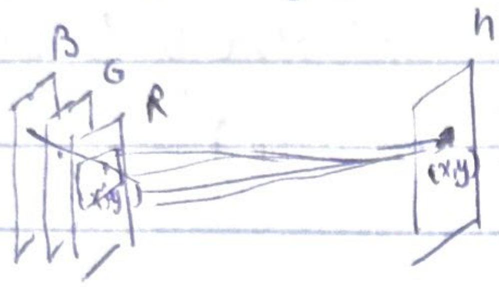
Convolution

in math. $(f*g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau$

In Convolutional NN:

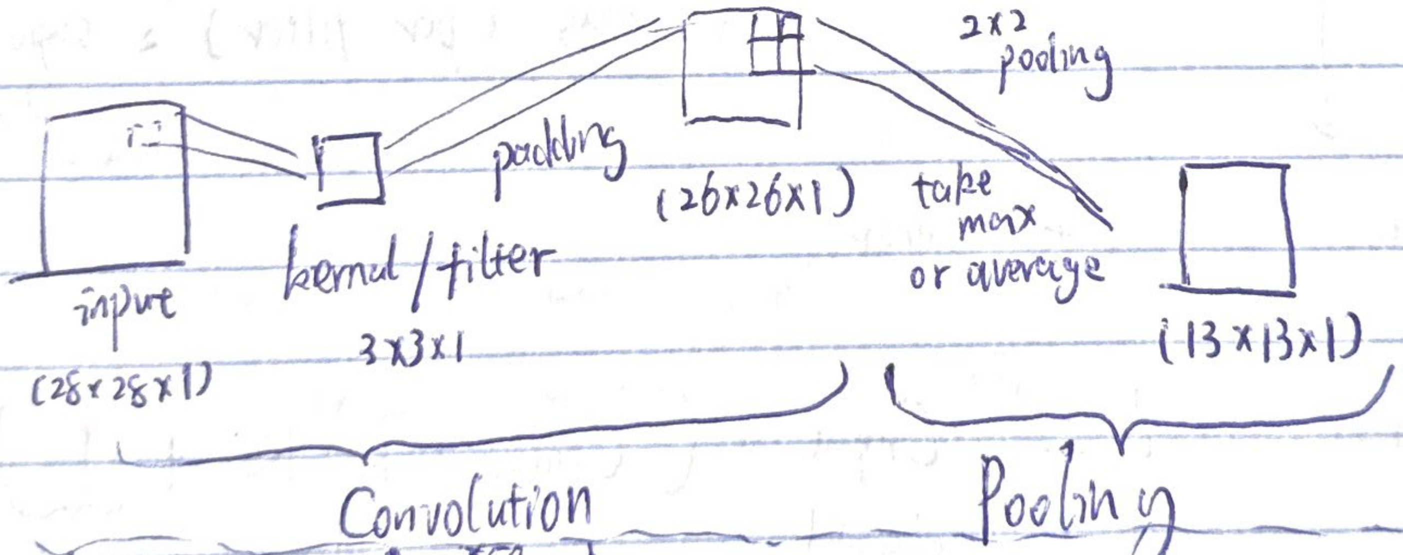
3 channels rgb

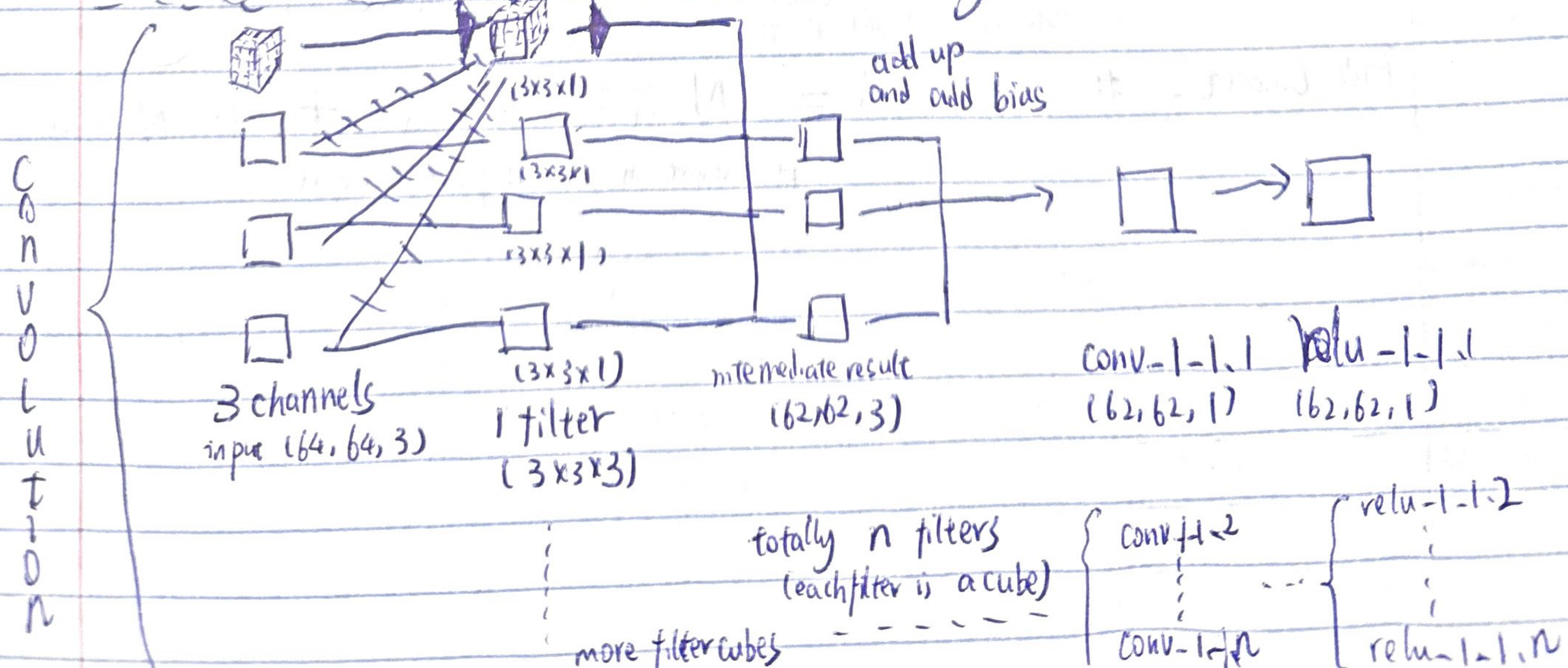


$$h(x,y) = \sum_{x'=x-1}^{x+1} \sum_{y'=y-1}^{y+1} I^{R}(x',y') W_h^{R}(x'y' \leftrightarrow xy)$$

$$+ \sum_{x'=x-1}^{x+1} \sum_{y'=y-1}^{y+1} I^{G}(x',y') W_h^{G}(x'y' \leftrightarrow xy)$$

$$+ \sum_{x'=x-1}^{x+1} \sum_{y'=y-1}^{y+1} I^{B}(x',y') W_h^{B}(x'y' \leftrightarrow xy)$$





Output Dimensions After Convolution

$$= \left\lfloor \frac{(W - F + 2P)}{S} \right\rfloor + 1$$

W = input width / height

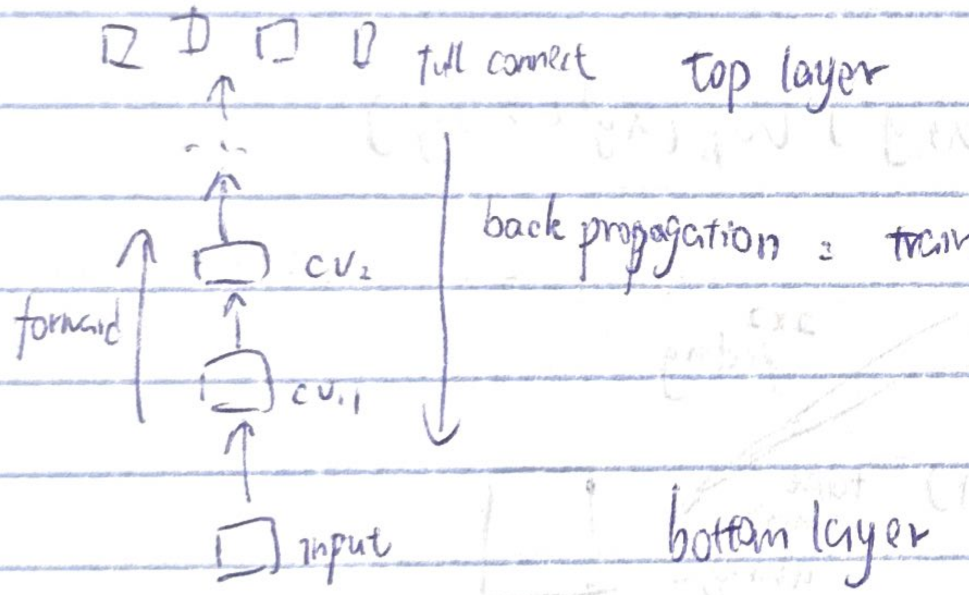
F = kernel / filter size

P = padding (0 for "valid" padding) add extra pixels outside existing input

S = stride (move the kernel by S each time)

Output Dimension After ~~Convolution~~ pooling

$$= \left\lfloor \frac{(W - F)}{S} \right\rfloor + 1 \qquad \text{typically } S = F$$



weight (per filters) = matrix # param = ~~[illegible]~~ → $C_{output} \cdot (K)^2$

bias (per filter) = single value to shift result to allow network fit better
# param = 1 (per filter)

Conv: # param trained $= \underset{\text{channel}}{C_{input}} \cdot (\underbrace{C_{output} \cdot K \cdot K}_{\text{kenel size}} + \underbrace{1}_{\text{bias term}})$

weight

Full Connect = # param trand $= \overbrace{\underset{\text{\# paramenta in input}}{N_{input}} \cdot \underset{\text{\# paramer in output}}{N_{output}}} + \overbrace{1 \cdot N_{output}}$