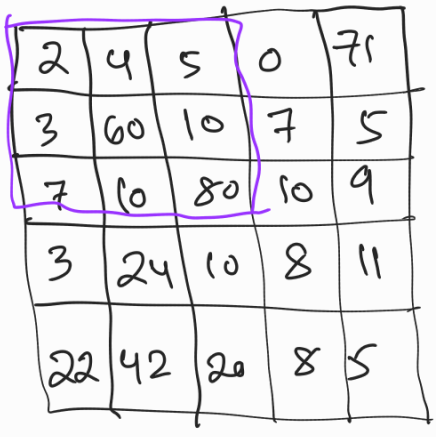
Q1. Perform the following convolution ?

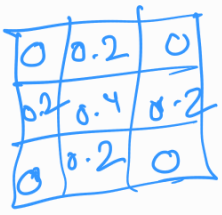


2	4	5	0	71
3	60	10	7	5
7	10	80	10	9
3	24	10	8	11
22	42	20	8	5

5x5 input

(X)

$*$



0	0.2	0
0.2	0.4	0.2
0	0.2	0

3x3 filter

(W)

Ans: You should do the full calculation either by Hand or using a program.

I am just doing the first window calculation: (marked in this color)

3x3

$*$

3x3

$=$ $0.2 \times 0.4 + 0.2 \times 0.3 + 0.4 \times 60 + 0.2 \times 10 + 0.2 \times 10$

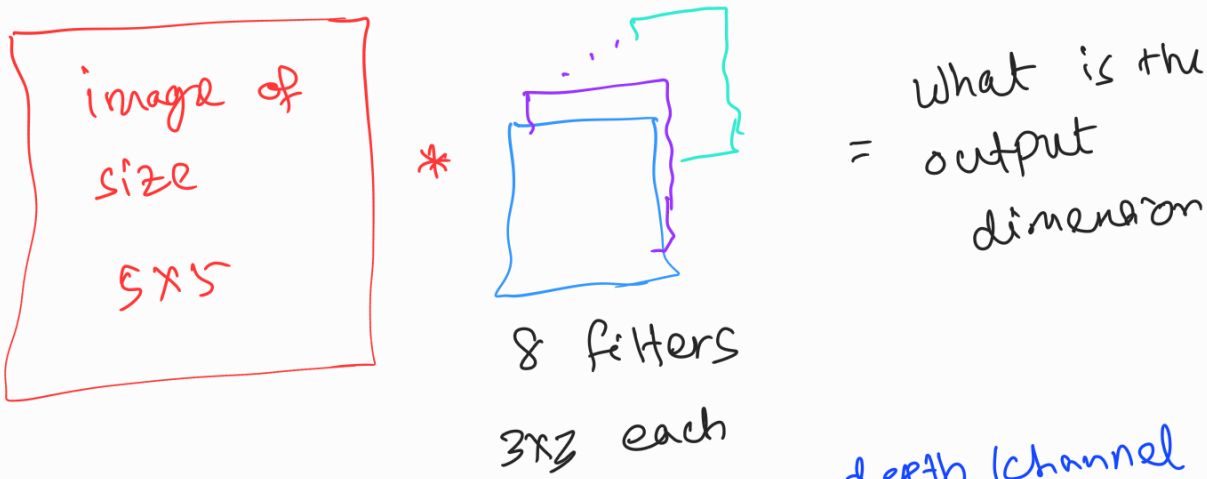
$=$  a scalar value

1x1

Q2. what is the final output dimension (assume stride = 1 on both X, Y axis)

Ans: 3x3 (verify this)

Q.3



Ans: Each filter is 3x3x[1] ← depth (channel)

produces a feature map. (output) see Q1.

8 filters will produce [8] feature maps.

Q4 Read about batch normalisation and how it helps Deep network training (lowers no. of epochs required, helps generalising better).

Hint: consider a numerical with batch size 3. Each feature map dimension is 3x3. Calculate

z-score $z = \frac{x - \mu}{\sigma}$ ... [read]

Q5 Solve some numericals on

* average pooling
* max pooling.
* L_2 - norm pooling

Hint: Consider a convolution output of dimension 5×5 and using pooling size of 3×3. what will be the output dimension ?

What are differences between these three outputs.

— x — (more on next page)

If you are wondering why my hand-writing is so bad, I want say that I can really write nicely on real-black board.

This is my new tablet (expensive one :-)) and I am learning to write on this:

Convolution with tensors as input:

That is: $\underline{X}$ is not just $h \times w$

h: height
w: width.

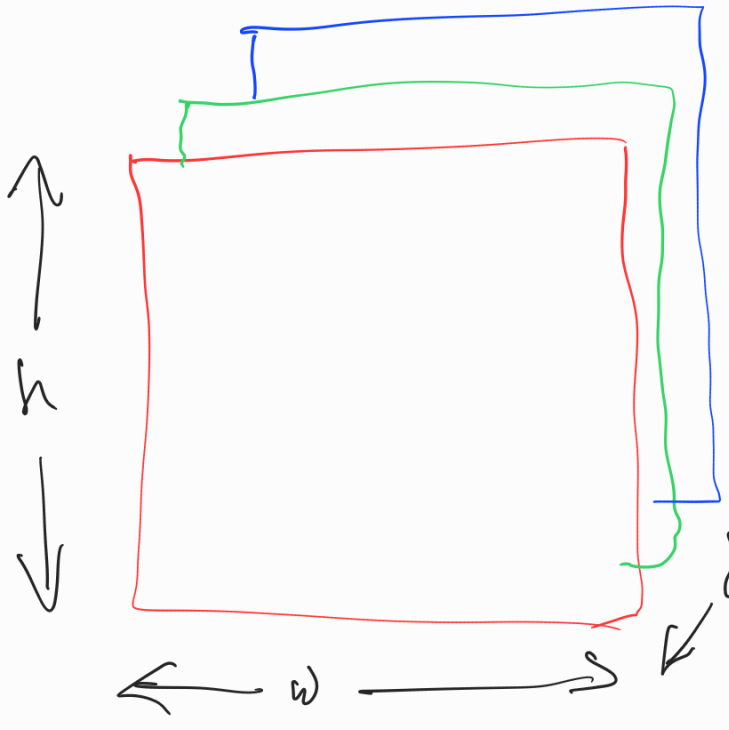
It has a depth as well.

i.e. $[\underline{X}]$

$h \times w \times d$.

E.g. Consider an image with 3-channels

$\Rightarrow$ it has 3-depth (red, green, blue)



Now, a filter cannot be $h_1 \times w_1$

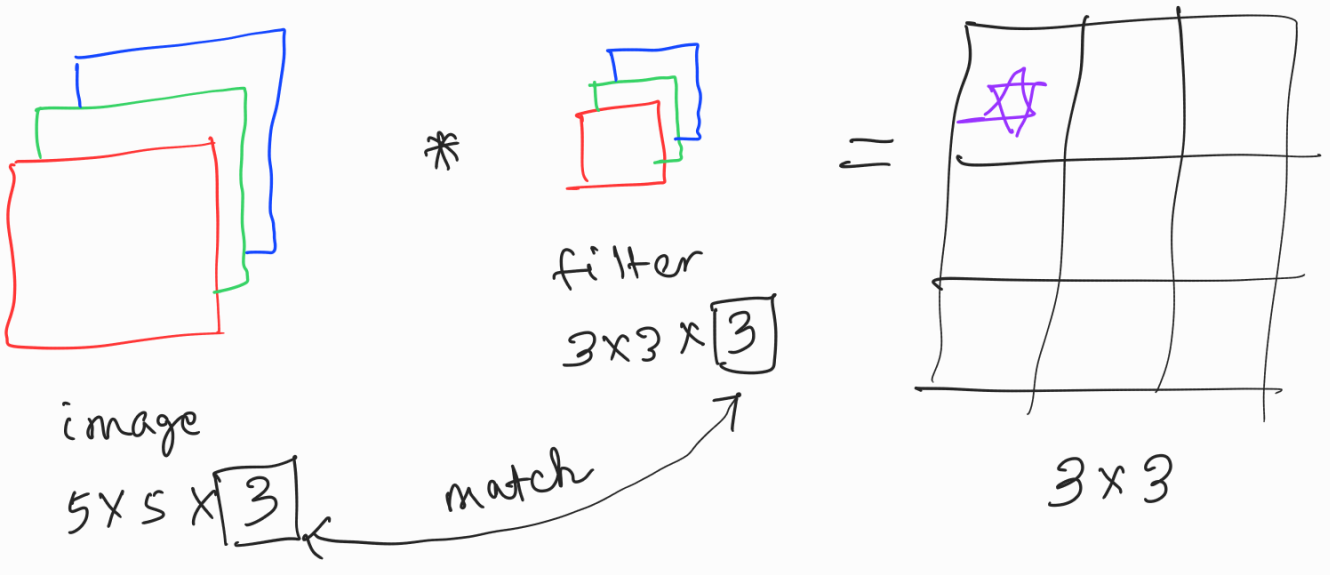
it MUST BE $h_1 \times w_1 \times d$

must match.

So, now a single filter will have a depth as well, which is equal to the depth of the input.

There can be several such filters. For ex: 32 filters of dimension $h_1 \times w_1 \times d$.

Let's now look at how to do the convolution with depth now:



✡ : 

produces a single number n_1 + 

produces a single number n_2

+ 

produces a single number n_3

$= n_1 + n_2 + n_3$