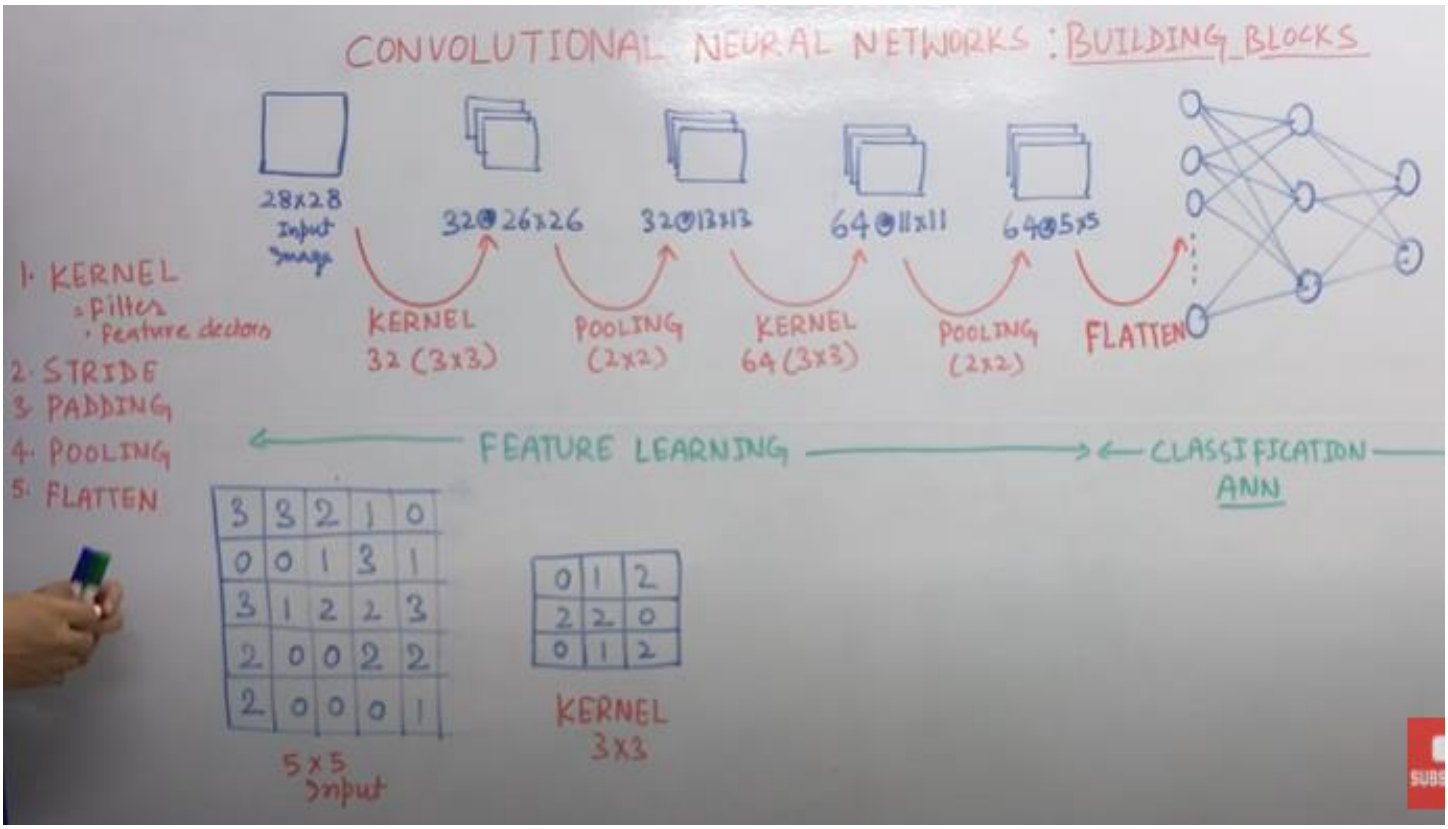
CNN model

I. Kernel or Filter or feature detector
II. Stride
III. Padding
IV. Pooling
V. Flatten



Kernel

Kernel is nothing but a filter that is used to extract the feature from the images.
The kernel is a matrix that moves over the input data, performs the dot product with the sub region of input data, and gets the output as the matrix of dot product.
Kernel moves on the input data by the stride value.

Step1:

Input matrix

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

(5*5)

Kernel matrix

0	**1**	**2**
2	**2**	**0**
0	**1**	**2**

(3*3)

Convolutional matrix as resultant

3*0 +3*1+2*2+0*2+0*2+1*0+3*0+1*1+2*2= 3+4+1+4=12

Step2

Input matrix:

3	**3**	**2**	**1**	**0**
0	**0**	**1**	**3**	**1**
3	**1**	**2**	**2**	**3**
2	**0**	**0**	**2**	**2**
2	**0**	**0**	**0**	**1**

Kernel image

0	**1**	**2**
2	**2**	**0**
0	**1**	**2**

Convolution matrix as resultant:

3*0+2*1+1*2+0*2+1*2+3*0+1*0+2*1+2*2= 2+2+2+2+4= 12

Step3

Input matrix

3	**3**	**2**	**1**	**0**
0	**0**	**1**	**3**	**1**
3	**1**	**2**	**2**	**3**
2	**0**	**0**	**2**	**2**
2	**0**	**0**	**0**	**1**

Kernel matrix

0	1	2
2	2	0
0	1	2

Convolutional matrix resultant

2*0+1*1+0*2+1*2+3*2+1*0+2*0+2*1+3*2 =1+2+6+2+6=17

Step4

Input matrix

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Kernel matrix

0	1	2
2	2	0
0	1	2

Convolutional matrix resultant

0*0+0*1+1*2+3*2+1*2+2*0+2*0+0*1+0*2 =2+6+2=10

Step5

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Kernel matrix

0	1	2
2	2	0
0	1	2

Convolutional matrix resultant

0*0+1*1+3*2+1*2+2*2+2*0+0*0+0*1+2*2=1+6+2+4+4=17

Step6

Input matrix

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Kernel matrix

0	1	2
2	2	0
0	1	2

Convolutional resultant matrix

1*0+3*1+1*2+2*2+2*2+3*0+0*0+2*1+2*2=3+2+4+4+2+4=19

Step 7:

Input matrix

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Kernel matrix

0	1	2
2	2	0
0	1	2

Convolutional resultant matrix

3*0+1*1+2*2+2*2+0*2+0*0+2*0+0*1+0*2=1+4+4=9

Step8:

Input matrix

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Kernel matrix

0	1	2
2	2	0
0	1	2

Resultant convolutional matrix:

1*0+2*1+2*2+0*2+0*2+2*0+0*0+0*1+0*2=2+4=6

Step9:

Input matrix:

3		3	2	1	0
0		0	1	3	1
3		1	2	2	3
2		0	0	2	2
2		0	0	0	1

Kernel matrix:

0	1	2
2	2	0
0	1	2

Convolutional resultant matrix:

2*0+2*1+3*2+0*2+2*2+2*0+0*0+0*1+1*2=2+6+4+2=14

Final convolutional matrix

12	12	17
10	17	19
9	6	14

This case stride=1

Size of output =[input – kernel]\stride +1

= [5-3]\1+1

= 3

Stride:

Filter moves across the image left to right, top to bottom, with a one pixel column change on the horizontal movement, then one pixel row change on the vertical movement.

The amount of movement between application of the filter to the input image reffered to as stride.

By default stride in two dimensions (1,1) for the height and width of movement.

Padding:

Padding is a term relevant to convolutional neural networks as it refers to **the amount of pixels added to an image when it is being processed by the kernel of a CNN.**

0	**0**	**0**	**0**	**0**	**0**	**0**
0	**3**	**3**	**2**	**1**	**0**	**0**
0	**0**	**0**	**1**	**3**	**1**	**0**
0	**3**	**1**	**2**	**2**	**3**	**0**
0	**2**	**0**	**0**	**2**	**2**	**0**
0	**2**	**0**	**0**	**0**	**1**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**

Output matrix = [input-kernel+2p]\stride +1

= [5-3+2*1]\1 +1

= [2+2]+1

= 5

Pooling:

Pooling is required to down sample the detection of feature in the feature map.

Two method of pooling:

I. Average pooling
II. Max pooling

Example of max pooling(2*2)

12	13	23	11	2
33	33	33	22	2
22	4	44	33	44
44	4	99	45	33
77	55	66	55	33

Answer =33

12	**13**	**23**	**11**	**2**
33	**33**	**33**	**22**	**2**
22	**4**	**44**	**33**	**44**
44	**4**	**99**	**45**	**33**
77	**55**	**66**	**55**	**33**

Answer=33

Final max pooling matrix

33	**33**
44	**99**

Average pooling layer example:

12	**13**	**23**	**11**	**2**
33	**33**	**33**	**22**	**2**
22	**4**	**44**	**33**	**44**
44	**4**	**99**	**45**	**33**
77	**55**	**66**	**55**	**33**

= 12+13+33+33\4
= 66.25

Concept of using pooling layers is reducing dimension

Flatten layers:
Once the pooled feature map is obtained, the next layers is flatten,
Its involve the whole pooled feature matrix into single column which is fed into neural network for processing.