12.12.23

CSE-461

"Lecture -13"

"Intro to CNNs and Object Detection"

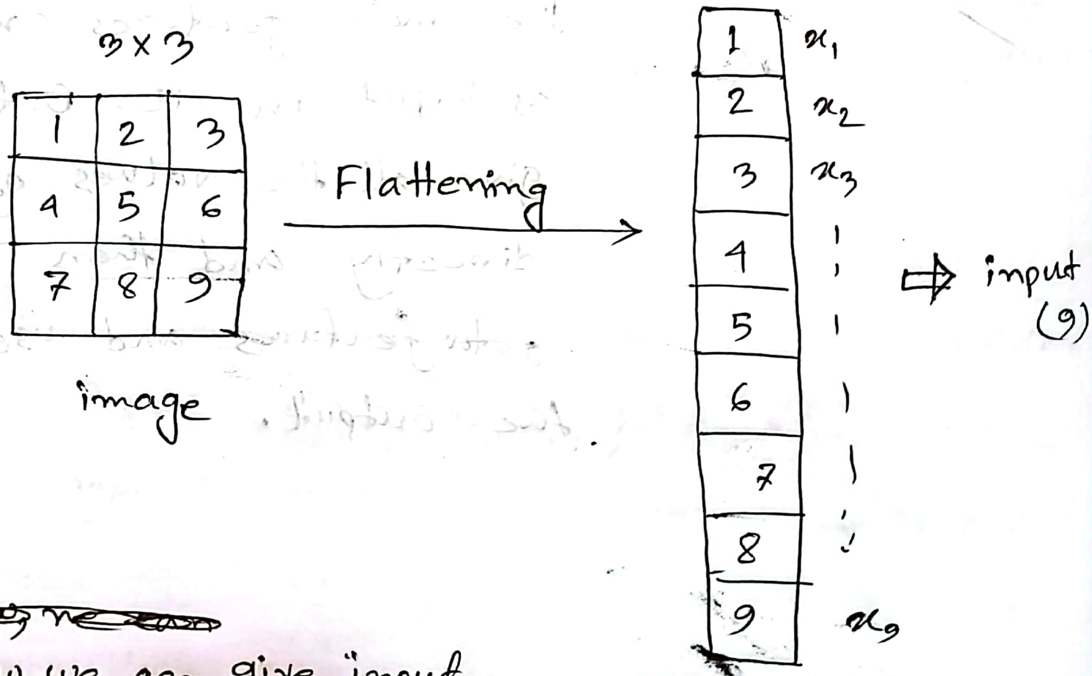
→ Input features are two types,

(a) Features are given (uses ML)

(b) Raw data is given not features (uses DL)

→ Now, how to give input? "Pixels as Features"

(a) we can do Flattening of our image.



~~Now, we can~~

now we can give input.

তারপর

→ But we have a problem. The problem is ~~dime~~ dimensionality.

if we have 3x3 image, then input = 9

" " " 12x12 " " " = 144

⋮ ⋮

" " " 680x420 " " " = 856,800

→ So, image size বাড়লে input features বাড়তেছে and for that ~~we h~~ the cost is increasing and also time consuming.

→ Second is, একটা image থেকে আমরা actually কোনো feature পাবো না। একটা image থেকে object detect করা কঠিন। So, image flatten করলে structural information থাকতেছে না।

→ So, to Solve these two Problems we will use 'Convolution'

→ Convolution:

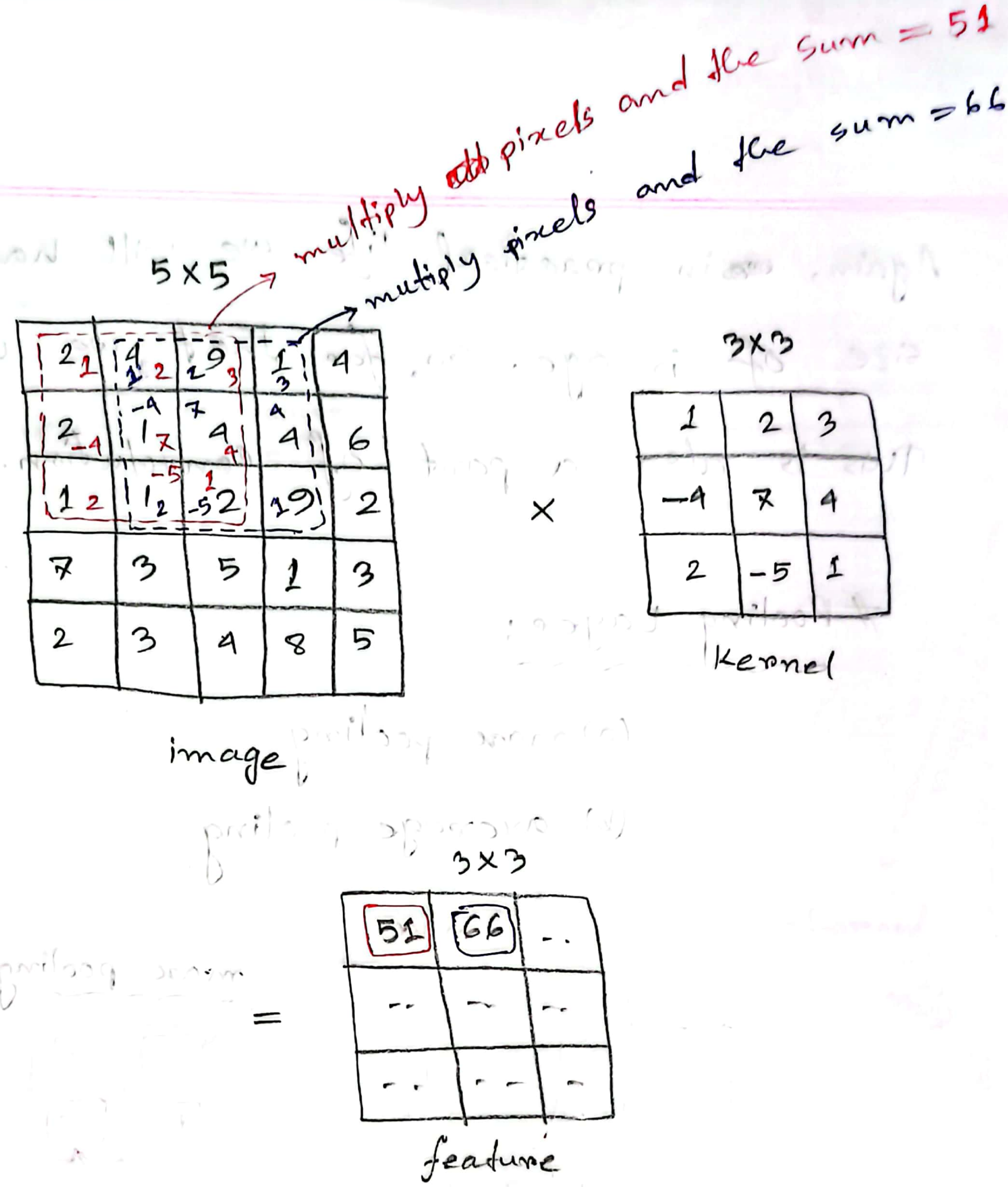
Here, we have a image and a filter or Kernel. So, we will place the kernel in our main image in the upper left corner. Then ~~value of~~ the image and the kernel এর pixel value গুলো multiply করে sum করবো and সেই value পাবো সেটা feature Matrix এর upper left pixel এ বসবে।

Then kernel কে 1 ঘর right এ বসাবো and then same ভাবে করে একটা feature Matrix পাবো।

→ So, feature Matrix এর dimension কি হবে?

$$\Rightarrow (\text{image dimension} - \text{kernel dim}) + 1 = FM$$

$$\therefore \text{feature dim} = (n - k + 1)$$



multiply ~~all~~ pixels and the sum = 51

mutiply pixels and the sum = 66

2	4	9	1	4
2	1	4	4	6
1	1	2	9	2
7	3	5	1	3
2	3	4	8	5

image

1	2	3
-4	7	4
2	-5	1

Kernel

51	66	..
--	--	--
..	--	-

feature

here,

$n = 5$

$k = 3$

$$\therefore \text{feature} = (n - k + 1)$$
$$= (5 - 3 + 1)$$
$$= 3$$

So, convolution করে আমরা feature কে small pixels এ convert করে দিচ্ছি and যার কারণে flatten করলে input feature কম যাবে as input.

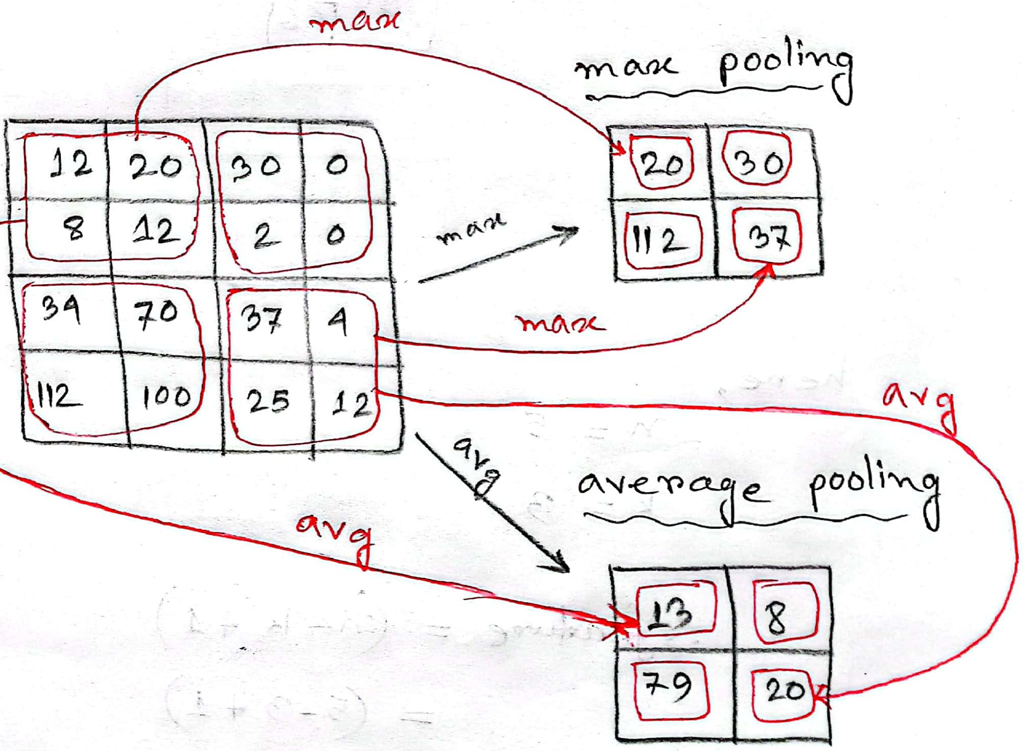
CNN will learn which Kernel should use.

Again, in practical life we will have a large size of image. So, for that we use Pooling. This is also a part of Convolution.

Pooling Layer:

(a) max pooling

(b) average pooling



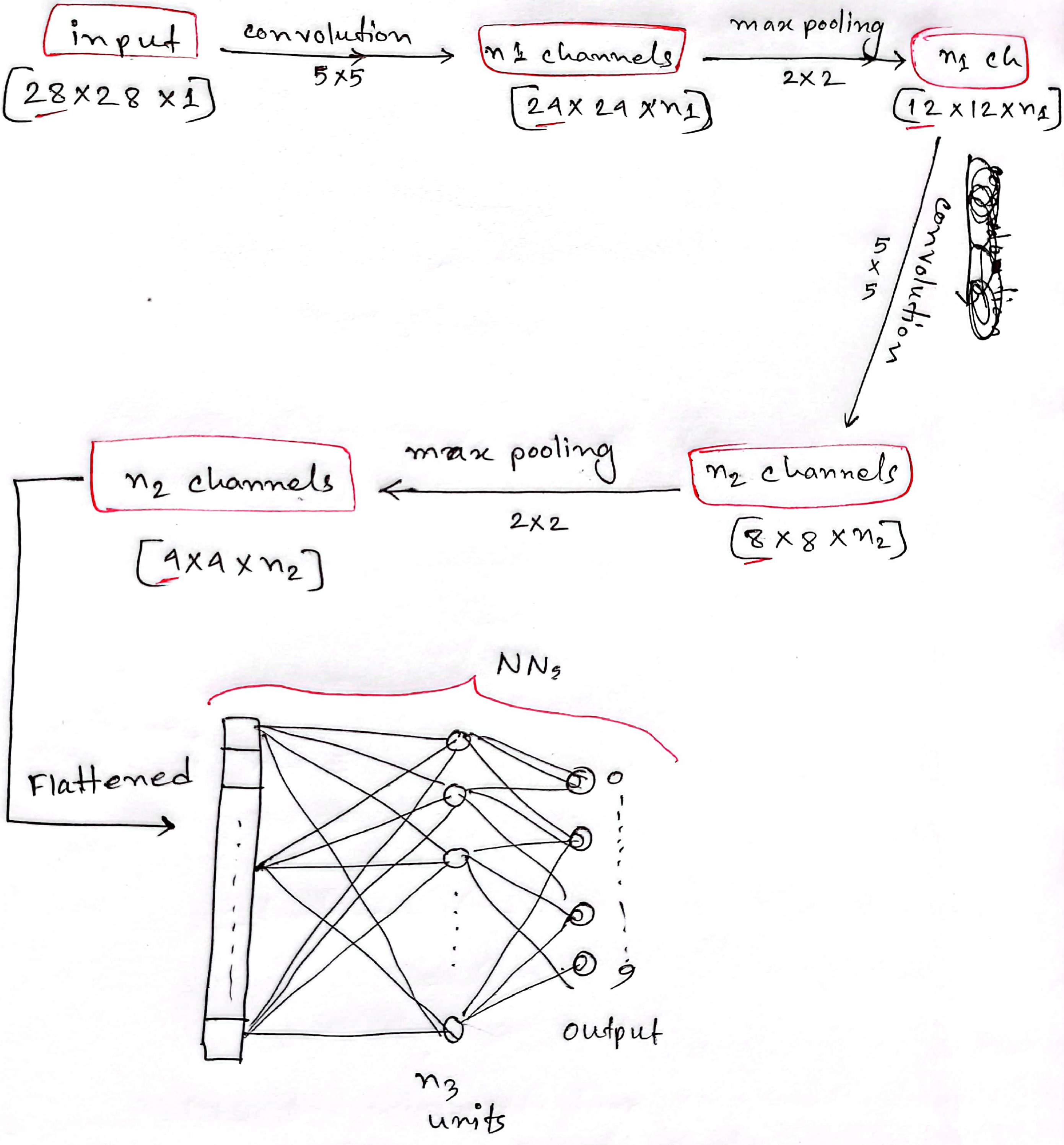
12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

13	8
79	20

→ By doing pooling we also reduce the number of parameters in case of Large Image. Also, retains major information.

→ Architecture of CNNs:


input
[28×28 ×1]
convolution
5×5
n1 channels
[24×24×n1]
max pooling
2×2
n1 ch
[12×12×n1]
convolution
5×5
n2 channels
[8×8×n2]
max pooling
2×2
n2 channels
[4×4×n2]
Flattened
NNs
0
9
output
n3
units