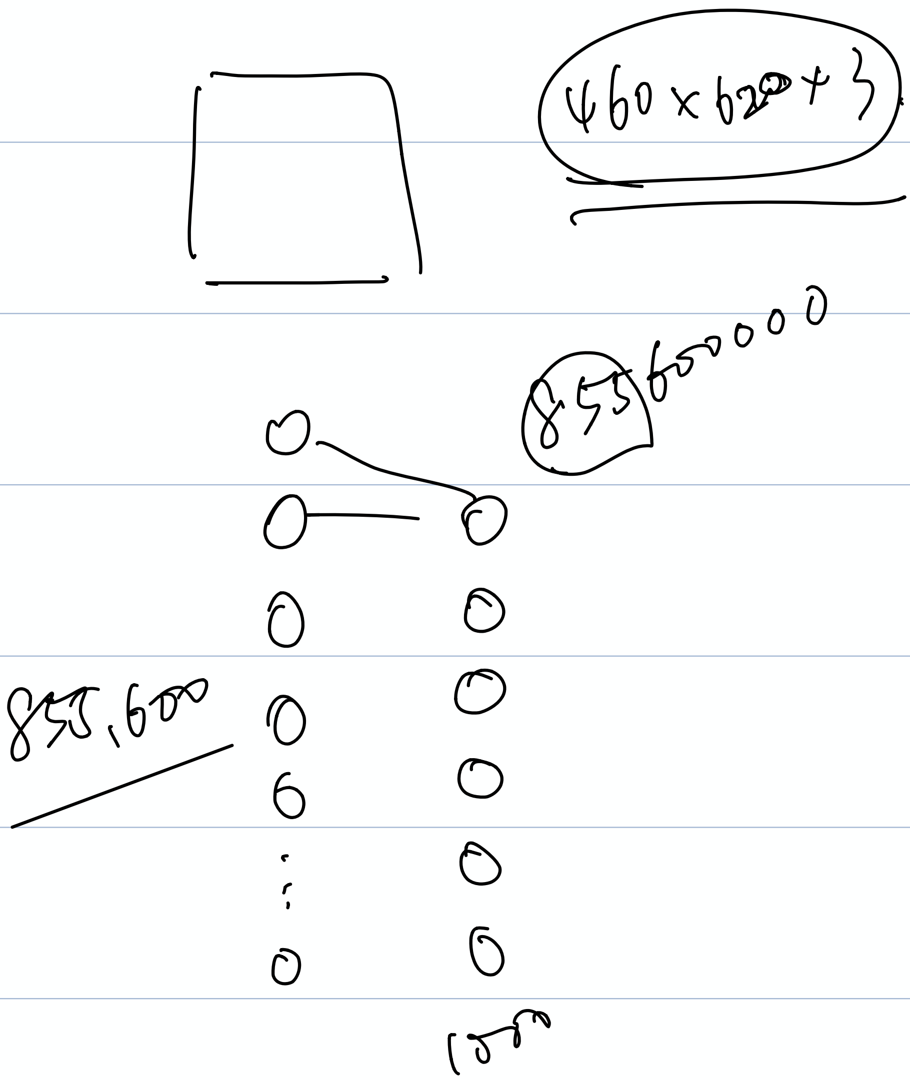
Why use convolutions?

64x64x3 image
1000x1000x3 image

} what will happen if we flatten them and only use fully connected layers?

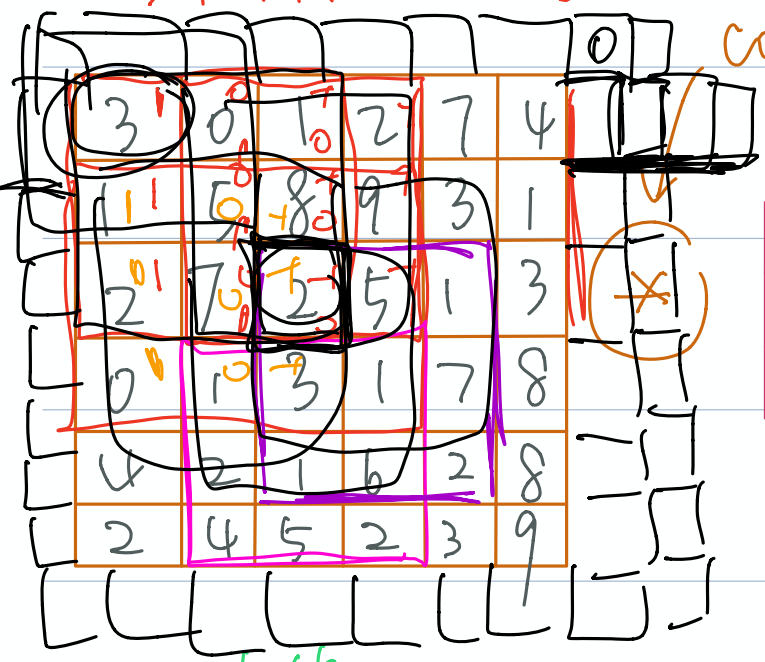


Filter and kernel

vertical edge detection

$3\times1+1\times1+2\times1+0\times0+5\times0+7\times0+1\times(-1)+8\times(-1)+2\times(-1)=-5$

convolution



6×6

$*$

1	0	-1
1	0	-1
1	0	-1

filter
kernel

3×3

$=$

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

4×4

also
3
10
3
Scharr filter

1	0	-1
2	0	-2
1	0	-1

Sobel filter

also can detect vertical edges

3×3
5×5
9×9
1×1

45° 30°
45° and 30°

we can learn filters that better suit the task

Padding

$(6\times 6) * (3\times 3) \longrightarrow (4\times 4)$

$(n\times n) * (f\times f) \longrightarrow (n-f+1 \times n-f+1)$

reason: ① shrinks for large model

② the edge pixels are given less weight

$p=0$

$6\times 6 \xrightarrow{\text{pad}} (8\times 8) * (3\times 3) \longrightarrow 6\times 6$

$n+2p \times 2p+n \quad f\times f$

$(n+2p-f+1 \times n+2p-f+1)$

valid conv: no padding

valid same.

Same conv: output size = input size.

$p = \frac{f-1}{2}$

also because there is a central filter

f is usually odd

Strided Convs:

$(n \times n) * (f \times f)$

padding : p

stride : s

floor operation

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

filter need to be entirely inside

$\lfloor 2.5 \rfloor$

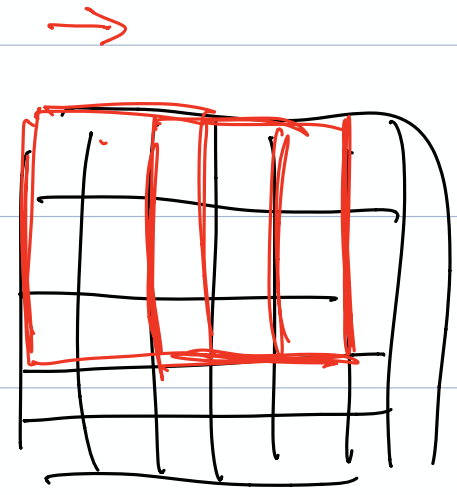
=

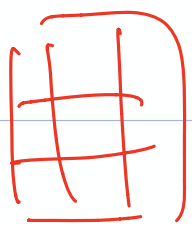
2

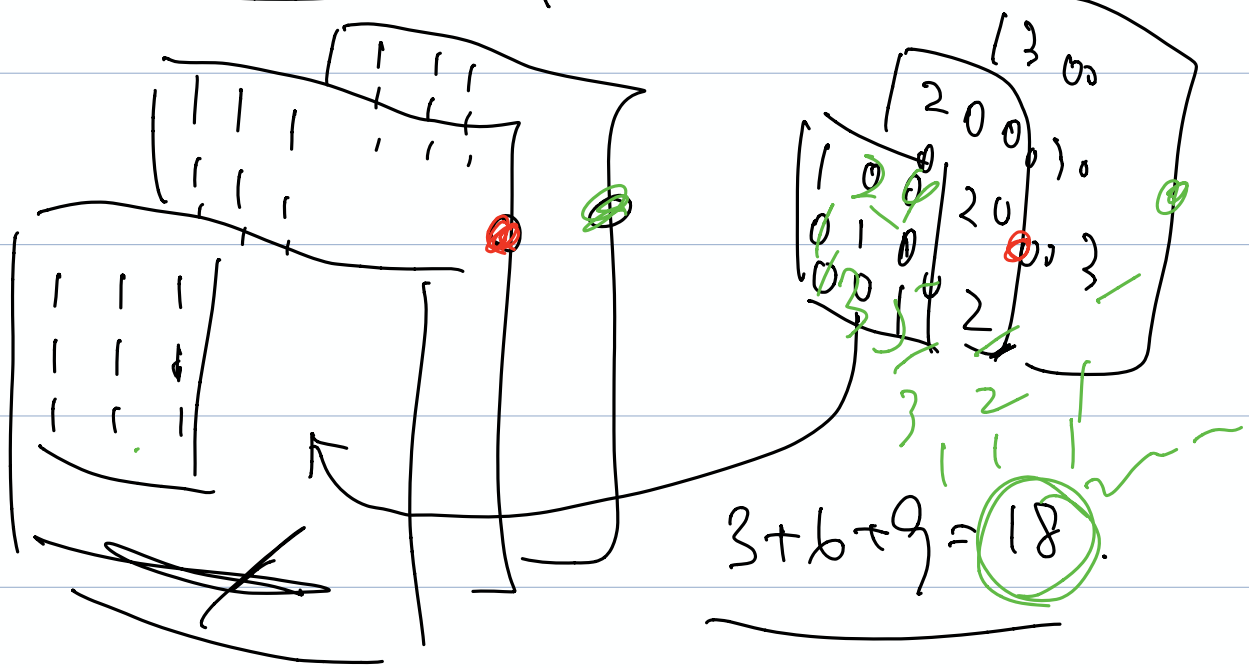
$\lfloor 2.9 \rfloor$

=

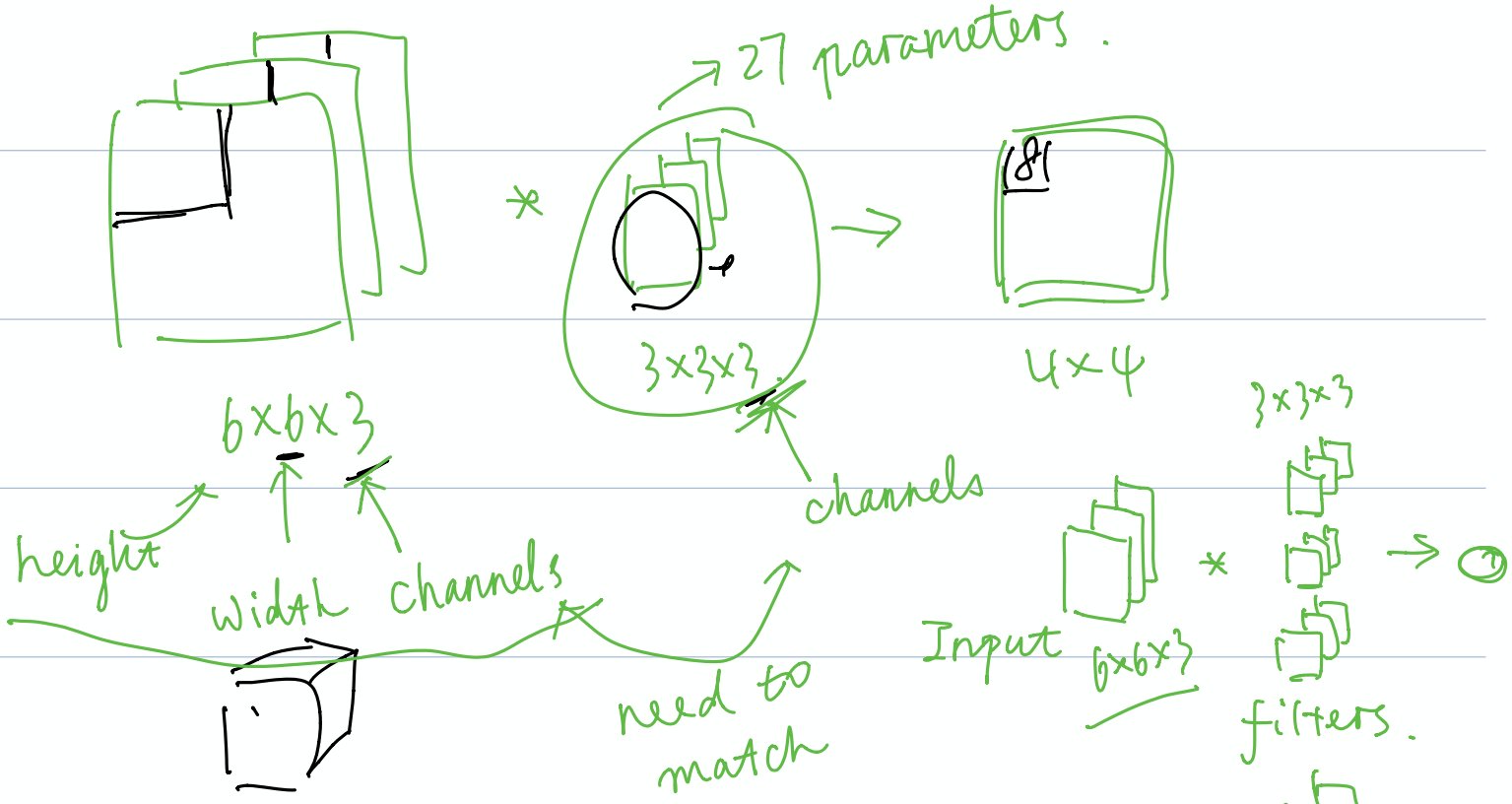
2







Conv over volumes:



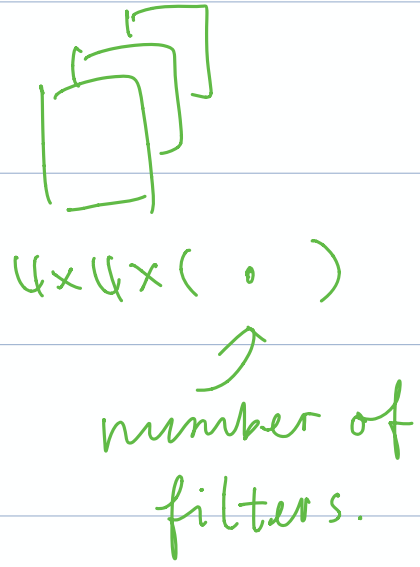
We can have multiple filters.

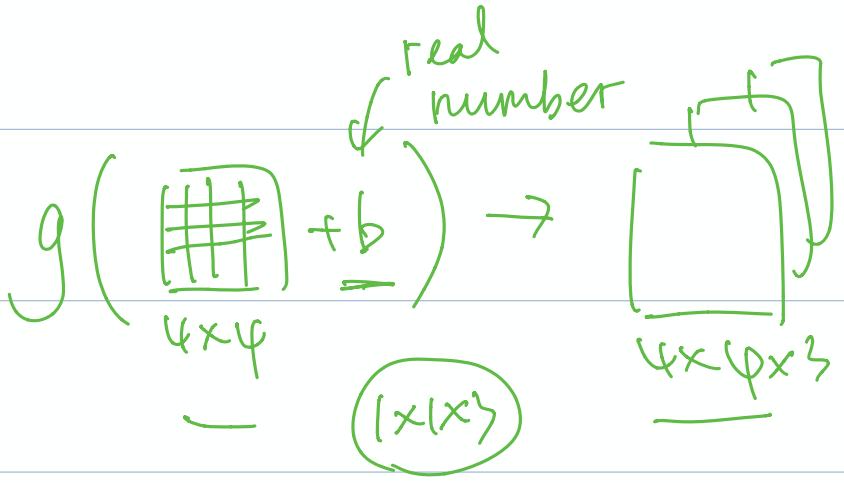
and output will be

$(n \times n \times n_c) * (f \times f \times n_c)$

$(f \times f \times n_c)$ ⋮ $(f \times f \times n_c)$ — $\#n_c'$

↳ $(n-f+1 \times n-f+1 \times n_c')$





$$g\left(a^{[0]} * w^{[1]} + b^{[1]}\right) = a^{[1]}$$

10 filters of 3x3x3 → 280 parameters

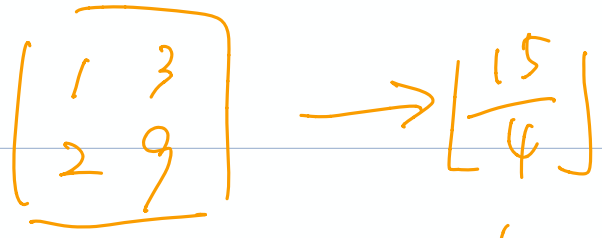
$f^{[l]}$ = filter size

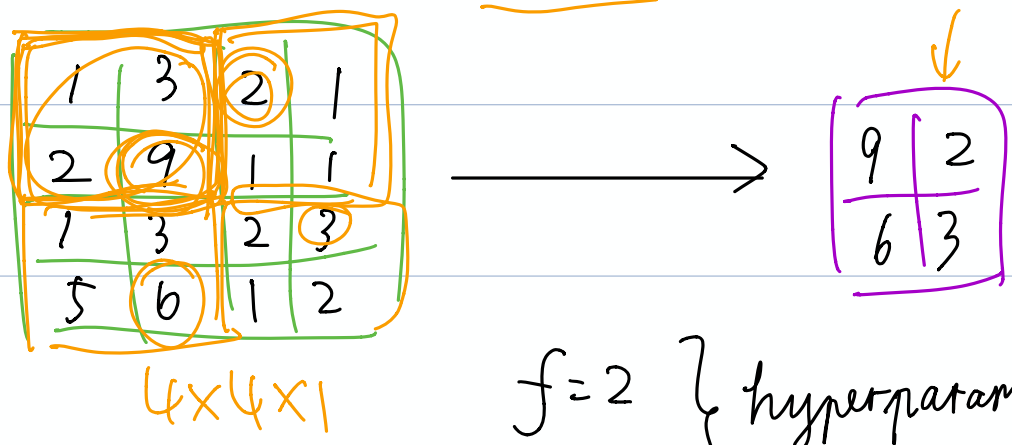
$p^{[l]}$ = padding

$s^{[l]}$ = stride

$n_c^{[l]}$ = # of filters

Pooling layer.



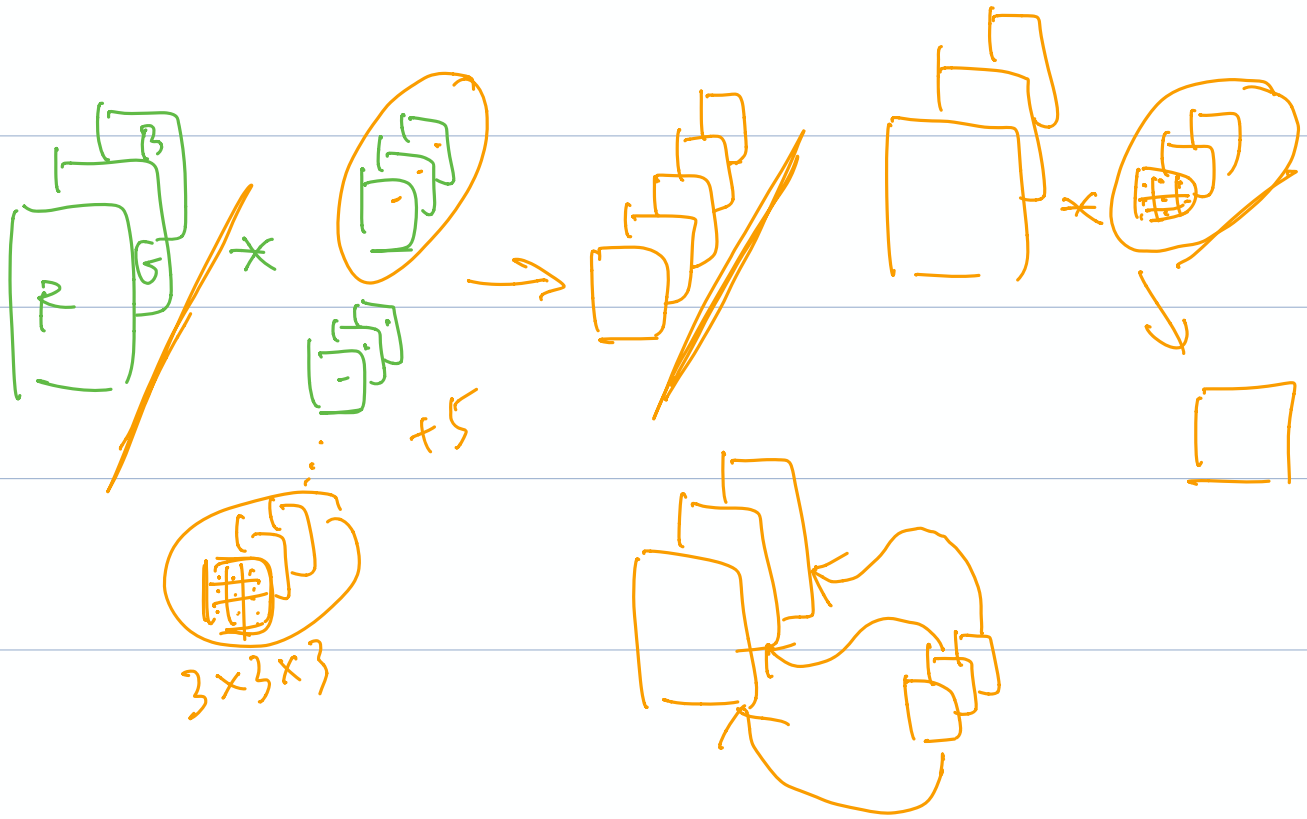


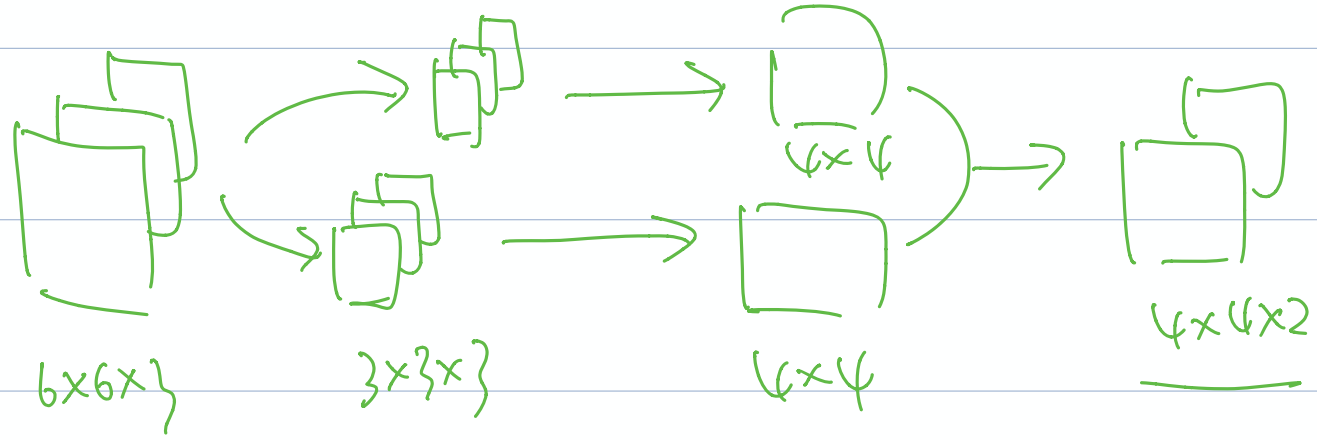
4×4×1

$f=2$
$S=2$
} hyperparameters

No parameters to learn

Average Pooling





ReLU $\left(\begin{array}{|c|c|c|} \hline 0 & -1 & 2 \\ \hline 3 & 4 & 5 \\ \hline -6 & 10 & 12 \\ \hline \end{array} + 1 \right) =$ $\begin{array}{|c|c|} \hline 1 & \\ \hline & \\ \hline 0 & 13 \\ \hline \end{array}$

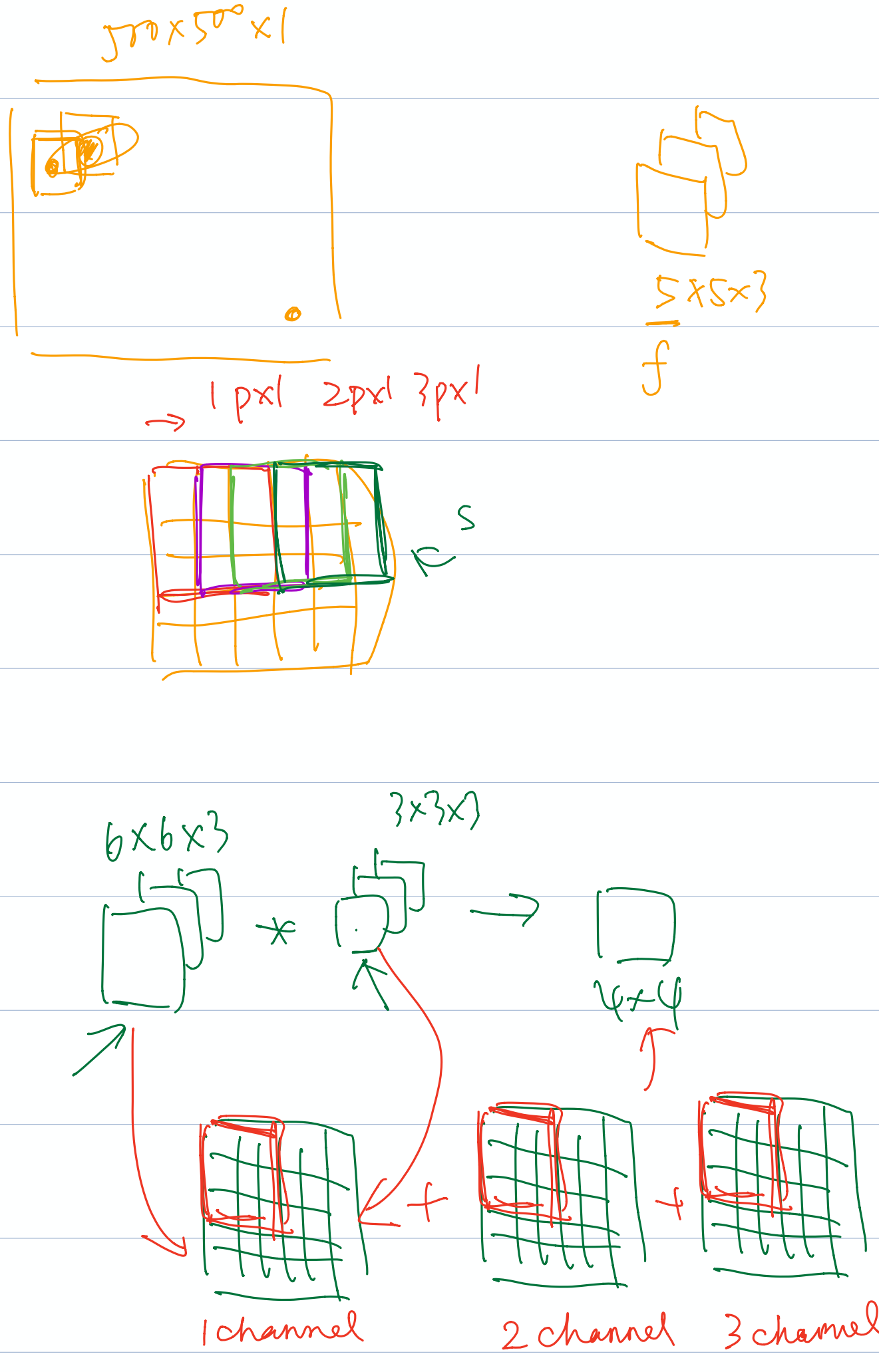
$$\begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$$

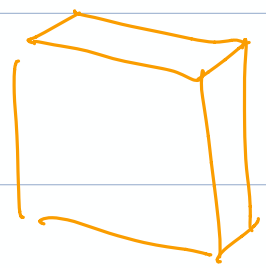
27

||

10 filters $3 \times 3 \times 3$

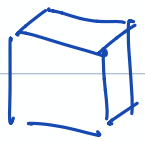
$27 \times 10 + 10 = 280$


5x5x3
f
1 pxl
2pxl
3pxl
s
6x6x3
3x3x3
4x4
1 channel
2 channel
3 channel



$39 \times 39 \times 3$

$*$

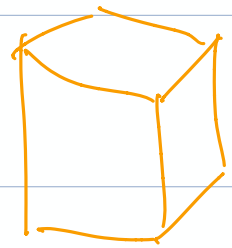


$3 \times 3 \times 3$

$S = 1$

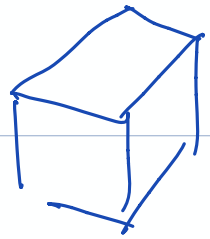
$P = 0.$

10 filters



$37 \times 37 \times 10$

$*$

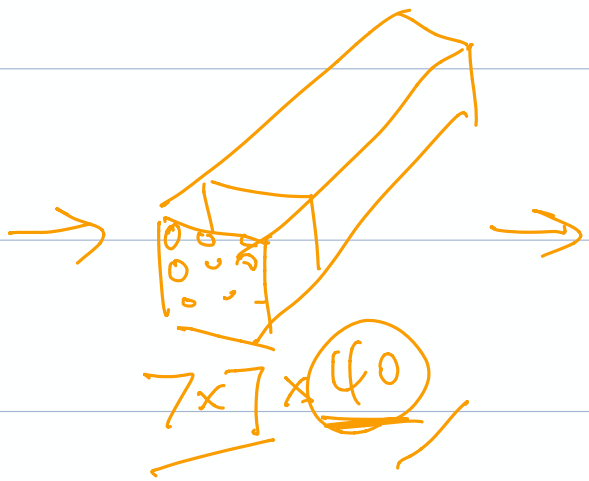


$5 \times 5 \times 10$

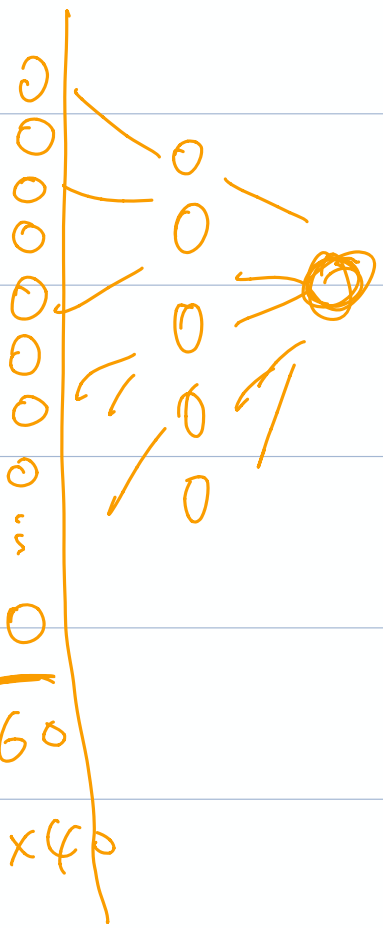
40 filters

$S = 2$

$P = 0$

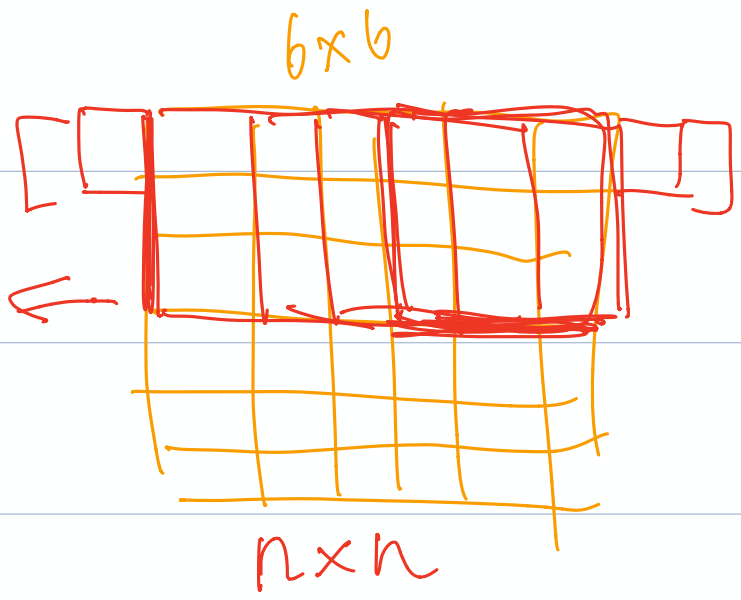


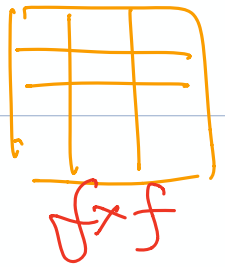
$7 \times 7 \times 40$



1960

$= 49 \times 40$





$(n-f+1)\times(n-f+1)$

$(n+2p)-f+1$ $\quad 6-3+1=4$

$p=1$ $\quad 8\times 8$ $\quad 3\times 3$ $\quad 8-3+1=6$

$\left\lfloor \frac{8-3}{2}+1 \right\rfloor =$ 2.5+1 $\left\lfloor \frac{n+2p-f}{s}+1 \right\rfloor$