NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

Lecture 36: CNN Architecture

CONCEPTS COVERED

Concepts Covered:

- CNN
  - CNN Architecture
  - Convolution Layer
  - Receptive Field
  - Nonlinearity
  - Pooling

Convolution

1 D Convolution

$$y(n)=\sum_{p=0}^{\infty}x(p)h(n-p)$$

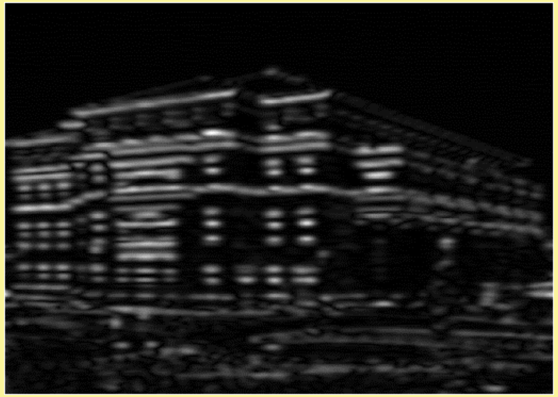
$$y(t)=\int_{0}^{\infty}x(\tau)h(t-\tau)d\tau$$

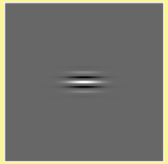
2 D Convolution

$$y(m,n)=\sum_{p=0}^{\infty}\sum_{q=0}^{\infty}x(p,q)h(m-p,n-q)$$

Finite Convolution Kernel

Feature at a point is local in nature





Convolution Kernel

1 D → 2A+1

$$y(n) = \sum_{p=-A}^{A} w(p)x(n-p)$$

2 D → (2A+1)x(2A+1)

$$y(m,n) = \sum_{p=-A}^{A}\sum_{q=-A}^{A} w(p,q)x(m-p,n-q)$$

Finite Convolution Kernel

0	0	X(0)	X(1)	X(2)	X(3)	.	X(n-2)	X(n-1)	X(n)	X(n+1)	X(n+2)	.
W(2)	W(1)	W(0)	W(-1)	W(-2)								
		Y(0)										

Finite Convolution Kernel

0	0	X(0)	X(1)	X(2)	X(3)	.	X(n-2)	X(n-1)	X(n)	X(n+1)	X(n+2)	.
	W(2)	W(1)	W(0)	W(-1)	W(-2)							
		Y(0)	Y(1)									

Finite Convolution Kernel

0	0	X(0)	X(1)	X(2)	X(3)	.	X(n-2)	X(n-1)	X(n)	X(n+1)	X(n+2)	.
		W(2)	W(1)	W(0)	W(-1)	W(-2)						
		Y(0)	Y(1)	Y(2)								

Finite Convolution Kernel

0	0	X(0)	X(1)	X(2)	X(3)	.	X(n-2)	X(n-1)	X(n)	X(n+1)	X(n+2)	.
			W(2)	W(1)	W(0)	W(-1)	W(-2)					
		Y(0)	Y(1)	Y(2)	Y(3)							

Finite Convolution Kernel

0	0	X(0)	X(1)	X(2)	X(3)	.	X(n-2)	X(n-1)	X(n)	X(n+1)	X(n+2)	.
						W(2)	W(1)	W(0)	W(-1)	W(-2)		
		Y(0)	Y(1)	Y(2)	Y(3)			Y(n-1)				

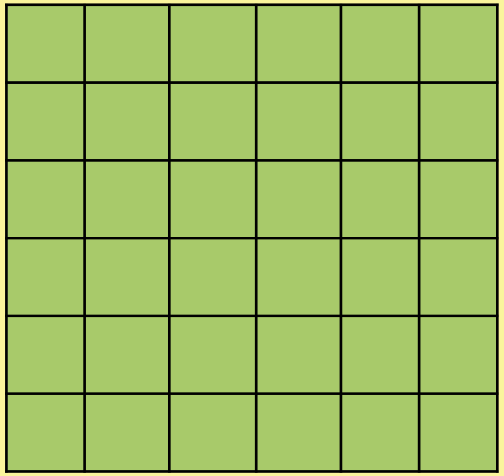
Finite Convolution Kernel

0	0	X(0)	X(1)	X(2)	X(3)	.	X(n-2)	X(n-1)	X(n)	X(n+1)	X(n+2)	.
							W(2)	W(1)	W(0)	W(-1)	W(-2)	
		Y(0)	Y(1)	Y(2)	Y(3)			Y(n-1)	Y(n)			

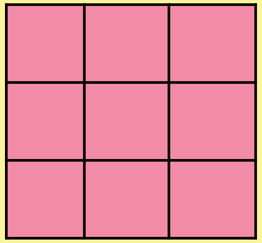
Finite Convolution Kernel

0	0	X(0)	X(1)	X(2)	X(3)	.	X(n-2)	X(n-1)	X(n)	X(n+1)	X(n+2)	.
								W(2)	W(1)	W(0)	W(-1)	W(-2)
		Y(0)	Y(1)	Y(2)	Y(3)			Y(n-1)	Y(n)	Y(n+1)		

2 D Convolution

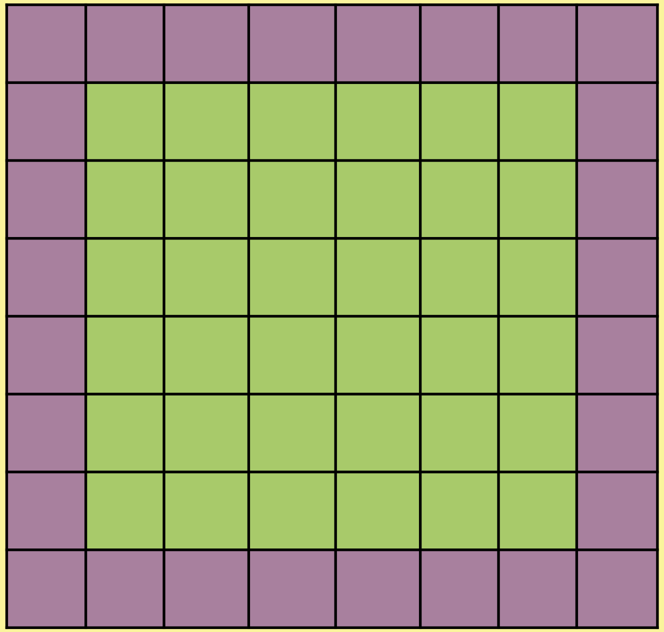


6 x 6 Image

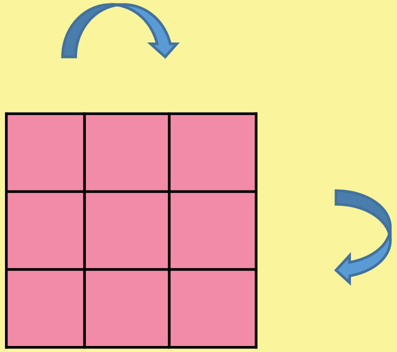


3 x 3 Kernel

2 D Convolution

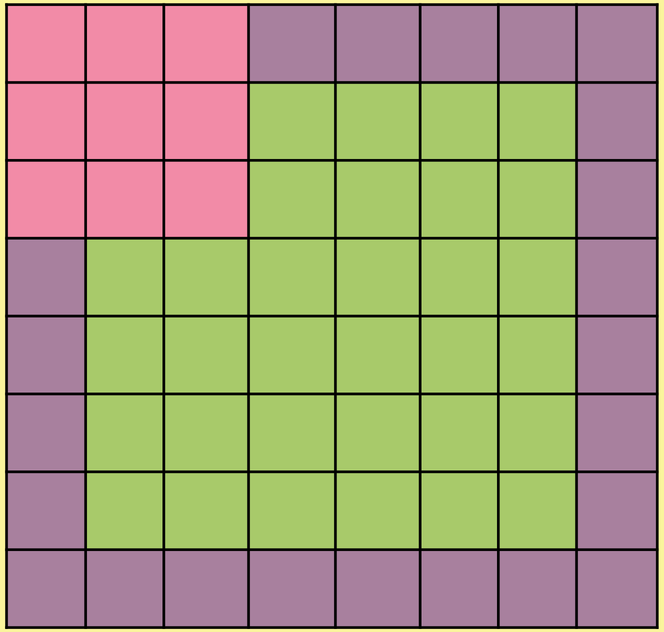


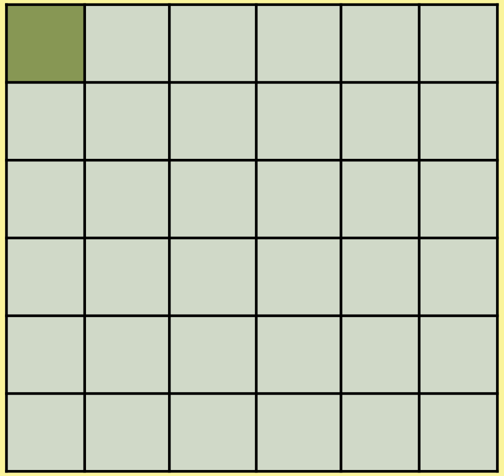
0 Padding



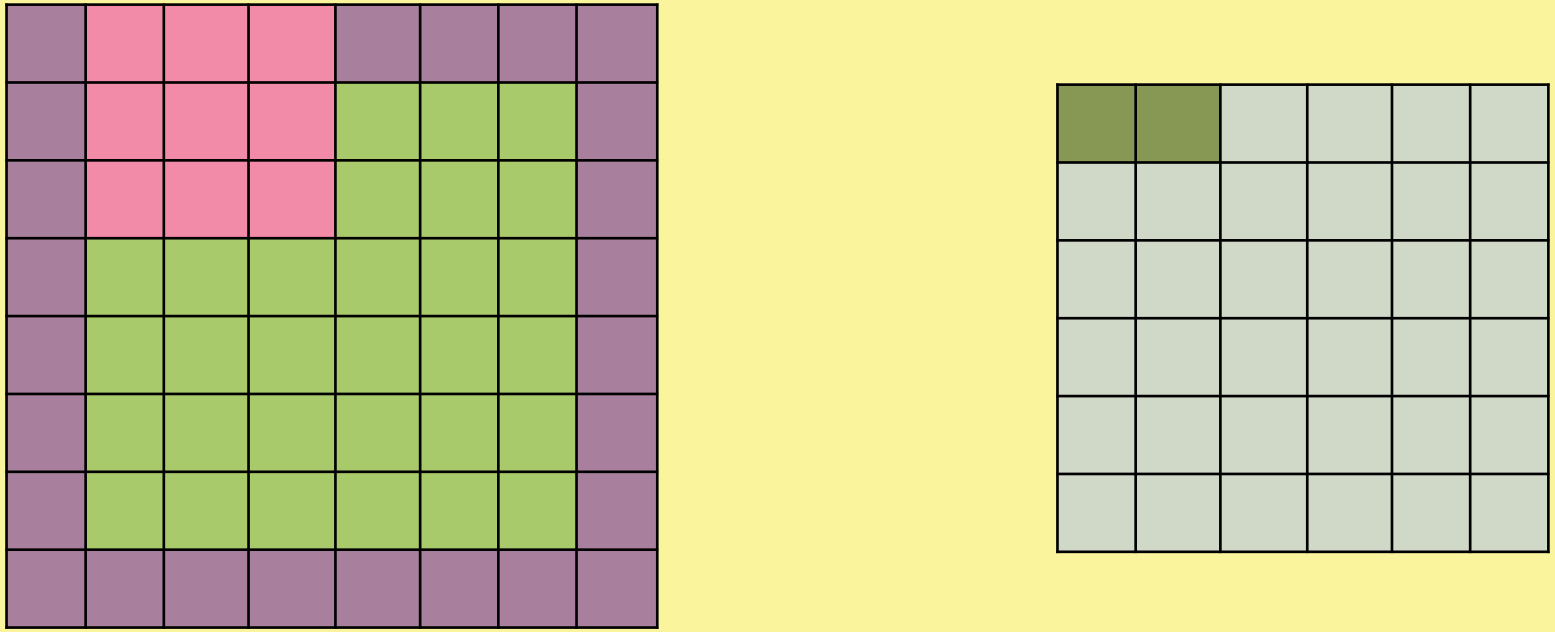
Flipping

2 D Convolution

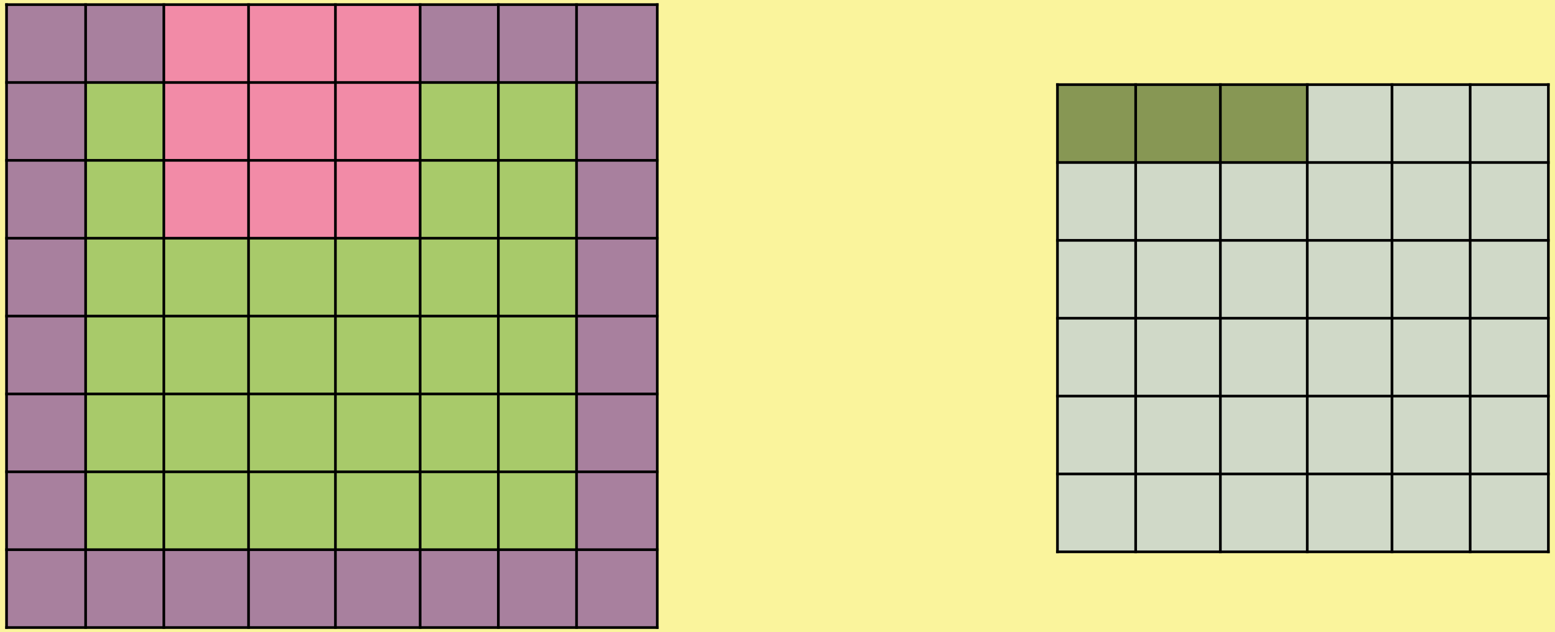




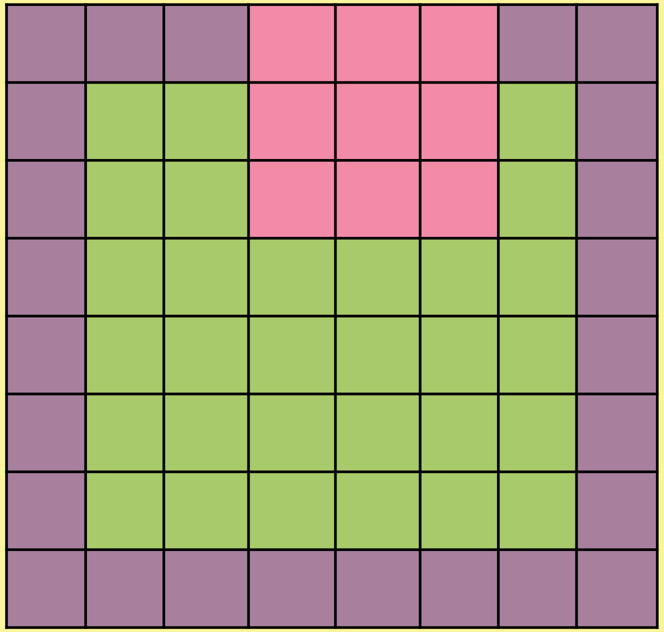
2 D Convolution

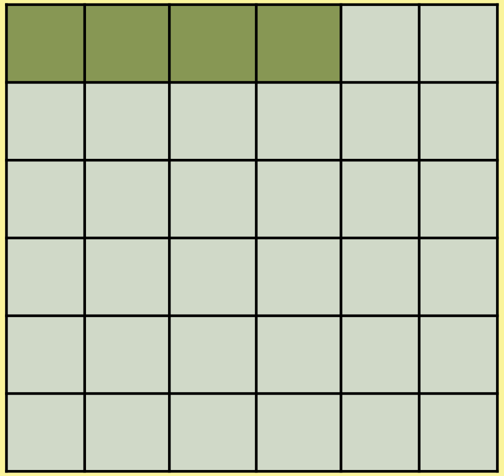


2 D Convolution

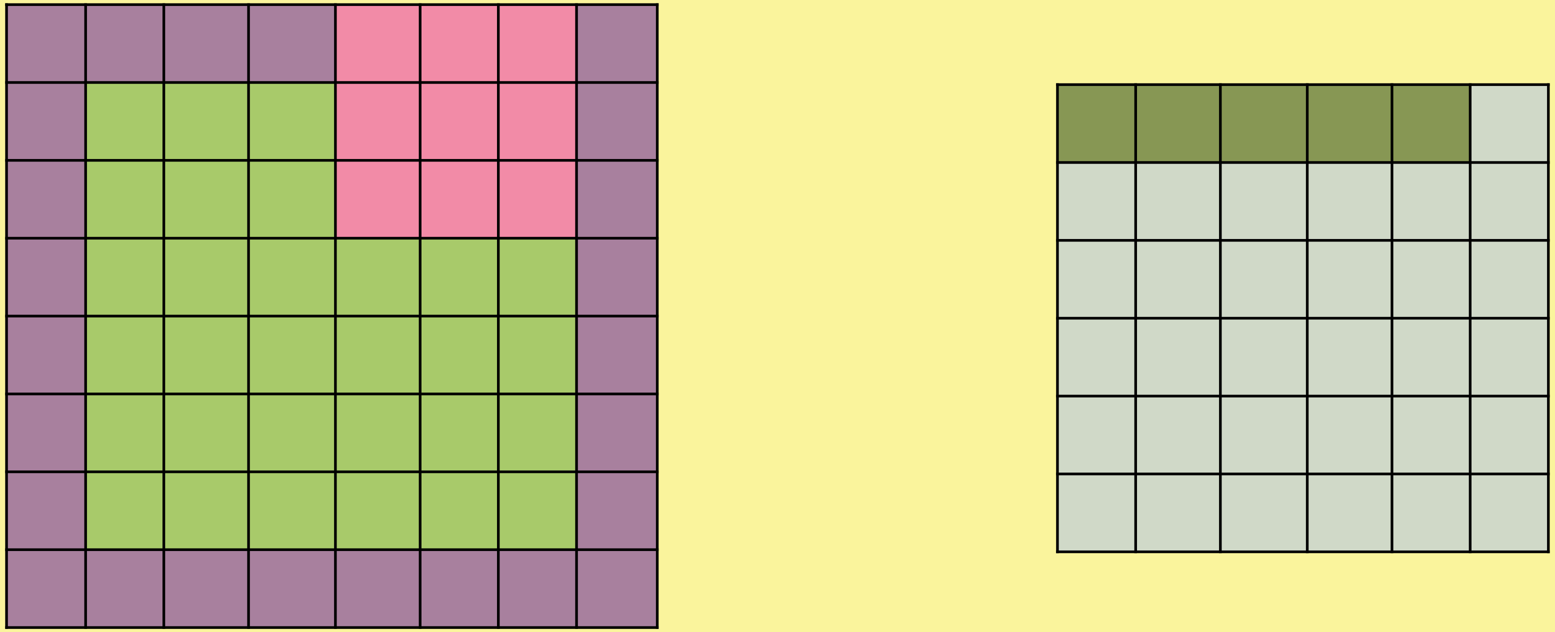


2 D Convolution

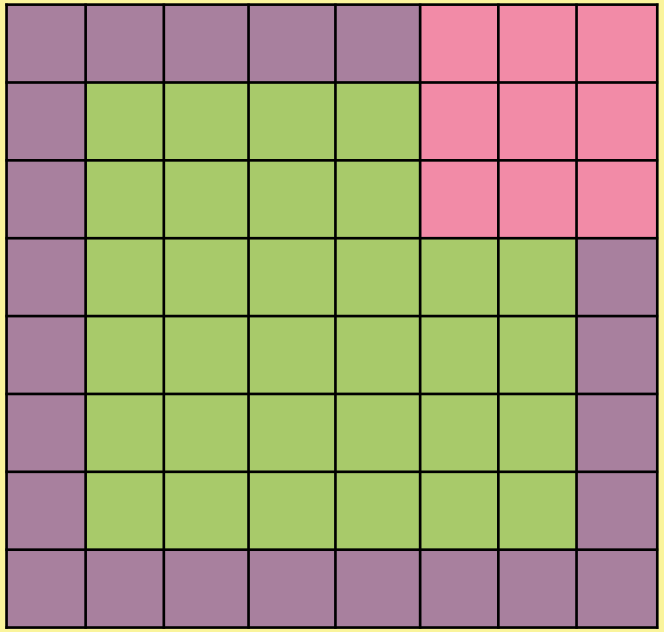


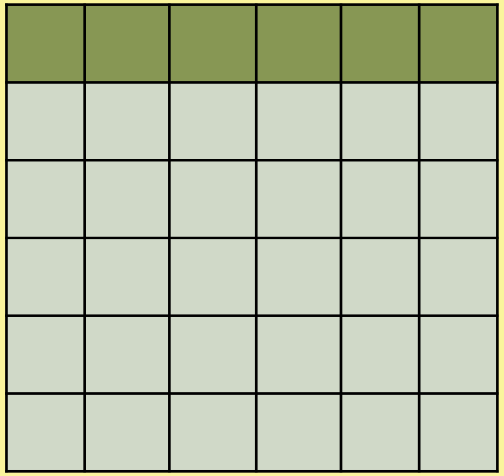


2 D Convolution

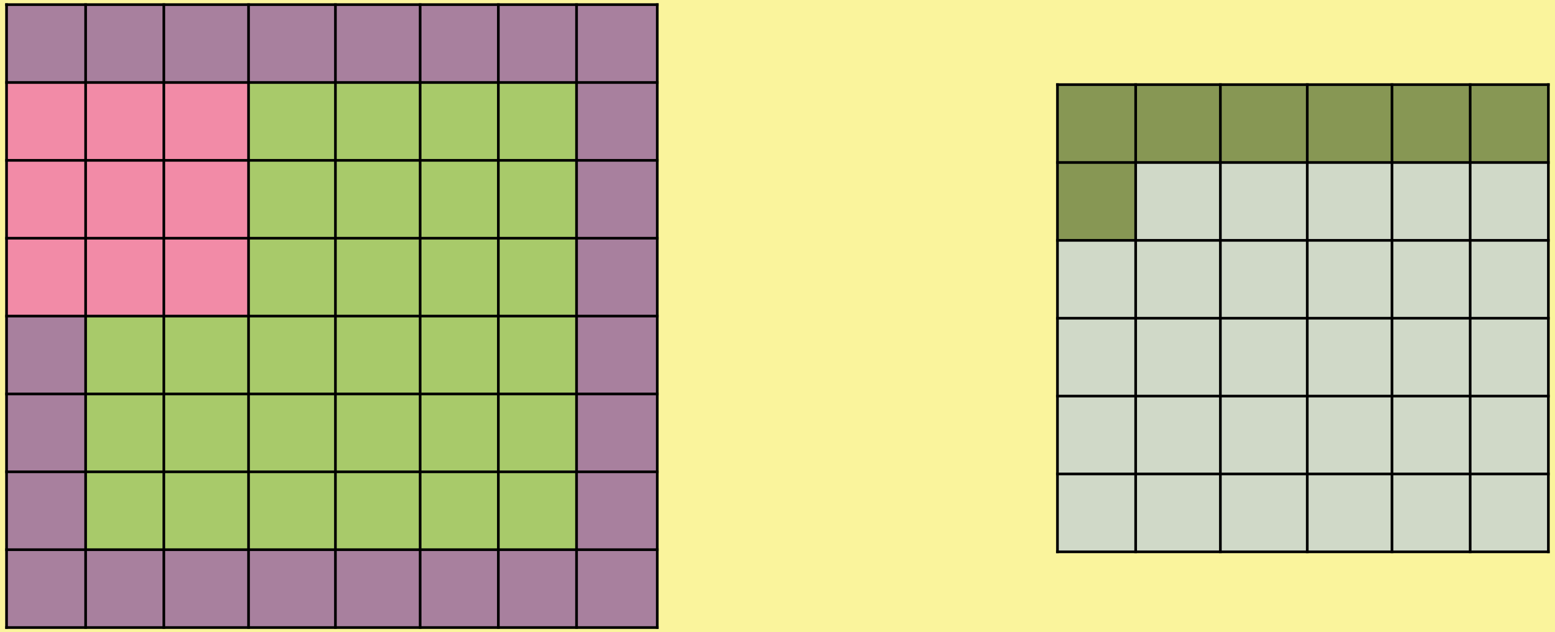


2 D Convolution

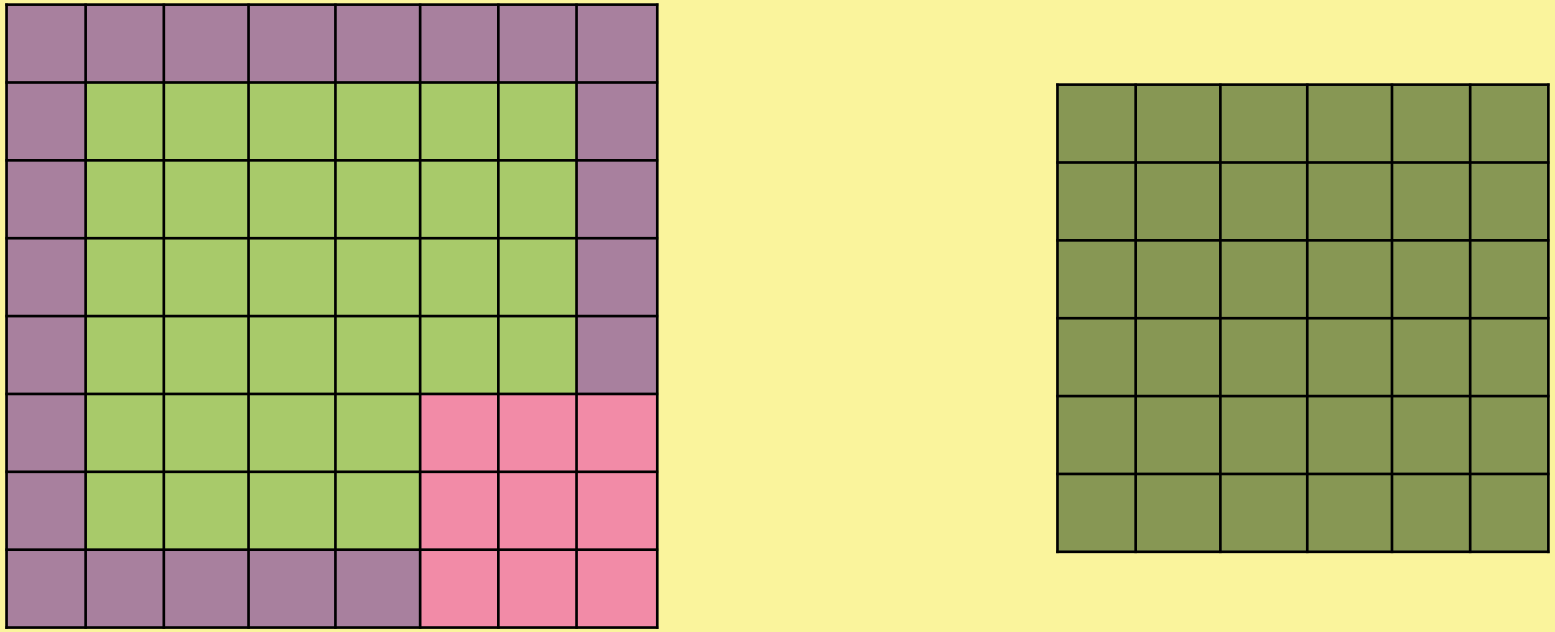




2 D Convolution

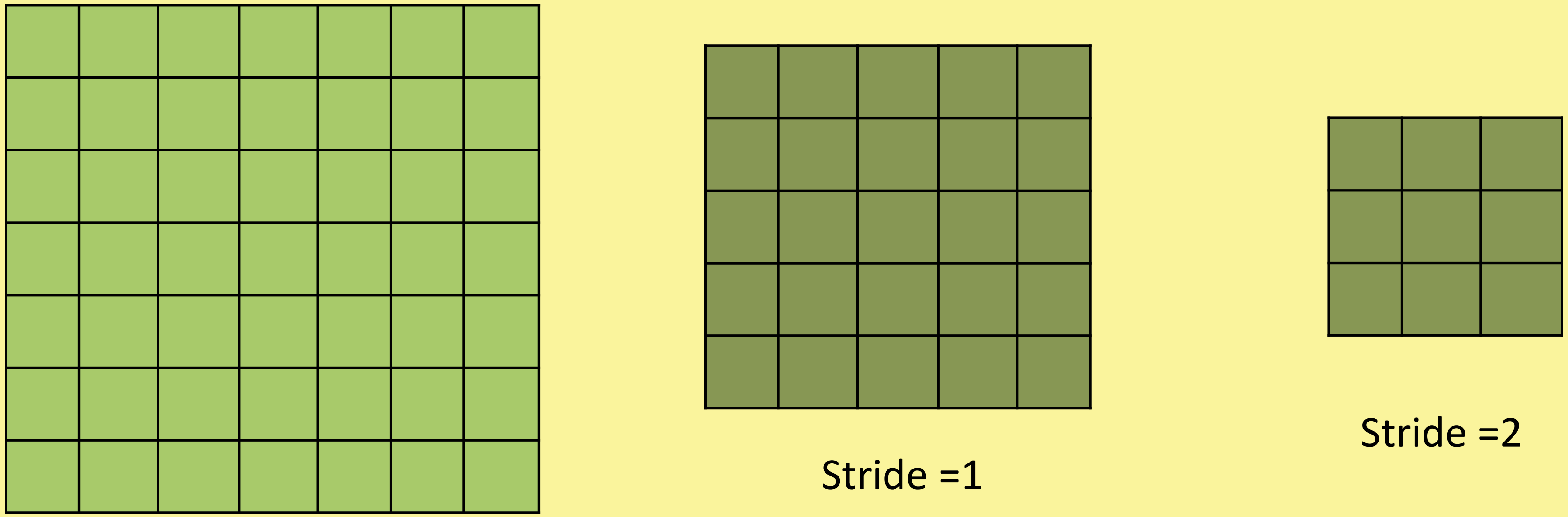


2 D Convolution



Stride

No. of steps the kernel is moved during convolution

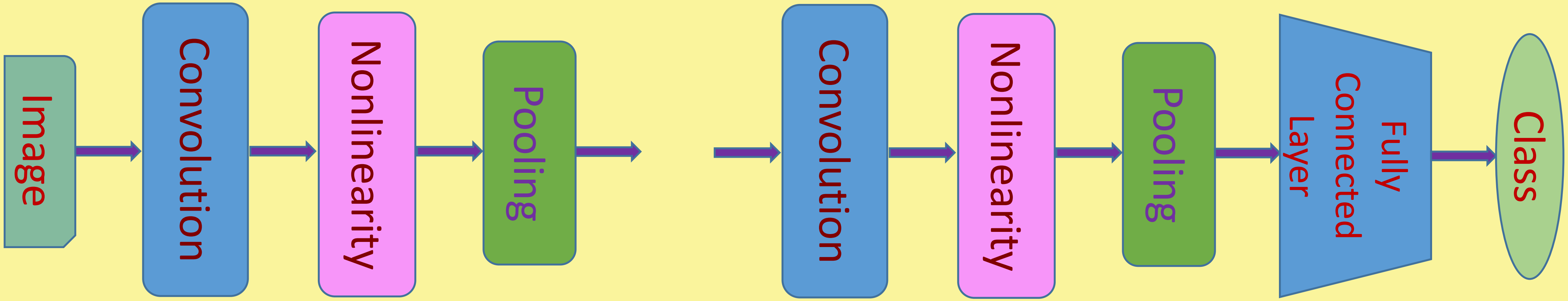


Stride =1

Stride =2

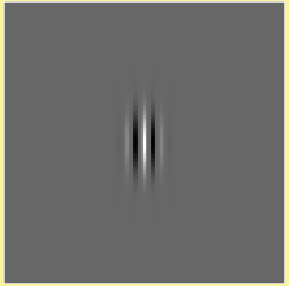
7 x 7 Input Image
3 x 3 Kernel

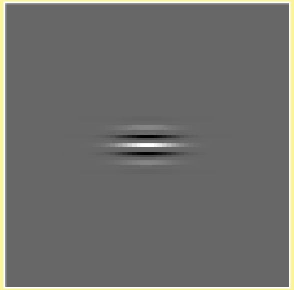
CNN Architecture



Convolution Layer: 3 D Convolution

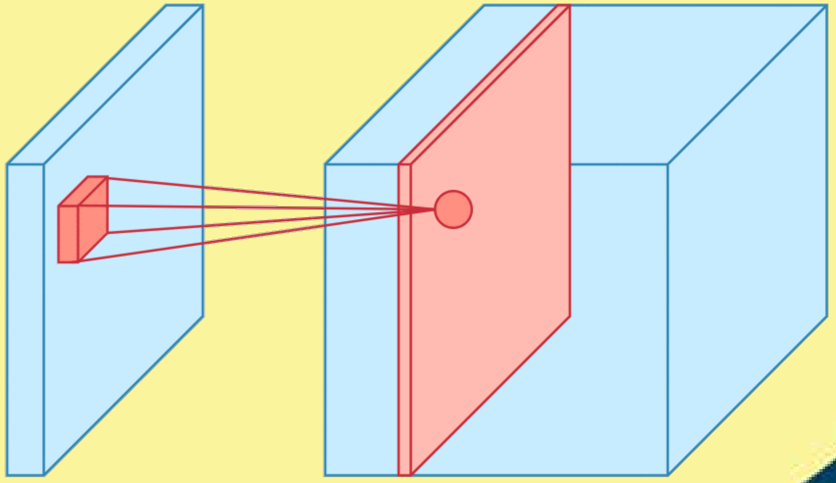
- Color image has 3 dimensions: height, width and depth (depth is the color channels i.e RGB)
- Filter or kernels that will be convolved with the RGB image could also be 3D
- For multiple Kernels: All feature maps obtained from distinct kernels are stacked to get the final output of that layer





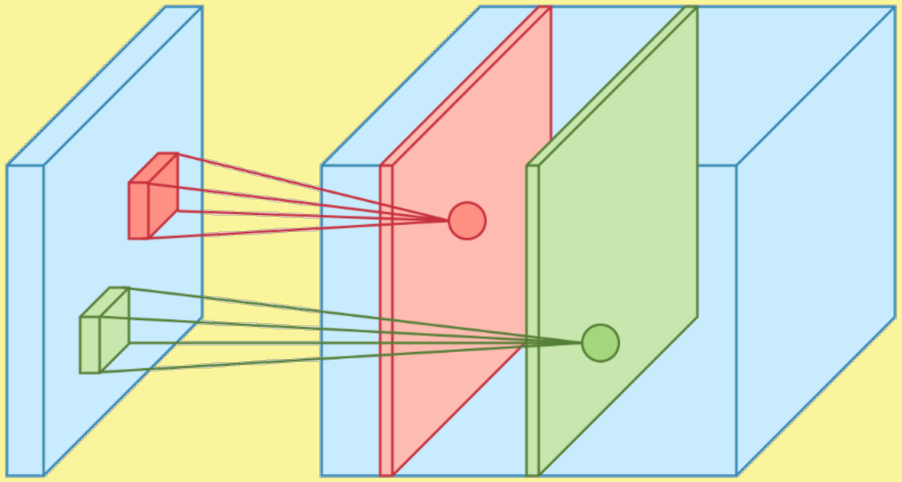
3 D Convolution-Visualization

- The kernel strides over the input Image.
- At each location $I(m,n)$ compute
  $$f(m,n) = \sum\sum w(p,q)I(p-m,q-n)$$
  collect them in the feature map.
- The animation shows the sliding operation at 4 locations, but in reality it is performed over the entire input.



Animation:- Arden Dertat
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

3 D Convolution-Visualization



- Red and green boxes are two different featured maps obtained by convolving the same input with two different kernels. The feature maps are stacked along the depth dimension as shown.

Figure: Arden Dertat
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

3 D Convolution-Visualization

- An RGB Image of size 32X32X3
- 10 Kernels of size 5x5x3
- Output featuremap of size 32x32x10

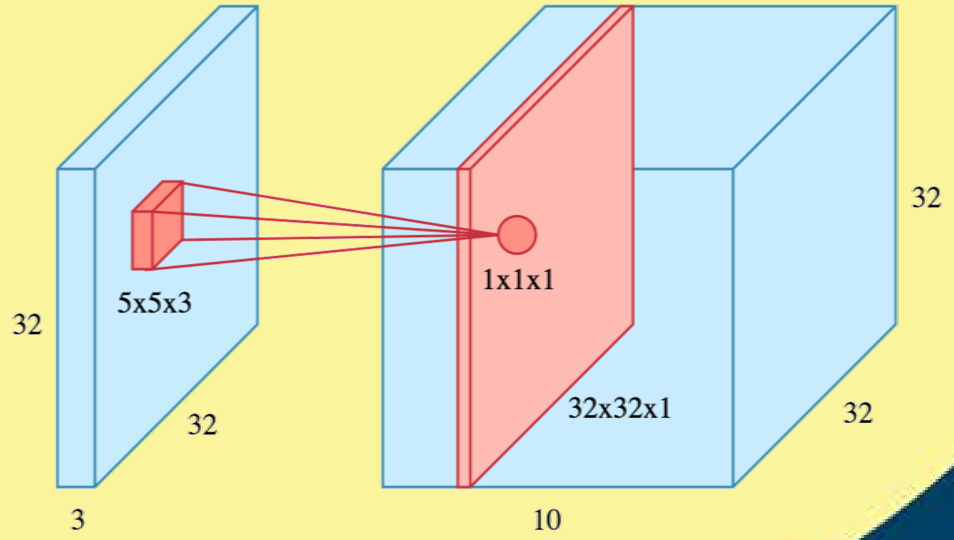


Figure: Arden Dertat
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Nonlinearity

- ReLU is an element wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero

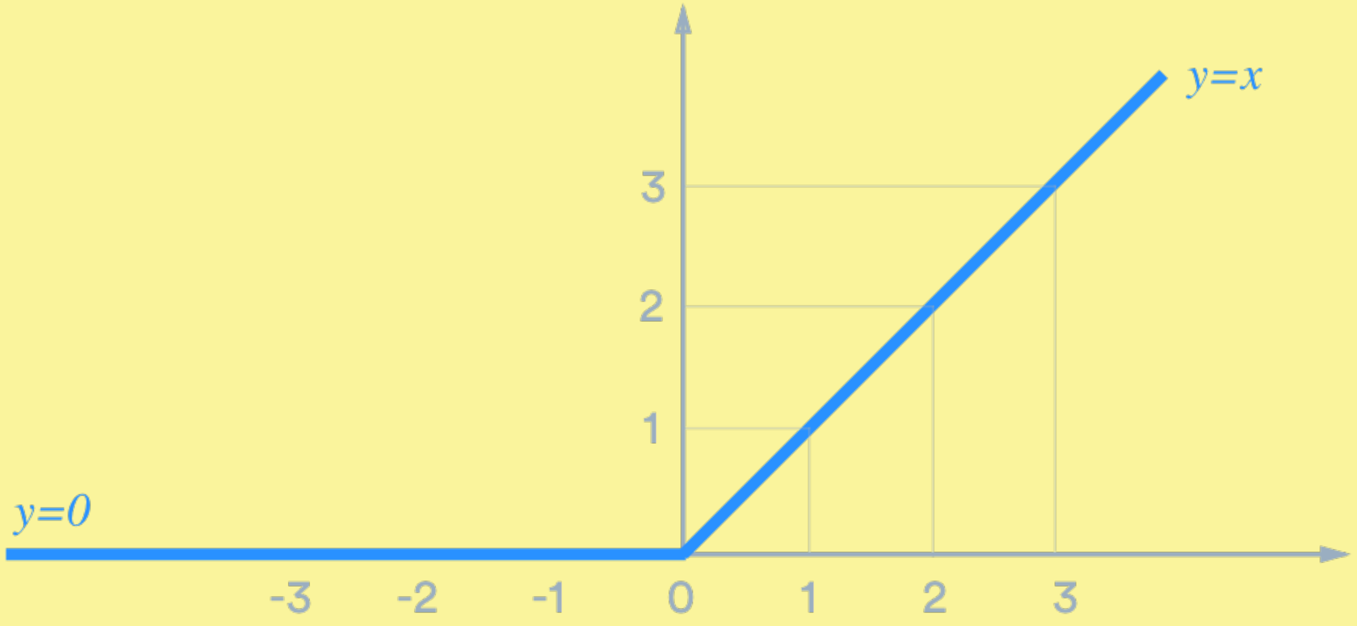


Figure: Arden Dertat
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Pooling

- Replaces the output of a node at certain locations with a summary statistic of nearby locations.
- Spatial Pooling can be of different types: Max, Average, Sum etc.
- Max Pooling report the maximum output within a rectangular neighborhood.
- Pooling helps to make the output approximately invariant to small translation.
- Pooling layers down sample each feature map independently, reducing the height and width, keeping the depth intact.
- In pooling layer stride and window size needs to be specified

Pooling

- Figure below is the result of max pooling using a 2x2 window and stride 2. Each color denotes a different window. Since both the window size and stride are 2, the windows are not overlapping

3	2	5	6
8	9	5	3
4	4	6	8
1	1	2	1

Max pool with 2x2 window with stride = 2 →

9	6
4	8

Figure: Arden Dertat
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Pooling

- Pooling reduces the height and the width of the feature map, but the depth remains unchanged as shown in figure
- Pooling operation is independently carried out across each depth

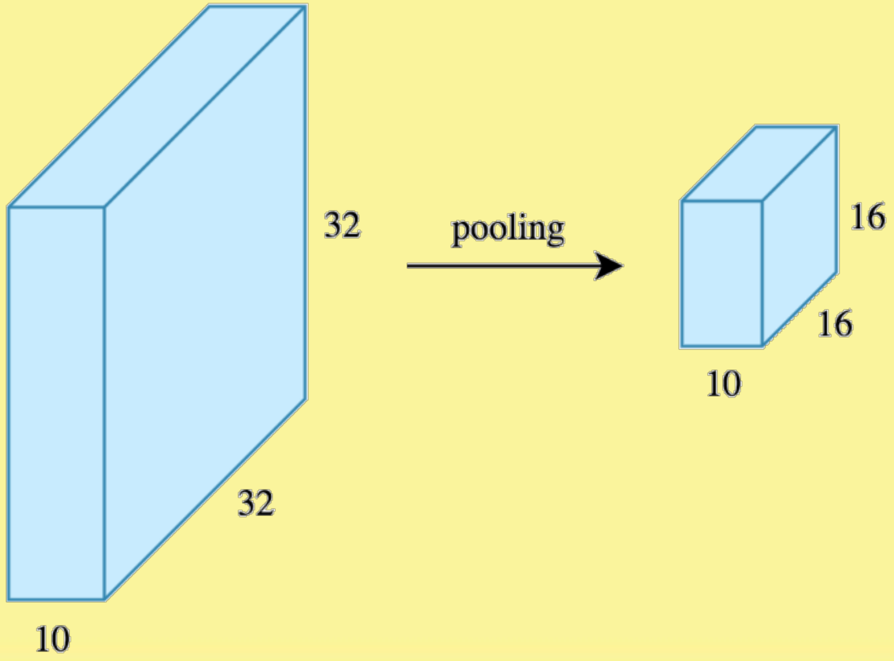
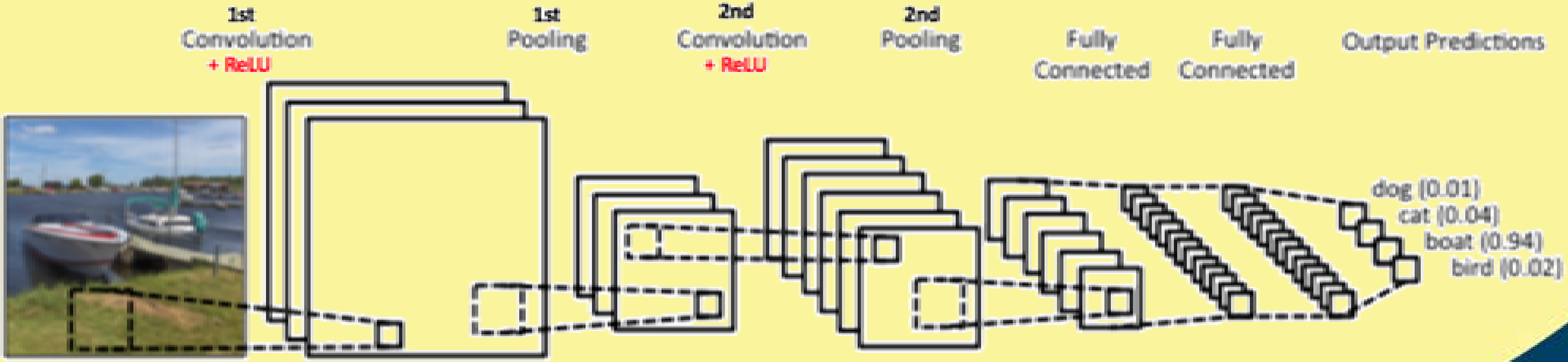


Figure: Arden Dertat
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Architecture



NPTEL ONLINE CERTIFICATION COURSES

Thank you