CNN Architecture

- modern CNN originate from LeNet
- Typical CNN have 2 stages
- stage 1: Conv + Pool layers. ← Feature transformer / finder
- stage 2: Fully connected layers ← non-linear image classifier.
  - must flatten 3D → 1D.

Pooling

- For downsampling / upsampling
  - output smaller image from larger image
  - eg: 512 x 512 → 128 x 128. (downsample by 3).
- Max vs Avg pooling
  - from filter, choose max / avg values.

1	2
5	6

← max = 6, avg = 3.5

1	2	3	4
5	6	7	8
16	15	14	13
12	11	10	9

→ max:

6	8
16	14

or avg:

3.5	5.5
13.5	11.5

Purpose:

1) less data to process
2) translation invariance (location does not matter)

Found, Not Found, Not Found, Not Found —max→ Found.

Pool size & Step

- Typically square.
- stride = steps to slide
- if stride < size, overlap occurs. (not common)

Conv-Pool Design

conv-pool → conv-pool → conv-pool

- conv-pool: simple lines & edges
- conv-pool: more complex features (eyes, lips)
- conv-pool: entire face, beard etc.

- After each conv-pool, image shrinks
- filter sizes stays the same, typically 3x3, 5x5, 7x7
- ∴ After each conv-pool layer, filter search for increasingly large patterns.
- As image shrinks, no. of feature maps ↑
  - spatial info ↓
  - No. features ↑

CNN Hyperparams

- Filter size & stride
- No. feature maps

Pool size & stride } ∃ existing conventions to follow.
1) follow conv-pool design
2) Increase No. feature maps 32 → 64 → 128.

Global max pooling 2D

- To allow for input image of different shape to be flattened.

  eg: Input 1: $32 \times 32 \times C_2$
  Input 2: $64 \times 64 \times C_2$
  } flatten does not work.

- Global max pooling always get output of $1 \times 1 \times C_2$
  - Takes max value of each feature map.

* Image cannot be too small eg 2×2.

Final activation fn:

Regression = none

Binary = sigmoid

multi = softmax