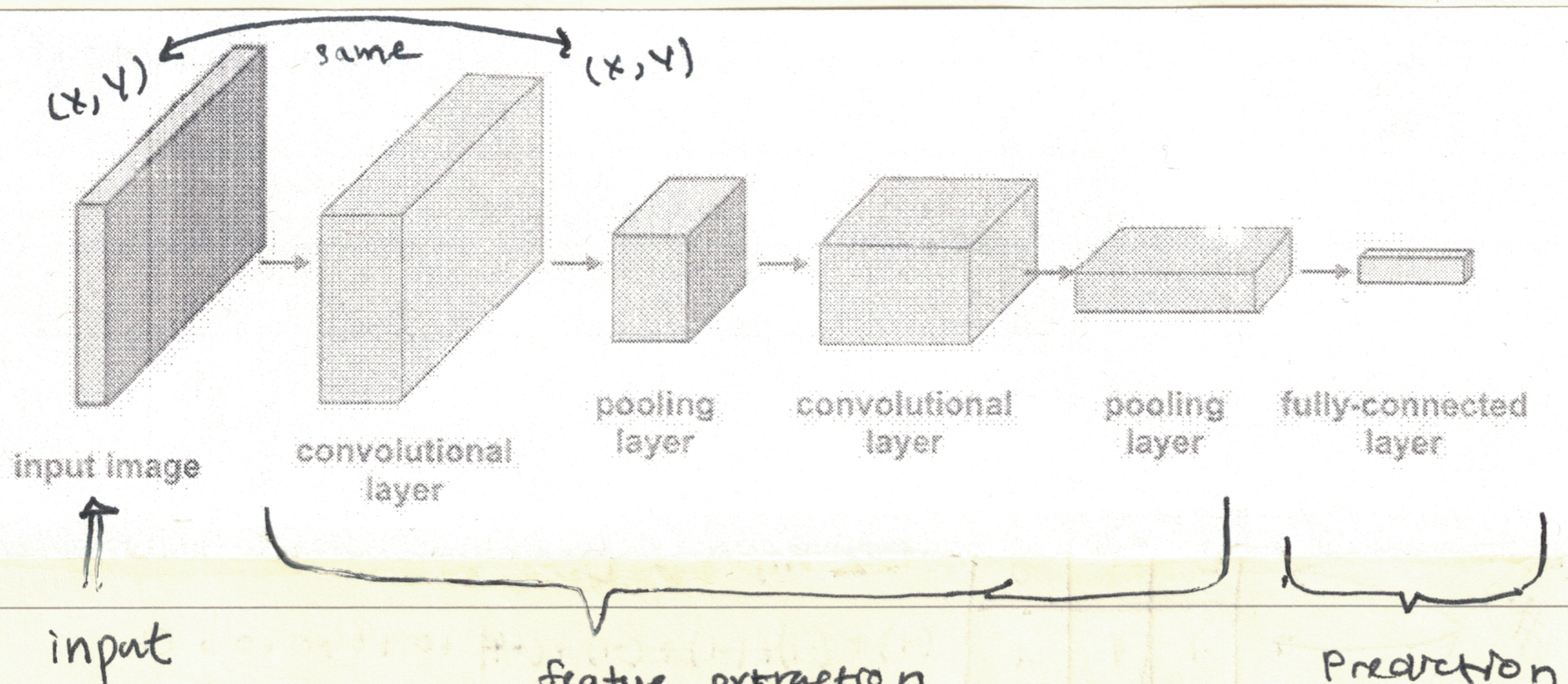


Isolation information from image)

4/3/2020

Transfer learning: implementing feature extraction from successful models and retraining the dense/fully-connected layer.

Dataset similarity

Dataset size	Similar	Different
Large	Fine-tune (2)	Fine-tune / retrain (3)
Small	End of ConvNet (1)	Start of ConvNet (4)

Cases:

1. Adjust end of ConvNet
   - ~~remove and replace~~ add dense layers (add) (not remove dense)
     - randomize dense layer weights → update weights
     - freeze weights in pretrained network - don't update weights
       ↳ prevent overfitting

Forward Pass: Propagation: All data travels forward through nodes to calculate the NN output

- Input Layer: Preprocessing are Hidden are non-Linearity
  - Equation:
    - * input: X · W → data → weight input to hidden
    - * Output: activation function(input) → sigmoid $\hat{y}$ are σ

- Hidden are output Layer: non-linearity / Linear
  - Equ:
    - * input: hidden output · W → output from hidden layer → weights from hidden to output layer
    - * output: activation / I ← or · input

Back propagation: data travels in opposite direction to fix/agust weights & bais

- Output layer:

  Equ input:

  Output error: $(y - \hat{y})$ ← target, output/prediction | label - prediction

  Output error term: $(y - \hat{y}) f'(a_k)$ | label - prediction (error) * derative of activation function of output (: gradient)

hidden: layer:

hidden error: $W_{jk}\delta$ are $W_{jk}\delta_k$ → weight, ← previous layer error term | Output layer terms · weights hidden to output

hidden error term: $\Sigma[W_{j,k}\delta_k] f'(h_j)$ | hidden layer error (error) * derative of activation function of hidden output (gradient)

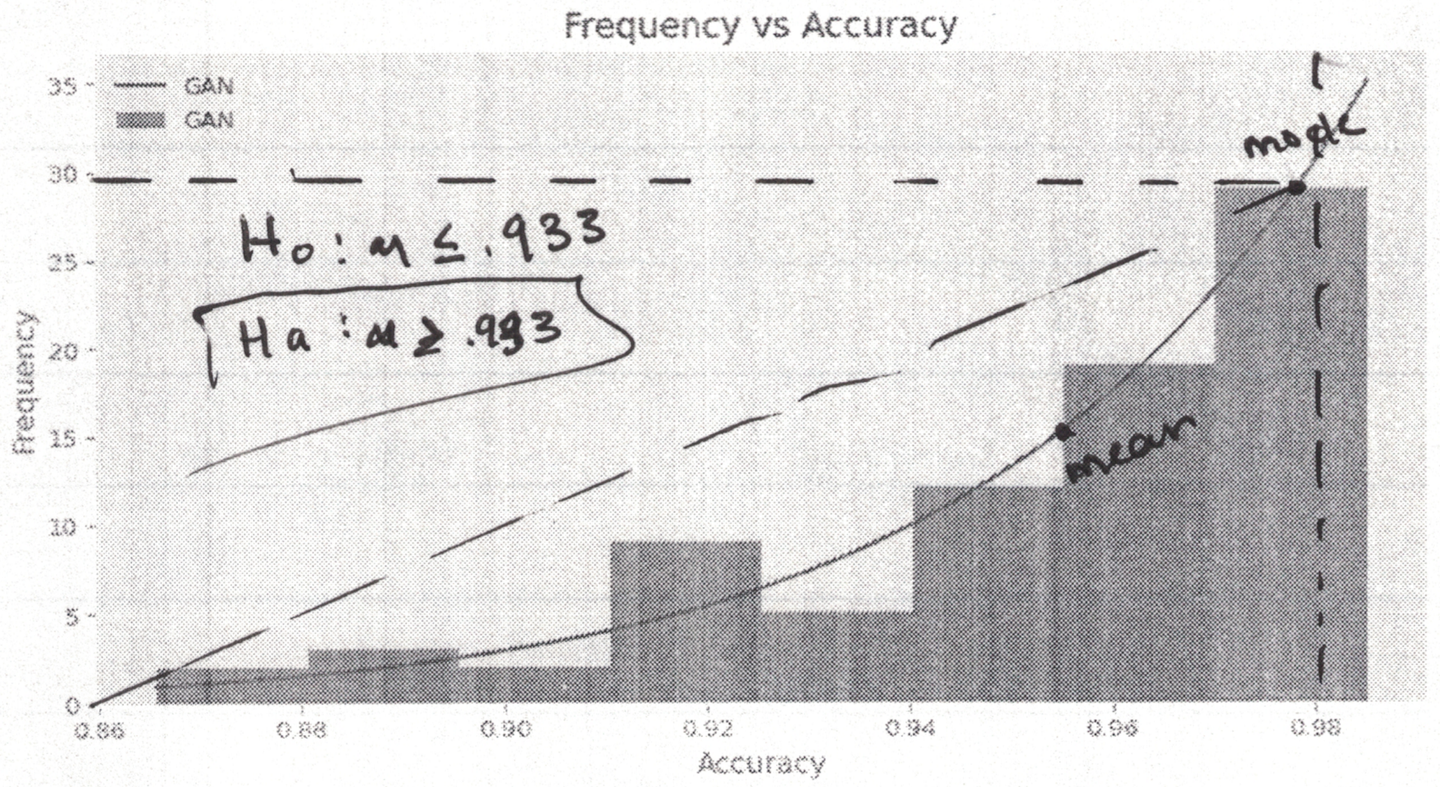
1/14/21

$\bar{X} = 0.9498$

$\sigma_x = 0.0293$

$t = \frac{0.9498 - 0.933}{\frac{0.0293}{\sqrt{149}}} = 6.99$

$P = 1.03 e^{-10}$



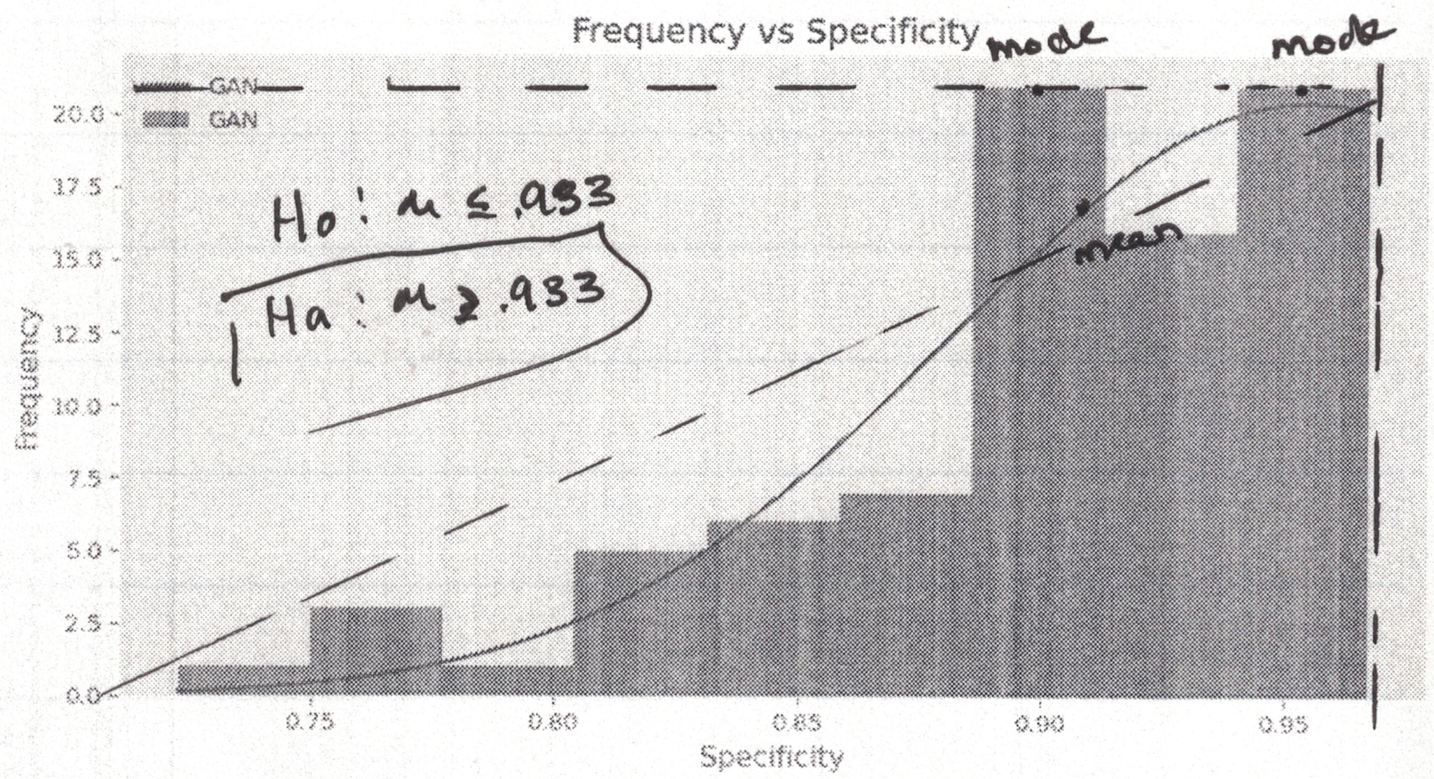
The distributions are subjected to the test, could happen that it is stable in performance

$\bar{X} = 0.9030$

$\sigma_x = 0.0547$

$t = \frac{0.903 - 0.933}{\frac{0.0547}{\sqrt{149}}} = -6.69$

$P = 1.65 e^{-10}$



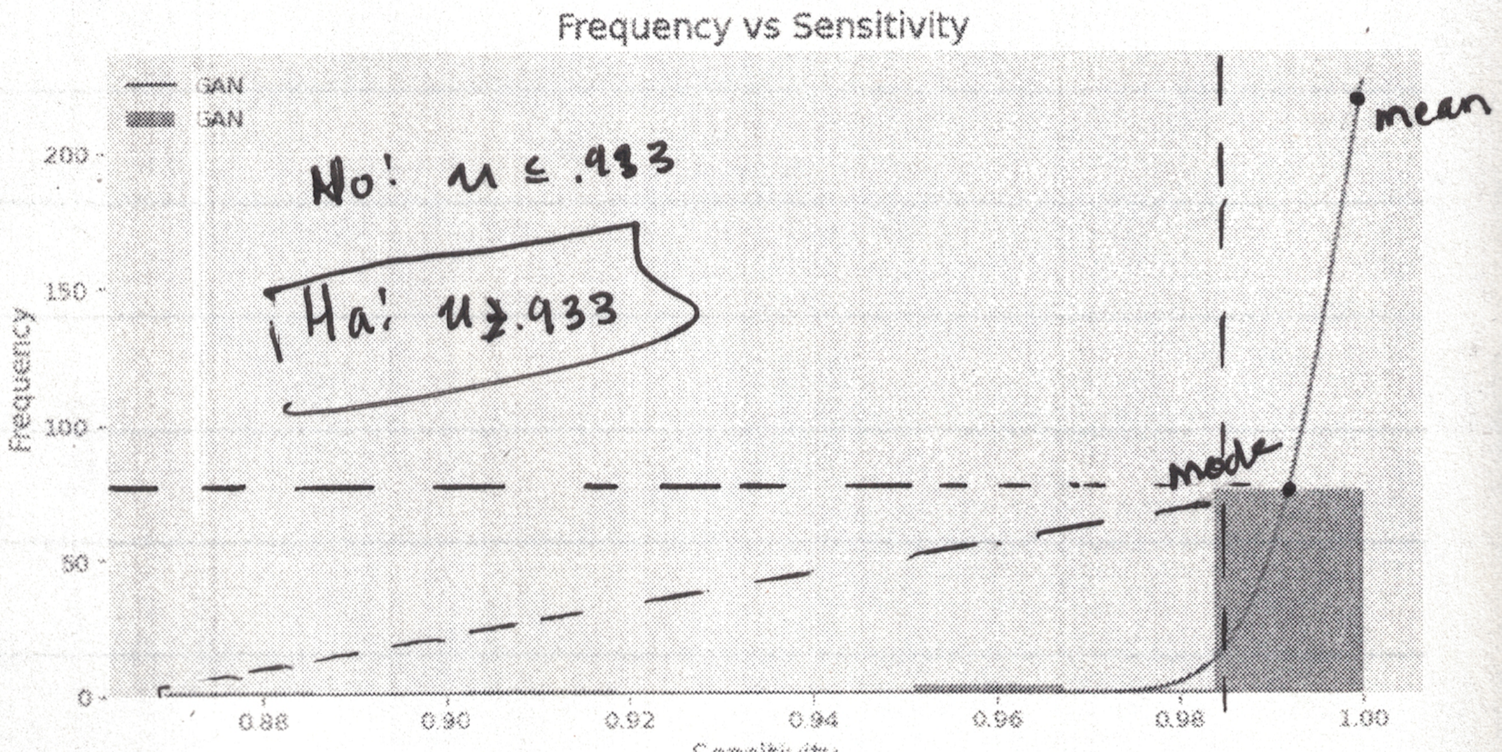
Sensitive is Maximized b/c we want to have all potential anomalies detected

$\bar{X} = 0.9952$

$\sigma_x = 0.0197$

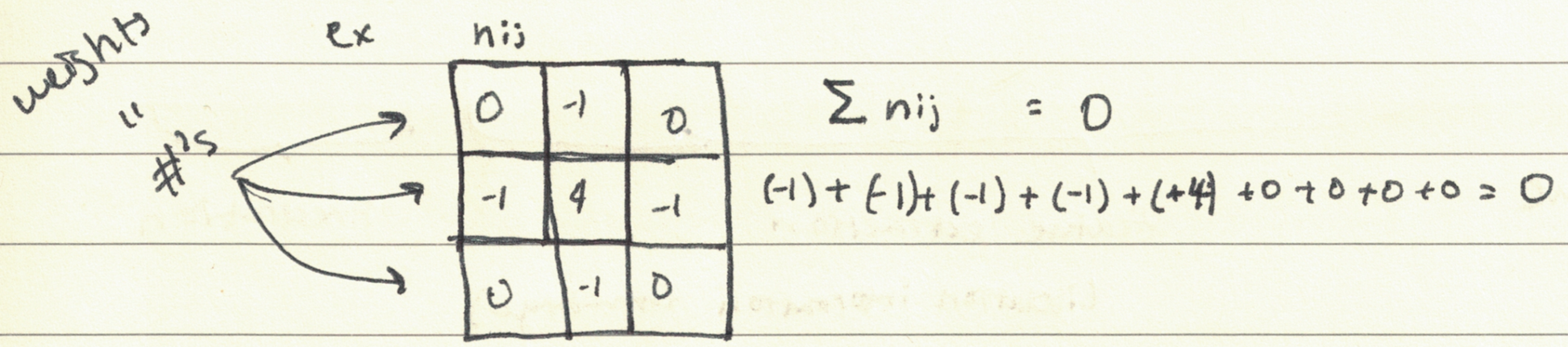
$t = \frac{0.9952 - 0.933}{\frac{0.0197}{\sqrt{149}}} = 38.5$

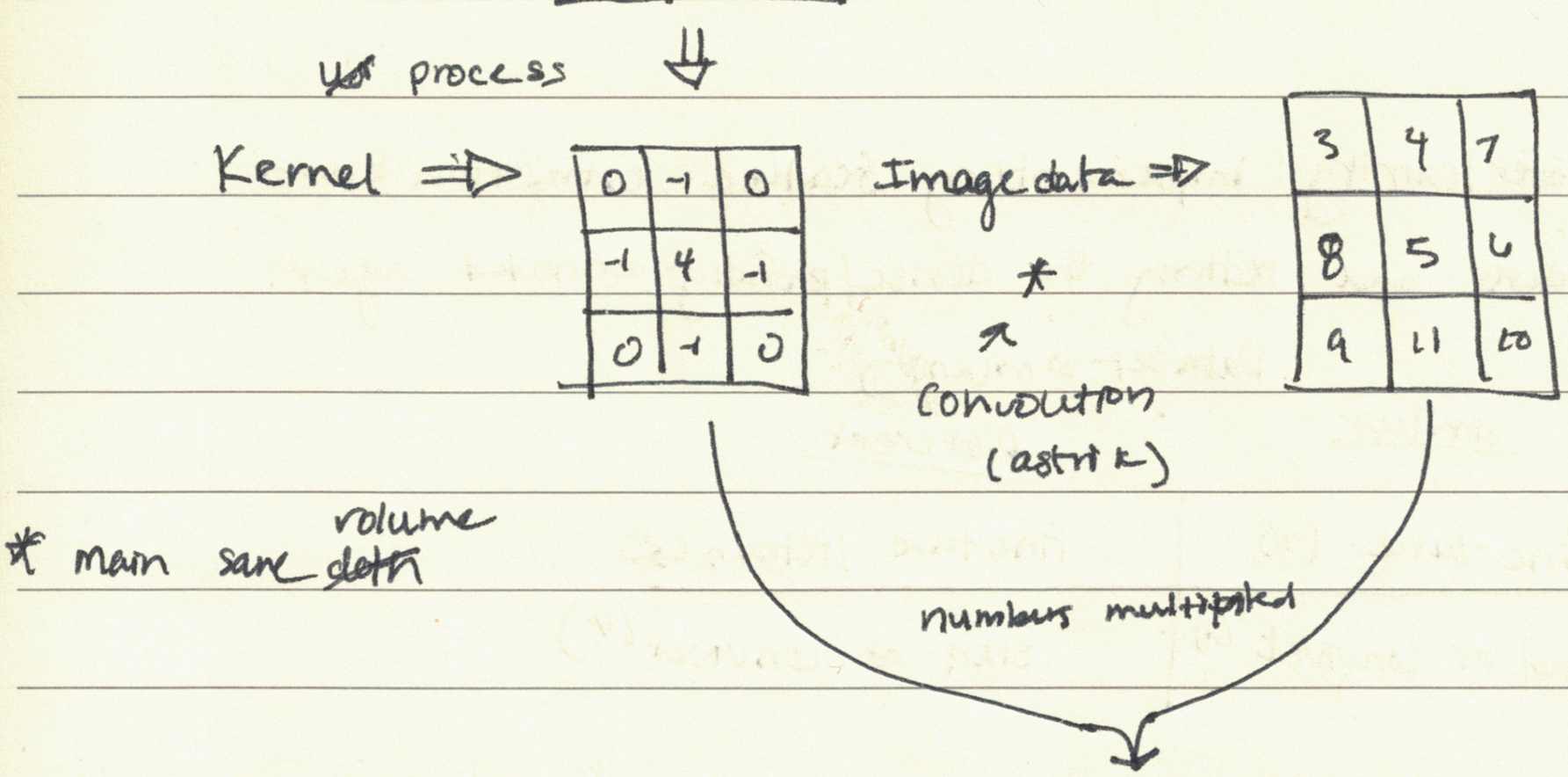
$P = 1.54 e^{-77}$



Filters & Edges in CNNs:

- low pass filters : block high frequencies (vice versa)
- using high pass filters - can block low pass freqs in image
  - used for edge detection
- convolutional kernal : edge detector → filter





* main same volume depth

Final ⇒

3	-4	0
-8	20	-6
0	-11	0

Σ nij = -9

(-4)+(-6)+(-8)+(-11)+20 = -9

Convolution Layer:

- applies many convolutional kernels
- stride the distance from pixel to pixel
  - 9 becomes center pixel value (5) in data image

Max pooling — used for decreasing complexity s/c overfitting

Avg Pooling

* capsel networks