Padding & Strides

Why we need padding?

(Image)

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

(5x5)

*

(filter)
(3x3)

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

=

(feature map)

6	-9	-8
-3	-2	-3
-3	0	-2

(3x3)

$2\times1 + 5\times1 + 3\times1 + 3\times0 + 2\times0 + 3\times-1 + 8\times-1 + 8\times-1 = -9$

2 problem

1. Feature map (3x3) size is small as compare to Image after using filter. If we use filter again on feature map then size of feature map will be reduce as compare to pervious feature map. (jitne convolutional layer (filter) apply hoge har layer ke bad utna size chhota hoga)
   Instant → loosing Data

2. Boader pixel (green Area) → less part in convolation as compare to middle part (orange)

   for eg:- pixel(0,0) → 7
   ↳ be only 1 time in convolution part

   pixel (3,3) → 2
   ↳ be multiple time in convolution part

So, In feature map → middle pixels → more important

border pixel → less importance

These two problem solve padding.

What is padding?

In Convolution

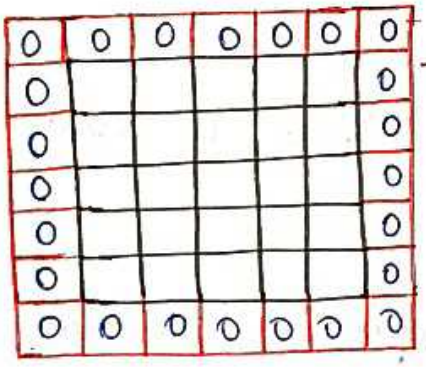
Image	filter	feature map
$n \times n$	$f \times f$	$(n-f+1)\ (n-f+1)$
5×5	3×3	$(5-3+1)\ (5-3+1)$
		3×3

but we want $n-f+1 = n$ → so Image Size = = feature map size

→ Not change in filter

→ change in Image size in padding

Add column and Row



→ Added column and Row And this boundary is called padding. Value of padding is 0.

After Padding ↴

→ Padding

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Image (5×5)
with padding (7×7)

*

0	-1	0
-1	5	-1
0	-1	0

Kernel
(3×3)

=

(5×5)
feature map

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

$5+2(1)-3+1=5$

↳ Padding

Keras

- Valid → Normal convolutional layer
- Same → Padding

Strides

1 pixel

pixel 1 move → filter move and reach at last pixel

pixel move right →

0	0	0	0	0	0	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

filter

1 pixel

(7×7)
Image

0	0	1
1	0	0
0	1	1

(3×3)
filter

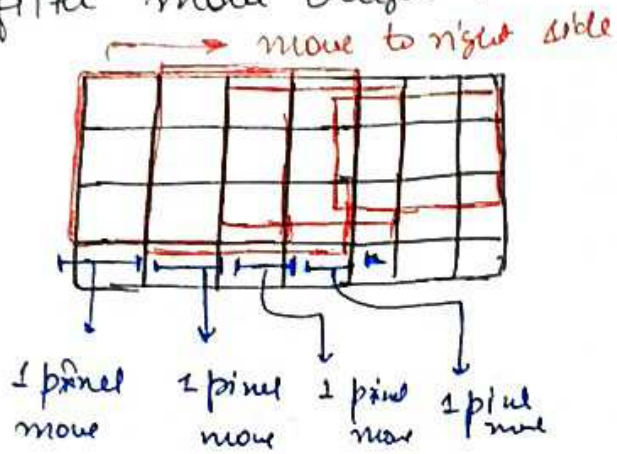
Stride = (1, 1)

- 1 pixel move right →
- 1 pixel move down ↓

→ filter move 1 pixel down ↓

Strides = (1, 1)

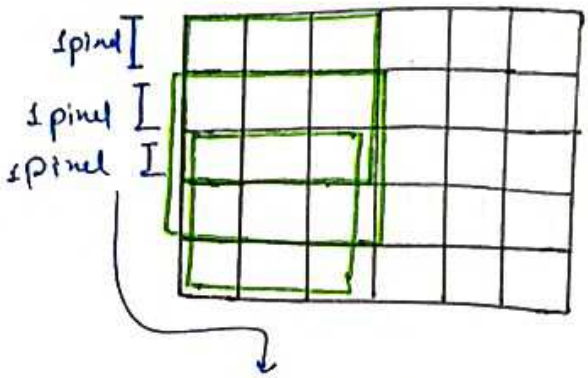
filter move right →



Strides = (1, 1)

right → pixel

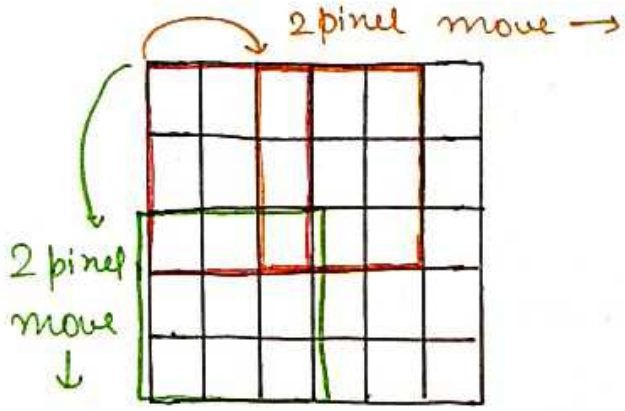
filter move down ↓



Strides = (1, 1)

move down ↓

We can change strides as well. and also write Strides = (2,2)



Stride = (2, 2)

move 2 pixel right →

move 2 pixel down ↓

feature map $(n-f+1) \rightarrow \left[\frac{n-f}{s} + 1\right]$ → with stride
→ without padding

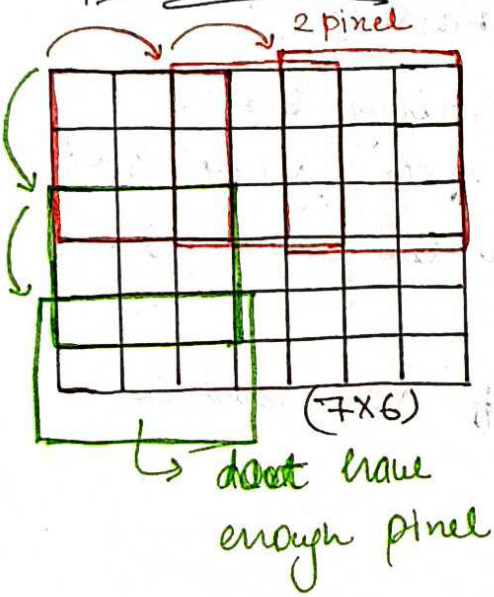
with padding and Strides → $\left[\frac{n+2p-f}{s} + 1\right]$

Strides value ↑↑ = feature map size ↓↓

Which means information loss while using stride or more than 1 stride.

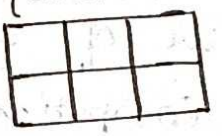
If stride value more than 1 is called strided convolution.

Special Case



→ let Stride = 2 → filter skip 2 pixel

feature map



→ because of don't have enough pixel

$$\left[\frac{n-f}{s}+1\right] \qquad \left[n-f+1\right]$$

$$\frac{6-3}{2}+1 = (1.5)+1 \qquad \frac{7-3}{2}+1 = 2+1$$

floor rule

$= 1$ $\qquad = 3$

2 $\qquad$ (2×3)

Why strides are required?

Reasons →

1. When you just want high level feature and not want low level feature.

low level feature means capture details of the image

↳ when stride value is small.

for example → We are working on computer vision problem and I only want some important feature (Not in detail feature) so, I use higher value of stride.

If I want details feature of image then I use lower value of strides.

2. Computing Power → Training Fast

Increase value of stride for Training Fast but Now, Computing Machine specs very good So, we can use default value (stride=1).

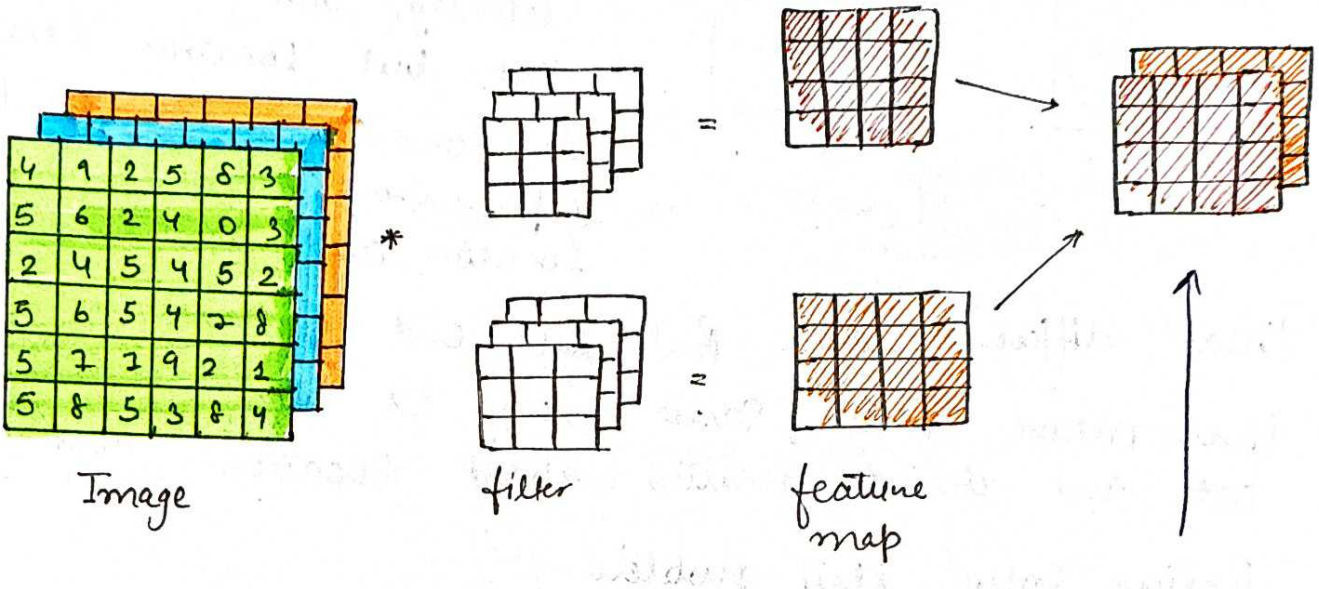
* We can give different value of row and col

stride = (2, 1)

Pooling Layer

The problem with convolution

1. Memory Issue
2. Translation Variance



let

Image [228 x 228 x 3]

filter → 100

feature map → (226 x 226) x 100

Store in 32bit float storage

1 batch for 1 training ← 19MB

If batch = 100

↳ 1.59 GB → we can not store in storage

* We need to reduce size of feature Map.

Possible solutions

1. Increase the value of stride to reduce size of feature map.

2. Pooling solve → Memory Prob
↳ Translation Variance

Translation Variance

↳ dependent on location

location changed

Let say, both are same cat but location changed. In convolution layer → tied up with (dependent) on location and both image treat different way. But our end goal is to both picture treat same way cz both are same cat and doesn't matter about location. And Pooling solve this problem.

↳ down sample your feature map.

Pooling

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

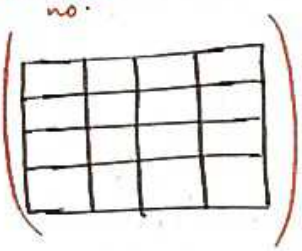
*

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

= relu

-ve → 0

+ve = same no.

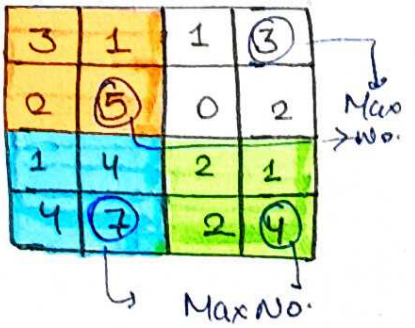


apply "relu" for non-linearity adding

After Non-linearity → apply pooling → famous (Max pooling method)

- Max pooling
- Avg pooling
- Min pooling
- L2 pooling
- Global pooling

let assume this featuremap



Size of pooling → (2,2)

Stride → 2

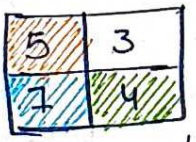
type → Max

→ Take first (2,2) box → Max Numbe = 5
↳ Orange box

→ Stride = 2 → Skip 2 pixel → reach box → Max no. = 3
↳ white box

→ Move down → Skip 2 pixel → reach blue box → Max no. = 7

→ Move right → Skip 2 pixel → reach Green box → Max no. = 4



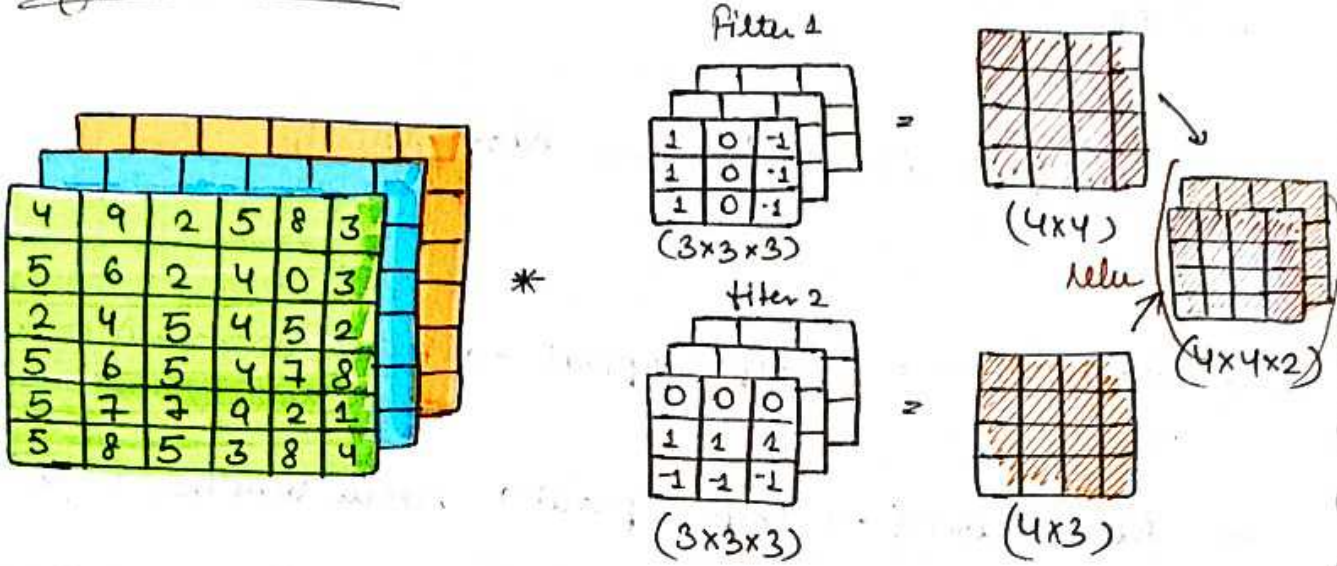
(2×2) ↳ featuremap (size reduce)

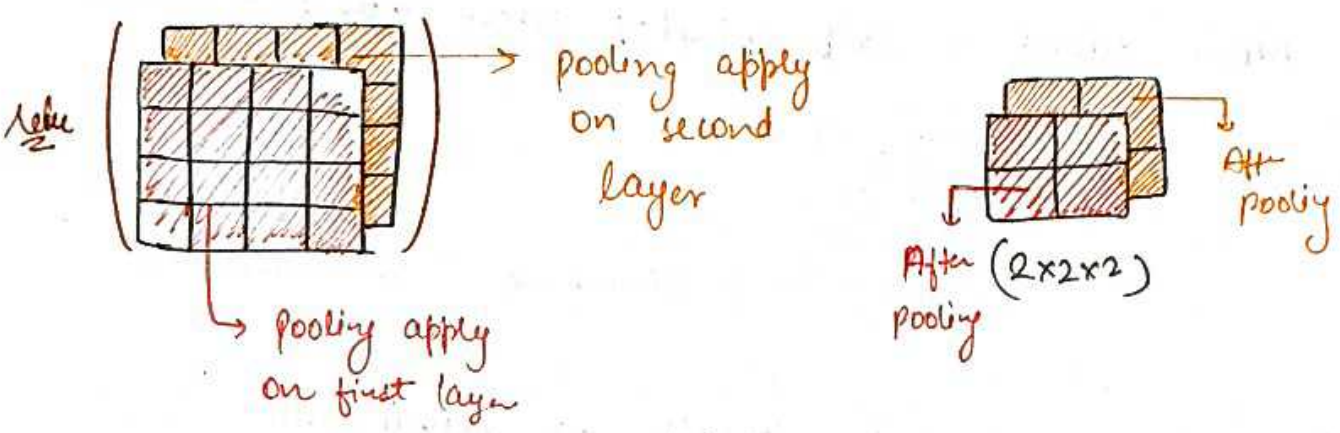
Pooling → Low level details → eliminate
= Max ↳ higher level details → Take

Pooling → Min pooling → extract Min no.
↳ Avg pooling → extract Avg of first box (4 pixel No.)
↳ L2 pooling → extract L2 Norm

for Demo and Visualization visit " deep lizard - Maxpool demo"

Pooling on Volumes





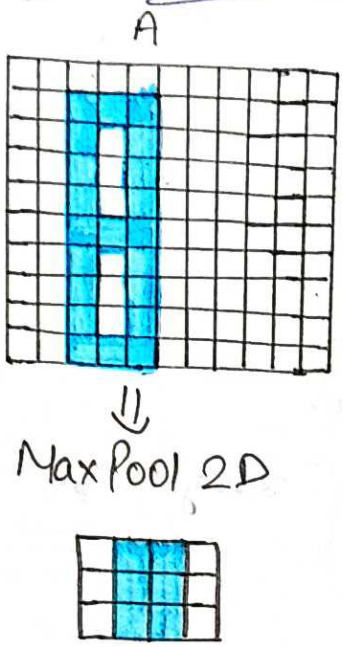
Advantage of Pooling

1. Reduced size

Image → filter → feature map → Pooling

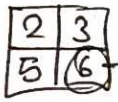
228 x 228 x 3 | 3x3x3, 100 filter | 226 x 226 x 100 | 113 x 113 x 100

2. Translation invariance



3 Enhanced Features (only in case of Max pooling)

Enhance edges of the ~~feature~~ Image



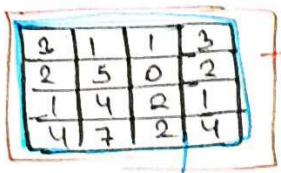
(feature enhance)

4. There's No need of training

Cause pooling → just agrigate the number or finding max number from feature map.

Types of Pooling

1. Max pooling
2. Avg Pooling
3. Global pooling
   a) Global Max
   b) Global avg Pooling



→ Find max of feature map → 7 (Global Max)

7

→ Find Avg of feature map

Disadvantage

Image Segmentation → require location

So, Pooling not need in Image segmentation