SRH Hochschule Heidelberg
Analytics 4
Dealing with Images

Academic Researcher: Ashish Chouhan
External Dozent: Ajinkya Patil

Date of Lecture: 25.05.2021

Advantages of dealing with Images

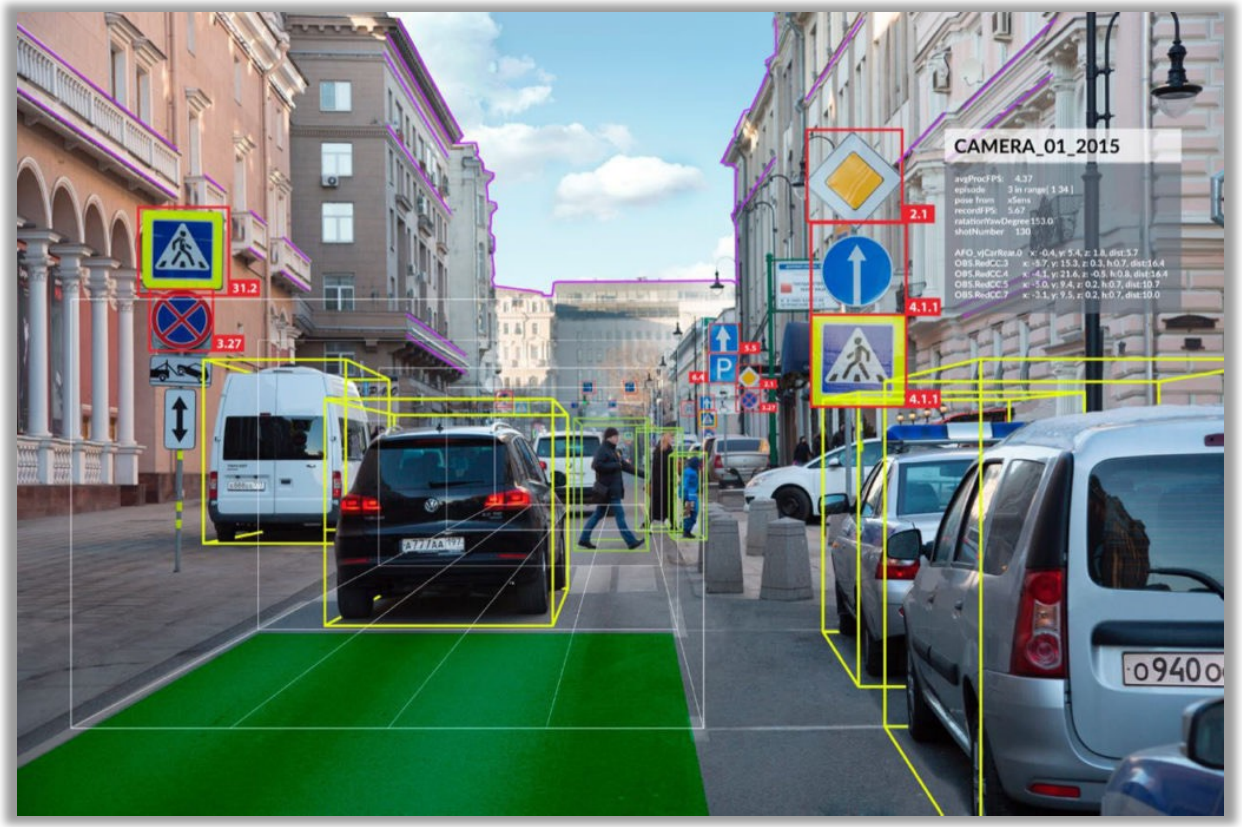
- Image Recognition and Classification
  - Automated Image Organization

Advantages of dealing with Images

- Image Recognition and Classification
  - Automated Image Organization
  - Automated Product Tagging

Advantages of dealing with Images

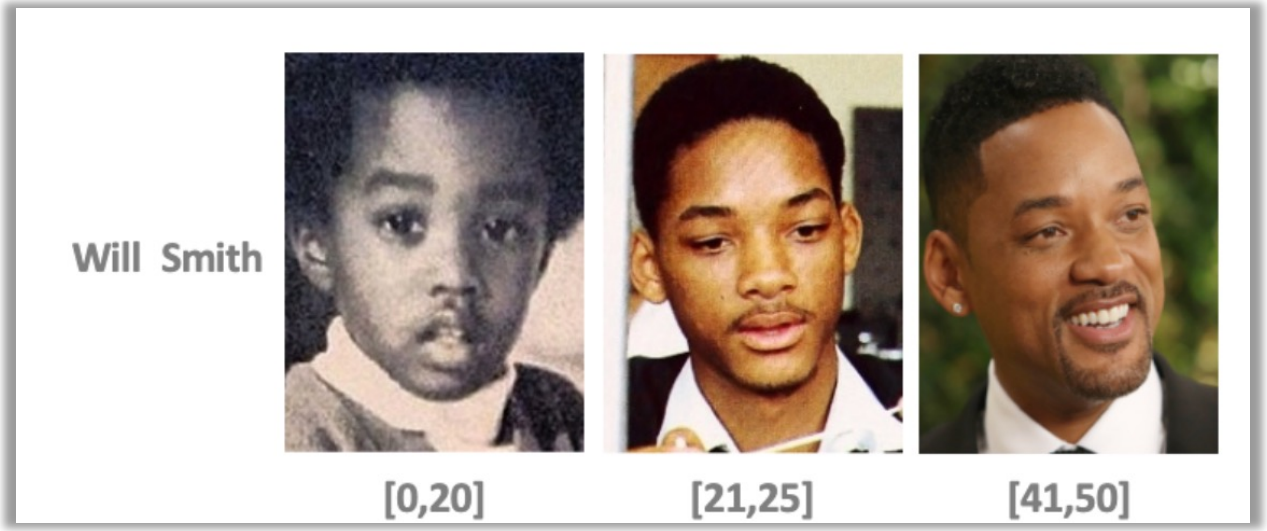
— Self driving cars



Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J. and Zhang, X., 2016. **End to end learning for self-driving cars.** *arXiv preprint arXiv:1604.07316*.

Advantages of dealing with Images

— Face recognition



Zhao, J., Cheng, Y., Cheng, Y., Yang, Y., Zhao, F., Li, J., Liu, H., Yan, S. and Feng, J., 2019, July. **Look across elapse: Disentangled representation learning and photorealistic cross-age face synthesis for age-invariant face recognition.** In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, pp. 9251-9258).

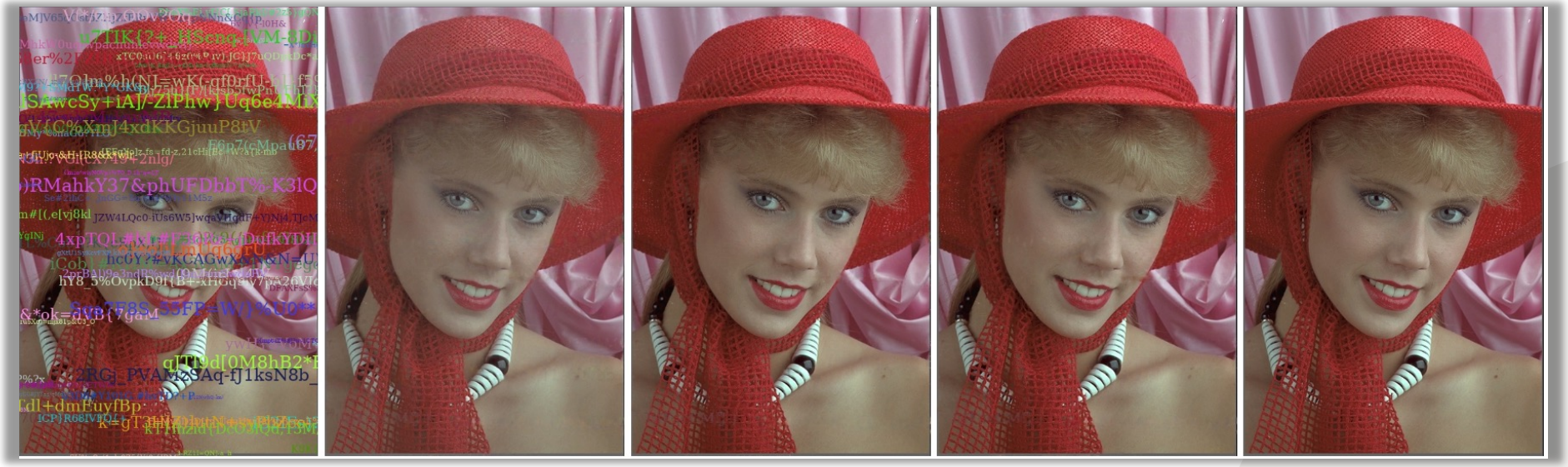
Advantages of dealing with Images

— Is your selfie good?

http://karpathy.github.io/2015/10/25/selfie/

Advantages of dealing with Images

— Image Denoising



Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M. and Aila, T., 2018. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*.

Github: https://github.com/NVlabs/noise2noise

Advantages of dealing with Images

— Automatic Colorization

https://tinyclouds.org/colorize/

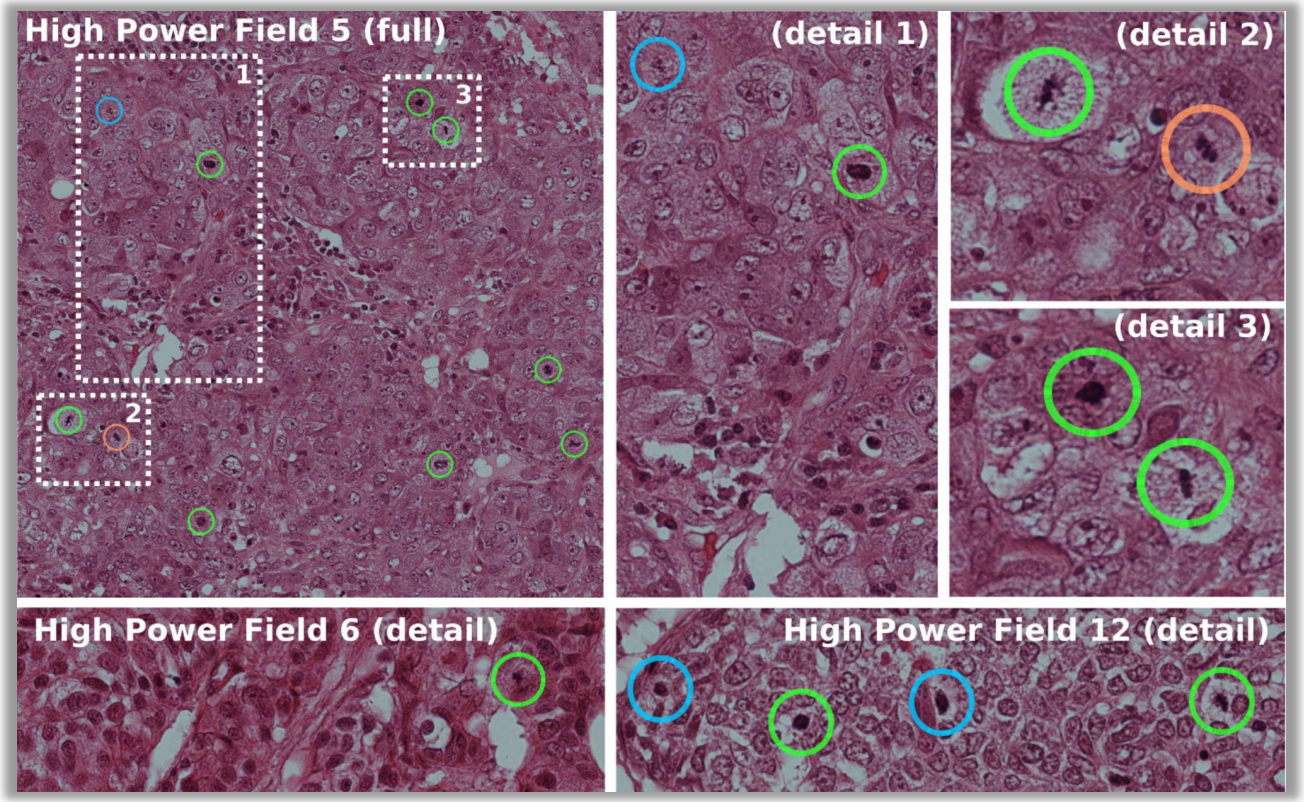
Advantages of dealing with Images

— New types of art is created

Elgammal, A., Liu, B., Elhoseiny, M. and Mazzone, M., 2017. **CAN: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms**. *arXiv preprint arXiv:1706.07068*.

Advantages of dealing with Images

— Detecting Mitosis



Giusti, A., Caccia, C., Cireşari, D.C., Schmidhuber, J. and Gambardella, L.M., 2014, April. **A comparison of algorithms and humans for mitosis detection.** In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)* (pp. 1360-1363). IEEE.

What is an Image?

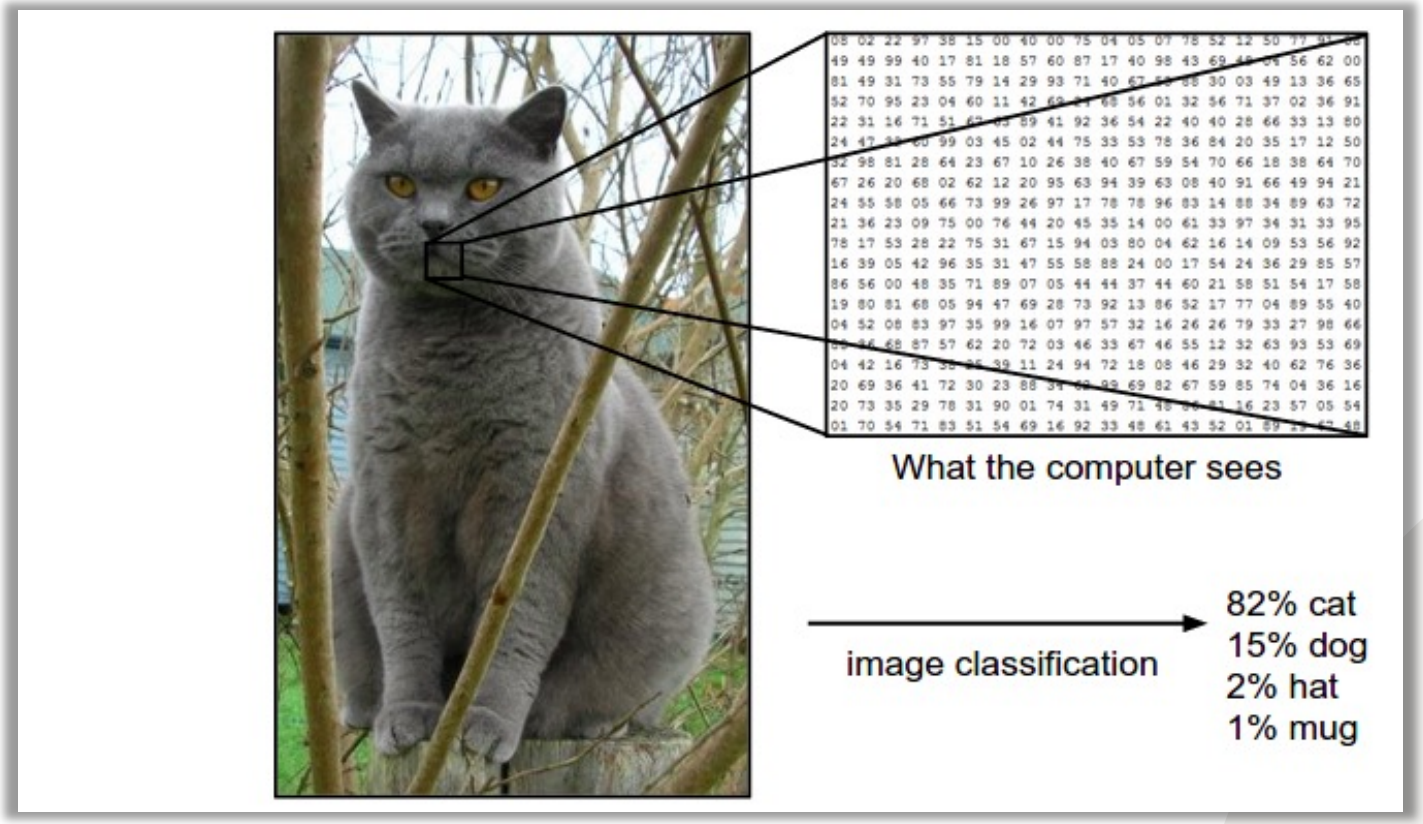


Image is represented as matrix of pixel values.

What is an Image?

- A grey scale image
  - 1 Channel
  - White and Black color pixel values represented in 2-dimensional matrix.

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

What is an Image?

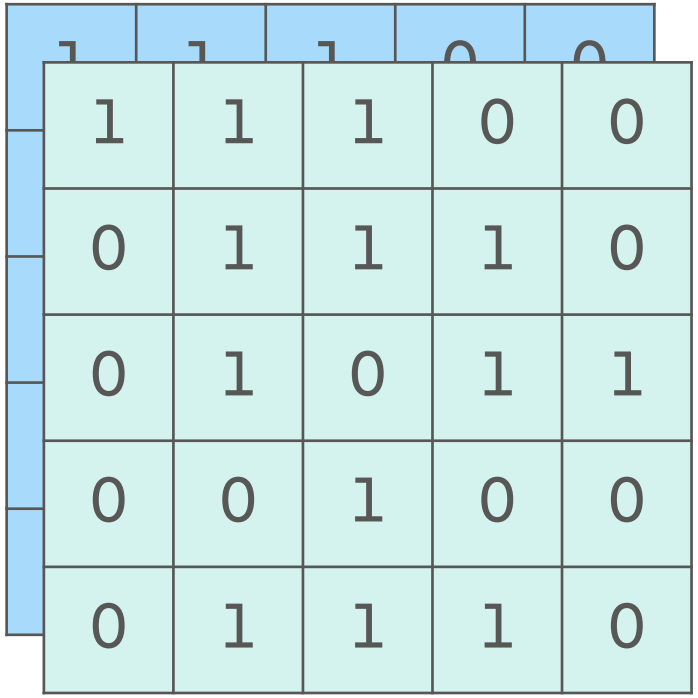
- A color image
  - 3 Channels (Red Green Blue)
  - All three channels color pixel values represented in 2-dimensional matrix, overlaped to create a color image.

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

Blue

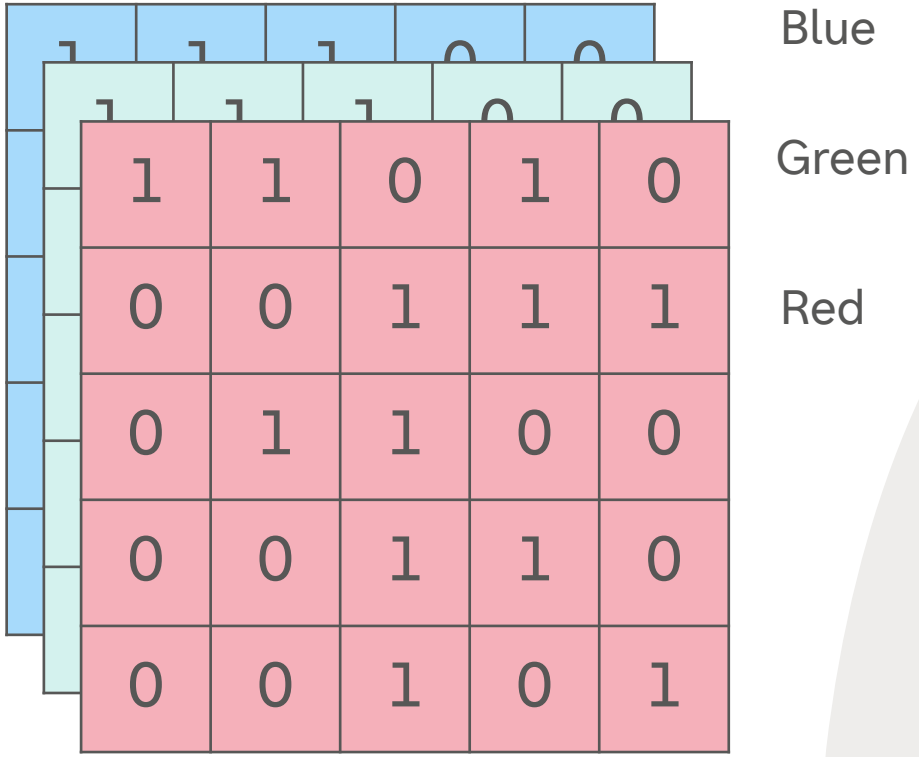
What is an Image?

- A color image
  - 3 Channels (Red Green Blue)
  - All three channels color pixel values represented in 2-dimensional matrix, overlaped to create a color image.



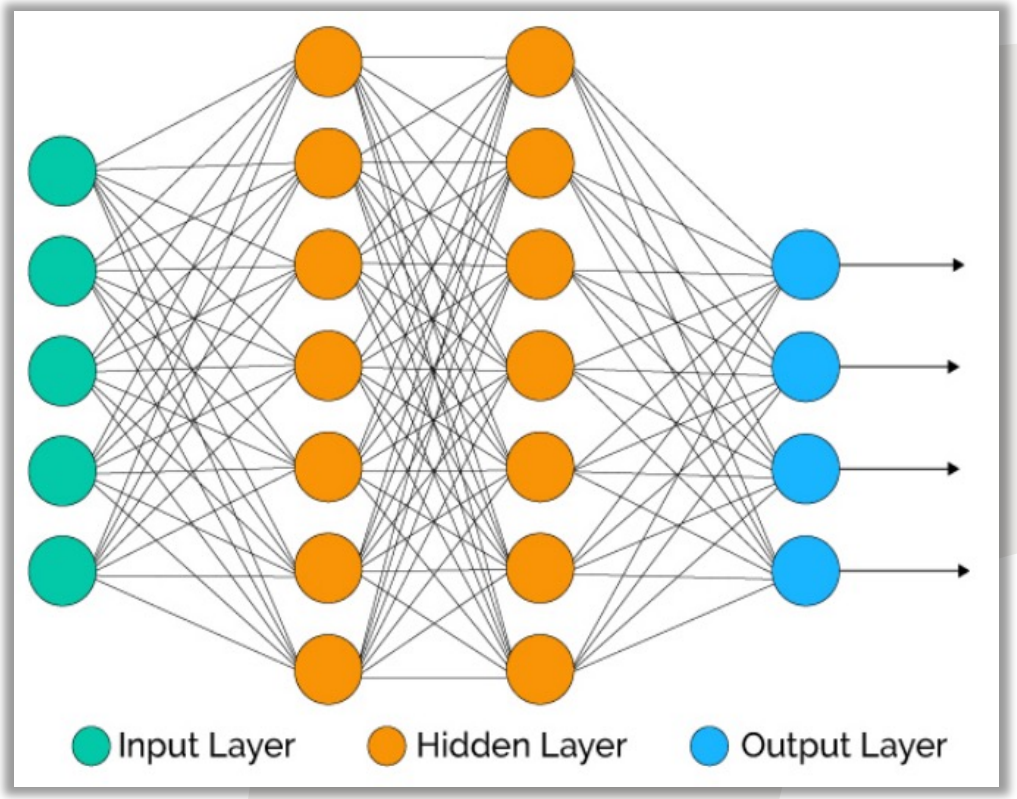
What is an Image?

- A color image
  - 3 Channels (Red Green Blue)
  - All three channels color pixel values represented in 2-dimensional matrix, overlaped to create a color image.



Dealing with Images

- Each pixel is one input neuron
- Image is 128 px * 128 px size (Gray scale image)
  - Input number of neurons: 128 * 128 = 16,384
- Consider a color image (3 Channels):
  - Input number of neurons: 128 * 128 * 3 = 49,152



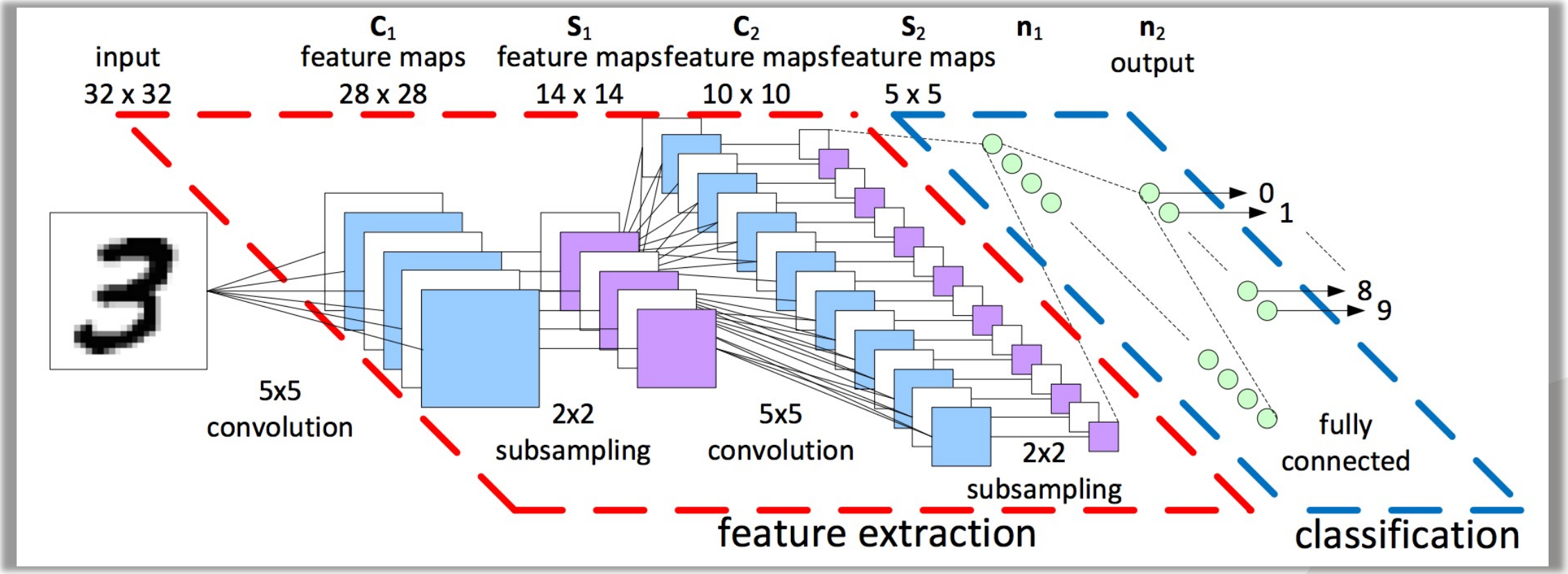
Dealing with Images

- Each pixel is one input neuron
- Image is 128 px * 128 px size (Gray scale image)
  - Input number of neurons: 128 * 128 = 16,384
- Consider a color image (3 Channels):
  - Input number of neurons: 128 * 128 * 3 = 49,152
- Work with larger images:
  - Image size is 1000 * 1000 * 3 = 3000000 = 3 million input neurons
  - Complexity: if 1000 hidden neurons then, Number of weights = 3 million * 1000
  - Weight Matrix = 3 million * 1000 = 3 billion parameters

Dealing with Images

Drawbacks of processing large images:

- Memory requirements will be higher
- Dataset requirement will be higher to train such a neural network.



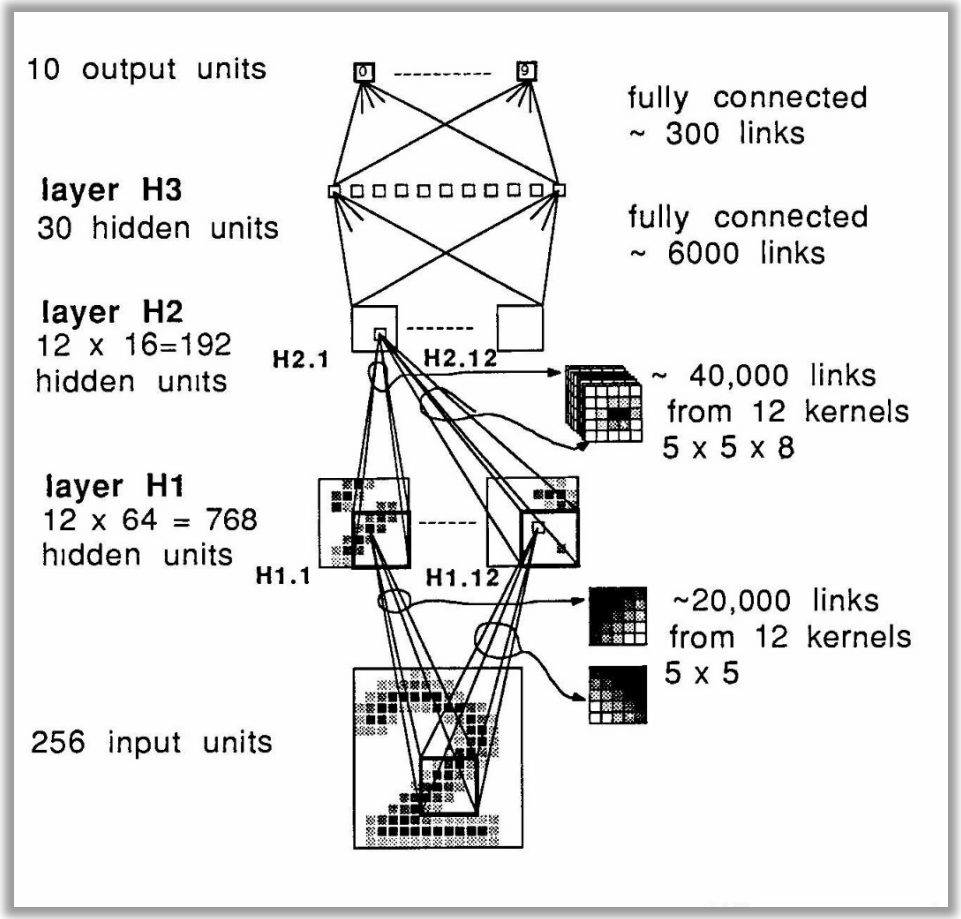
Convolutional Neural Network

What is Convolutional Neural Network?

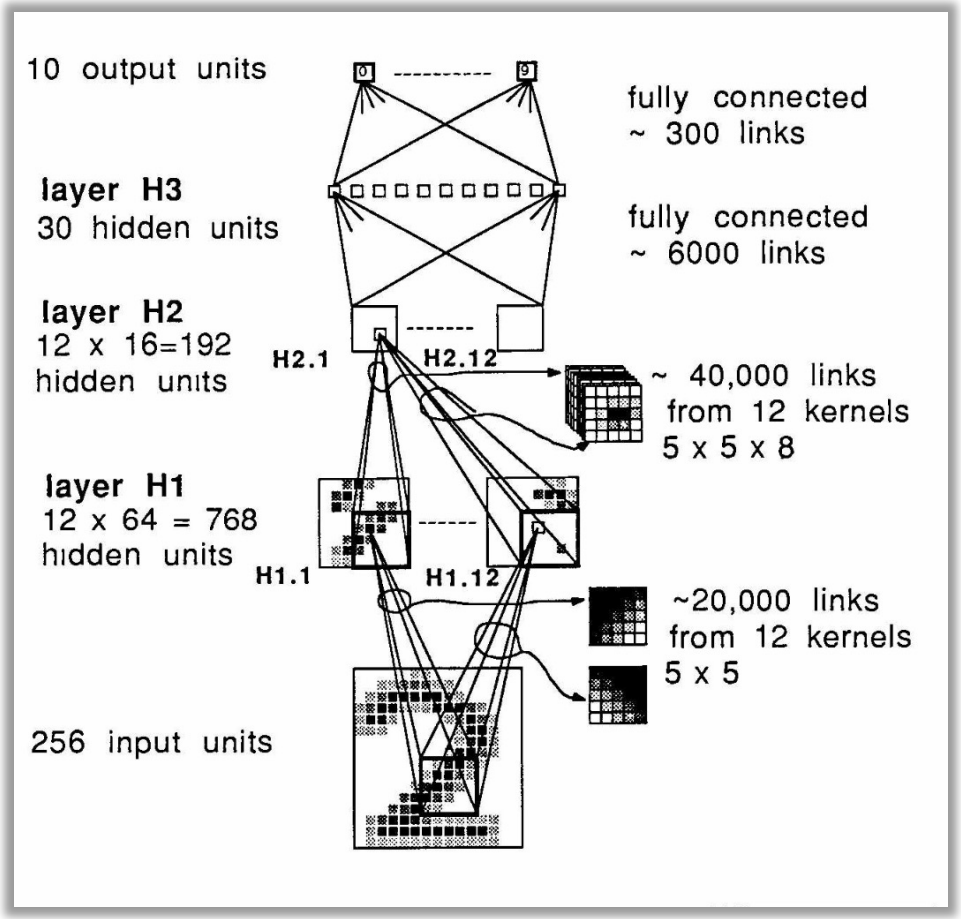
Convolution is a mathematical operation on two functions that produces a third function.

$$f(x)*g(x)=t(x)$$

What is Convolutional Neural Network?



What is Convolutional Neural Network?



LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. **Backpropagation applied to handwritten zip code recognition.** *Neural computation*, *1*(4), pp.541-551.

What is Convolutional Neural Network?

Demo 1:

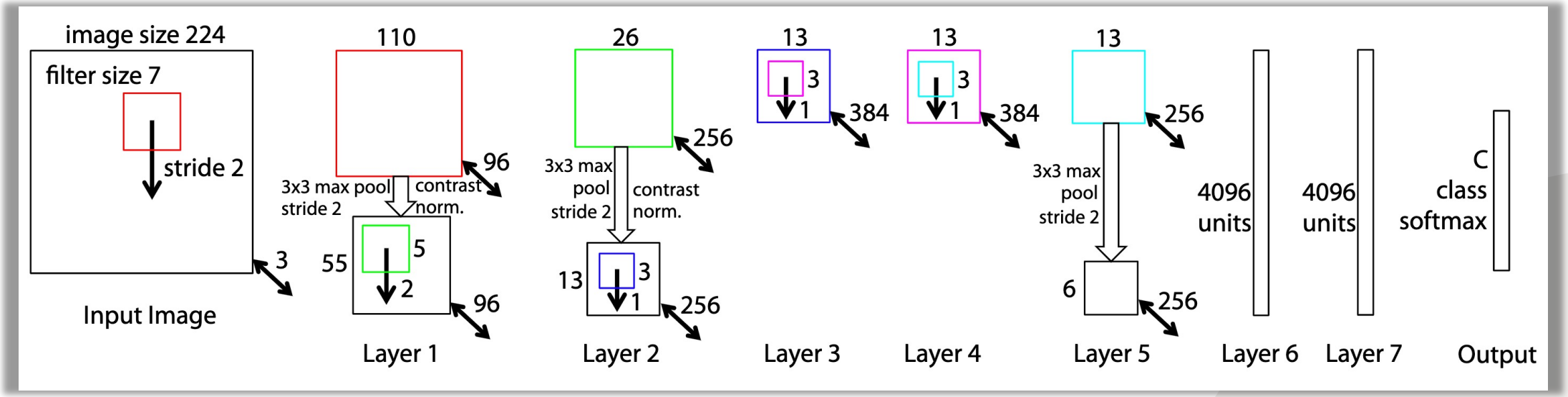
https://www.cs.ryerson.ca/~aharley/vis/conv/

Harley, A.W., 2015, December. **An interactive node-link visualization of convolutional neural networks.** In International Symposium on Visual Computing (pp. 867-877). Springer, Cham.

Demo 2:

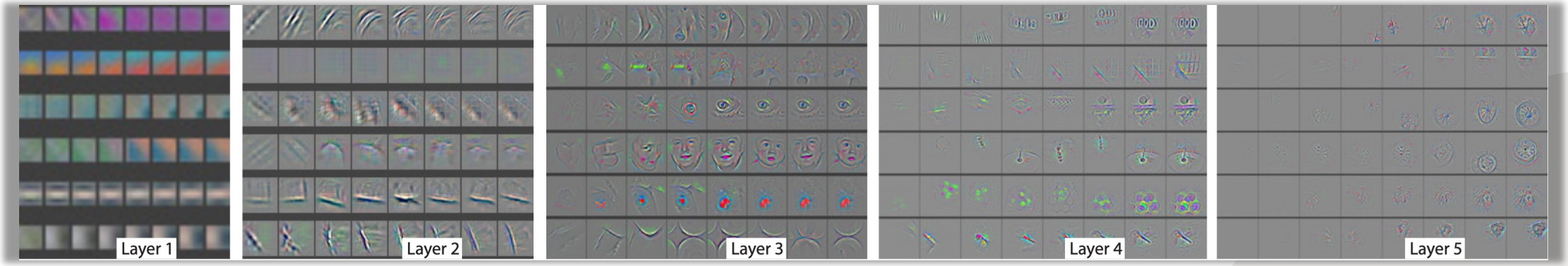
https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

What is Convolutional Neural Network?



Zeiler, M.D. and Fergus, R., 2014, September. **Visualizing and understanding convolutional networks. In *European conference on computer vision*** (pp. 818-833). Springer, Cham.

What is Convolutional Neural Network?



Zeiler, M.D. and Fergus, R., 2014, September. **Visualizing and understanding convolutional networks. In *European conference on computer vision*** (pp. 818-833). Springer, Cham.

How Convolutional Neural Network Work?

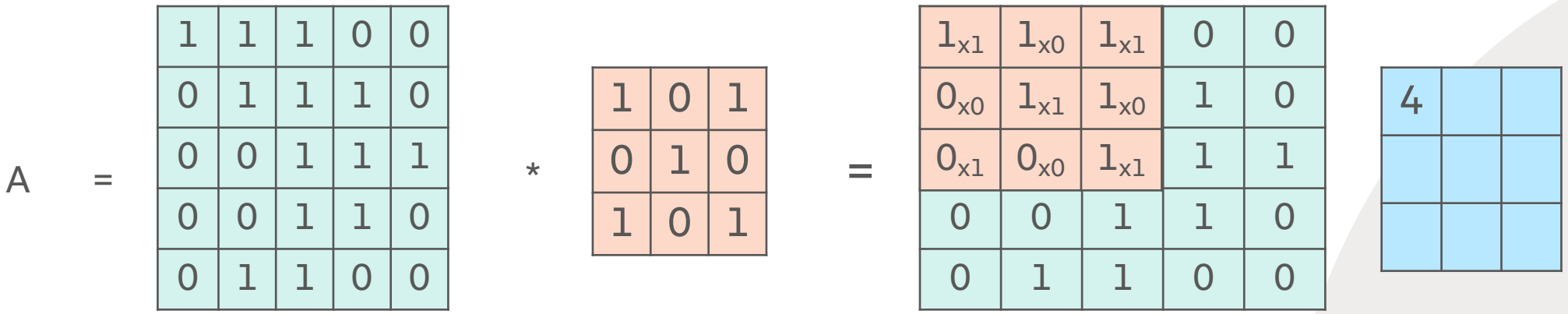
- Convolution Layer
- Pooling or Sub sampling
- Flatenning
- Non-Linearity (ReLU, Tanh)
- Classification or Fully Connected Layer

How Convolutional Neural Network Work?

- **Convolution Layer**
- Pooling or Sub sampling
- Flatenning
- Non-Linearity (ReLU, Tanh)
- Classification or Fully Connected Layer

Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



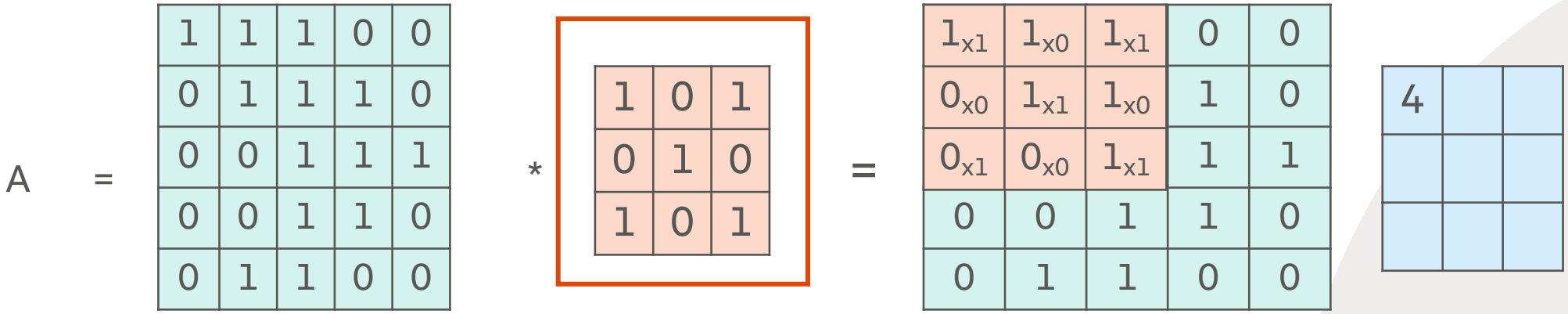
Input image

Filter/Kernel /Feature Detector

Convolved Feature/Activation Map/Feature Map

Filters

- Filters is the matrix with the random values that help to extract feature from the image.
- Different size of filters can be used based on input image.



Input image

Filter/Kernel /Feature Detector

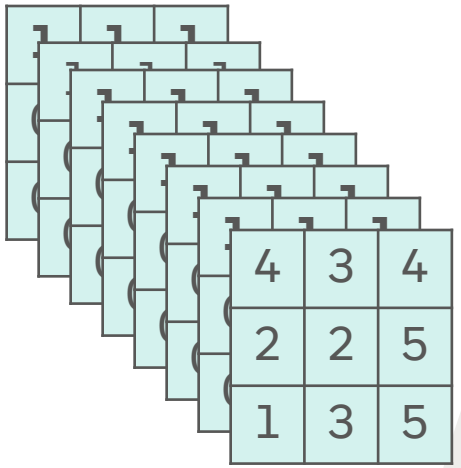
Convolved Feature/Activation Map/Feature Map

Filters

- Number of filters is an important parameter in the convolution layer.
- Considering a random number of 8 filters with filter size 3*3.
- Thus, number of weights = 3*3 (Filter size) * 8 (number of filters) = 72 weights

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

=>



Filters

— Vertical Sobel Filter

1	0	-1
2	0	-2
1	0	-1

Filters

— Horizontal Sobel Filter

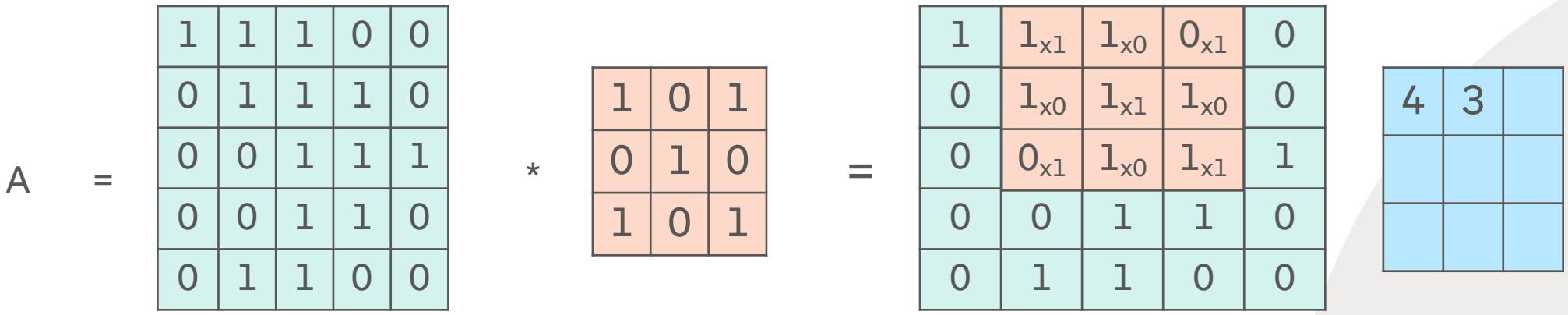
1	2	1
0	0	0
-1	-2	-1

Filters

- Instead of handpicking these weight values.
- Initialize these weight value with random number and use them as parameters and learn them during back propagation.
- Weight values can be very well learnt through backprop.

Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



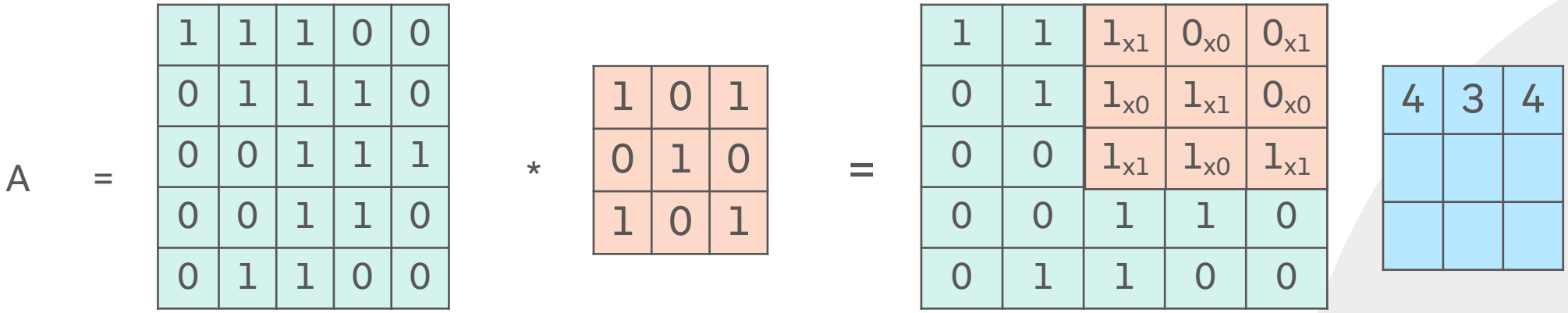
Input image

Filter/Kernel /Feature Detector

Convolved Feature/Activation Map/Feature Map

Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



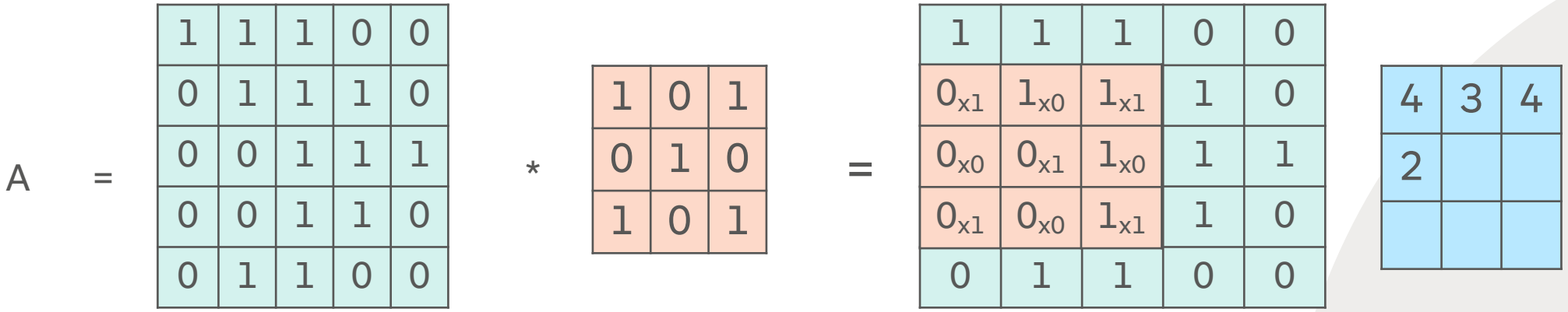
Input image

Filter/Kernel /Feature Detector

Convolved Feature/Activation Map/Feature Map

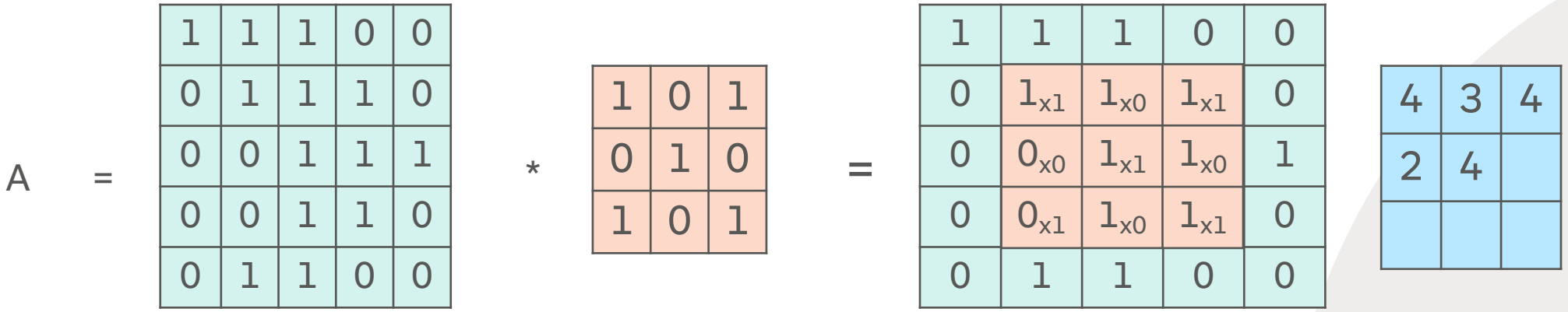
Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



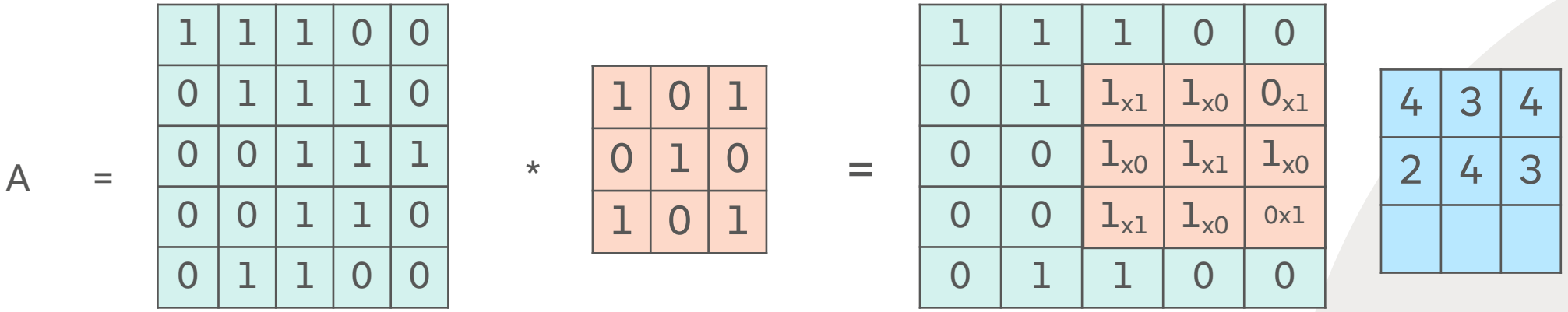
Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



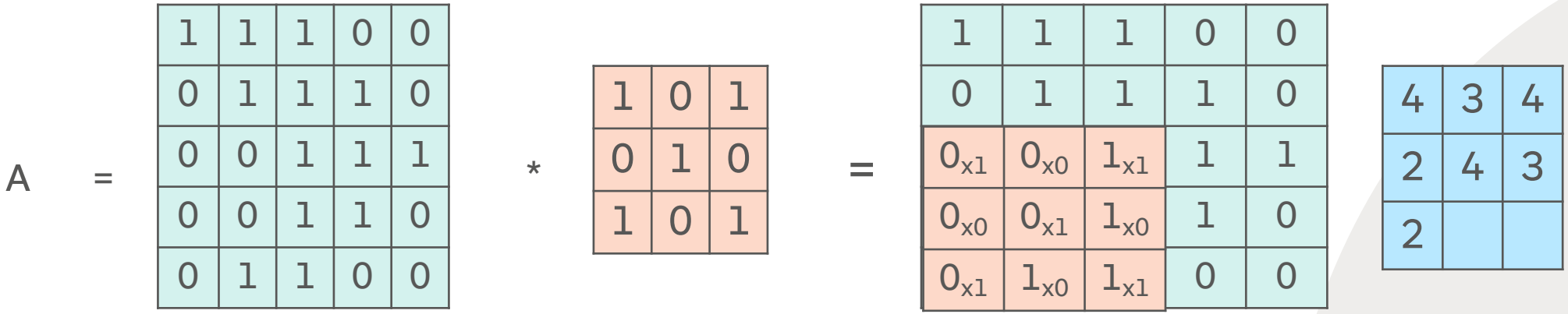
Input image

Filter/Kernel /Feature Detector

Convolved Feature/Activation Map/Feature Map

Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



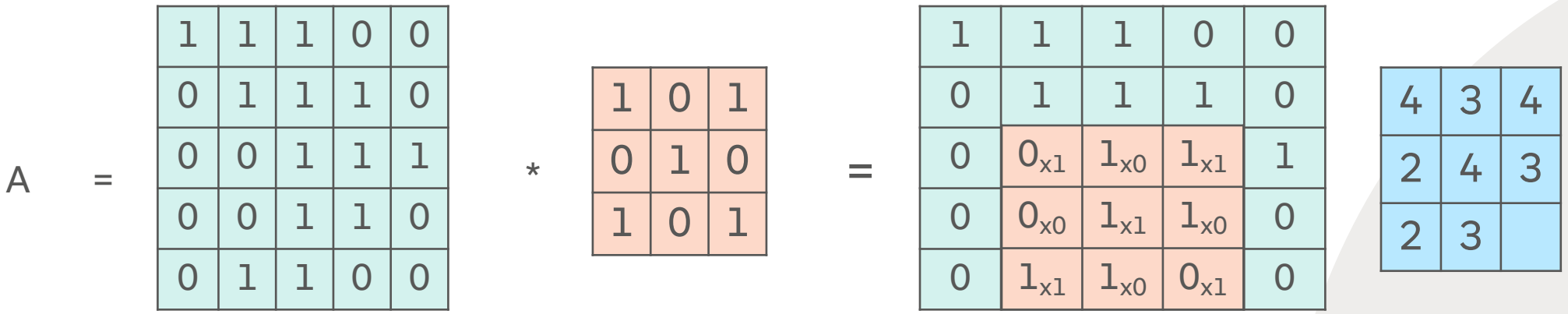
Input image

Filter/Kernel /Feature Detector

Convolved Feature/Activation Map/Feature Map

Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



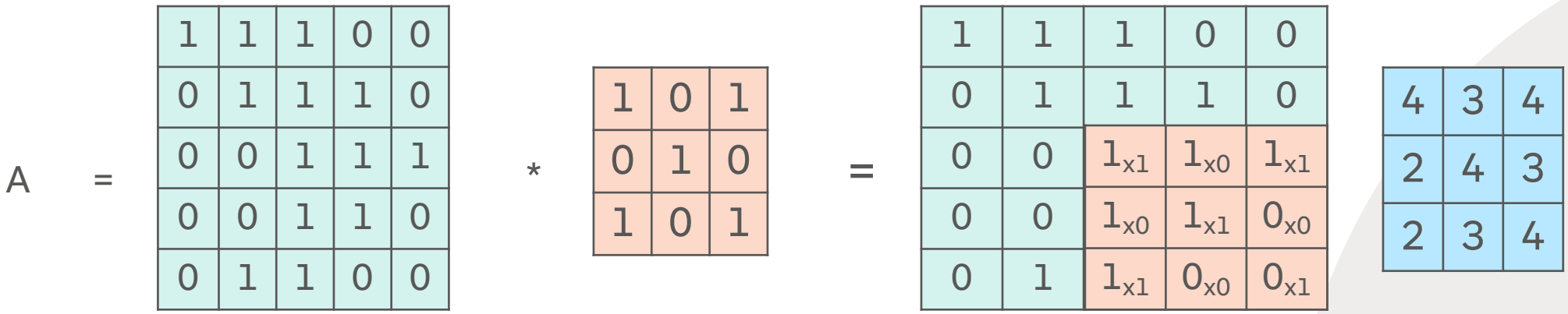
Input image

Filter/Kernel /Feature Detector

Convolved Feature/Activation Map/Feature Map

Convolution step

- Purpose of Convolution is to extract significant features from the input image
- Learns small features from the input data.



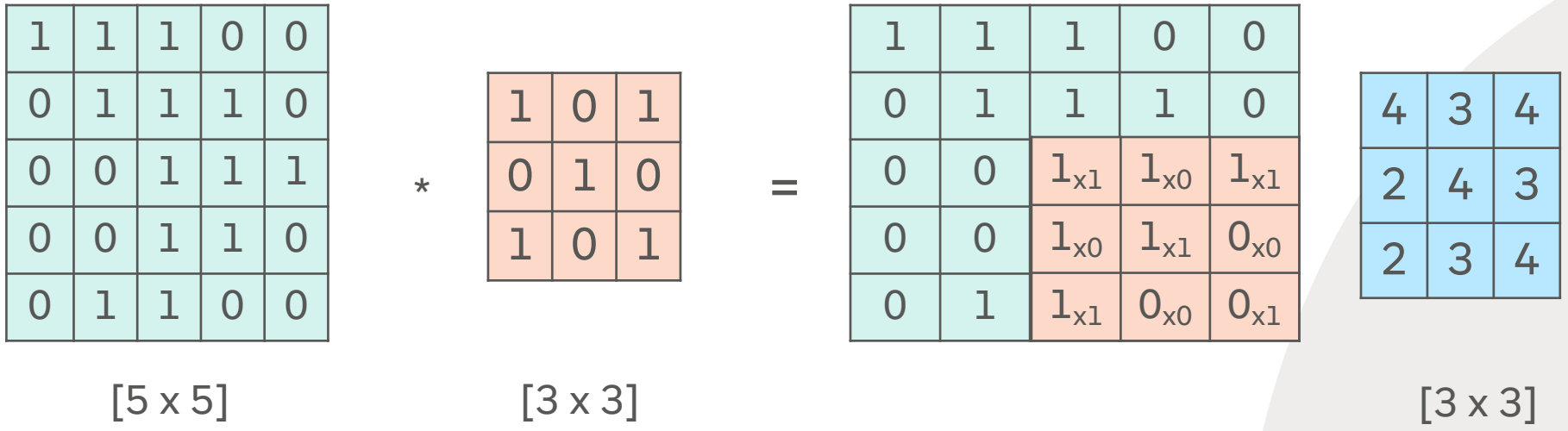
Input image

Filter/Kernel /Feature Detector

Convolved Feature/Activation Map/Feature Map

Convolution step

— Input Matrix [5 x 5] convolve with filter matrix [3 x 3] = Output Matrix [3 x 3]



Convolution step

— A generic formula can be considered:

$$[N \times N] * [f \times f] = [(N-f+1) \times (N-f+1)]$$

Where, $[N \times N]$ is the input image matrix dimension and $[f \times f]$ is the matrix dimension for the convolving filter.

Convolution step

Drawback 1:

- Drawback of convolving operation is that the Input Image dimension shrinks as there is reduction in the output matrix dimension.

  Example: [5 x 5] got reduced to [3 x 3]

- At the end of the neural network it might shrink to [1 x 1]
- **Image Shrinking is the important drawback**

Convolution step

Drawback 1:

- Drawback of convolving operation is that the Input Image dimension shrinks as there is reduction in the output matrix dimension.

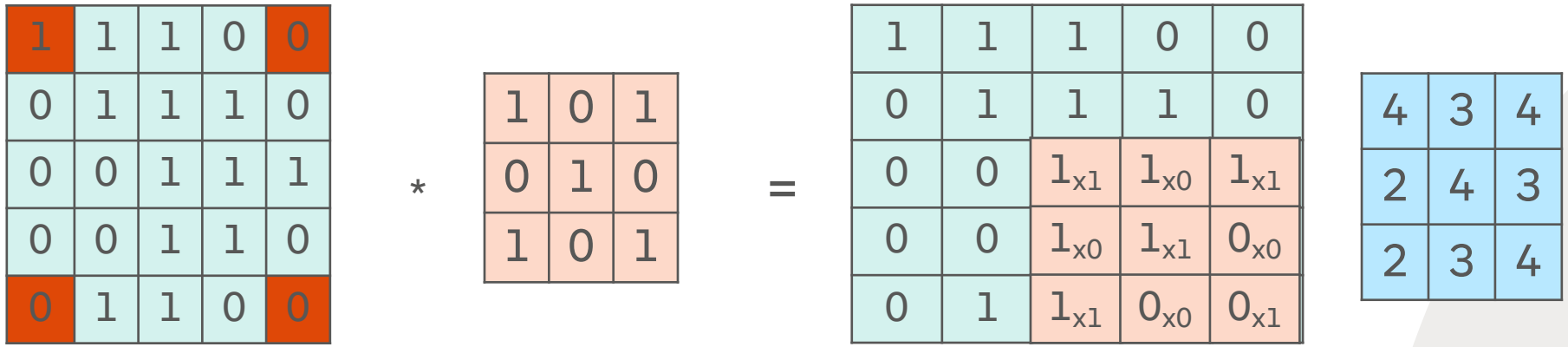
  Example: [5 x 5] got reduced to [3 x 3]

- At the end of the neural network it might shrink to [1 x 1]
- **Image Shrinking is the important drawback**
- **Ideal scenario:** Image Shrinking does not take place for every detection of edges or features.

Convolution step

Drawback 2:

— Pixel at the corners of the image is used only once.



[5 x 5] [3 x 3] [3 x 3]

Padding

— Shrinking output image and Pixel importance at the edge is overcome using **Padding**.

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

Padding

— Shrinking output image and Pixel importance at the edge is overcome using **Padding**.

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

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

Padding

- As an output we can get the original image size.

$$[N \times N] * [f \times f] = [(N-f+1) \times (N-f+1)]$$

Input image [5 x 5] padded with one pixel is converted into [7 x 7] matrix

- Output matrix = [(7 – 3 + 1) x (7 - 3 + 1)] = [5 x 5]
- The equation when padding is added is modified as follows:

$$[(N+2p) \times (N+2p)] * [f \times f] = [(N+2p-f+1) \times (N+2p-f+1)]$$

Where, N is the input image dimension, p is the padding pixel value, and f is the filter matrix dimension.

Valid Convolution / Same Convolution

Valid Convolution / Same Convolution

— **Valid Convolution**

$$[N \times N] * [f \times f] = [(N-f+1) \times (N-f+1)]$$

— **Same Convolution:** Padding is involved so that output size is the size as the input size.

$$[(N+2p) \times (N+2p)] * [f \times f] = [(N+2p-f+1) \times (N+2p-f+1)]$$

Valid Convolution / Same Convolution

What is the ideal value of Padding pixel?

Valid Convolution / Same Convolution

What is the ideal value of Padding pixel?

— Solution:

$$(N+2p-f+1)=N$$

$$p= ((f-1))/2$$

Therefore, filter size value (f) is always **ODD** number.

Because f is **EVEN** then asymmetric padding, then **NO** central position in the image matrix.

Strided Convolution

- Stride = Hoping of the filter over the input image = 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

*

1	0	1
0	1	0
1	0	1

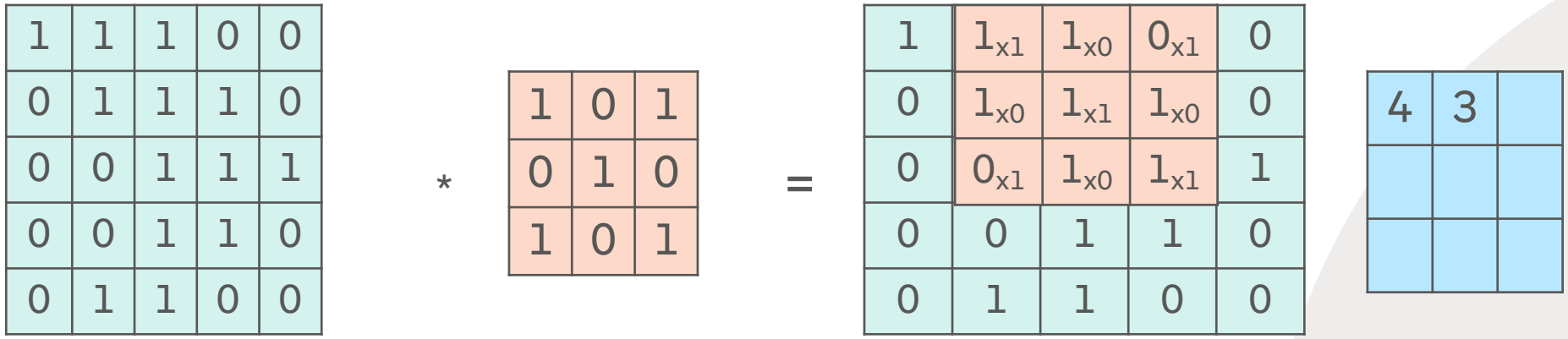
=

1_{x1}	1_{x0}	1_{x1}	0	0
0_{x0}	1_{x1}	1_{x0}	1	0
0_{x1}	0_{x0}	1_{x1}	1	1
0	0	1	1	0
0	1	1	0	0

4		

Strided Convolution

— Stride = Hoping of the filter over the input image = 1



Strided Convolution

— Stride = Hoping of the filter over the input image = 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

*

1	0	1
0	1	0
1	0	1

=

1	1	1_{x1}	0_{x0}	0_{x1}
0	1	1_{x0}	1_{x1}	0_{x0}
0	0	1_{x1}	1_{x0}	1_{x1}
0	0	1	1	0
0	1	1	0	0

4	3	4

Strided Convolution

— Stride = Hoping of the filter over the input image = 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

*

1	0	1
0	1	0
1	0	1

=

1	1	1	0	0
0_{x1}	1_{x0}	1_{x1}	1	0
0_{x0}	0_{x1}	1_{x0}	1	1
0_{x1}	0_{x0}	1_{x1}	1	0
0	1	1	0	0

4	3	4
2		

Strided Convolution

— Stride = Hoping of the filter over the input image = 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

*

1	0	1
0	1	0
1	0	1

=

1	1	1	0	0
0	1_{x1}	1_{x0}	1_{x1}	0
0	0_{x0}	1_{x1}	1_{x0}	1
0	0_{x1}	1_{x0}	1_{x1}	0
0	1	1	0	0

4	3	4
2	4	

Strided Convolution

— Stride = Hoping of the filter over the input image = 2

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

=

1_{x1}	1_{x0}	1_{x1}	0	0
0_{x0}	1_{x1}	1_{x0}	1	0
0_{x1}	0_{x0}	1_{x1}	1	1
0	0	1	1	0
0	1	1	0	0

4	4

Strided Convolution

— Stride = Hoping of the filter over the input image = 2

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

=

1	1	1_{x1}	0_{x0}	0_{x1}
0	1	1_{x0}	1_{x1}	0_{x0}
0	0	1_{x1}	1_{x0}	1_{x1}
0	0	1	1	0
0	1	1	0	0

4	4

Strided Convolution

— Stride = Hoping of the filter over the input image = 2

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

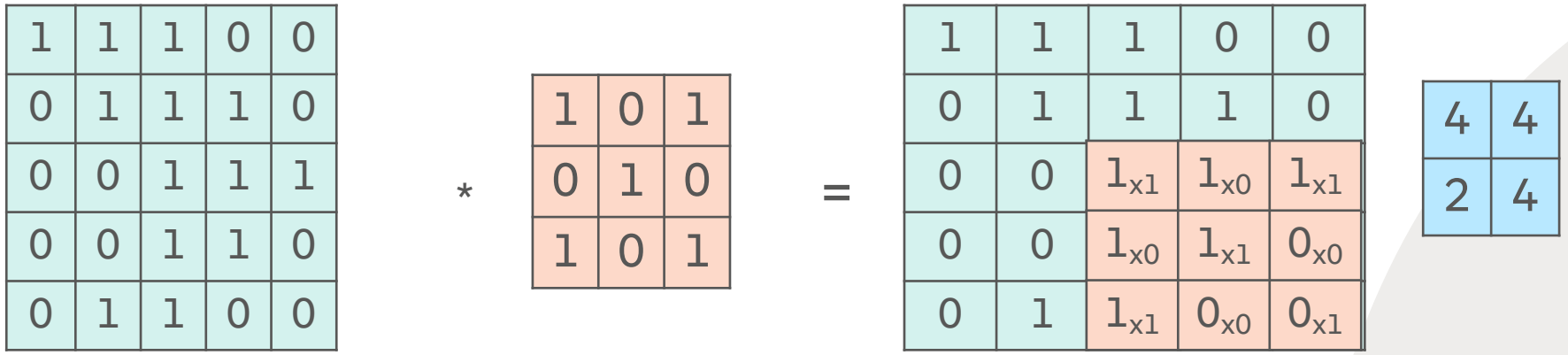
=

1	1	1	0	0
0	1	1	1	0
0_{x1}	0_{x0}	1_{x1}	1	1
0_{x0}	0_{x1}	1_{x0}	1	0
0_{x1}	1_{x0}	1_{x1}	0	0

4	4
2	

Strided Convolution

— Stride = Hoping of the filter over the input image = 2



Strided Convolution

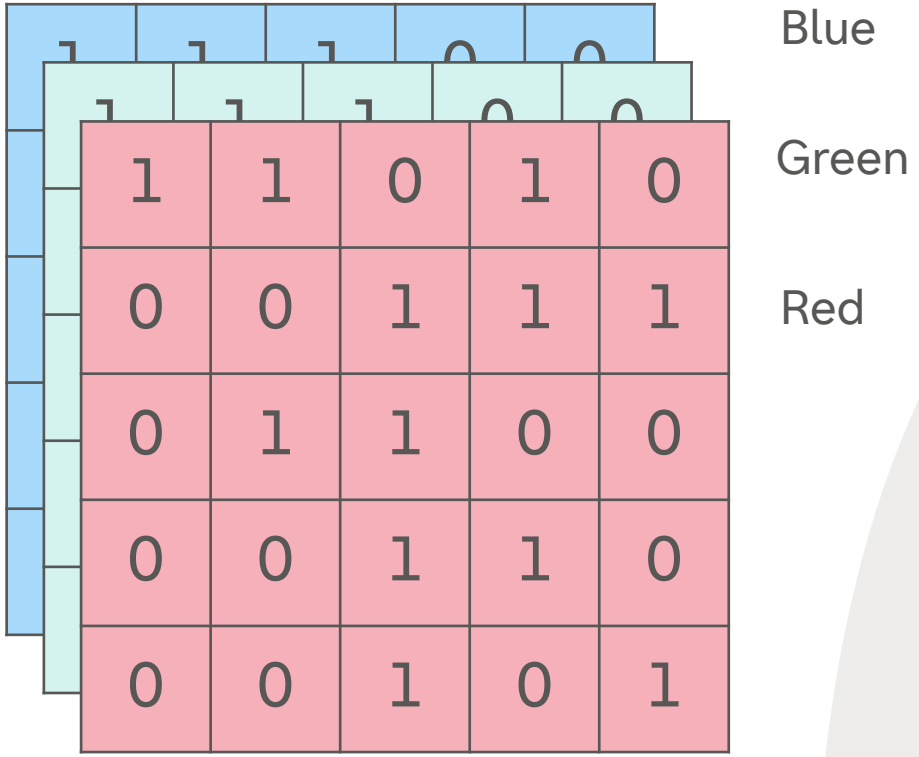
- Combining all the concepts learnt till now:
  - If the input image is $[N \times N]$ and filter size is $[f \times f]$, where p is the padding pixel value and s is the stride value then the final equation looks like:

$$[(N+2p) \times (N+2p)] * [f \times f] = [((N+2p-f)/s)+1) \times ((N+2p-f)/s)+1)]$$

  - If the stride makes the filter goes outside, then the computation is not taken place.

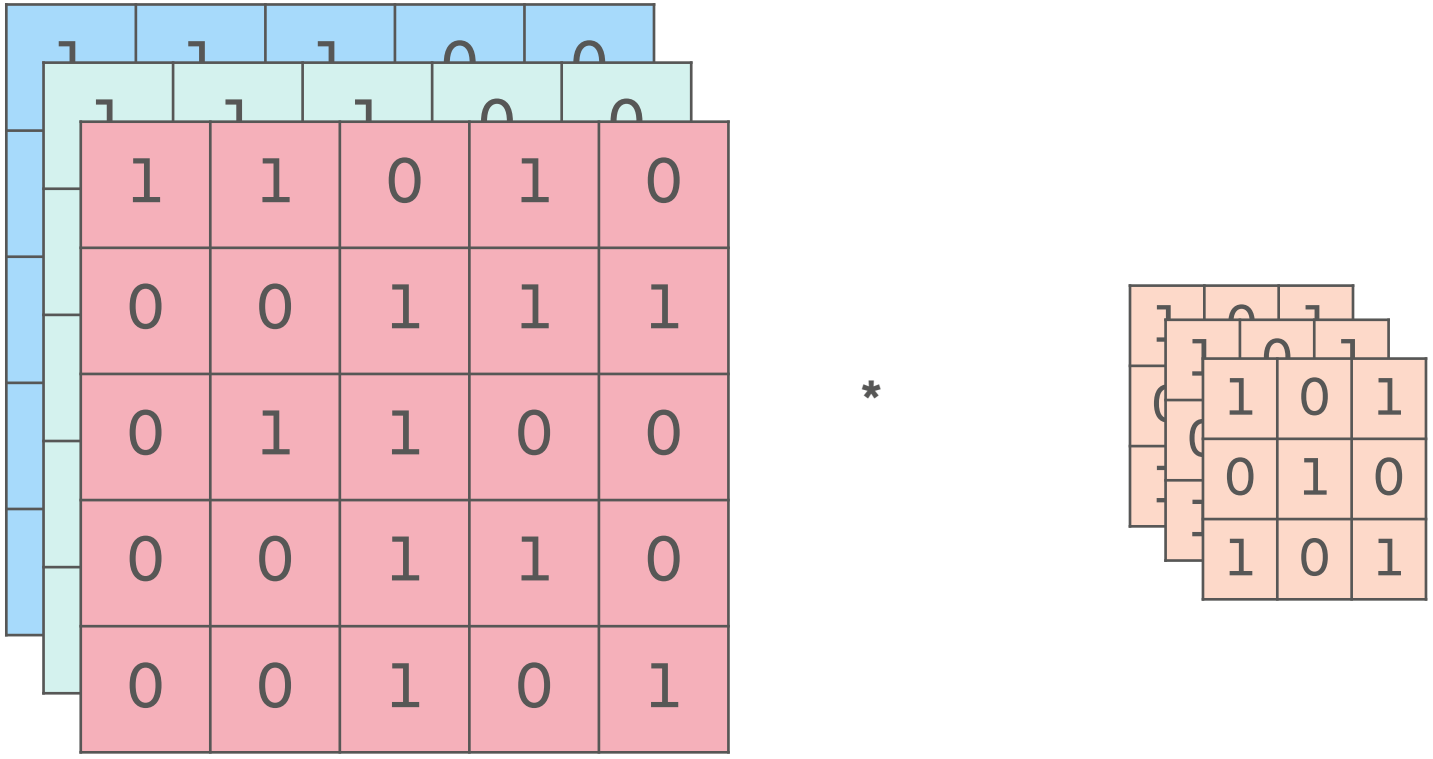
Convolution on RGB Images

— [5 x 5 x 3] => 5 is the height and width and 3 are the number of channels (RGB)



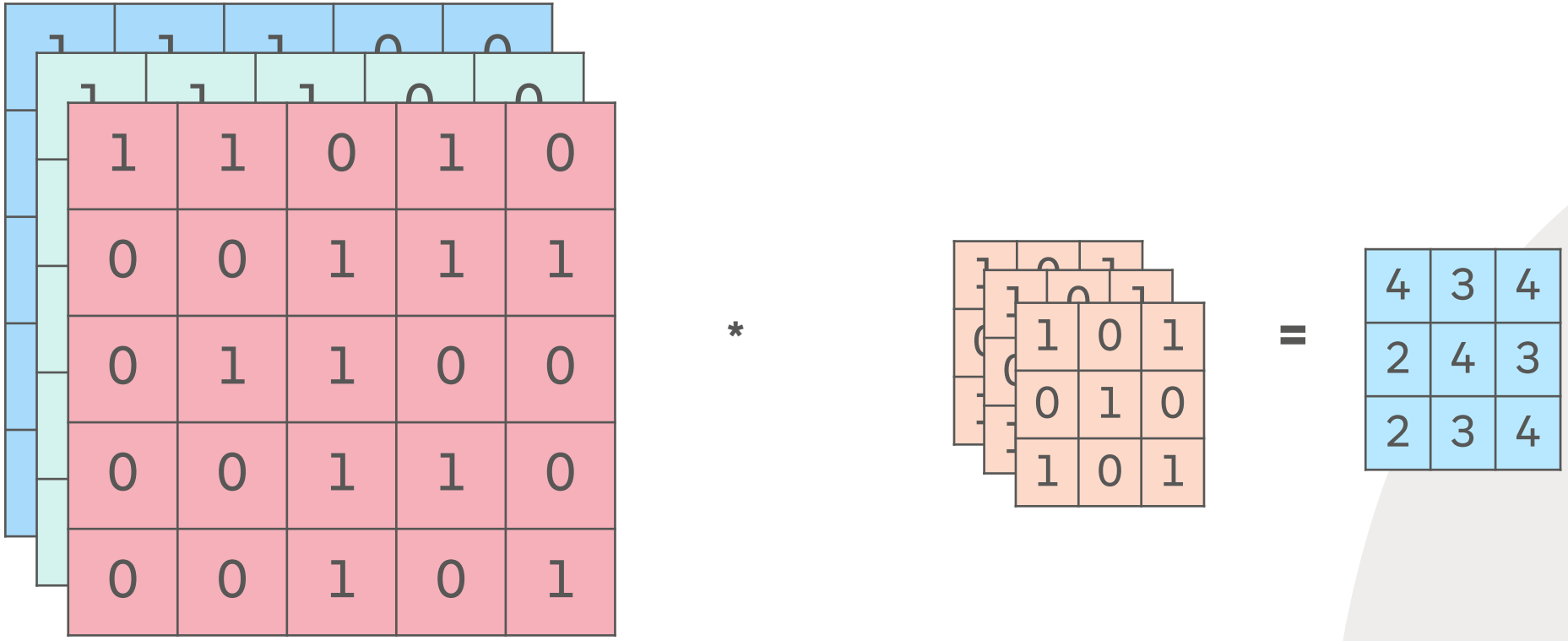
Convolution on RGB Images

— [5 x 5 x 3] * [3 x 3 x 3] => 3 is the height and width and 3 are the number of channels (RGB)



Convolution on RGB Images

— Output will be [3 x 3 x 1] that is calculated by multiplying each filter with the respective channel matrix and sum them up to calculate the value.

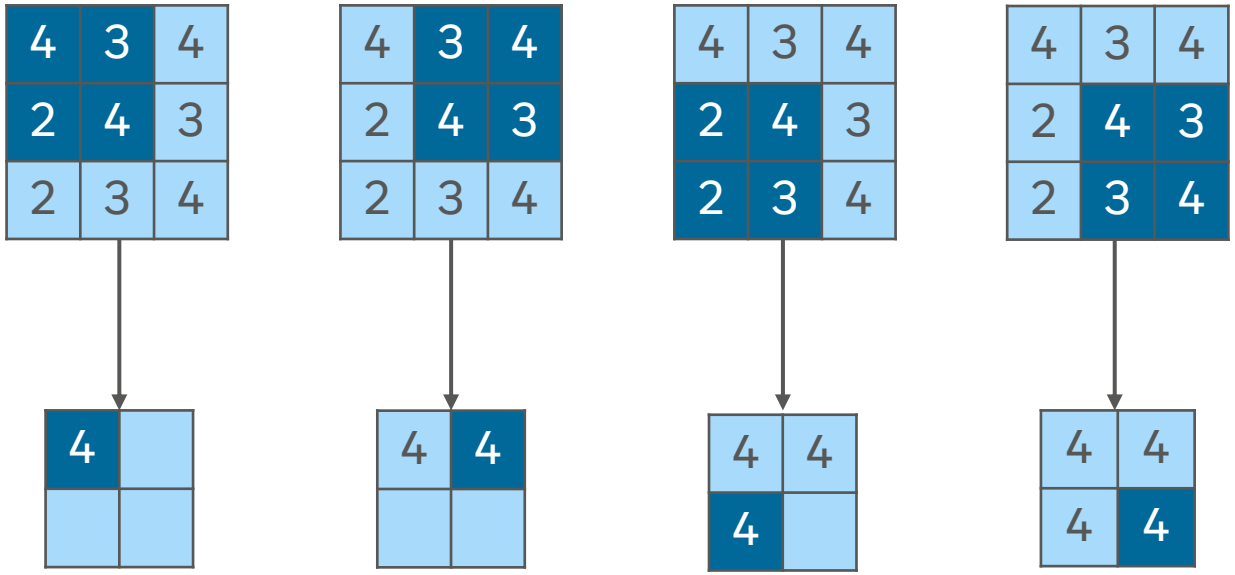


How Convolutional Neural Network Work?

- Convolution Layer
- **Pooling or Sub sampling**
- Flatenning
- Non-Linearity (ReLU, Tanh)
- Classification or Fully Connected Layer

Pooling (Sub sampling/Down sampling)

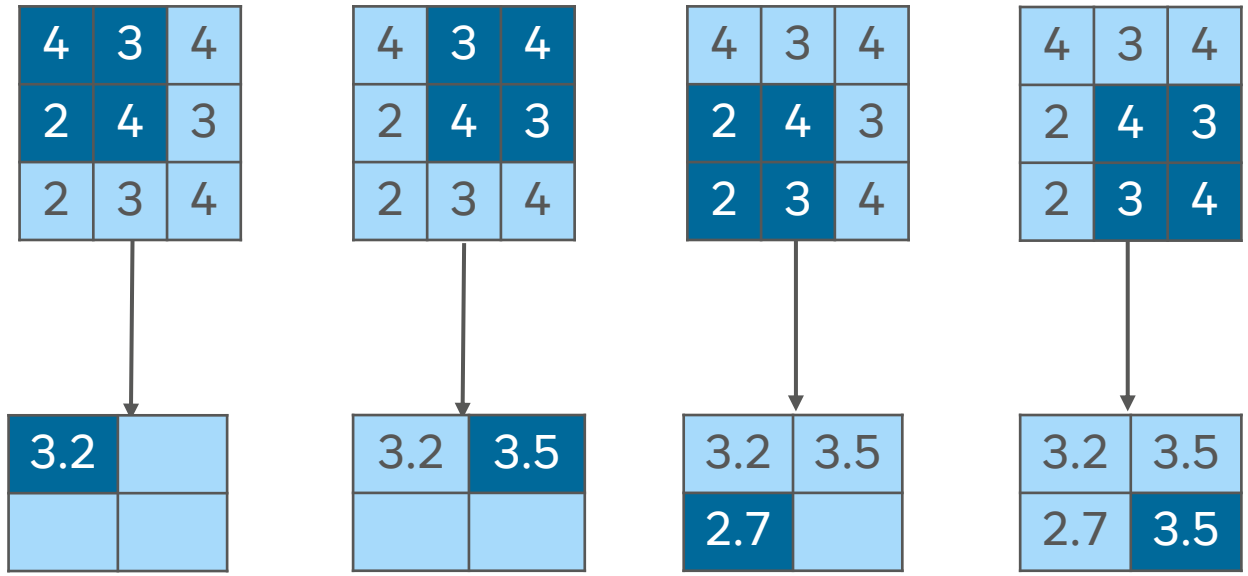
- Reduces the dimensionality of the feature map but also retains the most significant information.
- Different types of Pooling techniques based on decided pool size.



Max Pooling

Pooling (Sub sampling/Down sampling)

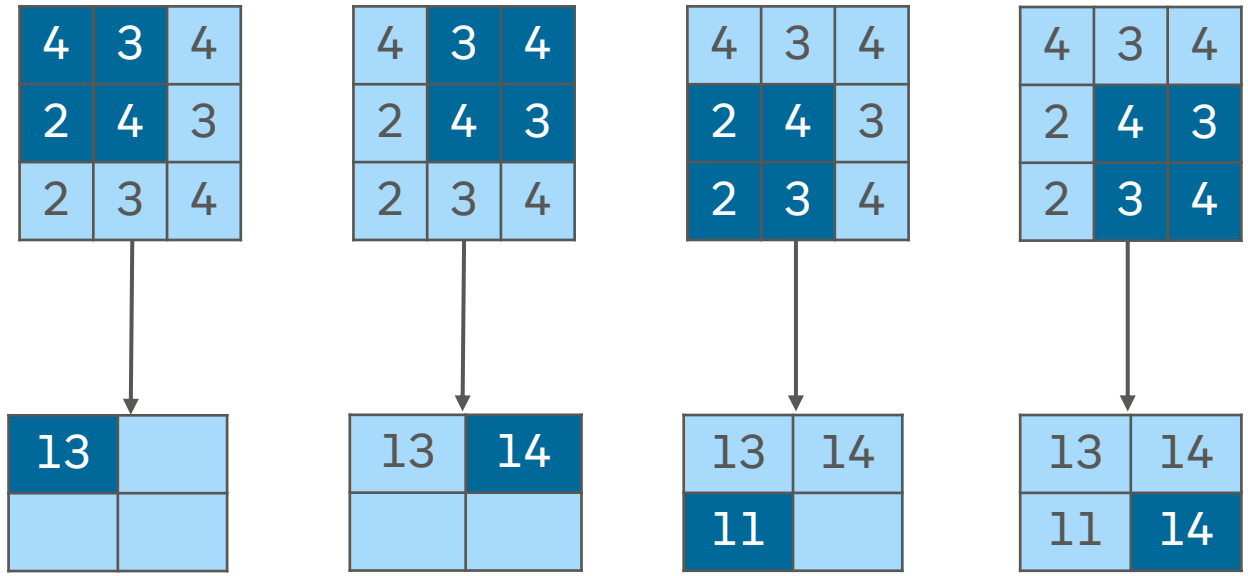
- Reduces the dimensionality of the feature map but also retains the most significant information.
- Different types of Pooling techniques based on decided pool size.



Average Pooling

Pooling (Sub sampling/Down sampling)

- Reduces the dimensionality of the feature map but also retains the most significant information.
- Different types of Pooling techniques based on decided pool size.



Sum Pooling

Pooling (Sub sampling/Down sampling)

- Most popular is **Max Pooling**.
- Advantages of Pooling is to makes the input representations (Feature dimension) smaller and more manageable.
- Reduces the number of parameters and computations in the network, therefore controlling overfitting.

How Convolutional Neural Network Work?

- Convolution Layer
- Pooling or Sub sampling
- **Flattening**
- Non-Linearity (ReLU, Tanh)
- Classification or Fully Connected Layer

Flattening

- After the convolution and pooling is performed, a final matrix is obtained.
- The obtained matrix can now be passed to the Fully connected layer for the classification purpose.
- As matrix cannot be passed directly to the Fully connected layer, Flattening is performed.
- Flatten is the concept of converting the matrix into a 1-D array value.

How Convolutional Neural Network Work?

- Convolution Layer
- Pooling or Sub sampling
- Flattening
- **Non-Linearity (ReLU, Tanh)**
- Classification or Fully Connected Layer

Non-linearity (ReLU)

- Rectified Linear Unit (Non-linear operation)
- Element wise operation
- Non-linearity in the neural network
- Convolution is a linear operation.

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \ * g[x-n_1\ , y-n_2]$$

How Convolutional Neural Network Work?

- Convolution Layer
- Pooling or Sub sampling
- Flattening
- Non-Linearity (ReLU, Tanh)
- **Classification or Fully Connected Layer**

Fully Connected Layer

- Multi Layer Perceptron is used as fully connected layer, SoftMax function at output layer
- Fully connected layer means every neuron from one layer is connected to each neuron of another layer.
- The output from convolution and pooling represent the high level feature of the input image.

Fully Connected Layer

- Fully connected layer uses these features in order to perform classification into various classes based on training dataset.
- Dropout is the process of dropping out some neurons from the neural networks during the training process. This dropout prevents the neural network from overfitting.

Convolutional Neural Network

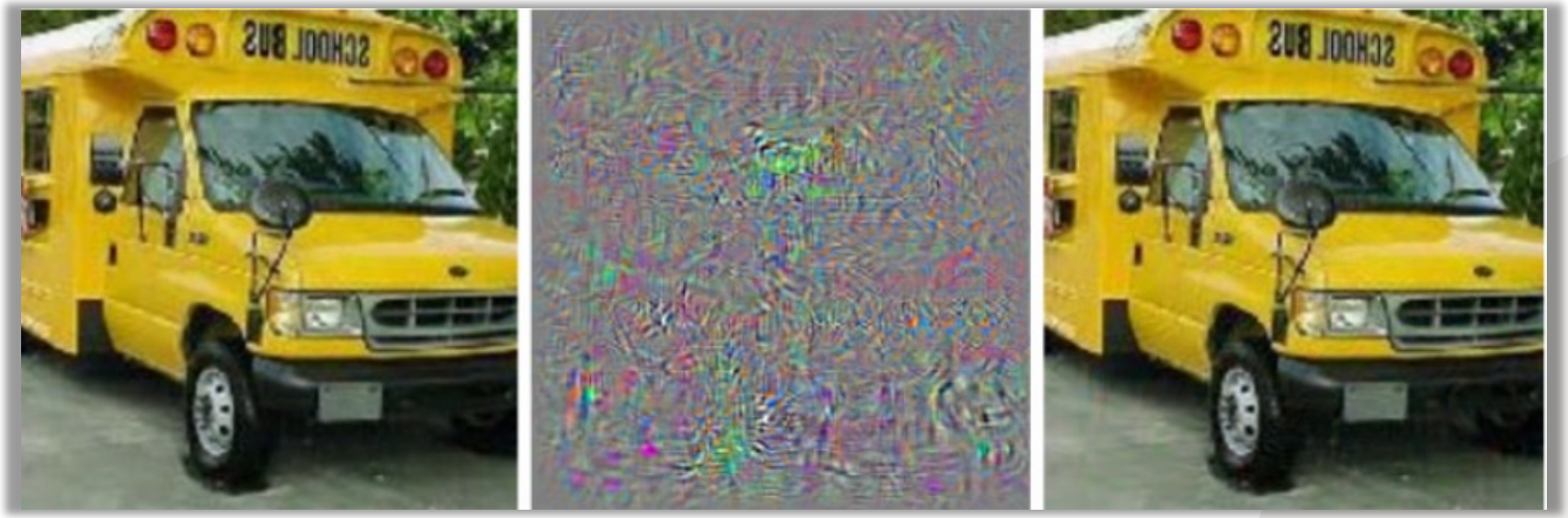
- Convolution and Pooling layer helps to identifying features from the image. (Feature Learning)
- Fully connected layer considers these features to classify into different categories. (Classification)
- Training dataset is large then Neural network will learn and generalize well to new images.
- Jonathan Huan et al. **“Speed/accuracy trade-offs for modern convolutional object detectors”**
  - Guide on how to select the right speed memory accuracy balance for a given application and platform

Hyper-Parameters of CNN

- Input Size
- Padding
- Kernel Size
- Stride
- Number of filters or Kernel

Drawbacks of Convolutional neural network

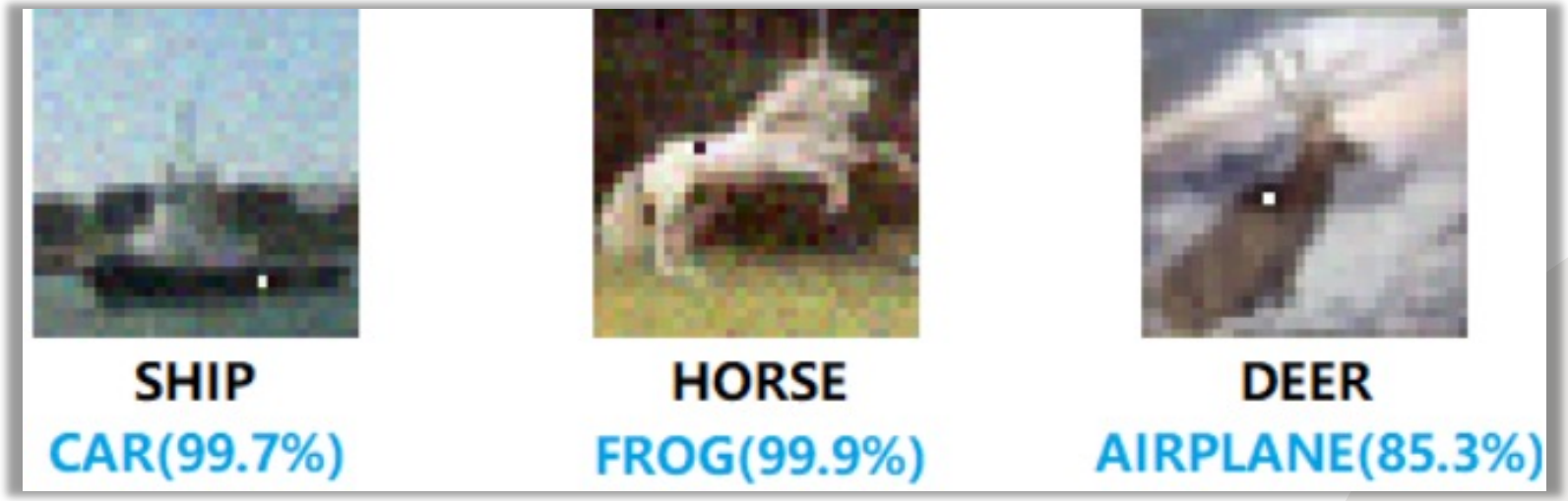
- As CNN are supervised learning method, require a large set of label data for training.
- CNN can be defeated by carefully adding a crafted noise into the image.



Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., 2013. **Intriguing properties of neural networks**. *arXiv preprint arXiv:1312.6199*.

Drawbacks of Convolutional neural network

— One Pixel Attack



Su, J., Vargas, D.V. and Sakurai, K., 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, *23*(5), pp.828-841.

Github: https://github.com/Hyperparticle/one-pixel-attack-keras

Convolutional Neural Network Explainer

URL:

https://poloclub.github.io/cnn-explainer/

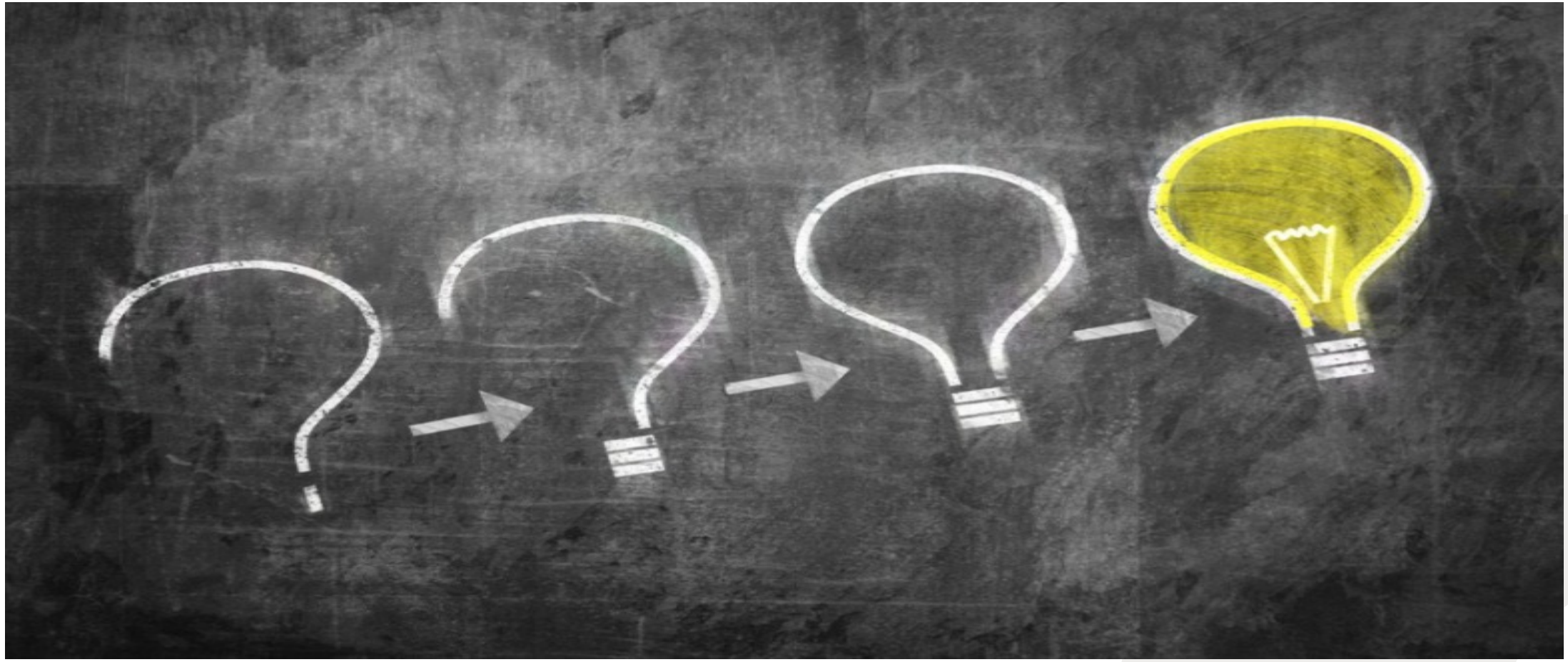
Relevant Paper:

Wang, Z.J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M. and Chau, D.H., 2020. **CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization.** *arXiv preprint arXiv:2004.15004*.

Github:

https://github.com/poloclub/cnn-explainer

Questions?



Vielen Dank für deine Aufmerksamkeit!

Kontakt:
Ashish Chouhan
SRH Hochschule Heidelberg
Ludwig-Guttmann-Straße 6
69123 Heidelberg
Phone: +49 6221 6799-224
Mail ID: ashish.chouhan@srh.de

Ajinkya Patil
Mail ID: ajinkya.patil.extern@srh.de
ajinkya.patil@sap.com