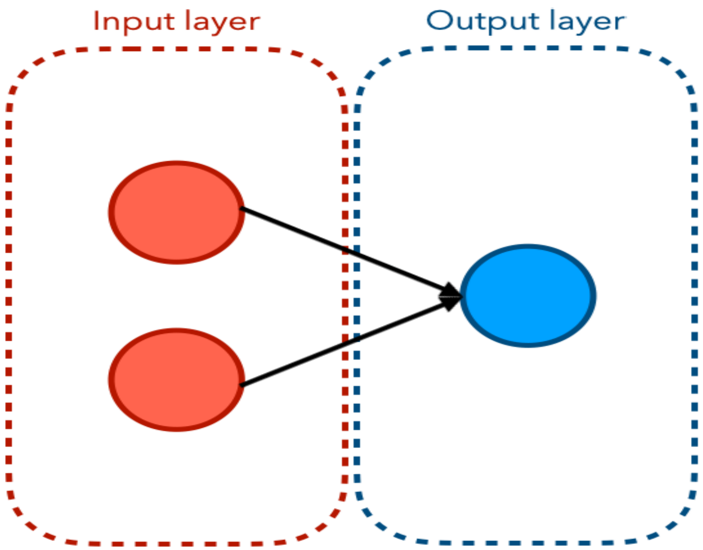
Ifrah Andleeb

Vaibhav Anilkumar Patel

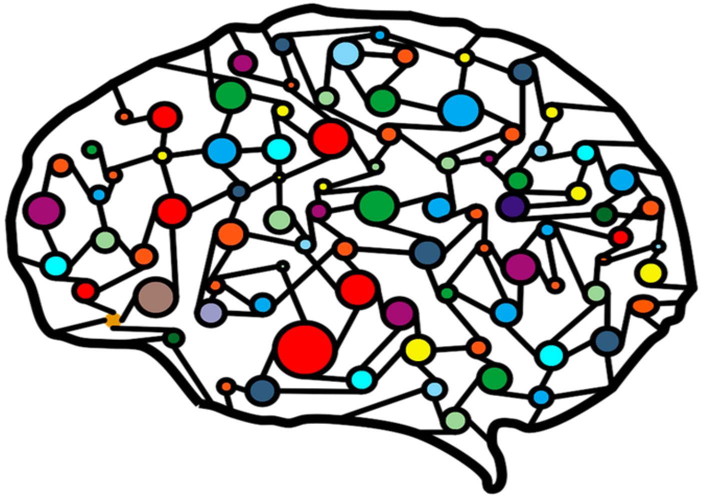
Date: 27-Oct-2023

What are Neural Networks ?

- A neural network is like a computer program inspired by the human brain.
- So, in a neural network, neurons receive, think about, and then share information, and by doing this collectively, they can solve all sorts of problems.

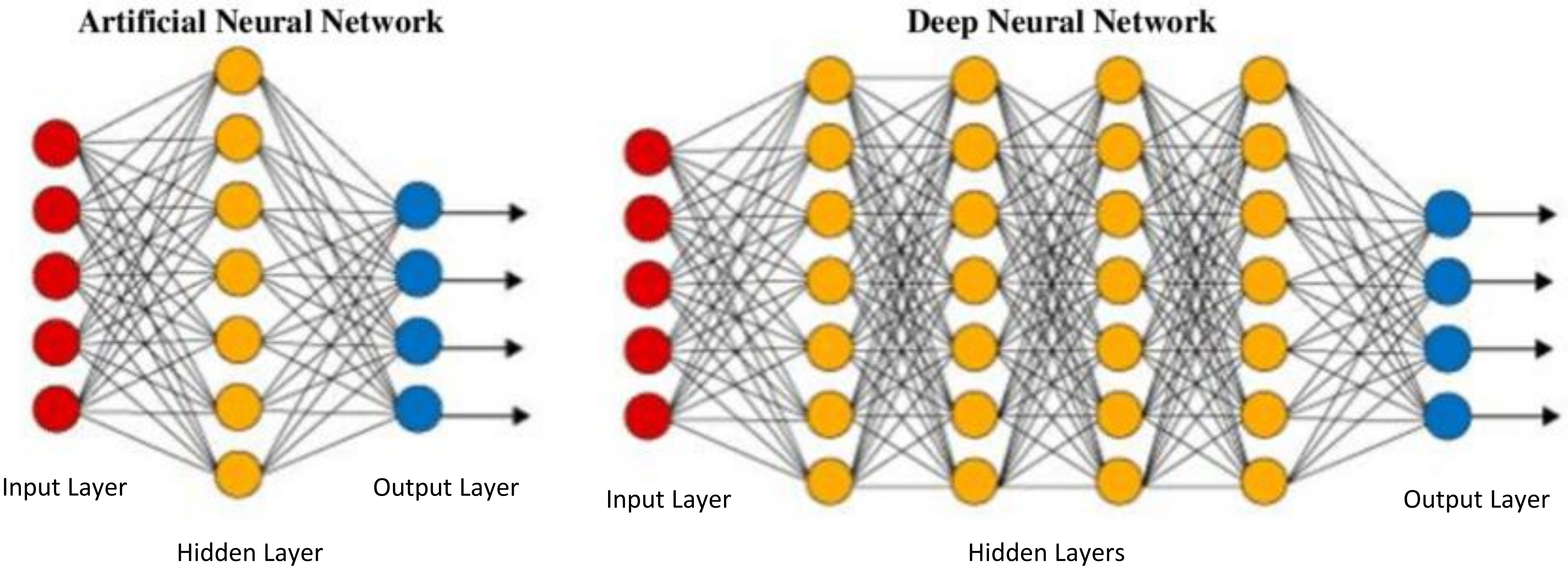


Neural Network Animated [https://tinyurl.com/28wmb4ez]



Brain Network [https://tinyurl.com/4fcta8j7]

Artificial Neural Networks vs Deep Neural Networks



Input Layer

Output Layer

Hidden Layer

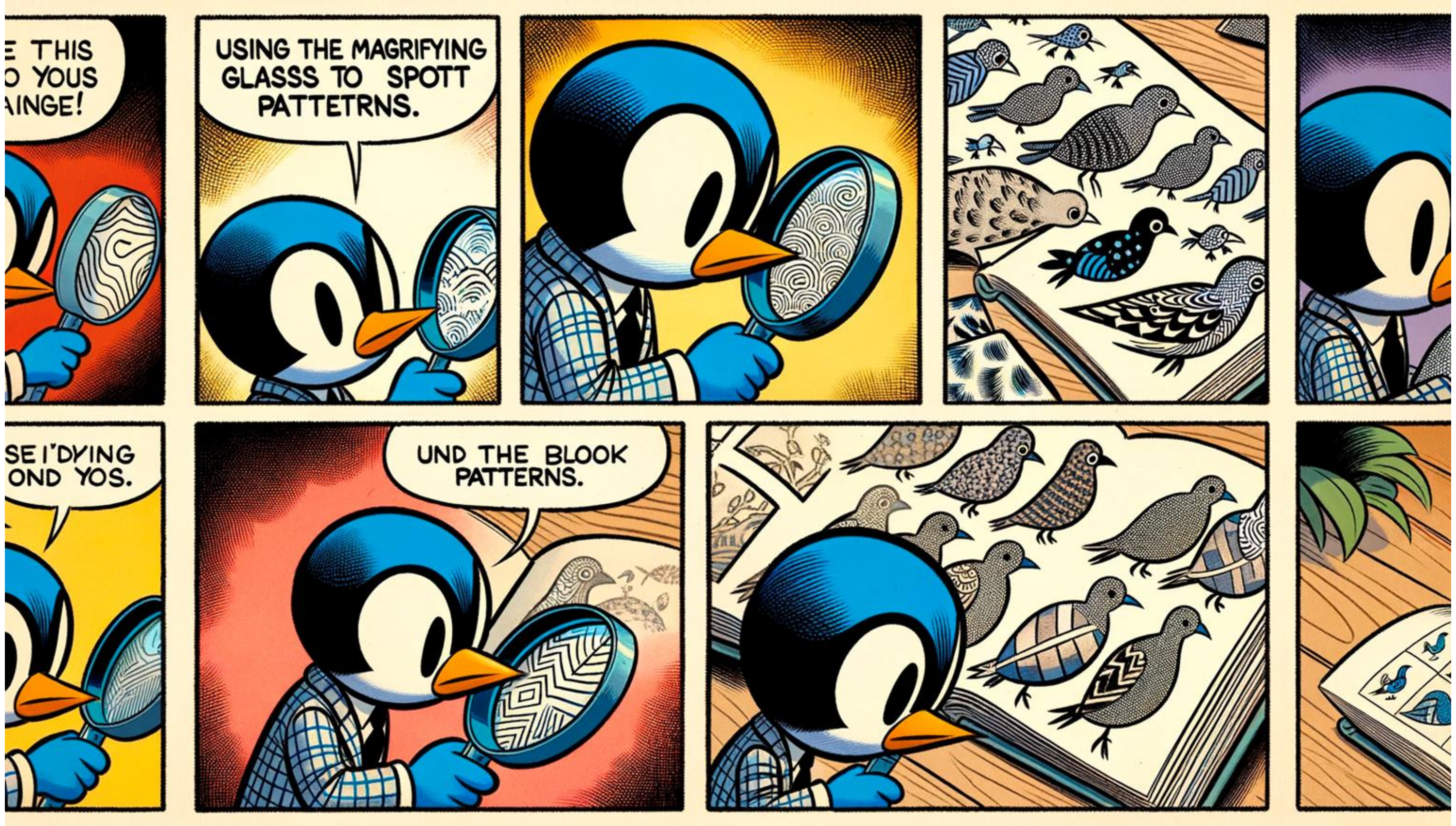
Input Layer

Output Layer

Hidden Layers

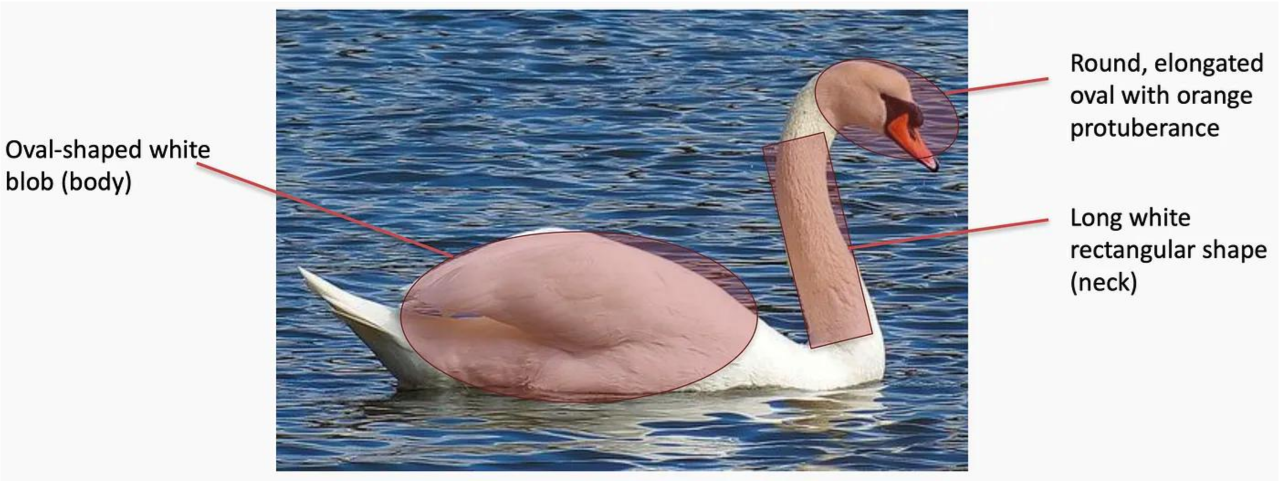
ANN vs DNN Architecture [https://www.datacamp.com/tutorial/introduction-to-deep-neural-networks]

Let’s learn more !



Recognizing birds with magnifying glass [https://openai.com/dall-e-3]

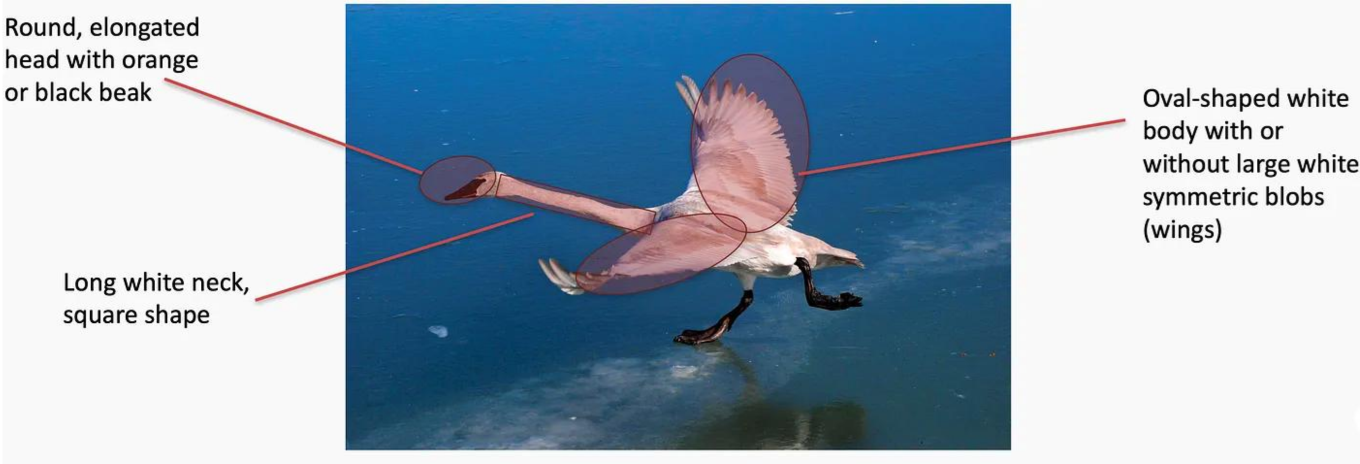
Image Analysis : How do you identify a Swan ?



Oval-shaped white blob (body)

Round, elongated oval with orange protuberance

Long white rectangular shape (neck)



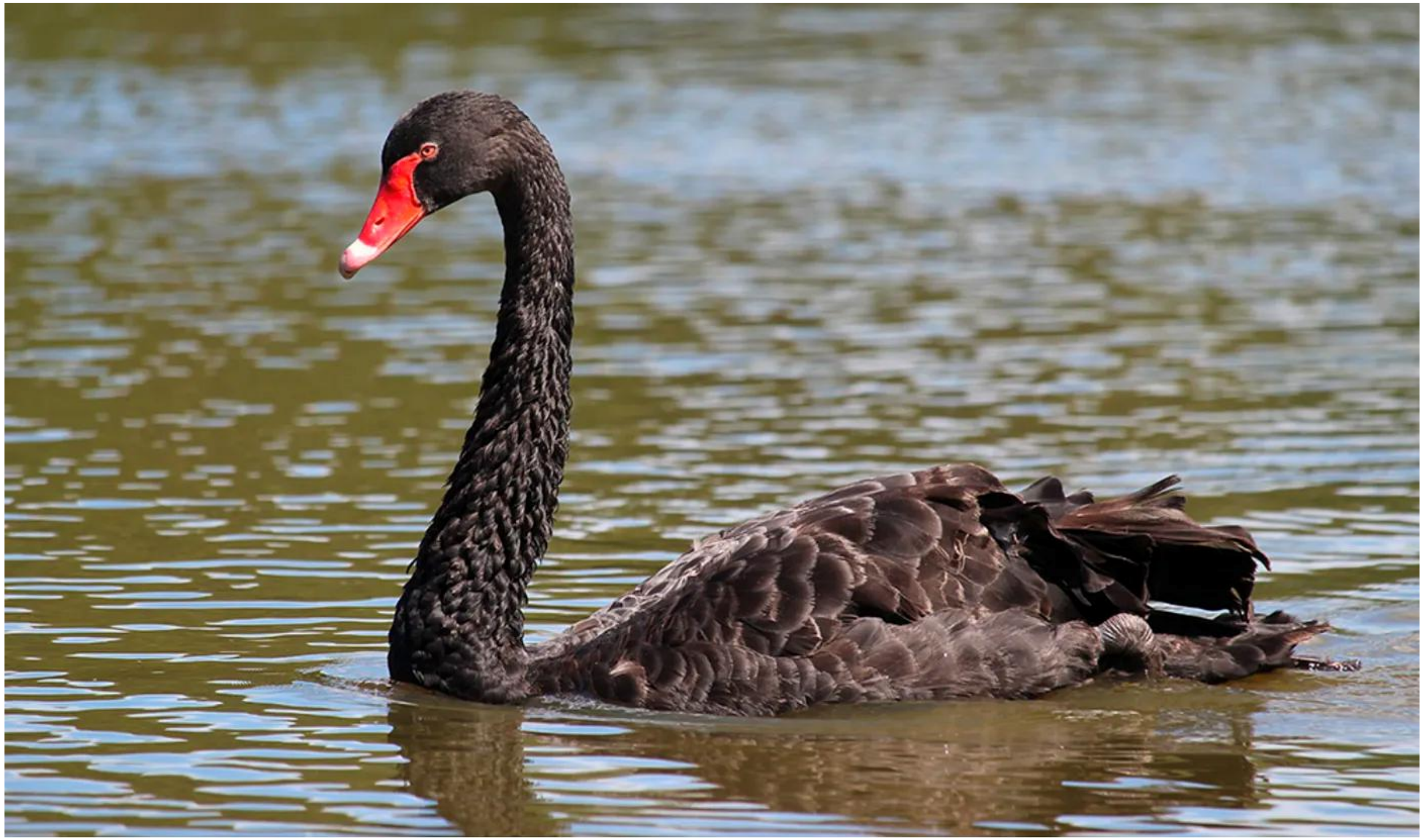
Round, elongated head with orange or black beak

Long white neck, square shape

Oval-shaped white body with or without large white symmetric blobs (wings)

Features identification in a bird [https://tinyurl.com/5n6r3xee]

Image Analysis : How do you identify a Swan ?



Don’t forget about those black swans! [https://tinyurl.com/5n6r3xee]

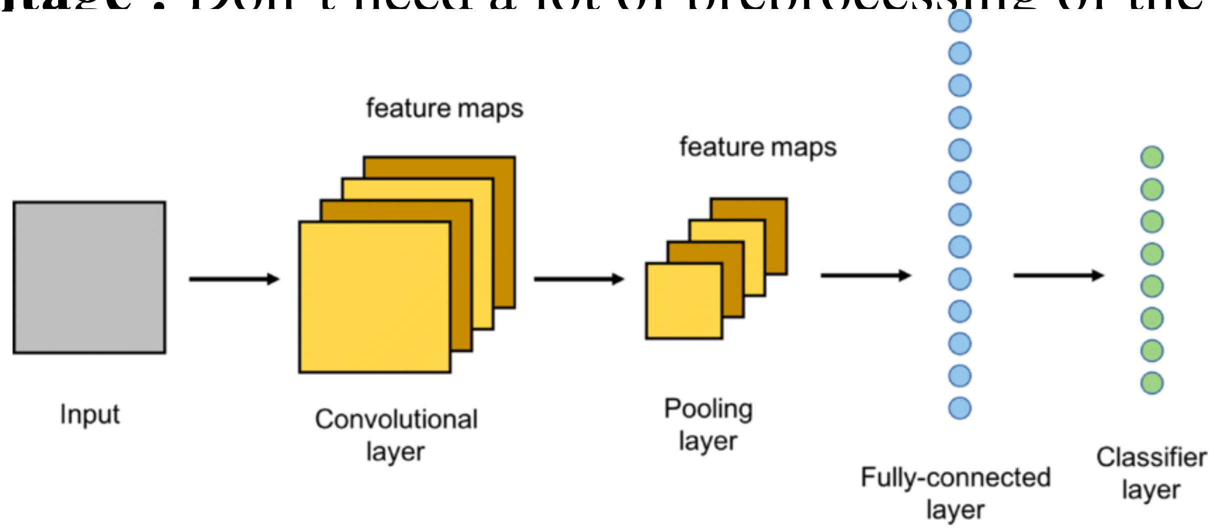
Image Analysis : How do you identify a Swan ?

Man in swan tent photographing swans

Worst-case scenarios [https://tinyurl.com/5n6r3xee]

What is CNN? and Why CNN?

- Deep Neural Network with multiple Layers, processes data with grids i.e. Images
- With other image processing algorithms, filters are typically created based on heuristics while CNN's can learn what characteristics in the filters are the most important.
- Saves a lot of time on trial and error work since we don't need as many parameters.
- **Huge advantage :** Don’t need a lot of preprocessing of the images.



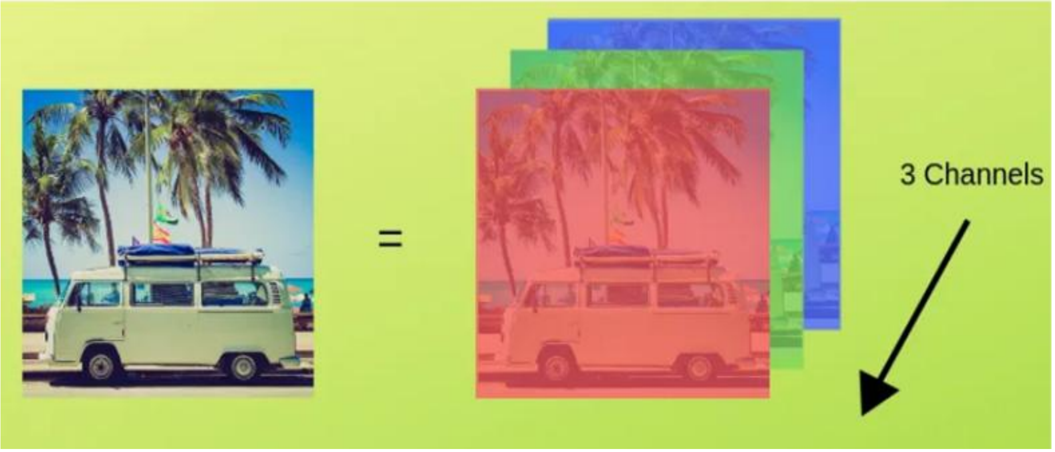
A simple CNN and its main layers [https://tinyurl.com/szpwhrpk}

Image Representation

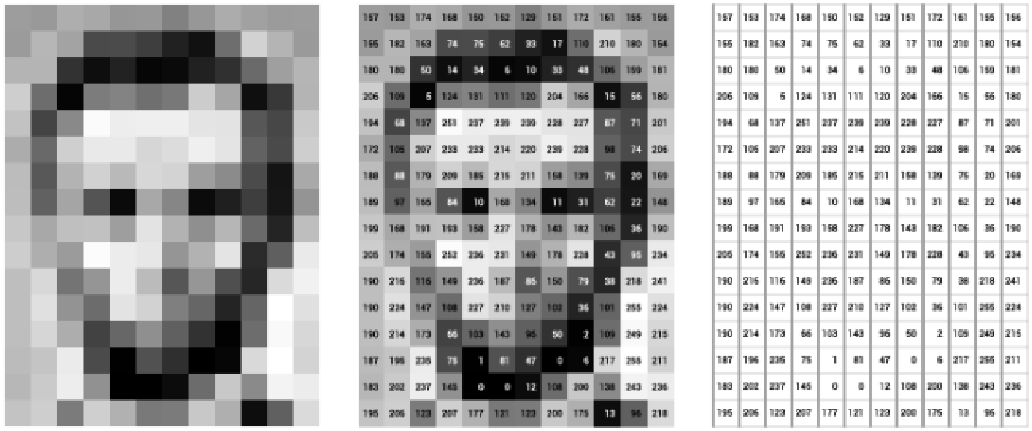
- **Plane/Channels**: Separate layers in images capturing specific information.

- An RGB (Red, Blue and Green) image is a matrix of pixel values having 3 planes.

- Whereas a grayscale image is the same but it has a single plane.



RGB Representation of an image - [https://tinyurl.com/yrw7zw8t]

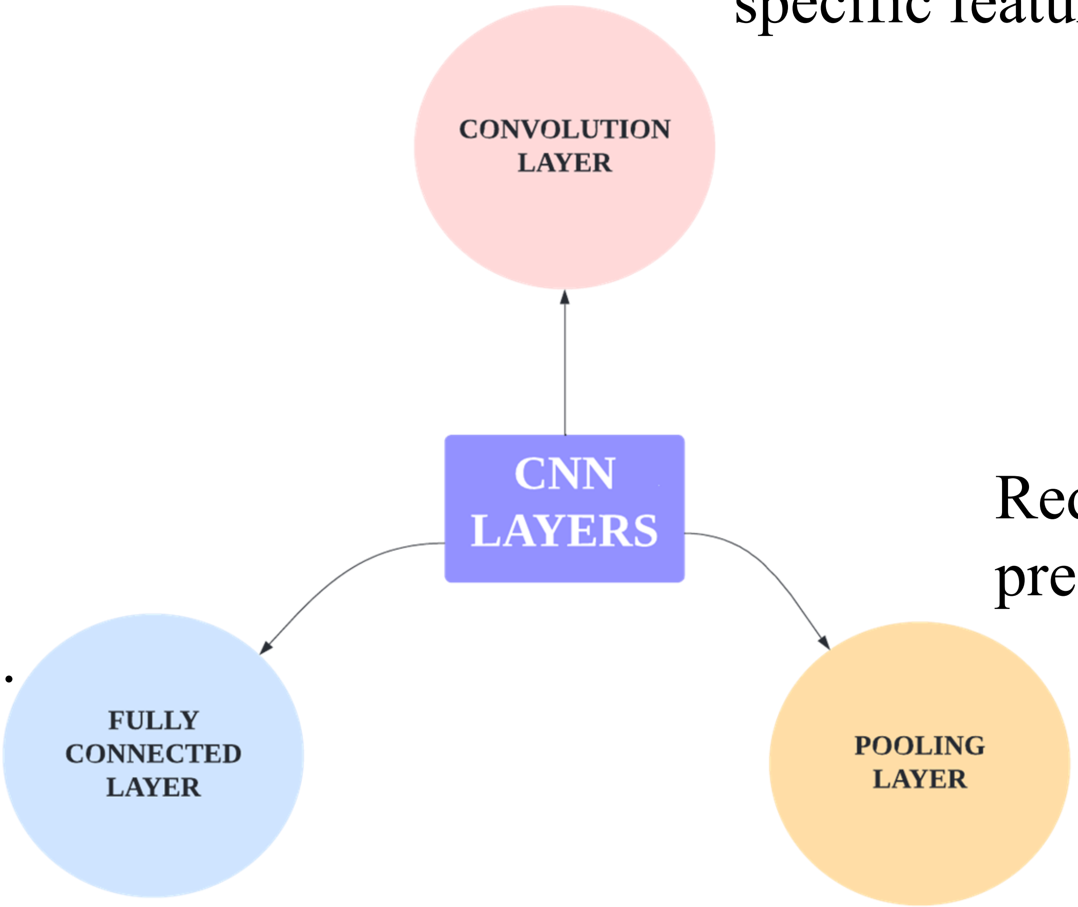


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Gray scale Representation of an image [https://www.v7labs.com/blog/image-recognition-guide]

Different Layers in CNN

Filters input data to detect specific features.

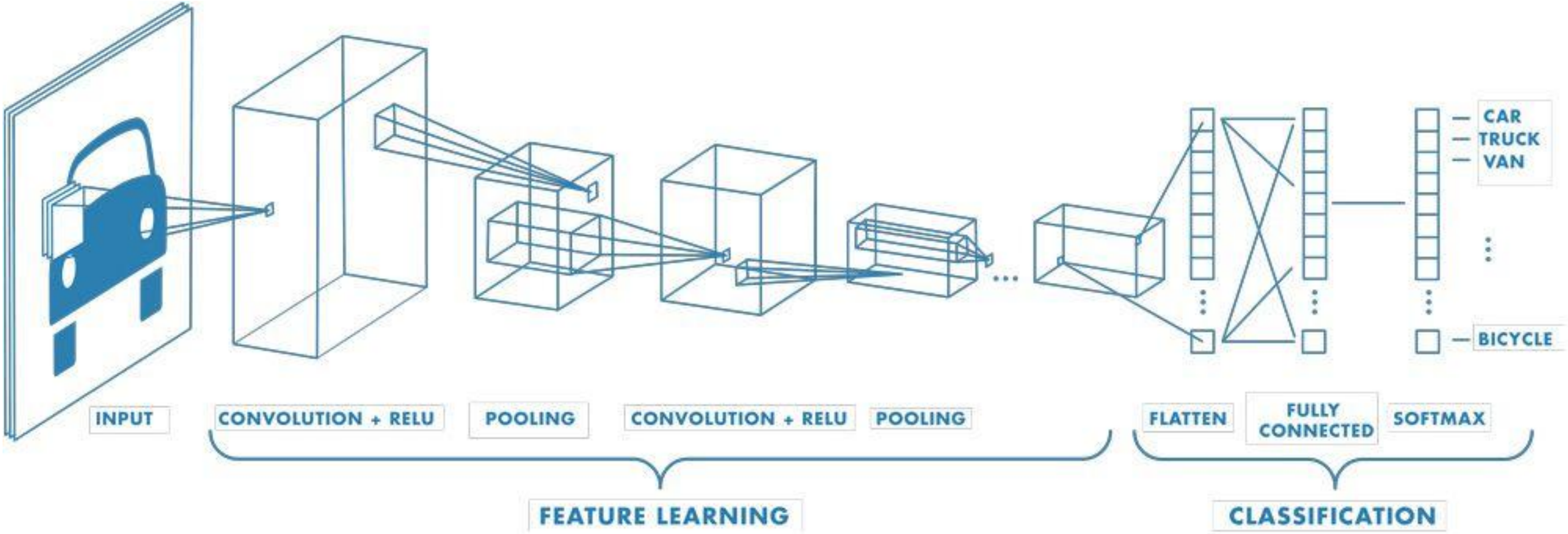


Reduces data size while preserving important features.

Links every neuron to previous layers' outputs.

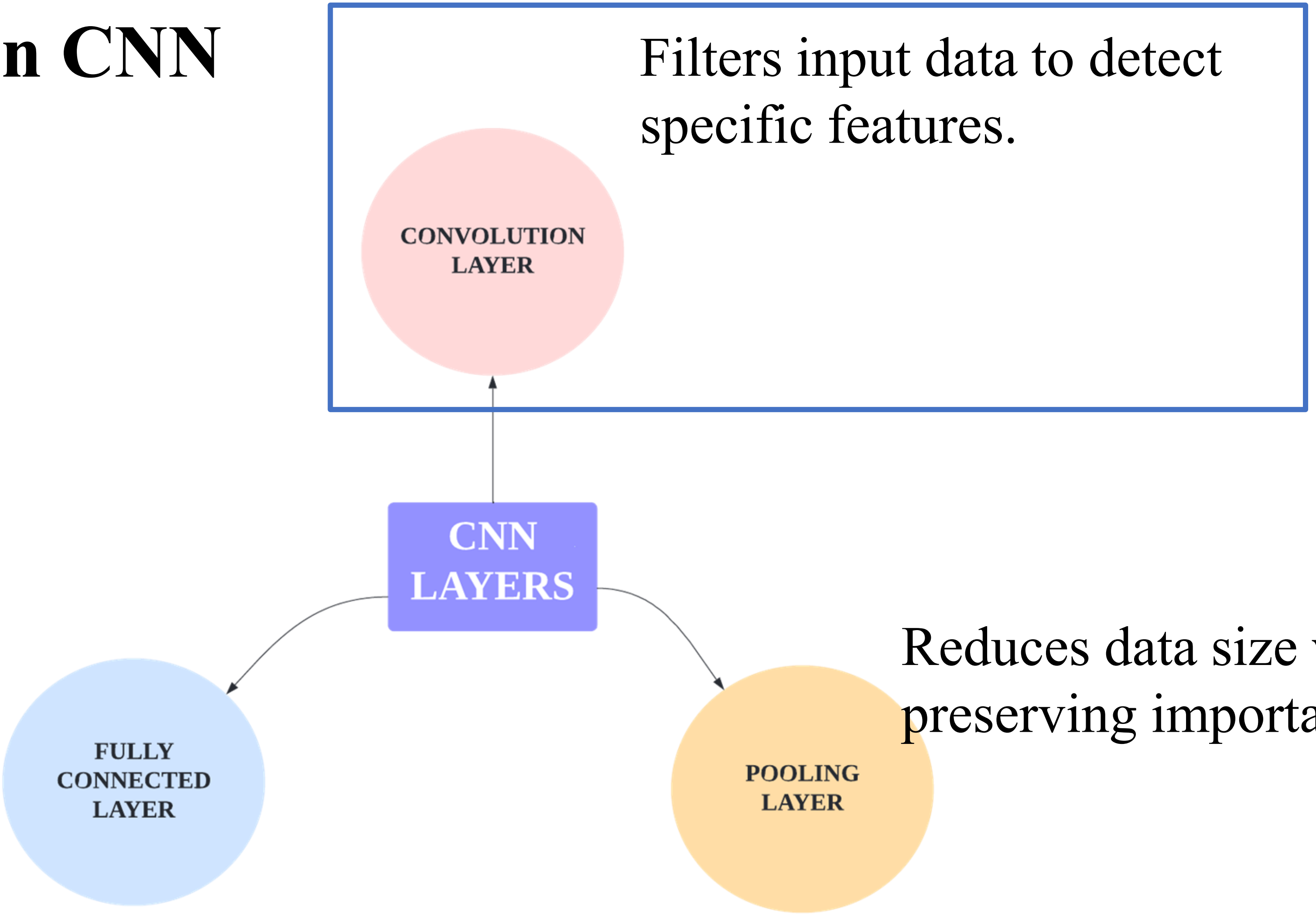
CNN Layers

Internal Workflow of CNNs



Basic CNN Architecture [https://tinyurl.com/5b3ecseh]

Different Layers in CNN



Filters input data to detect specific features.

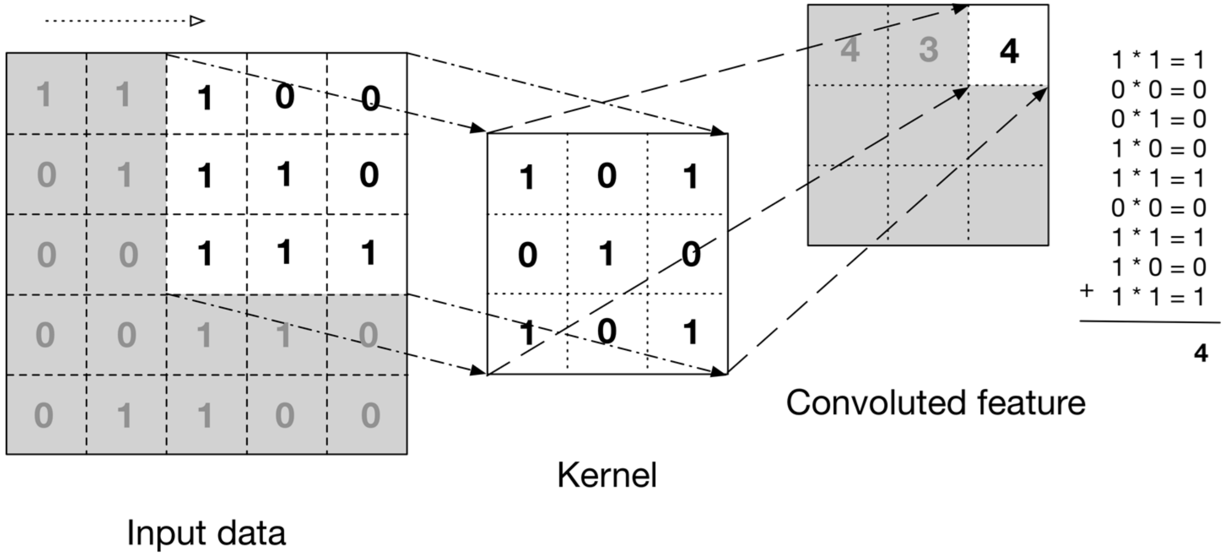
Links every neuron to previous layers’ outputs.

Reduces data size while preserving important features.

CNN Layers

Convolution Layer

- A convolution layer in a CNN applies a set of learnable filters (Kernel functions) to input image, detecting spatial features like edges, patterns or textures in an image.
- Convolutional Layers are vital for extracting meaningful features from raw data



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

1	0	1
0	1	0
1	0	1

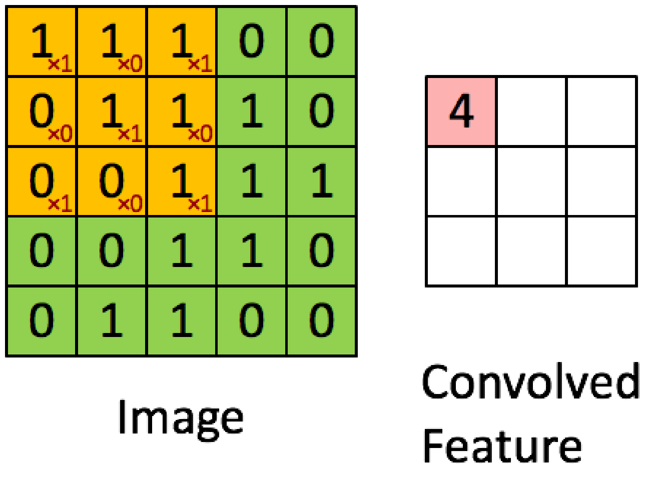
4	3	4

1 * 1 = 1
0 * 0 = 0
0 * 1 = 0
1 * 0 = 0
1 * 1 = 1
0 * 0 = 0
1 * 1 = 1
1 * 0 = 0
+ 1 * 1 = 1
4

Illustration of Convolution Operation [https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/]

Kernel

- **Kernel function:** A small filter that slide over an image to extract features from it.
- Places a filter/kernel over an array of image pixels & creates a convolved feature map.



$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

4		

Illustration of Convolution Operation [https://tinyurl.com/5u3vx4dk]

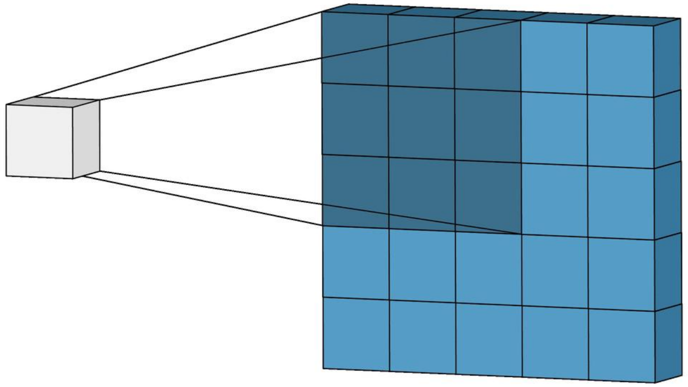
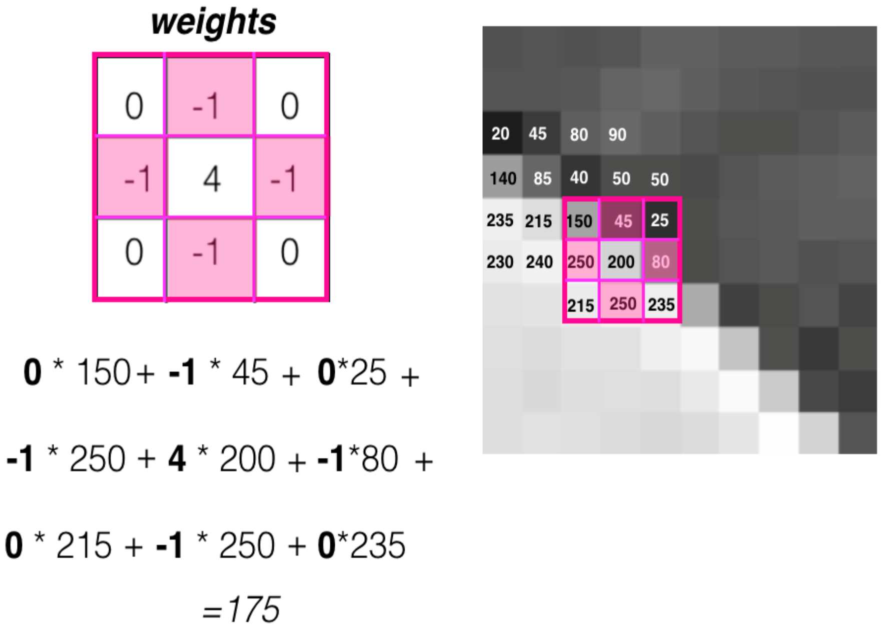


Illustration of Convolution Operation [https://tinyurl.com/3kjsmhcp]

Weights

- These weights determine what specific patterns or features the filter is looking for in the input data
- Weights are optimized during training



weights

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

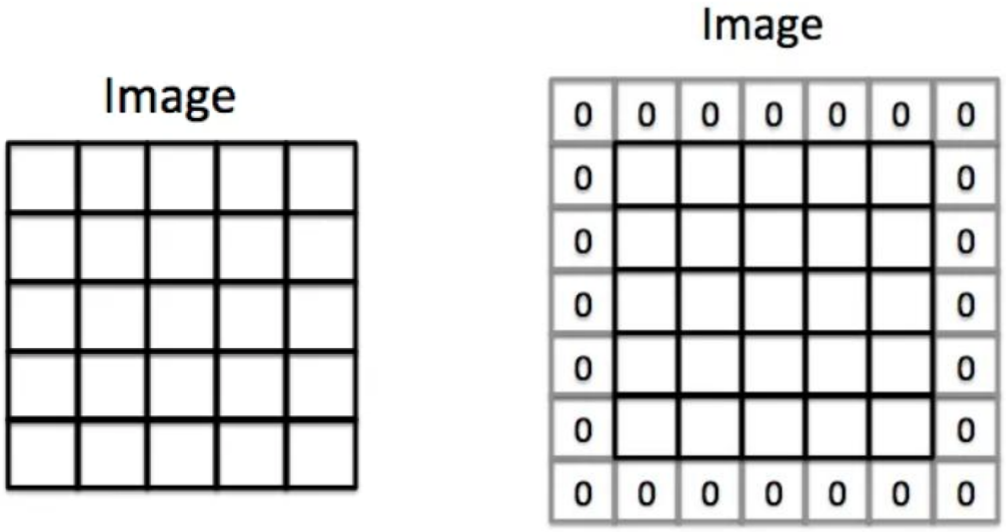
$$\mathbf{0} * 150 + \mathbf{-1} * 45 + \mathbf{0} * 25 + \mathbf{-1} * 250 + \mathbf{4} * 200 + \mathbf{-1} * 80 + \mathbf{0} * 215 + \mathbf{-1} * 250 + \mathbf{0} * 235$$

$$= 175$$

Weight Optimization [https://cezannec.github.io/Convolutional_Neural_Networks/]

Padding

- **Drawback of Kernel function:** Focuses more on the center than the corners
- **Padding is the solution! :** Adding extra pixels on the borders.
- **Zero padding :** A column and row of zeros are added on each side of the input image.



0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

Padding Operation [https://medium.com/machine-learning-algorithms/what-is-padding-in-convolutional-neural-network-c120077469cc]

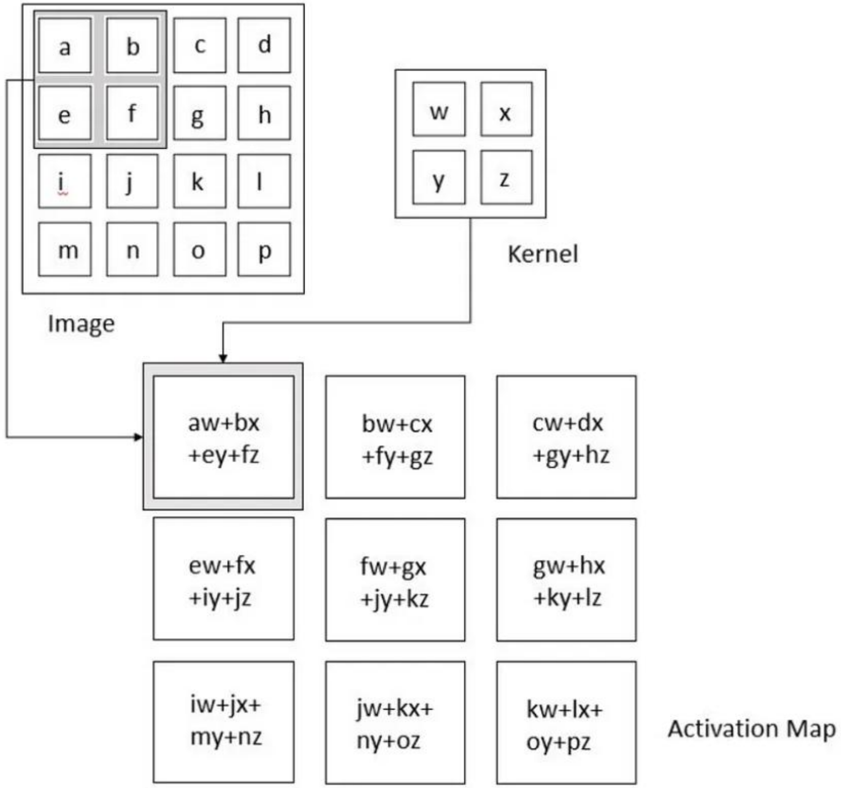
Convolution Layer

- Size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

where **W** = height/width of the image
D = number of channels
Spatial size (F) = Kernel dimensions (width x height)
Stride (S) = number of pixels by which a kernel or filter is moved across
Padding (P) = Any extra padded borders

- This will yield an output volume of size ***Wout*** x ***Wout*** x ***Dout***.



Convolution Operation

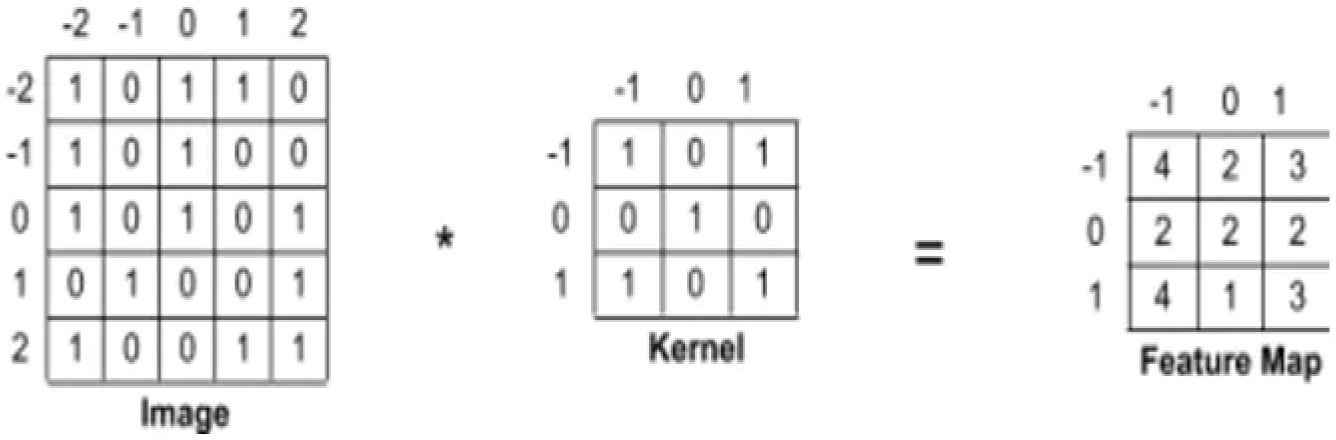
[https://doi.org/10.1007/s10710-017-9314-z]

Convolution Layer

- Dimension of the image (I) = W1 x W2 x Wc (height x width x channels)
- Dimension of the filter (K) = n1 x n2 x nc (height x width x channels)

  (Where nc = Wc)

$$\text{Feature Map } F_{[i,j]} = (I * K)_{[i,j]} \Rightarrow f[i,j] = \sum_{x}^{m_1} \sum_{y}^{m_2} \sum_{z}^{m_c} K_{[x,y,z]} I_{[i+x-1,j+y-1,z]}$$



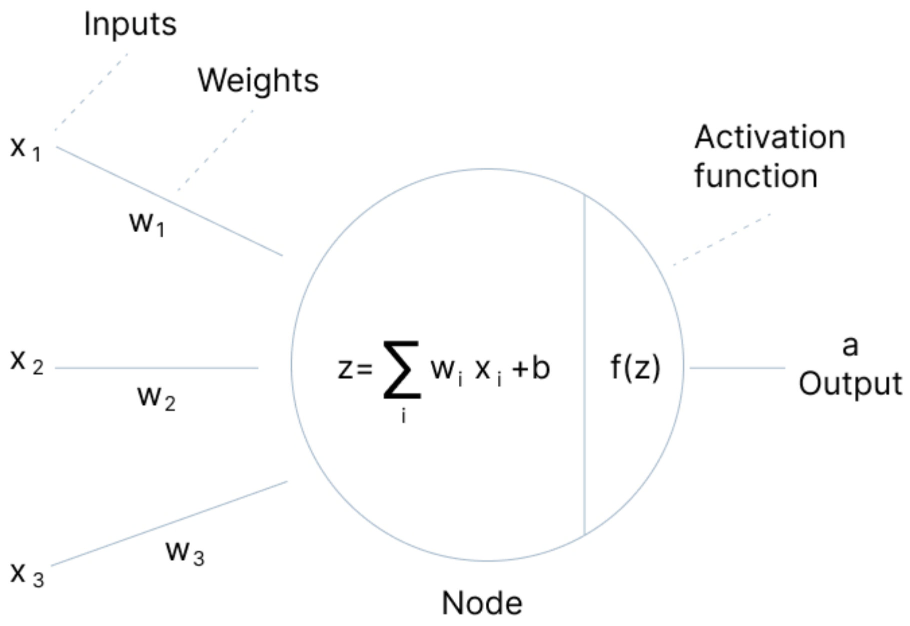
$$\begin{array}{c|ccccc} & -2 & -1 & 0 & 1 & 2 \\ \hline -2 & 1 & 0 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 1 & 1 \end{array} * \begin{array}{c|ccc} & -1 & 0 & 1 \\ \hline -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{array} = \begin{array}{c|ccc} & -1 & 0 & 1 \\ \hline -1 & 4 & 2 & 3 \\ 0 & 2 & 2 & 2 \\ 1 & 4 & 1 & 3 \end{array}$$

Calculation of values of the Feature Map

https://doi.org/10.1007/s11042-019-08385-4

Activation Functions in Convolution Layer

- Activation functions are used to learn complex patterns in data.
- Transform the weighted sum of inputs to determine the neuron's output or "activation."
- The Activation Functions can be basically divided into 2 types-
  1. Linear Activation Function
  2. Non-linear Activation Functions

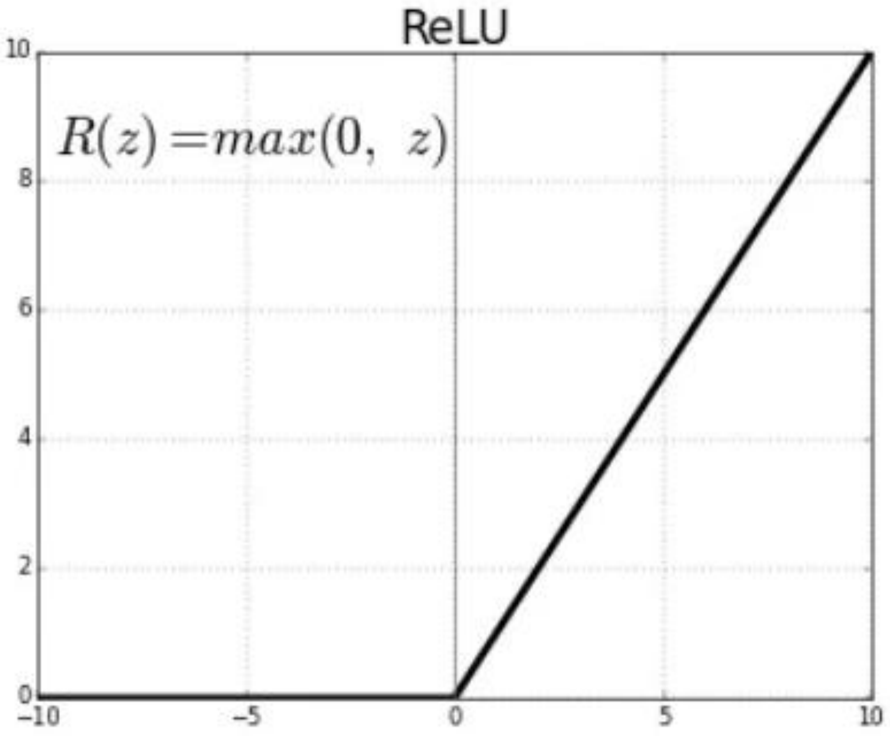


Activation Functions in CNNs [https://tinyurl.com/hbk3f5k9]

Rectified Linear Unit (Relu)

- It computes the function f(x)=max(0,x). In other words, the activation is simply thresholded at zero

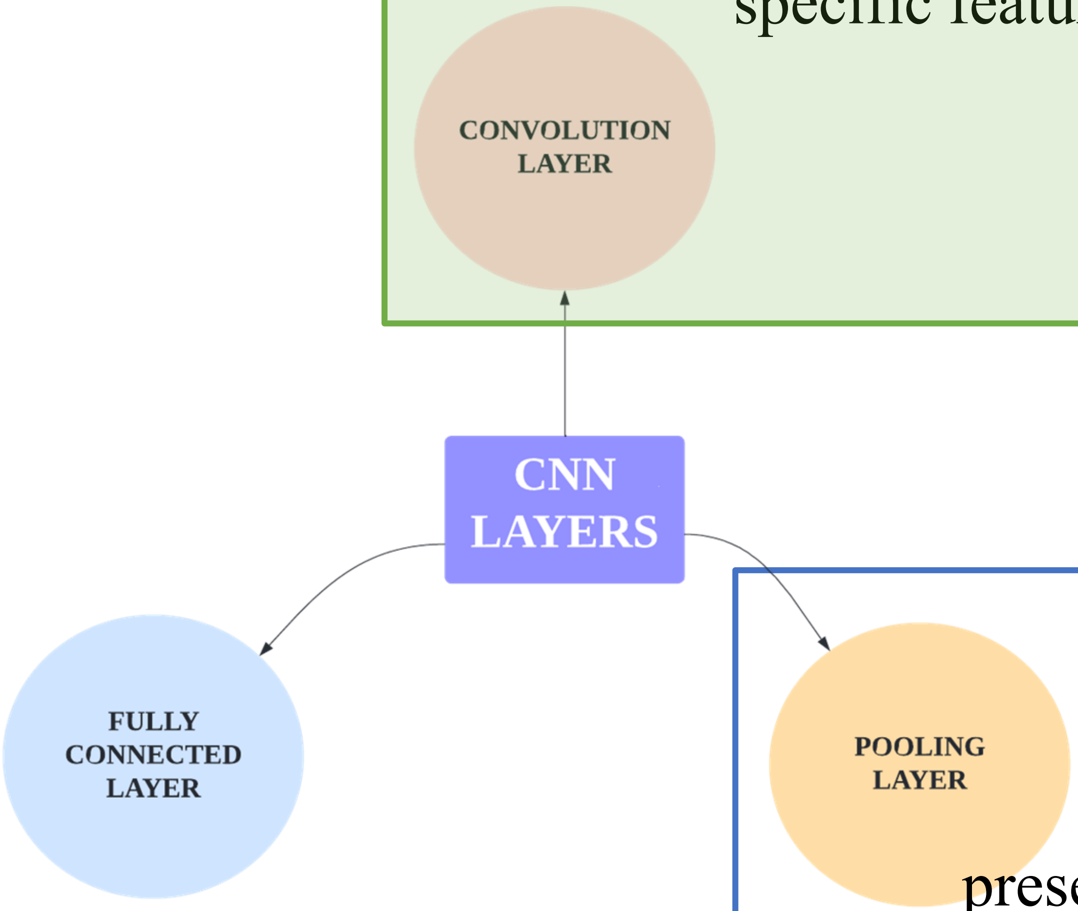
- The function returns 0 if it receives any negative input, but for any positive value x, it returns that value back.



Padding Operation [https://tinyurl.com/32fa5f5h]

Different Layers in CNN

Filters input data to detect specific features.



Links every neuron to previous layers’ outputs.

Reduces data size while preserving important features.

CNN Layers

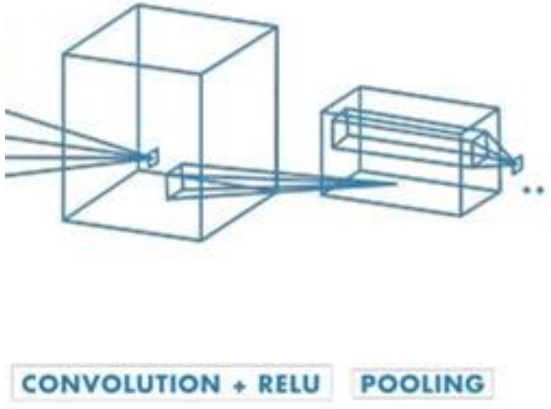
Pooling Layer

- Responsible for reducing the spatial size of the Convolved Feature
- Aim is to decrease computational power required to process the data by reducing the dimensions

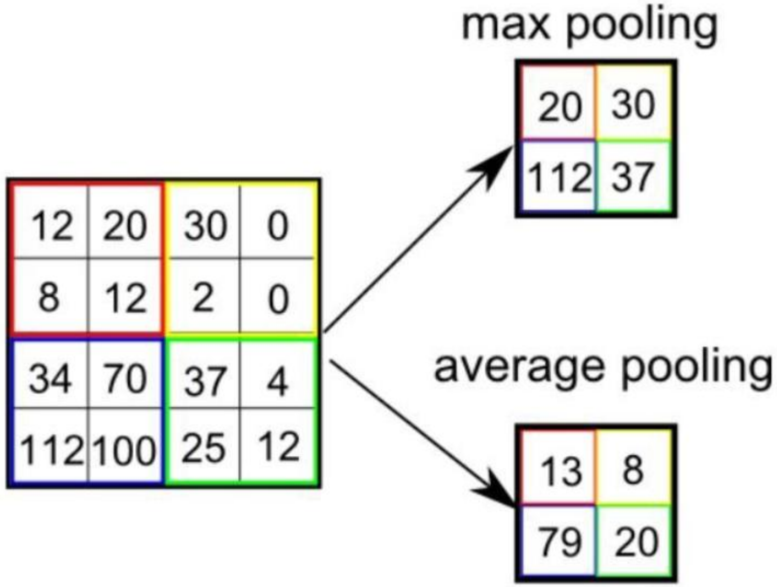
There are two types of pooling.

Average Pooling

- Returns the average of all the values from the portion of the image covered by the Kernel.
- Simply performs dimensionality reduction as a noise suppressing mechanism.



Basic CNN Architecture [https://tinyurl.com/5b3ecseh]



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

average pooling

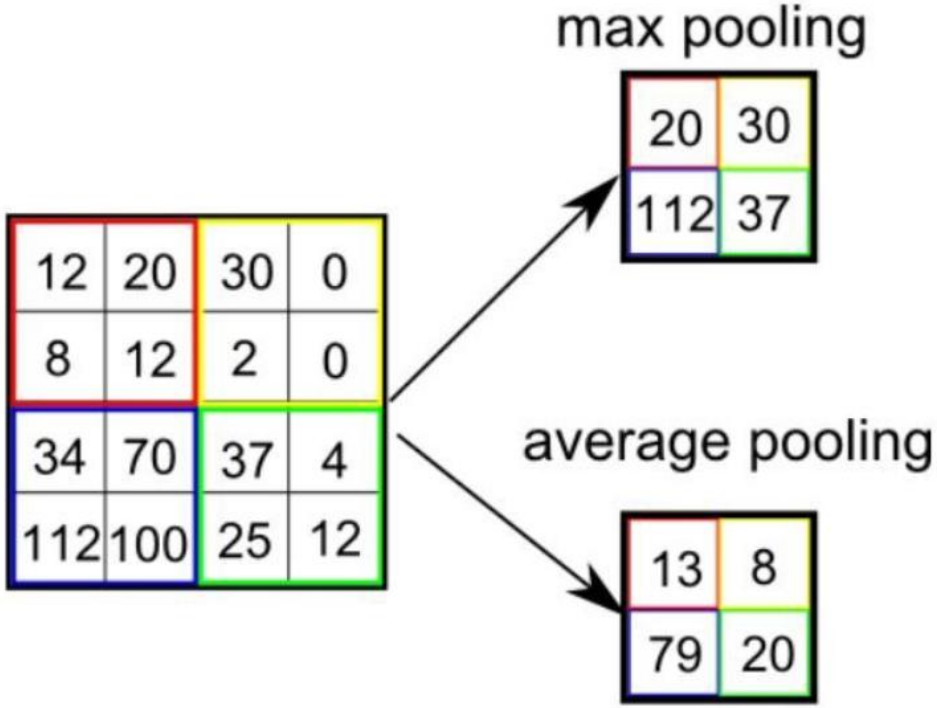
13	8
79	20

Illustration of Max Pooling and Average Pooling layers [https://tinyurl.com/5n6umca5]

Pooling Layer

Max Pooling

- Find the maximum value from a portion of the image covered by the kernel.
- Discards the noisy activations altogether
- Performs de-noising along with dimensionality reduction.



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

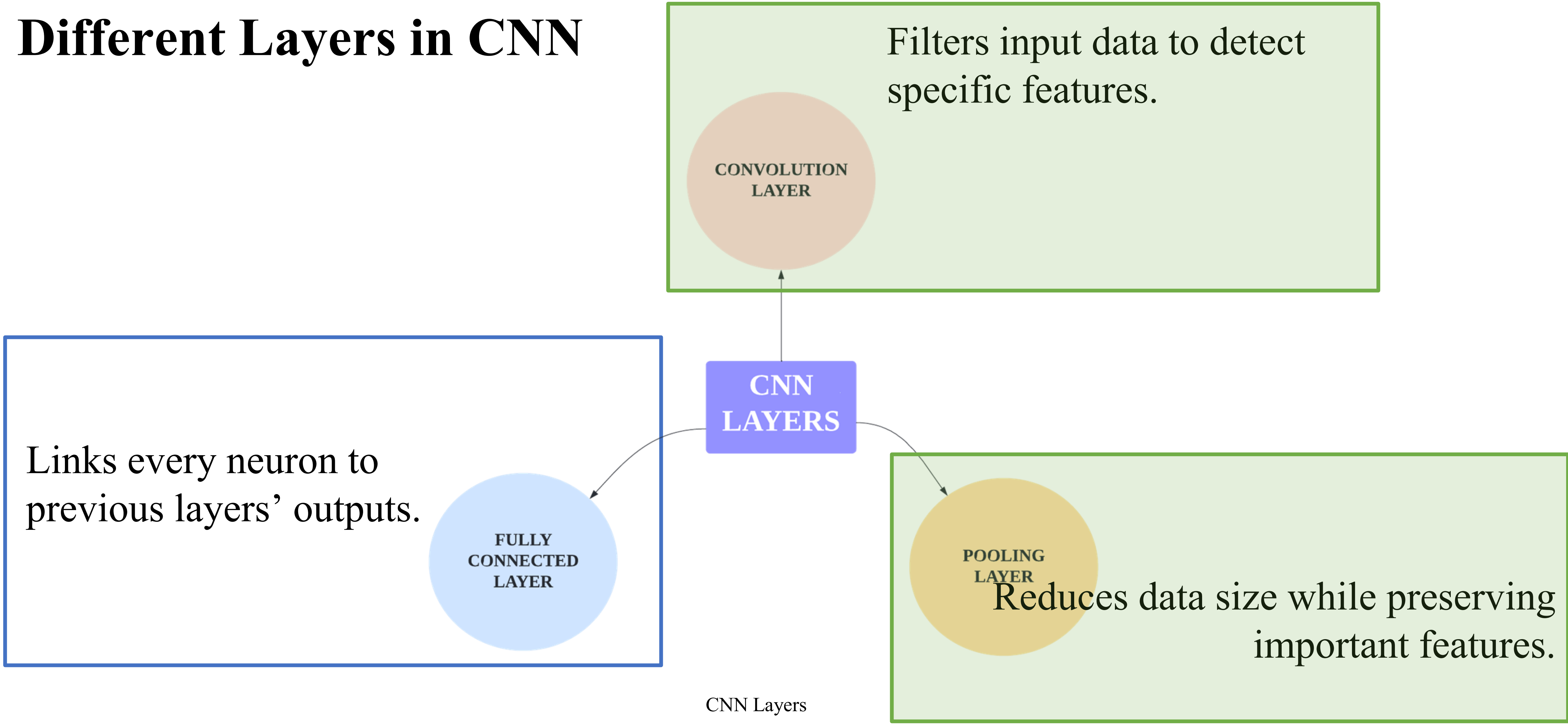
20	30
112	37

average pooling

13	8
79	20

Illustration of Max Pooling and Average Pooling layers [https://tinyurl.com/5n6umca5]

Different Layers in CNN



Filters input data to detect specific features.

Links every neuron to previous layers' outputs.

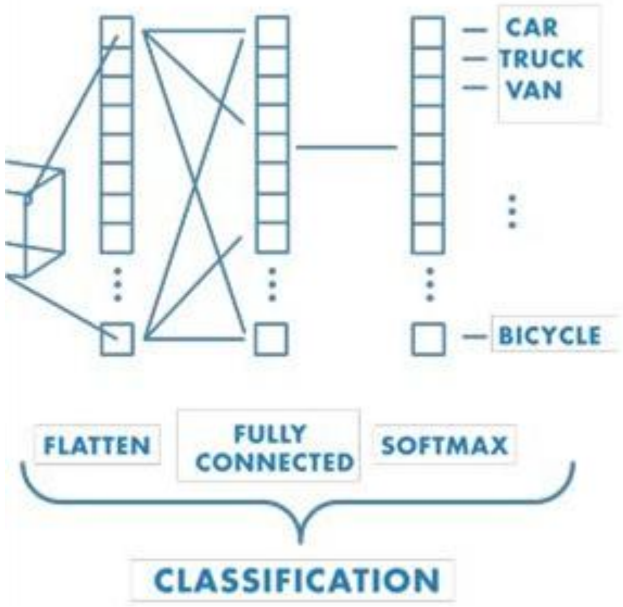
Reduces data size while preserving important features.

CNN Layers

Fully Connected Layer

- Once the features extracted by the convolution layers and downsampled by the pooling layers are created, they are mapped by a subset of fully connected layers to the final outputs of the network, such as the probabilities for each class in classification tasks.

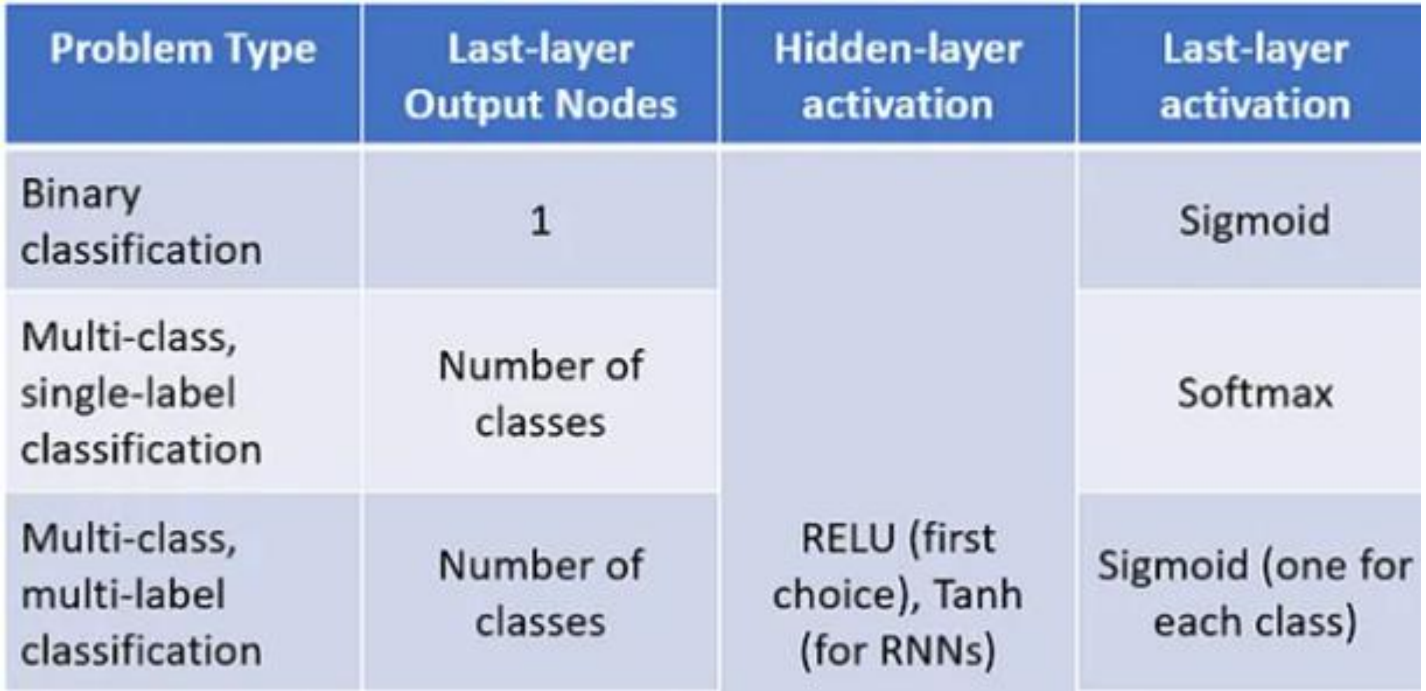
- The final fully connected layer typically has the same number of output nodes as the number of classes. Each fully connected layer is followed by a nonlinear function, such as ReLU



Basic CNN Architecture [https://tinyurl.com/5b3ecseh]

Activation Functions used in Fully Connected Layer

Table - Different Activation Functions used in Hidden and Output Layer

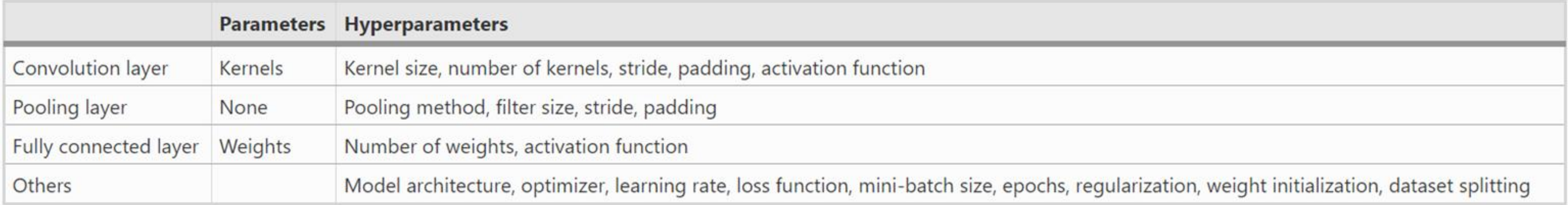


Problem Type	Last-layer Output Nodes	Hidden-layer activation	Last-layer activation
Binary classification	1	RELU (first choice), Tanh (for RNNs)	Sigmoid
Multi-class, single-label classification	Number of classes		Softmax
Multi-class, multi-label classification	Number of classes		Sigmoid (one for each class)

Different Activation functions [https://tinyurl.com/5n8cdwze]

A list of parameters and hyperparameters in a convolutional neural network (CNN)

Table - Summary for CNN Layers



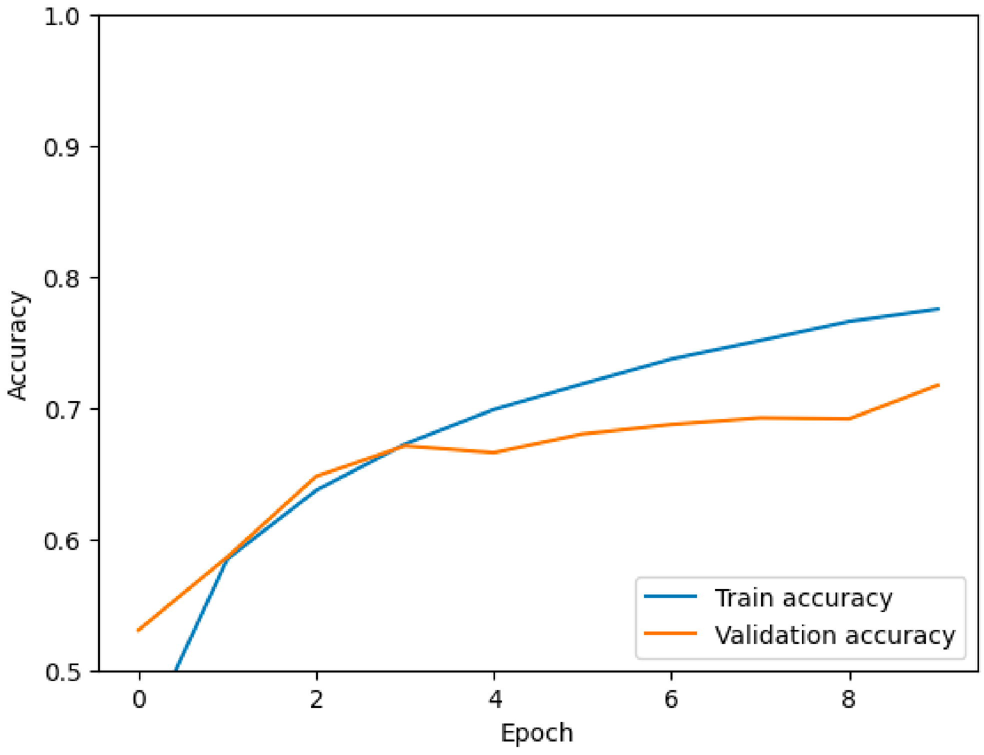
	Parameters	Hyperparameters
Convolution layer	Kernels	Kernel size, number of kernels, stride, padding, activation function
Pooling layer	None	Pooling method, filter size, stride, padding
Fully connected layer	Weights	Number of weights, activation function
Others		Model architecture, optimizer, learning rate, loss function, mini-batch size, epochs, regularization, weight initialization, dataset splitting

https://doi.org/10.1007/s13244-018-0639-9

Evaluation Metrics

Accuracy (Correct/Total Predictions)

- **Train Accuracy :** High training accuracy doesn't guarantee good performance on unseen data.
- **Test Accuracy (on unseen data):** High test accuracy == Model generalizes well to new data.
- **Validation Accuracy :** to tune hyperparameters and prevent overfitting during training
  - ○ provides an interim check on the model's performance.



Our Model’s Train Vs. Val Accuracy [https://tinyurl.com/mycnn01]

Evaluation Metrics

Classification Loss functions

- **Binary Cross Entropy Loss/Negative Log Likelihood**
  - As the predicted probability diverges from the actual label, cross-entropy loss increases.

$$CrossEntropyLoss = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

Binary Cross Entropy Loss [https://tinyurl.com/3w9y86ed]

  - Here,
    - yi = Value of the Actual Label
    - ŷi = Value of the Predicted Label

Evaluation Metrics

Classification Loss functions

- **Categorical Cross Entropy**
  - Same as binary, but for multiclass classification

$$\text{Loss} = -\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

where k is number of classes in the data

Categorical Cross Entropy Loss [https://tinyurl.com/3w9y86ed]

  - Here,
    - yj = Value of the Actual Label
    - ŷj = Value of the Predicted Label

Applications of CNN

1. **Decoding Facial Recognition**
   - Identifying every face and unique features
2. **Analyzing Documents**
   - Optical Character Recognition (OCR), Signature Verification
3. **Understanding Climate**
   - Cloud Detection & Classification, Extreme Weather Event Prediction
4. **Advertising**
   - Understanding buying patterns and data-driven personalized advertising
5. **Collecting Historic and Environmental Elements**
   - Identification of Historic Artifact and various species in the environment

Applications of CNN [https://openai.com/dall-e-3]

Implementation of CNN

Libraries for NN:

Deep Learning Libraries [https://tinyurl.com/52fnc5eu]

Demo Implementation : https://tinyurl.com/mycnn01

Paper References

[1] Heaton, J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genet Program Evolvable Mach 19, 305–307 (2018). https://doi.org/10.1007/s10710-017-9314-z

[2] Zaniolo, L., Marques, O. On the use of variable stride in convolutional neural networks. Multimed Tools Appl 79, 13581–13598 (2020). https://doi.org/10.1007/s11042-019-08385-4

[3] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

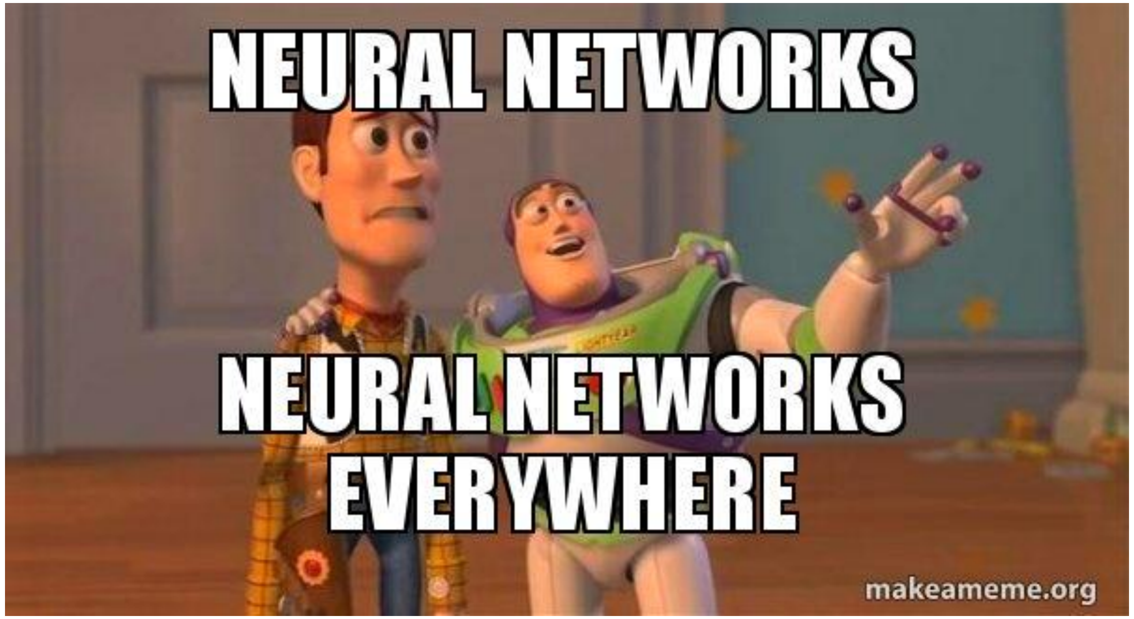
[4] Kumar, Vaibhav & L., M.. (2018). Deep Learning as a Frontier of Machine Learning: A Review. International Journal of Computer Applications. 182. 22-30. 10.5120/ijca2018917433.

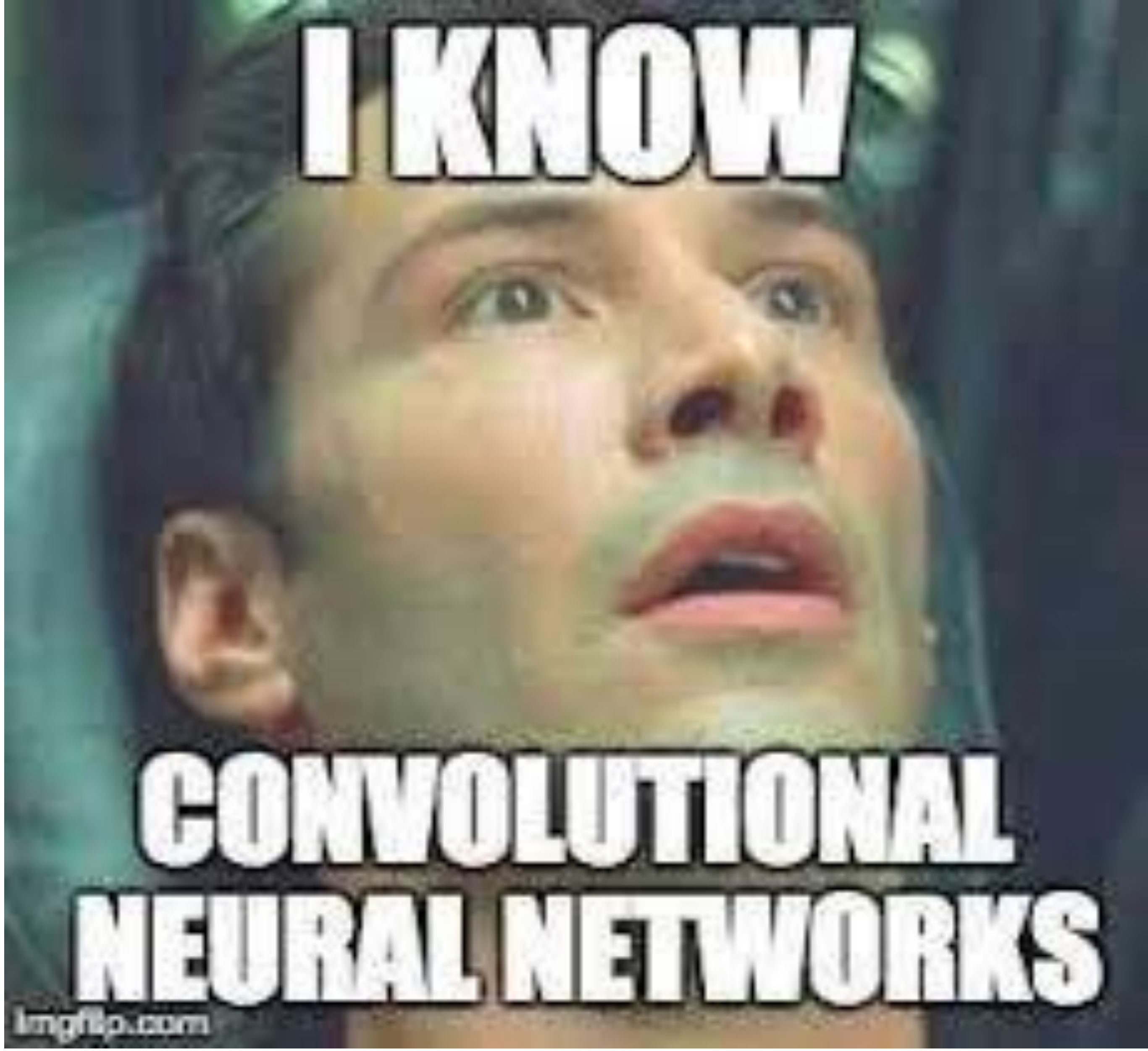
[5]Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

Video References

- What is a convolutional neural network (CNN)? [https://youtu.be/K_BHmztRTpA?si=ZMw-jJv6NYZCwYcb]
- Convolutional Neural Network Tutorial (CNN) | How CNN Works | Deep Learning Tutorial | Simplilearn[https://youtu.be/Jy9-aGMB_TE?si=Ob8qf05jDnoWBd7g]
- Simple explanation of convolutional neural network | Deep Learning Tutorial 23 (Tensorflow & Python)[https://youtu.be/zfiSAzpy9NM?si=qJ6vCc9cqpiDSAw7]

SAY DEEP LEARNING
ONE MORE TIME


NEURAL NETWORKS
NEURAL NETWORKS EVERYWHERE
makeameme.org


I KNOW
CONVOLUTIONAL NEURAL NETWORKS
imgflip.com