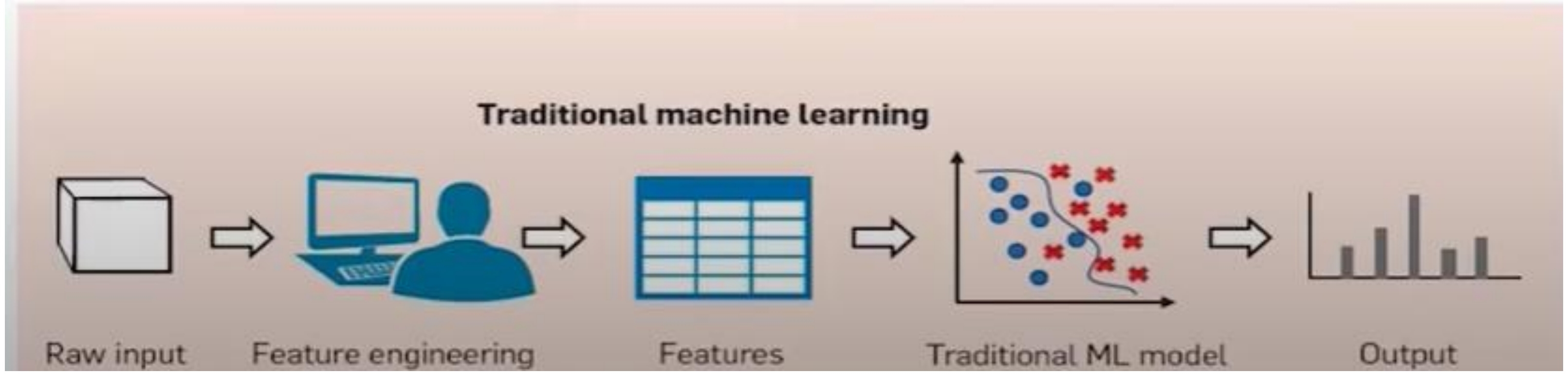
Deep Learning : Convolutional Neural Network (LeNet)

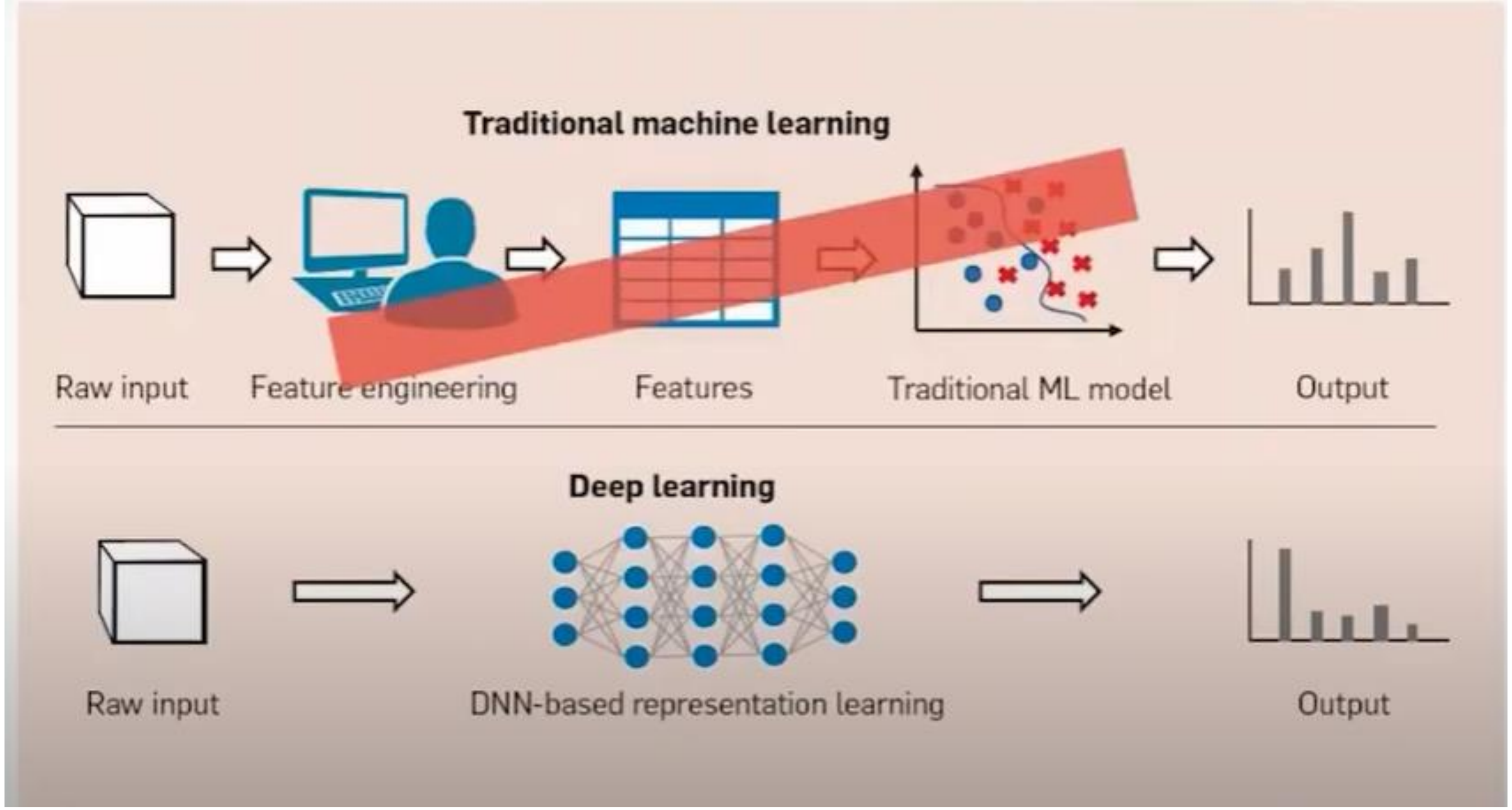
राष्ट्रीय प्रौद्योगिकी संस्थान सिक्किम
NATIONAL INSTITUTE OF TECHNOLOGY SIKKIM

Course Instructor:
Dr. Bam Bahadur Sinha
Assistant Professor
Computer Science & Engineering
National Institute of Technology
Sikkim

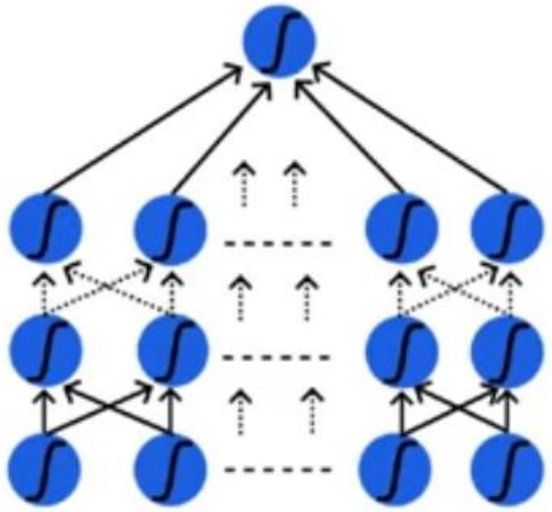
Traditional Machine Learning



Deep Learning



Deep Neural Networks



- UAT says that DNN are powerful function approximators
- Can be trained using backpropagation

- Prone to overfitting
- Gradient can vanish due to long chains

What Does The Convolution operation do?

x_0 $\qquad x_1 \qquad x_2$

w_{-2} $\qquad w_{-1} \qquad w_0$

$$s_t = \sum_{a=0}^{\infty} x_{t-a} w_{-a} = (x * w)_t$$

	w_{-6}	w_{-5}	w_{-4}	w_{-3}	w_{-2}	w_{-1}	w_0					
W	0.01	0.01	0.02	0.02	0.04	0.4	0.5					
X	1.00	1.10	1.20	1.40	1.70	1.80	1.90	2.10	2.20	2.40	2.50	2.70
S							1.80					

$$s_6 = x_6 w_0 + x_5 w_{-1} + x_4 w_{-2} + x_3 w_{-3} + x_2 w_{-4} + x_1$$

In 1-D case

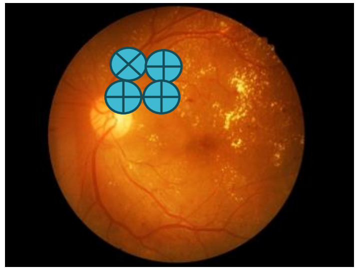
Convolution Operation – 2D Inputs

Input

a	b	c	d
e	f	g	h
i	j	k	ℓ

Kernel

w	x
y	z

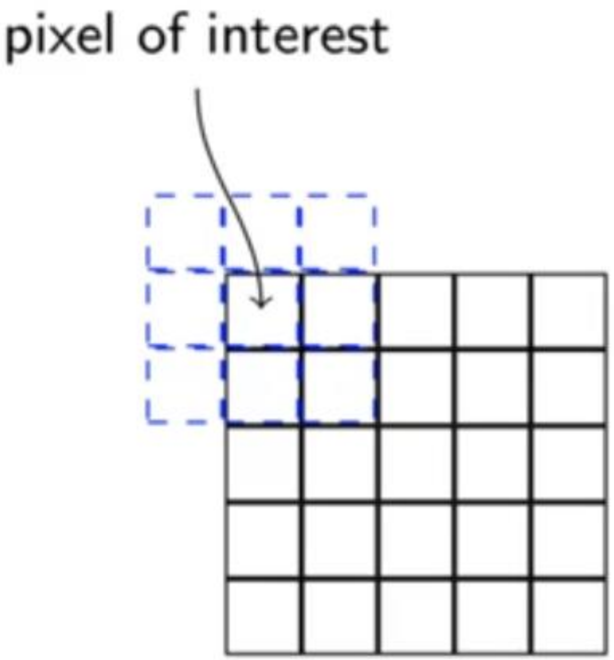


$$S_{ij} = (I * K)_{ij} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} I_{i+a,j+b} K_{a,b}$$

Output

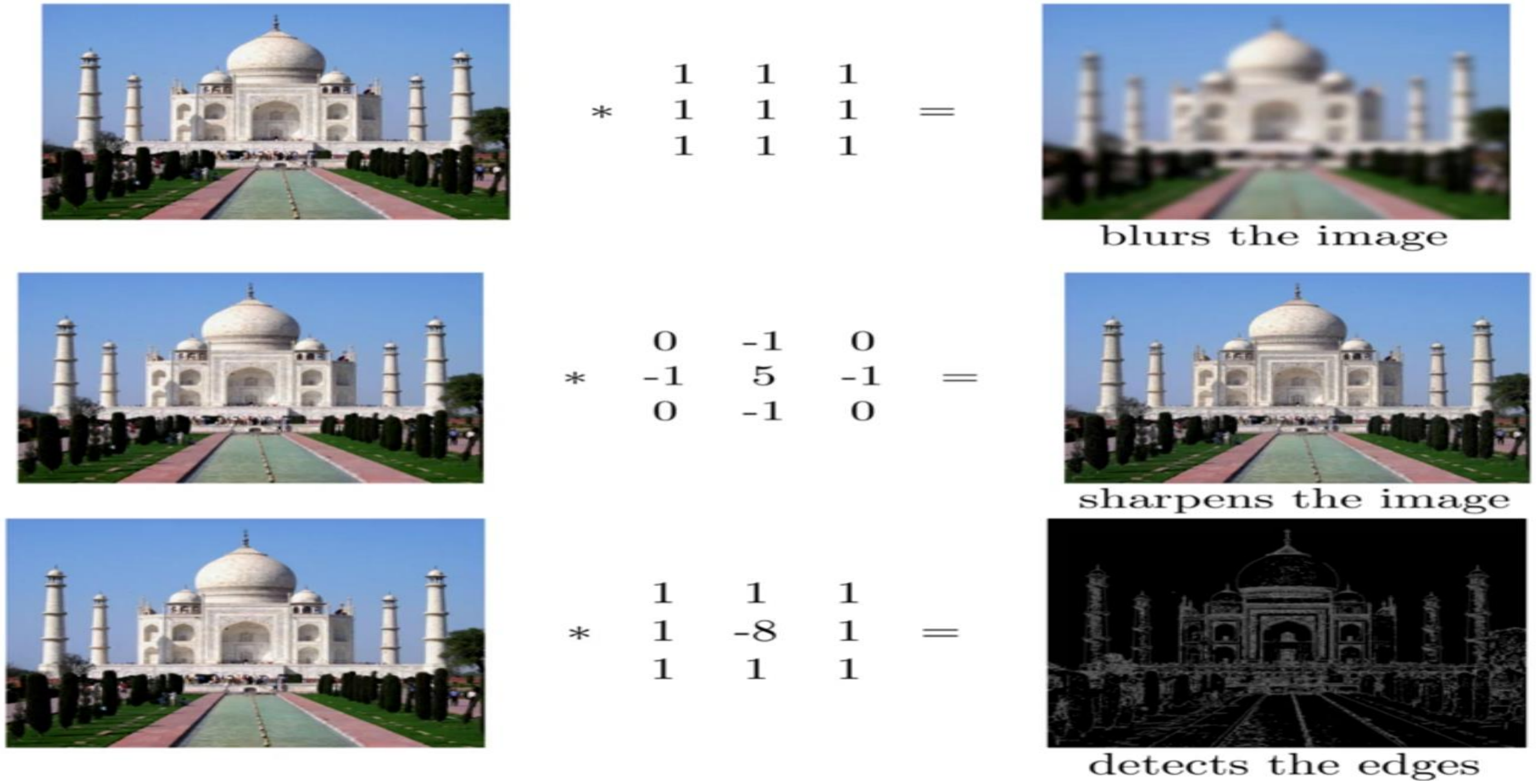
aw+bx+ey+fz	bw+cx+fy+gz	cw+dx+
ew+fx+iy+jz	fw+gx+jy+kz	gw+hx+

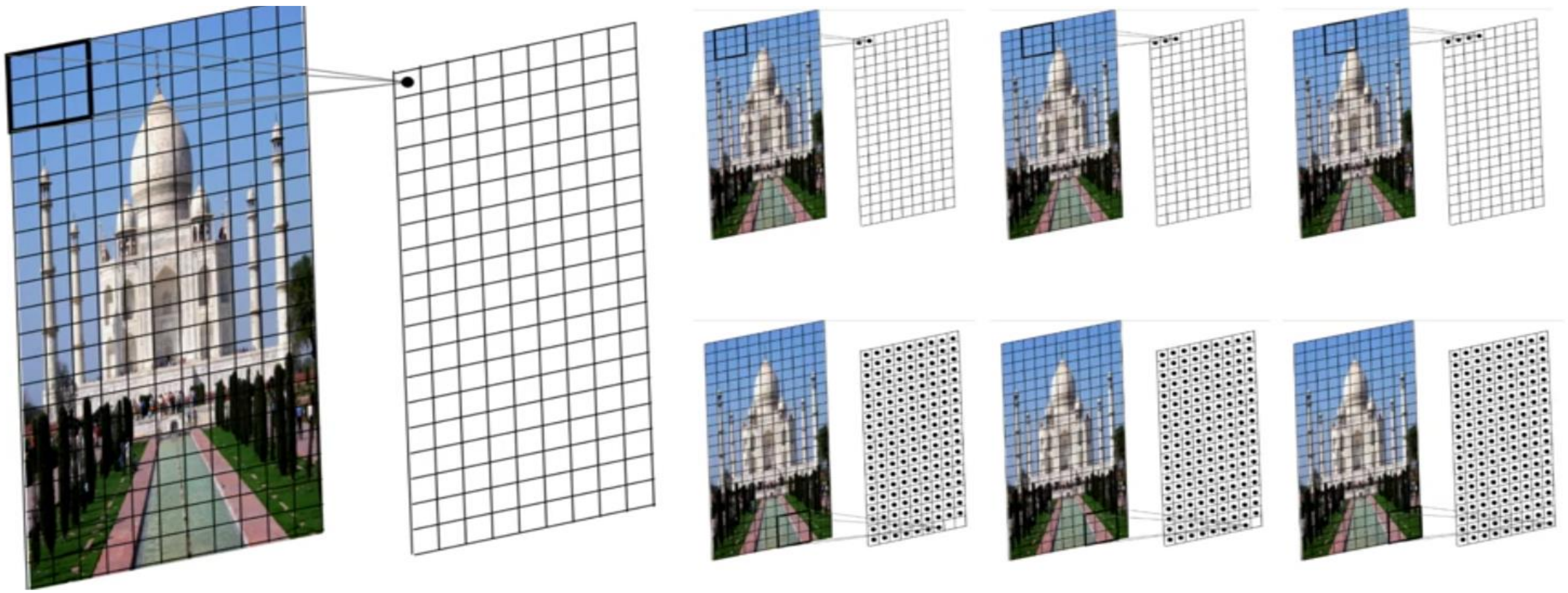
Convolution Operation (Considering previous neighbours)



$$S_{ij} = (I * K)_{ij} = \sum_{a=\lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b=\lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a,j-b} K_{\frac{m}{2}+a,\frac{n}{2}+b}$$

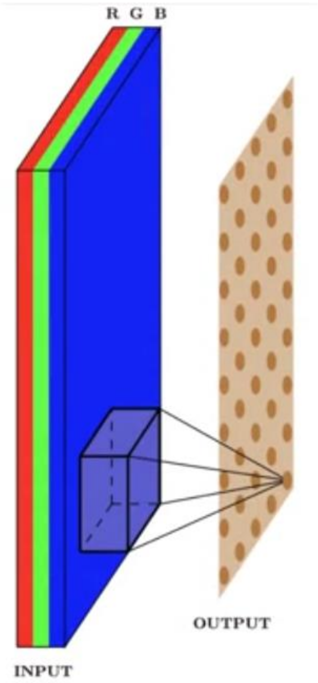
Convolution – In Practice





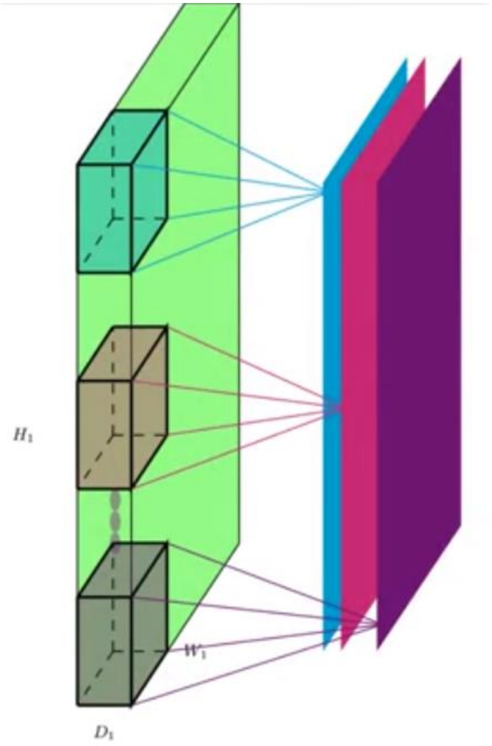
Convolution – In Practice

Convolution – In Practice



- Input is 3D
- Filter is also 3D
- But the convolution operation that we are performing is 2D
- We are only sliding vertically & horizontally and not along the path.
- This is because the depth of the filter is the same as the depth of the input.

Convolution – In Practice



- Can we apply multiple filters to the same image?

- Each filter applied to a 3D input will give a 2D output
- Combining the output of multiple such filters will result in a 3D input

Terminologies

Wi, Hi, Di
- Input Width,
- Input Height and
- Input Depth

F
- Spatial extent of the filter

K
- Number of Filters

- Output Width,
- Output Height and
- Output Depth

W, Ho, Do

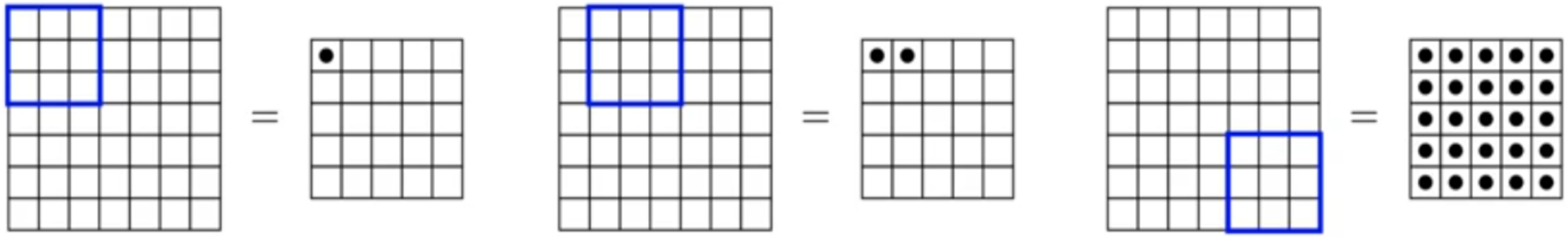
- Padding

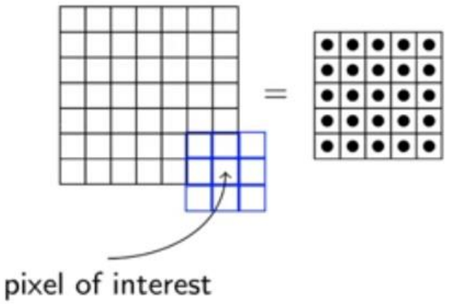
P

- Stride

S

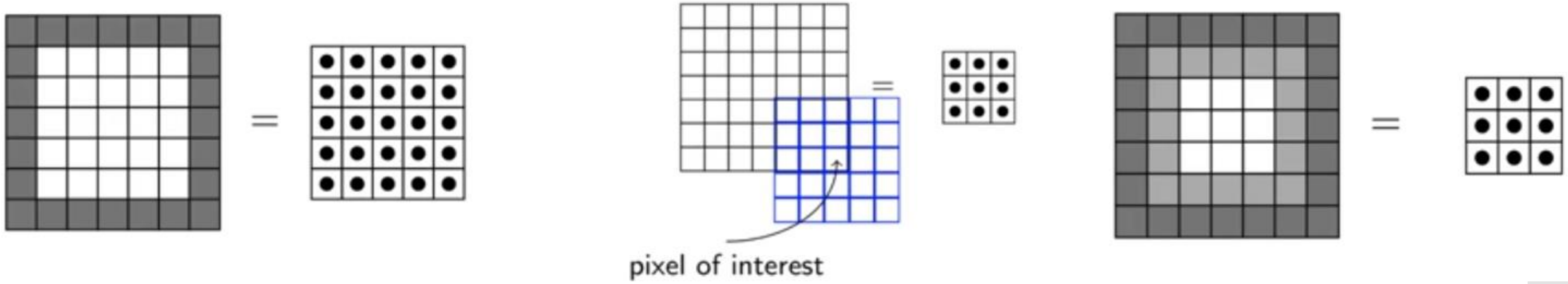
How do we Compute Wo, Ho, and Do





Size of output will be less than that of the input

How do we Compute Wo, Ho, and Do

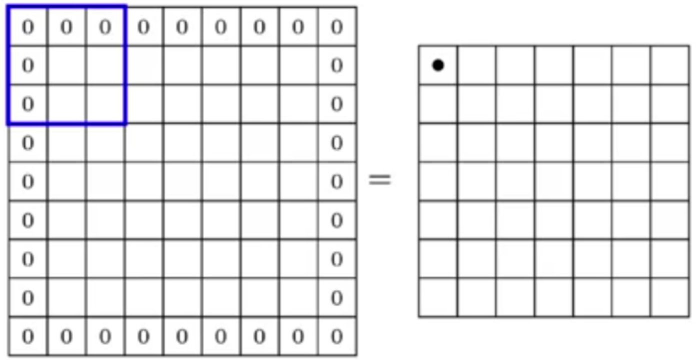


$$W_O = W_I - F + 1$$

$$H_O = H_I - F + 1$$

Size of output will be less than that of the input

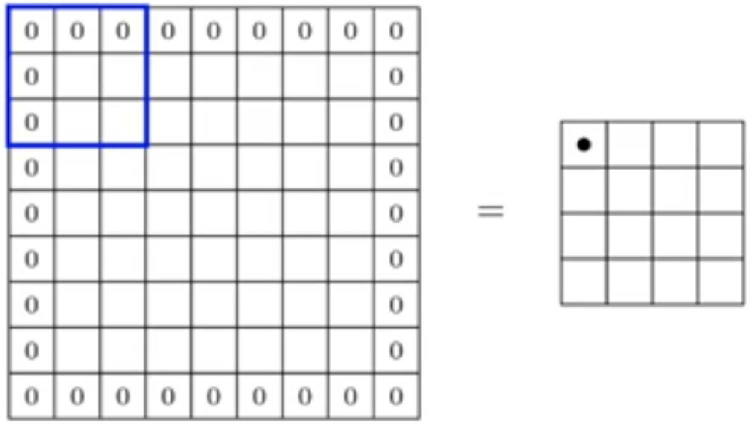
What if We Want the Output to be of same size as INPUT?

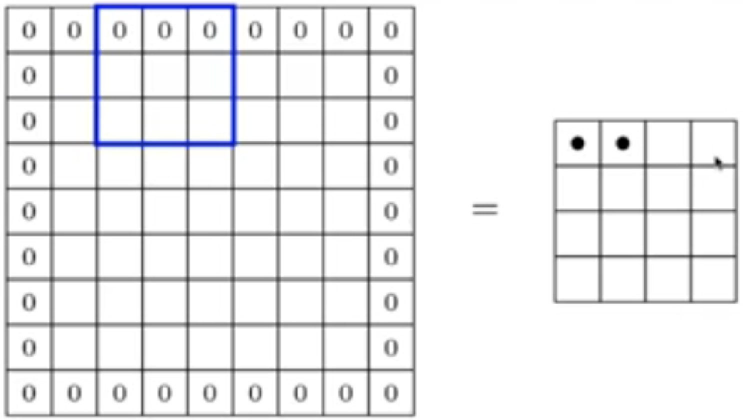


$$W_O = W_I - F + 2P + 1$$
$$H_O = H_I - F + 2P + 1$$

The bigger the kernel size, the larger is the padding required

What Does The Stride 'S' Do?



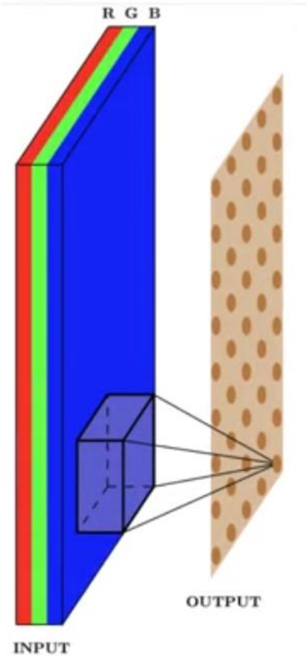


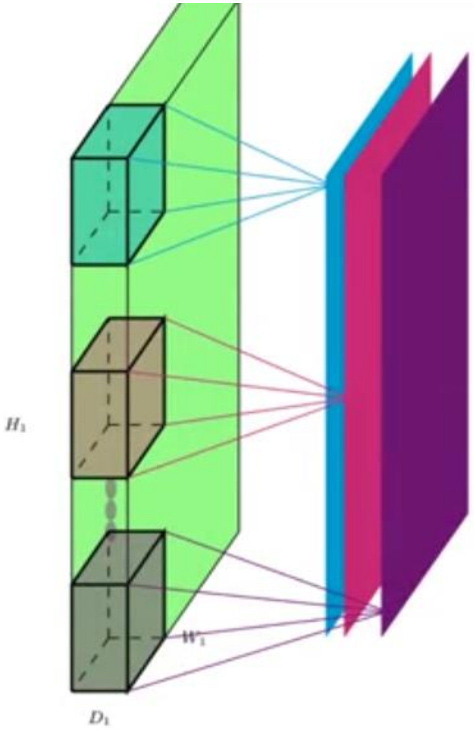
$$W_O = \frac{W_I - F + 2P}{S} + 1$$

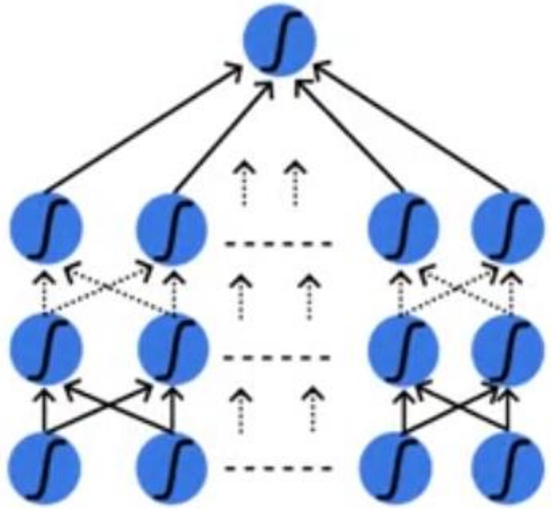
$$H_O = \frac{H_I - F + 2P}{S} + 1$$

Higher the stride, smaller will be the size of the output

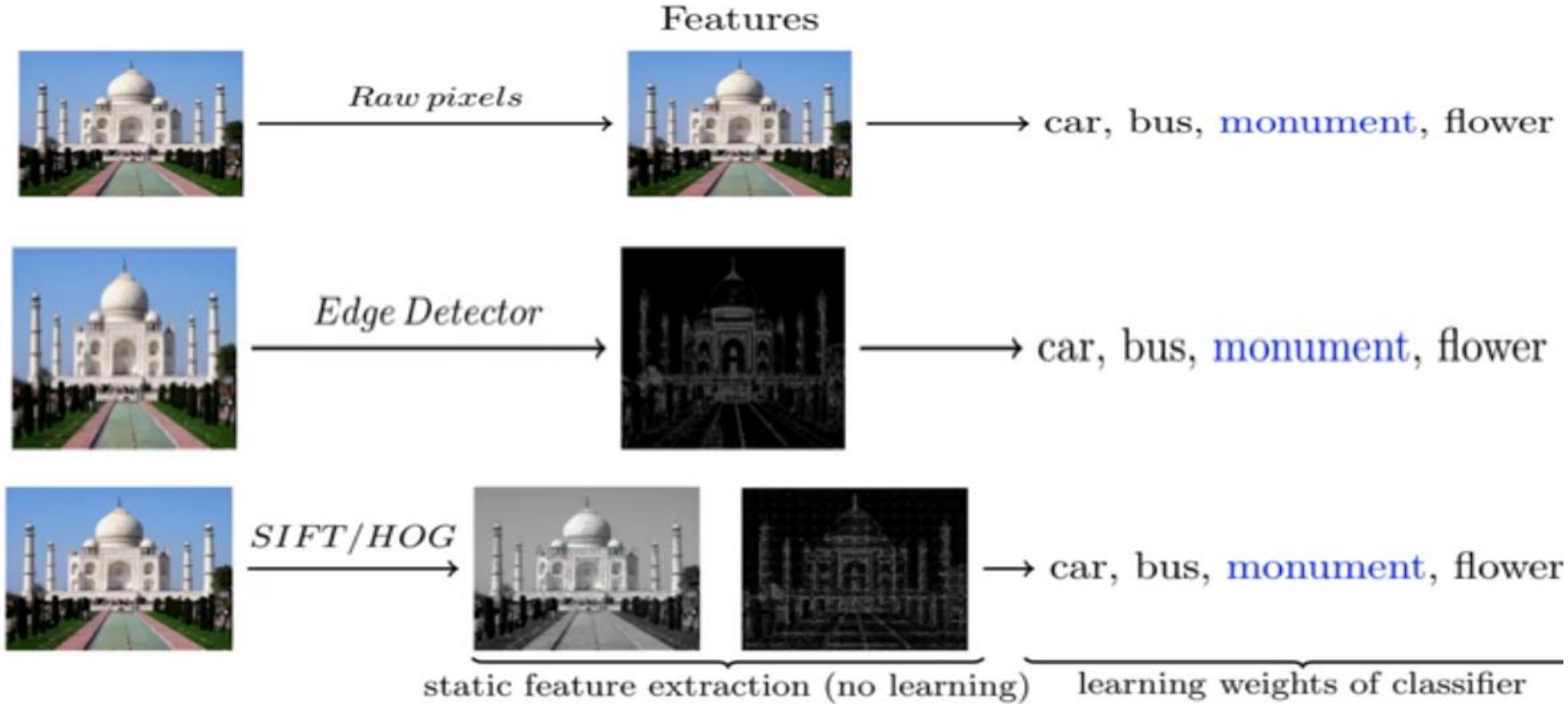
Relation Between Convolution Operation & Neural Network



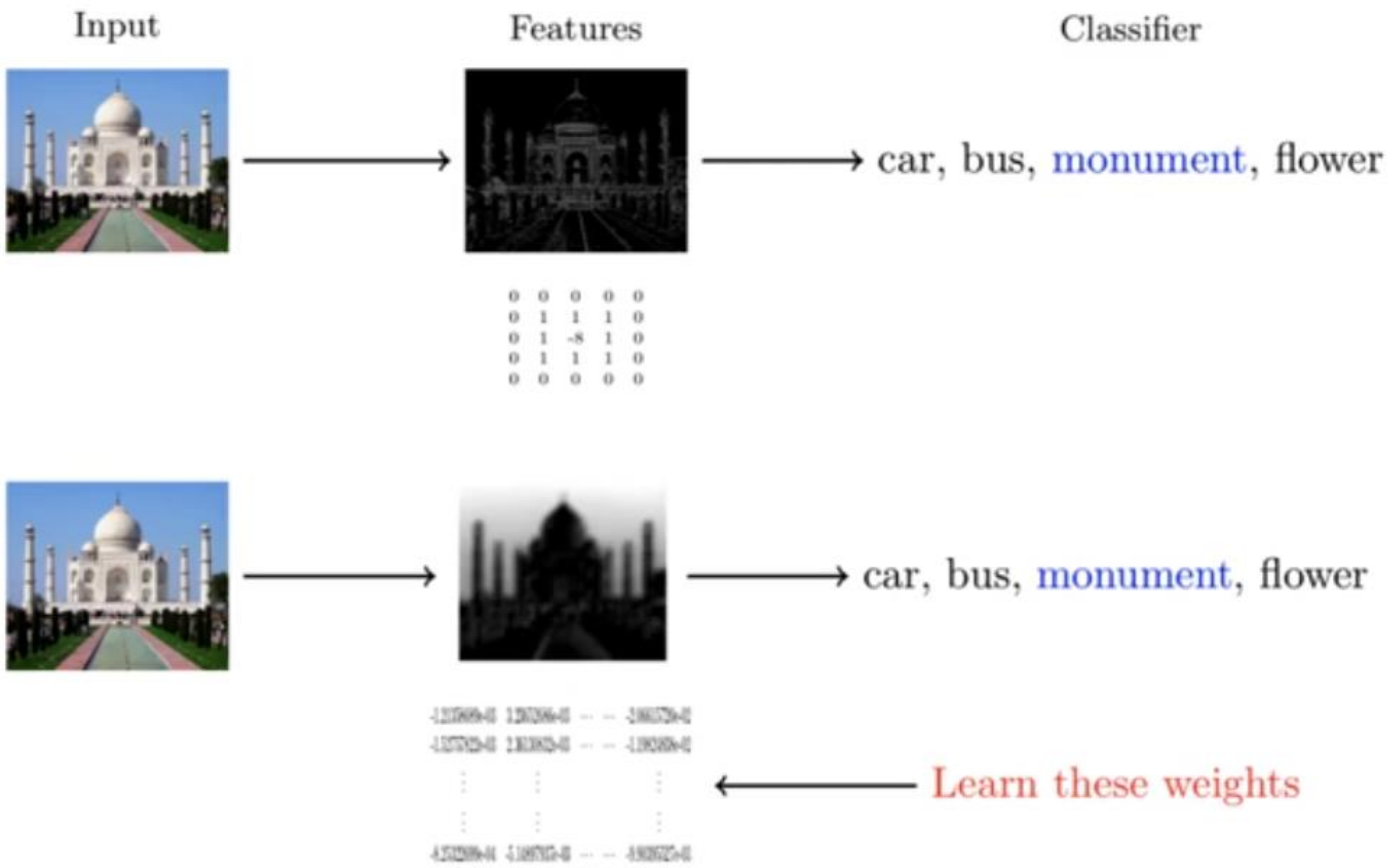




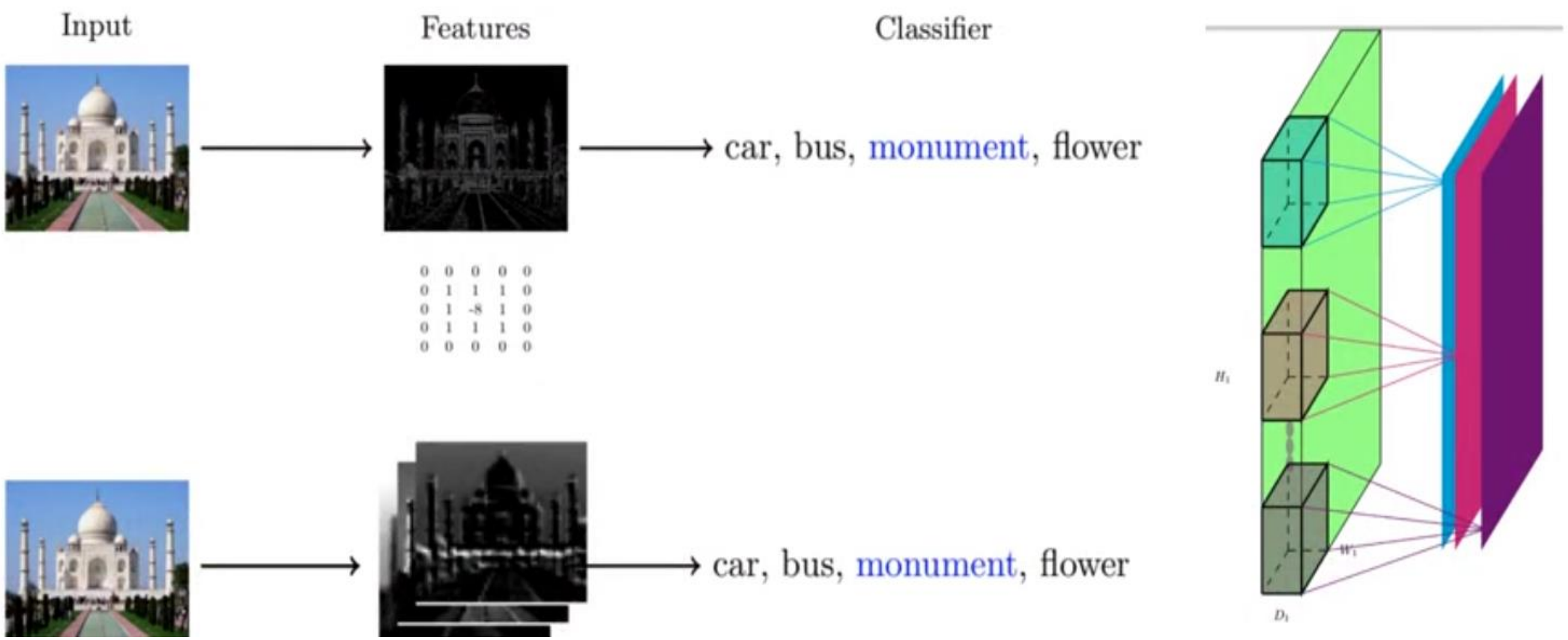
Convolutional Neural Network Vs Simple Neural Network



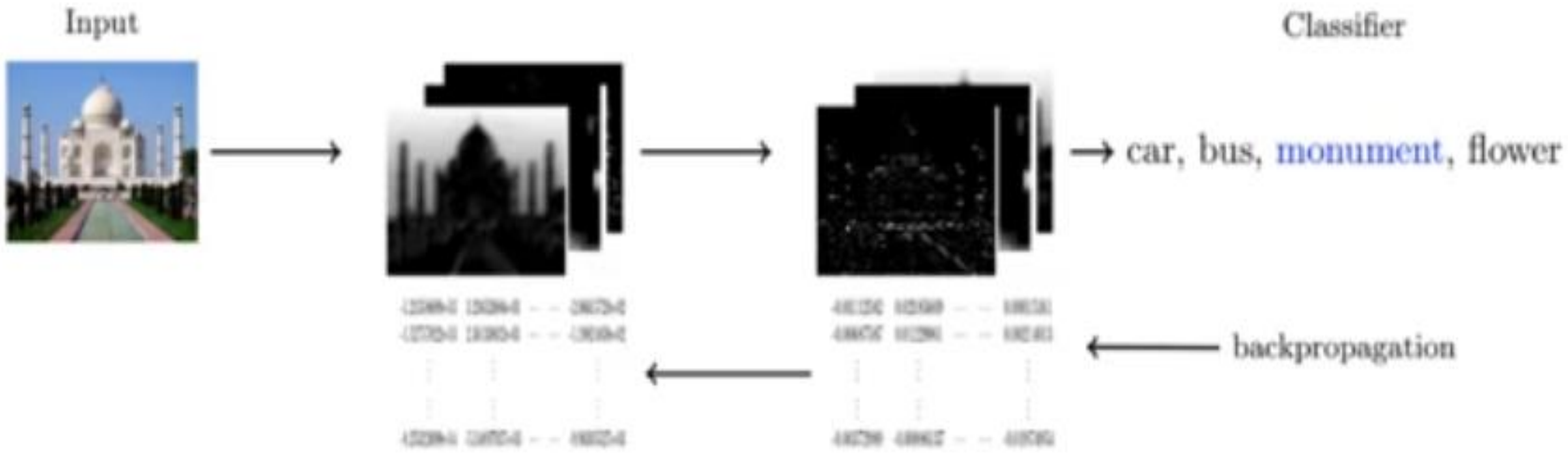
Let The Network Learn Feature Representations



Let The Network Learn MULTIPLE Feature Representations

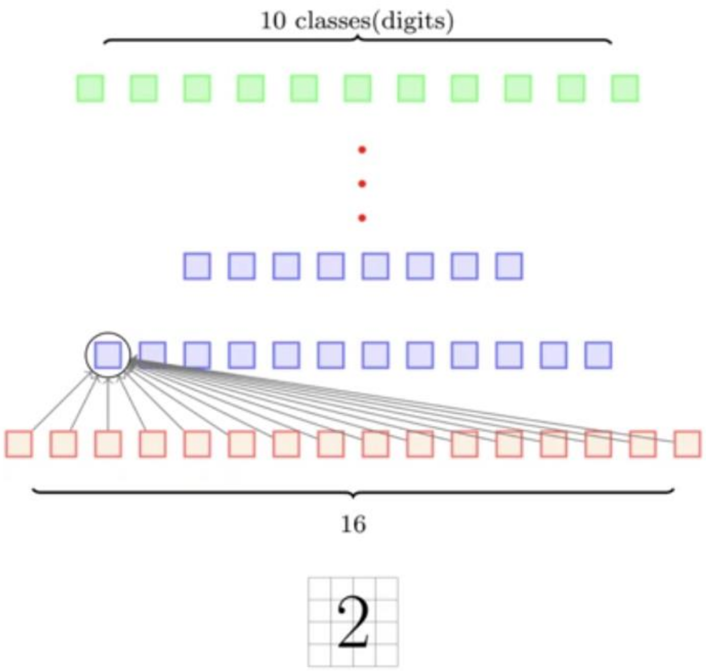


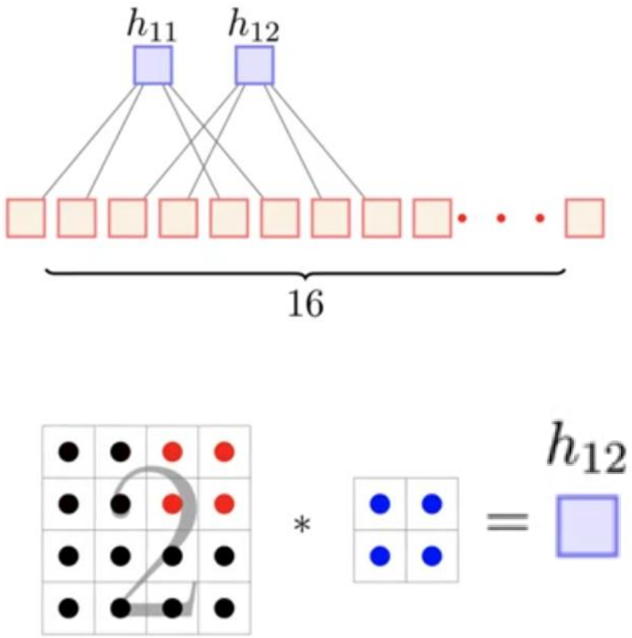
Let The Network Learn MULTIPLE layers of Feature Representations



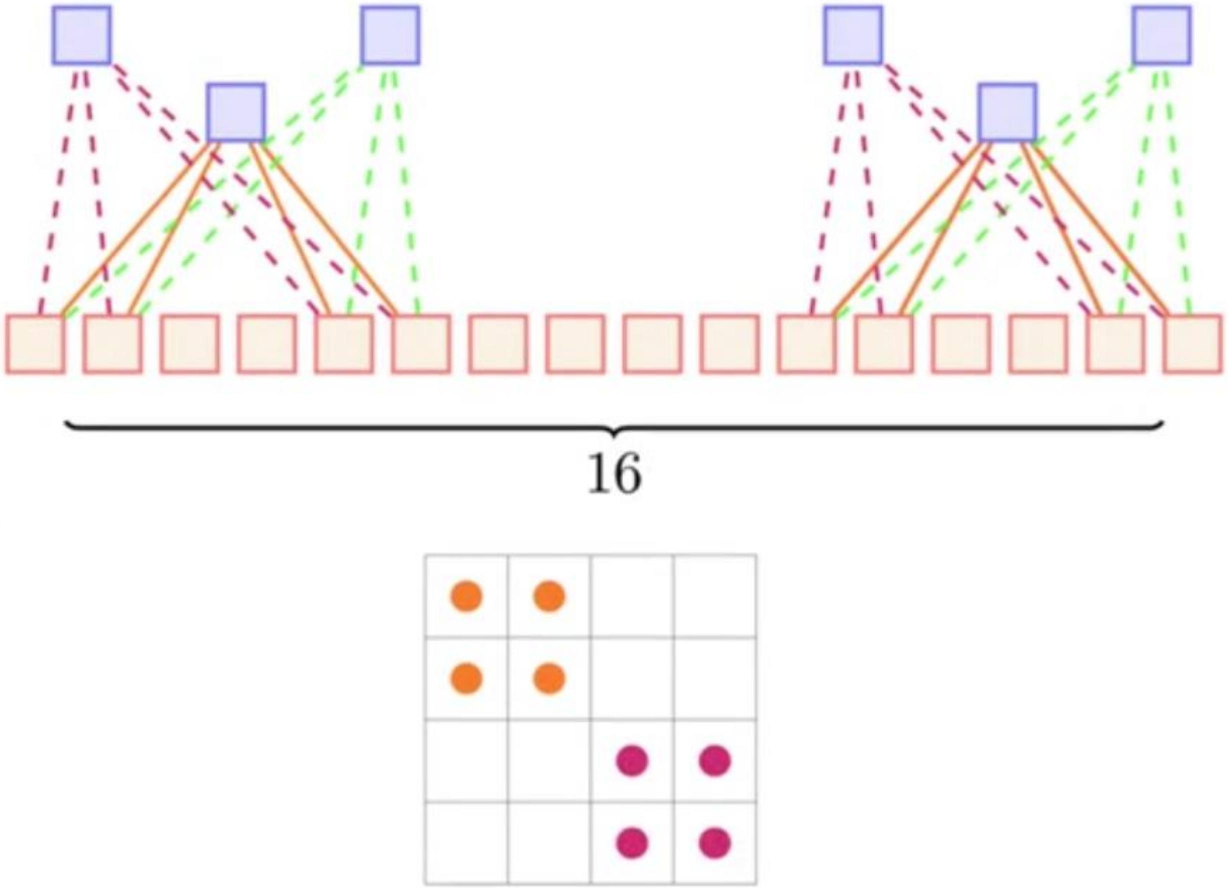
How is CNN different from a fully Connected Neural Network ?

Sparse Connectivity & Weight Sharing

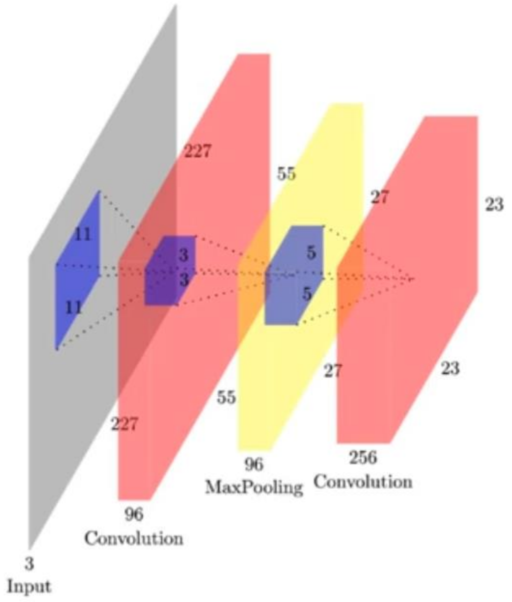


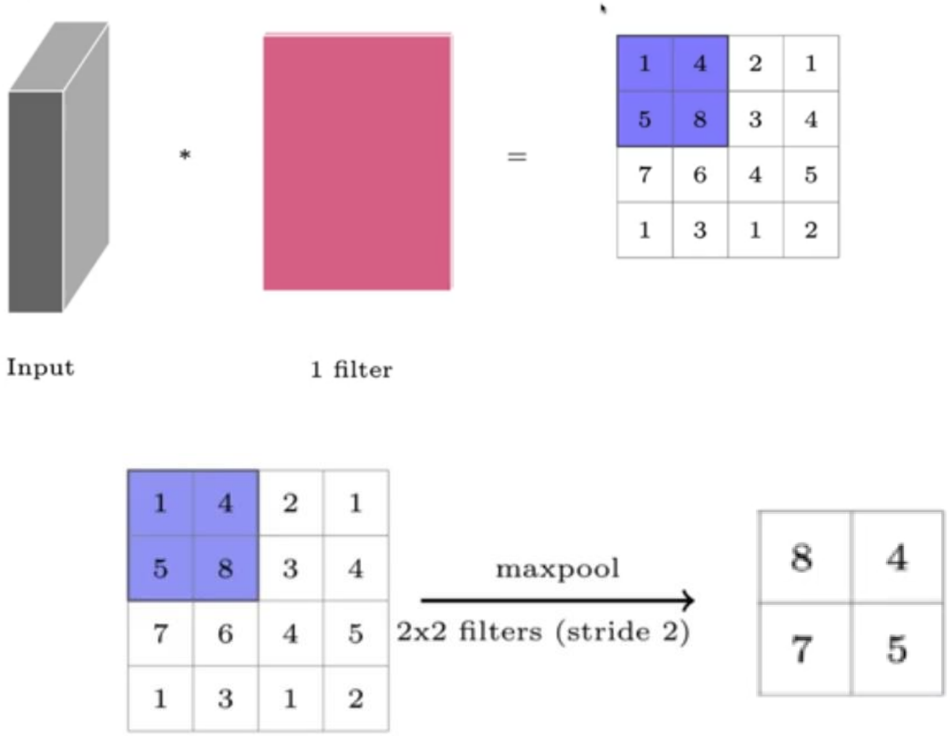


How is CNN different from a fully Connected Neural Network ?



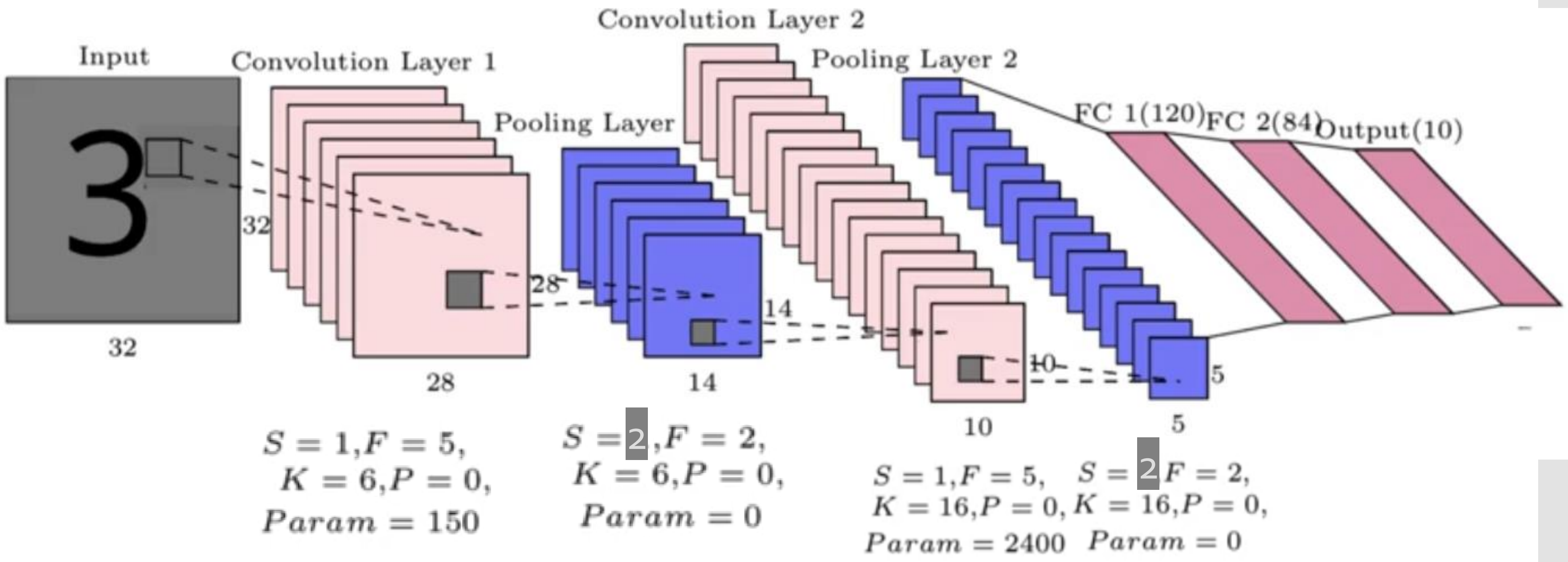
What Is The Max Pooling Operation



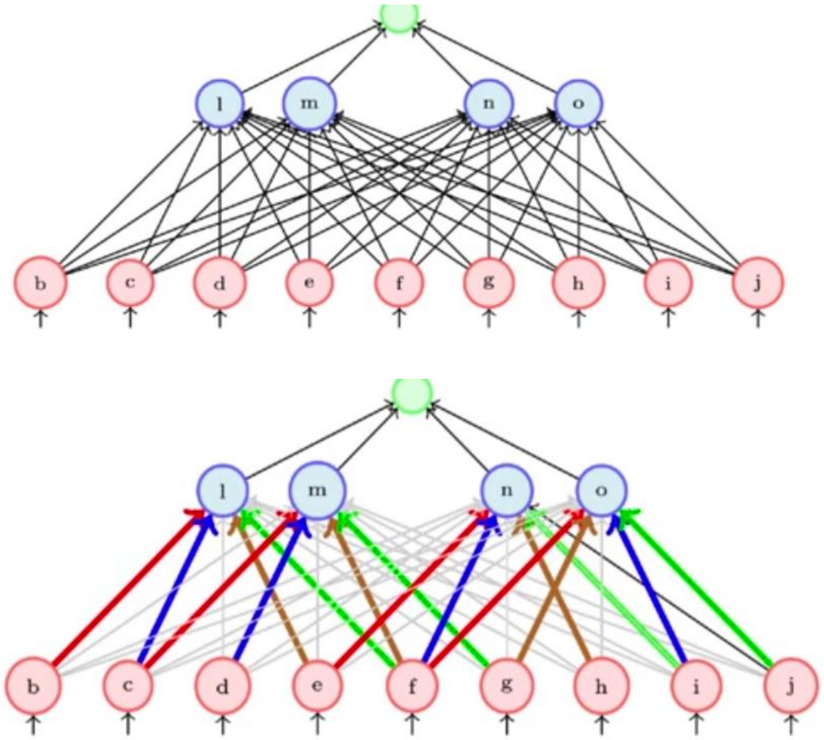


How To Use CNN for Image Classification

(LeNet Architecture)



How Do We Train A CNN Model?



- A CNN can be implemented as a Feedforward Network
- Only a few weights (in color) are active
- The rest of weights (in grey) are zero/inactive