

Yalong Pi (Texas A&M Institute of Data Science)

Address: Office 221C, John R. Blocker Building

Email: piyalong@tamu.edu

- B.S., Mechanical Engineering, 2007-2011
- M.S., Civil Engineering, 2011-2013
- Ph.D., Architecture Engineering, 2017-2020

- Assistant Research Scientist, 2020-present
- Architect, 2016-2017
- Project manager, 2013-2016



**Final project due
tomorrow at 4:00 PM**

Lab Day4c 3:01-4:00

Syllabus

- *Day 1: Classification Fundamentals and Convolutional Neural Network (CNN)*
- *Day 2: Data Augmentation, Evaluation, and Transfer-learning*
- *Day 3: Object Detection and Tracking*
- *Day 4: Segmentation and Autocoder*
- *Day 5: Generative Adversarial Networks (GANs) and Beyond*

Syllabus

- *Everyday (1:00-4:00 pm)*
 - *Quiz (40%)*
 - *Lecture*
 - *Lab*
- *Final project (60%) due: End of Day5*

1:30 - 2:00 auto coder

2:00 - 2:20 lab 4a

2:20 - 2:40 UI-net

2:40 - 3:10 lab 4b

3:10 - 3:30 MASK RCNN

3:30 - 4:00 Lab 4c

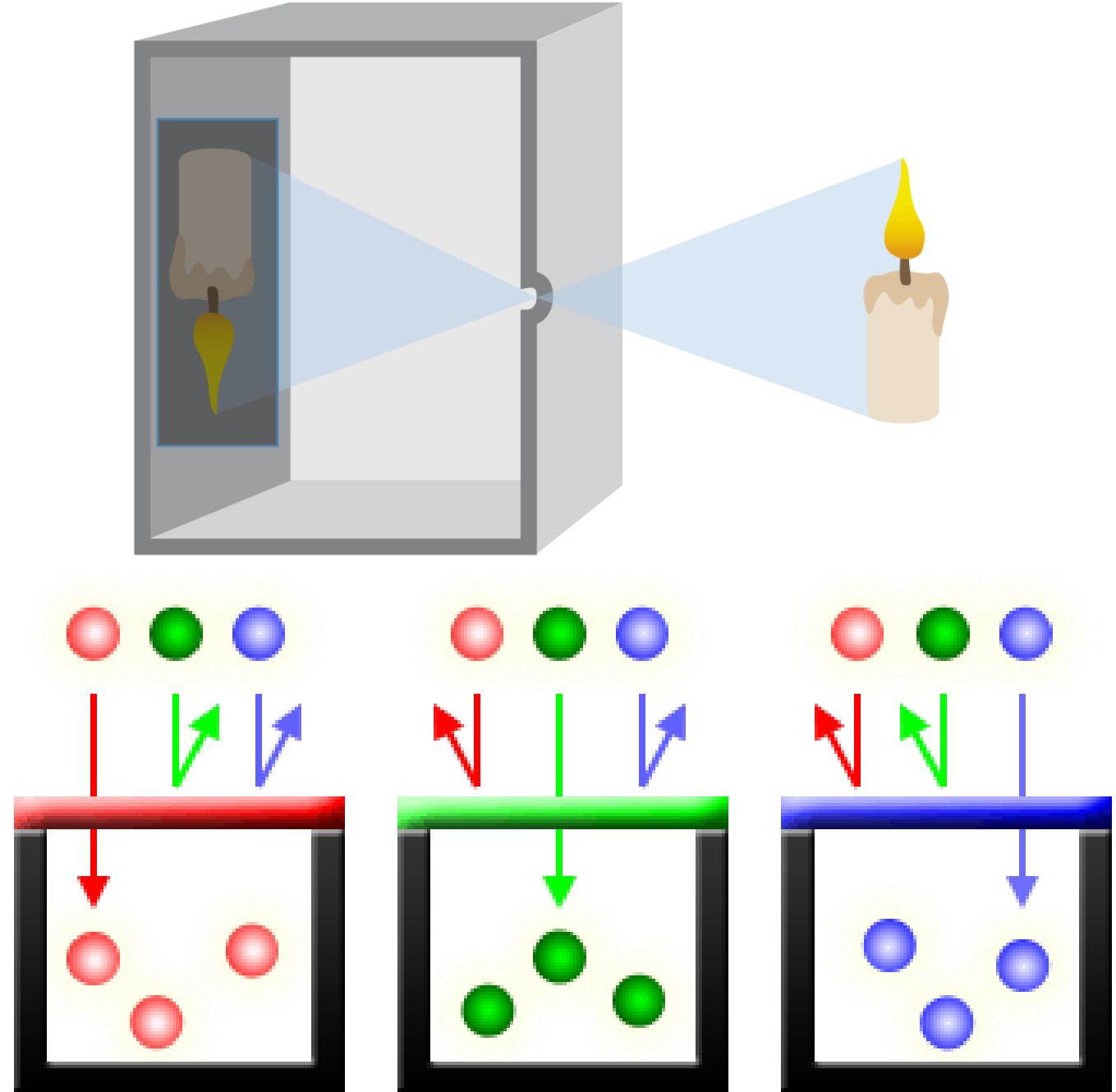
Course link

- <https://github.com/TAMIDSpiyalong/Machine-Learning-for-Computer-Vision>
- Canvas
- Howdy

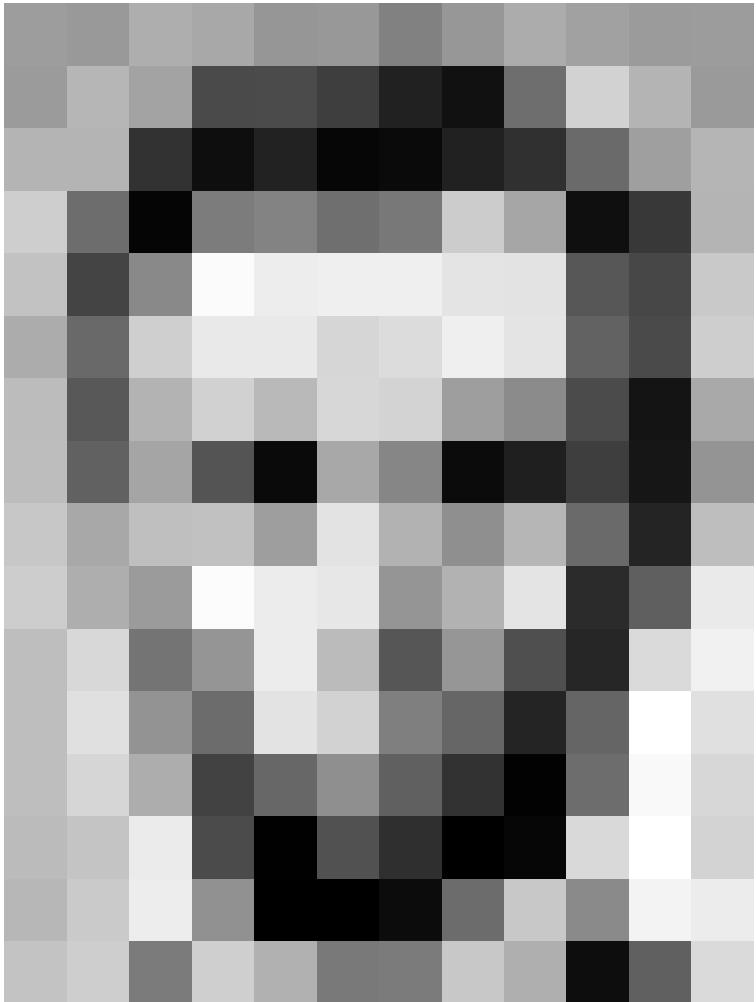
TensorFlow	Google Brain, 2015 (rewritten DistBelief)
Theano	University of Montréal, 2009
Keras	François Chollet, 2015 (now at Google)
Torch	Facebook AI Research, Twitter, Google DeepMind
Caffe	Berkeley Vision and Learning Center (BVLC), 2013



- Pinhole principle
- Traditional film
- Digital sensors (CCD and CMOS)
- Red Green Blue (RGB) channels



Grey Scale



157	153	174	168	150	152	129	151	172	161	155	166					
155	152	163	74	75	62	33	17	110	210	180	154					
180	180	50	14	34	6	10	33	48	106	189	181					
206	169	5	124	191	111	120	204	165	15	56	180					
194	63	137	251	237	239	239	228	227	87	71	201					
172	165	207	233	233	214	220	239	228	98	74	206					
188	88	179	209	185	215	211	158	139	75	29	169					
189	97	165	84	10	168	134	11	31	62	32	148					
199	168	191	163	158	227	178	143	182	105	36	190					
205	174	155	252	236	231	149	178	228	43	95	234					
190	216	116	149	236	187	85	150	79	36	218	241					
190	224	147	108	227	210	127	102	36	101	255	224					
190	214	173	66	103	143	95	50	2	109	249	215					
187	196	235	79	1	81	47	0	6	217	255	211					
183	202	237	145	0	0	12	108	209	138	243	236					
195	206	123	207	177	121	123	209	179	13	96	218					

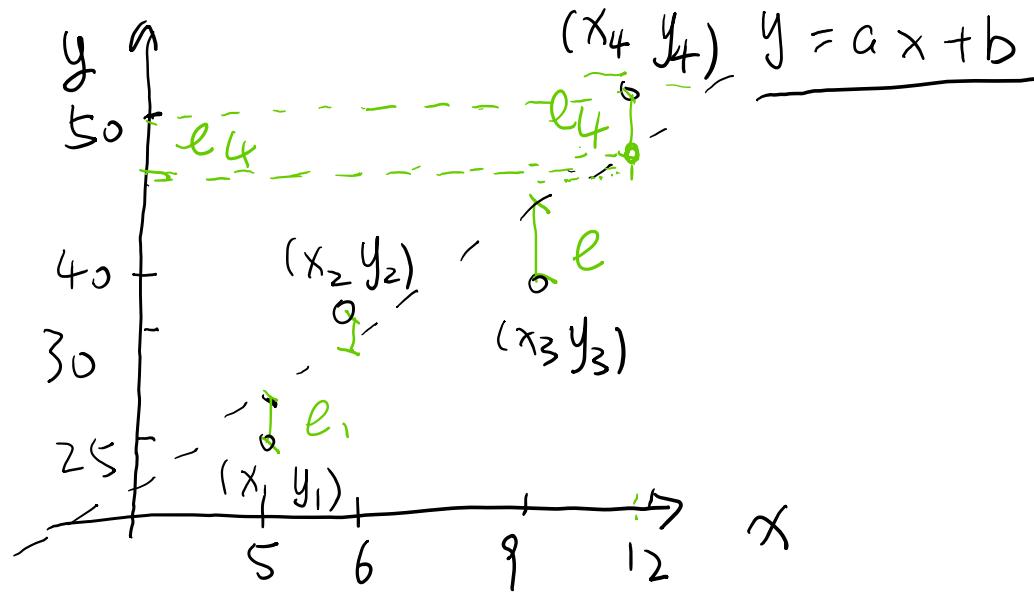
157	153	174	168	150	152	129	151	172	161	155	166					
155	152	163	74	75	62	33	17	110	210	180	154					
180	180	50	14	34	6	10	33	48	106	189	181					
206	169	5	124	191	111	120	204	165	15	56	180					
194	68	137	251	237	239	239	228	227	87	71	201					
172	165	207	233	233	214	220	239	228	98	74	206					
188	88	179	209	185	215	211	158	139	75	29	169					
189	97	165	84	10	168	134	11	31	62	32	148					
199	168	191	163	158	227	178	143	182	105	36	190					
205	174	155	252	236	231	149	178	228	43	95	234					
190	216	116	149	236	187	85	150	79	36	218	241					
190	224	147	108	227	210	127	102	36	101	255	224					
190	214	173	66	103	143	95	50	2	109	249	215					
187	196	235	79	1	81	47	0	6	217	255	211					
183	202	237	145	0	0	12	108	209	138	243	236					
195	206	123	207	177	121	123	209	179	13	96	218					

x	y
5	25
6	30
9	40
12	50

$$\begin{array}{c} 4 \times 1 \\ \left[\begin{array}{c} 25 \\ 30 \\ 40 \\ 50 \end{array} \right] = \left[\begin{array}{c} 5a + 1b \\ 6a + 1b \\ 9a + 1b \\ 12a + 1b \end{array} \right] + \left[\begin{array}{c} e_1 \\ e_2 \\ e_3 \\ e_4 \end{array} \right] \end{array}$$

$$Y = X \cdot A + E$$

$4 \times 1 \quad 4 \times 2 \quad 2 \times 1 \quad 4 \times 1$



$$\begin{array}{l} y_1 = a x_1 + b + e_1 \\ y_2 = a x_2 + b + e_2 \\ y_3 = a x_3 + b + e_3 \\ y_4 = a x_4 + b + e_4 \end{array}$$

4×2

$$X = \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 9 & 1 \\ 12 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} a & b \end{bmatrix}$$

$$Y = \underbrace{X \cdot A}_{4 \times 1} + \underbrace{E}_{4 \times 2} \quad \Rightarrow \quad A = (\underbrace{X^T X}_{2 \times 4})^{-1} \underbrace{X^T Y}_{2 \times 1}$$

$$X^T X = \begin{bmatrix} 5 & 6 & 9 & 12 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}_{2 \times 4} \begin{bmatrix} 5 & 1 \\ 6 & 1 \\ 9 & 1 \\ 12 & 1 \end{bmatrix}_{4 \times 2} = \begin{bmatrix} 286 & 32 \\ 32 & 4 \end{bmatrix}_{2 \times 2}$$

$$(X^T X)^{-1} = \frac{1}{286 \times 4 - 32 \times 32} \begin{bmatrix} 4 & -32 \\ -32 & 286 \end{bmatrix} = \frac{1}{120} \begin{bmatrix} 4 & -32 \\ -32 & 286 \end{bmatrix}$$

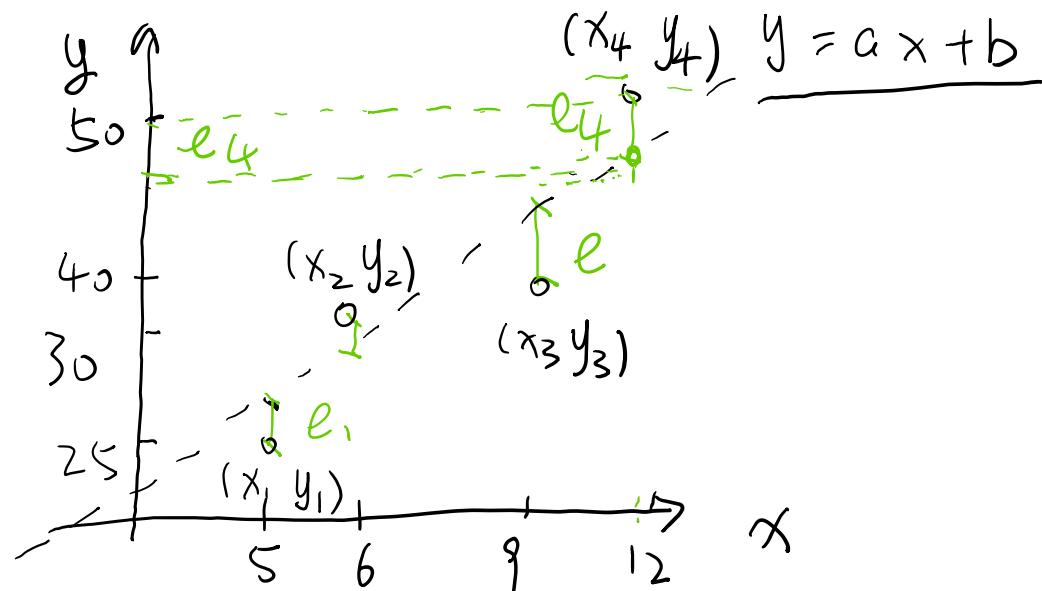
$$X^T Y = \begin{bmatrix} 5 & 6 & 9 & 12 \\ 1 & 1 & 1 & 1 \end{bmatrix}_{2 \times 4} \begin{bmatrix} 25 \\ 30 \\ 40 \\ 50 \end{bmatrix}_{4 \times 1} = \begin{bmatrix} 1265 \\ 145 \end{bmatrix}_{2 \times 1}$$

$$A = (\underline{X}^T \underline{X})^{-1} \underline{\underline{X}}^T \underline{Y} = \frac{1}{120} \begin{bmatrix} 4 & -32 \\ -32 & 286 \end{bmatrix} \begin{bmatrix} 1265 \\ 145 \end{bmatrix}$$

$$= \frac{1}{120} \begin{bmatrix} 2 \times 2 \\ 420 \\ 990 \end{bmatrix} = \begin{bmatrix} 2 \times 1 \\ 3.5 \\ 8.25 \end{bmatrix} = \begin{bmatrix} 9 \\ b \end{bmatrix}$$

$$\bar{E} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}$$

$$MSE = \frac{1}{4} (e_1^2 + e_2^2 + e_3^2 + e_4^2)$$



$$MSE = \frac{1}{4} (e_1^2 + e_2^2 + e_3^2 + e_4^2)$$

$$a = 1 \quad b = 0$$

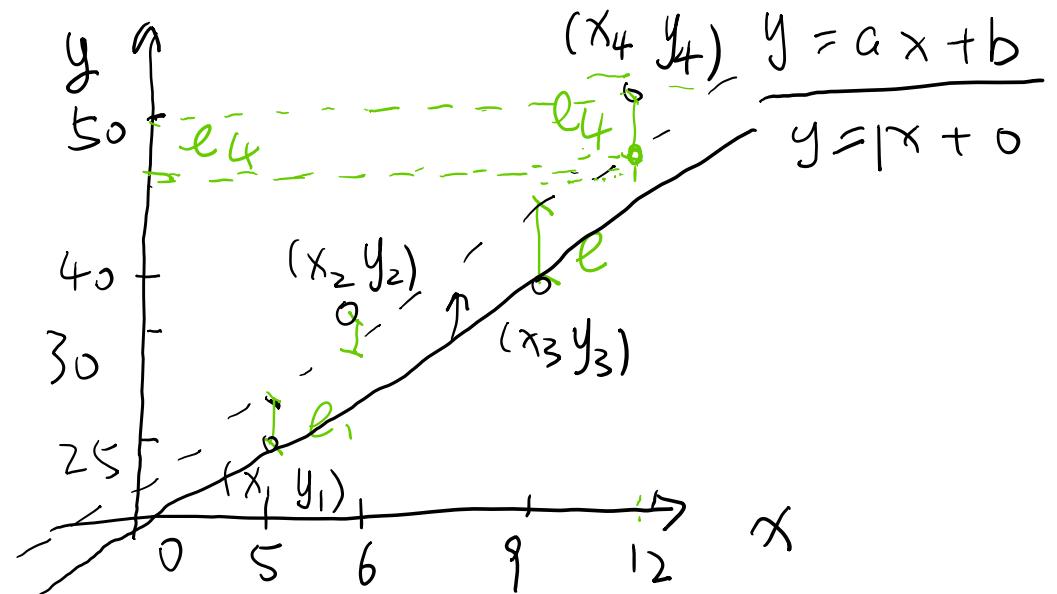
$$\text{new_}b = b - \frac{\partial MSE}{\partial b} \cdot LR \xrightarrow{10^{-2}}$$

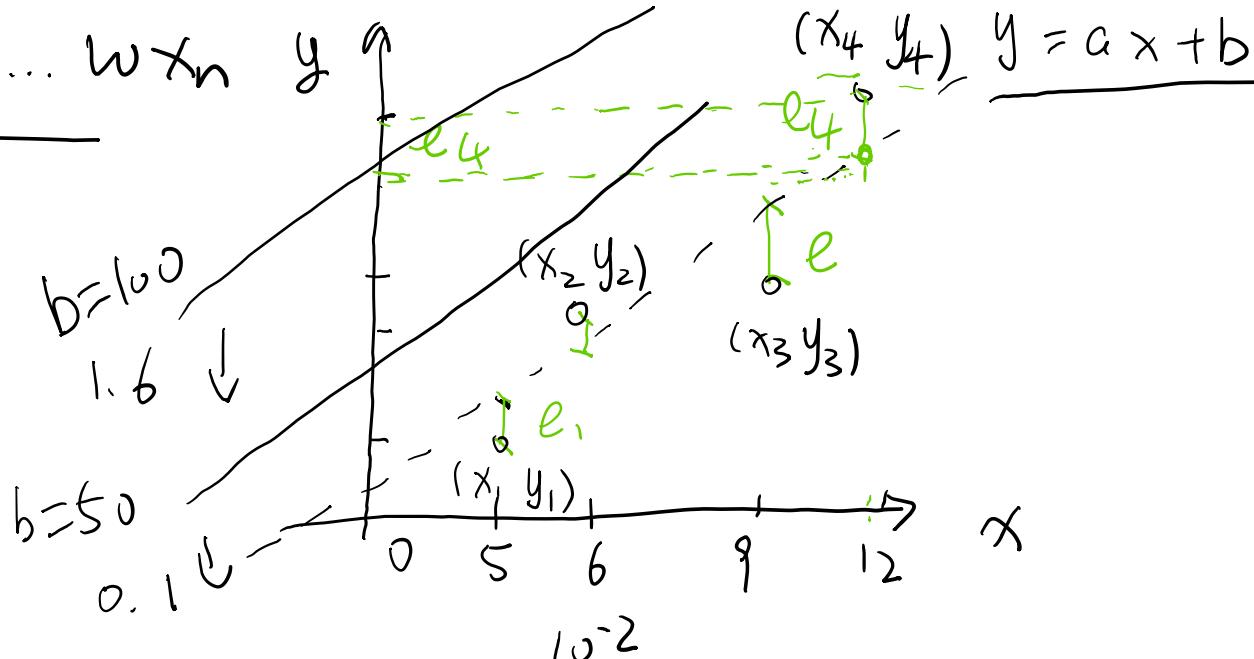
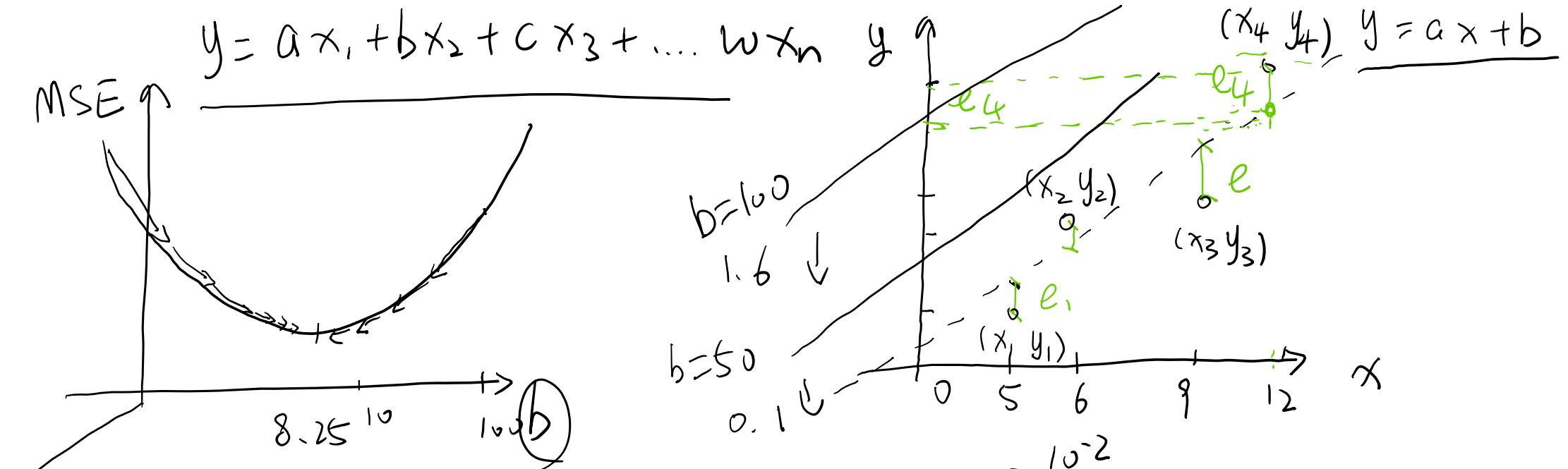
for e_1 only

$$\begin{aligned} MSE &= \frac{1}{4} (e_1^2) = (y_1 - (ax_1 + b))^2 = (25 - (5a + b))^2 \\ &= \underline{625} - \underline{250a} - \underline{50b} + \underline{25a^2} + \underline{10ab} + \underline{b^2} \end{aligned}$$

$$LR \cdot \frac{\partial MSE}{\partial b} = -50 + 10a + 2b = -40 \cdot LR = -0.4 \Rightarrow \text{new_}b = 0.4$$

$$\text{new-new_}b = \text{new_}b - \frac{\partial MSE}{\partial b} \cdot LR \quad \xrightarrow{\text{loop}}$$





$a = 1 \quad b = 50 \quad \text{new_}b = b - \frac{\partial MSE}{\partial b} \cdot LR = 10 - (-50 + 10 + 100) \times 0.01$

$$= 10 - \underline{60 \times 0.01} \quad -0.6$$

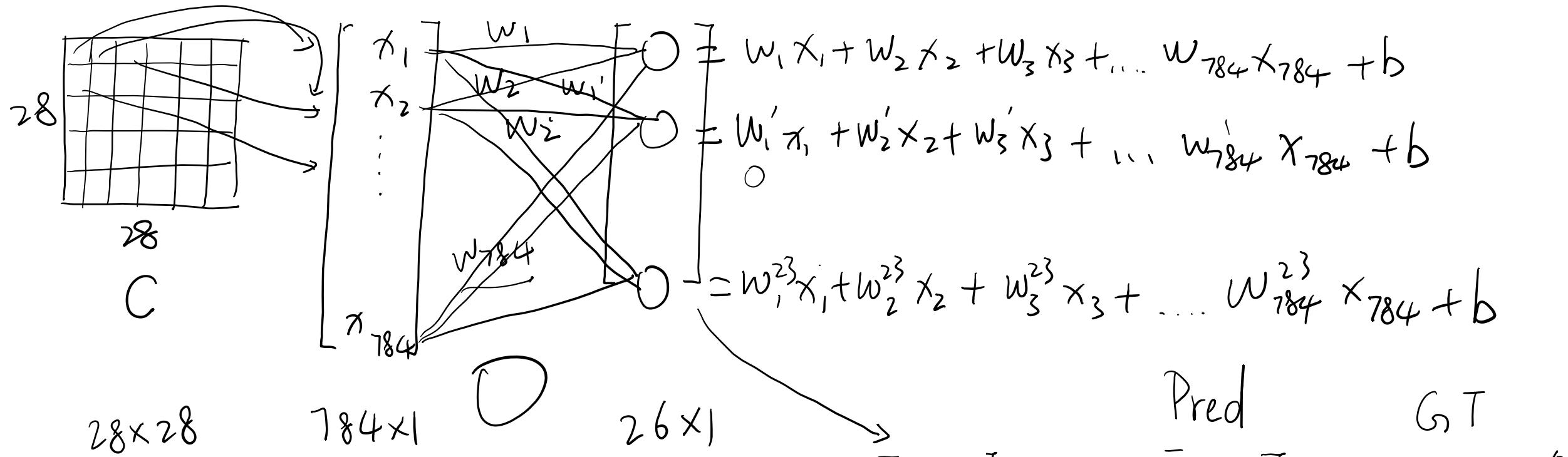
$a = 1 \quad b = 100 \quad \text{new_}b = b - \frac{\partial MSE}{\partial b} \cdot LR = 100 - (-50 + 10 + 200) \times 0.01$

$$= 100 - \underline{160 \times 0.01} \quad \sim 1.6$$

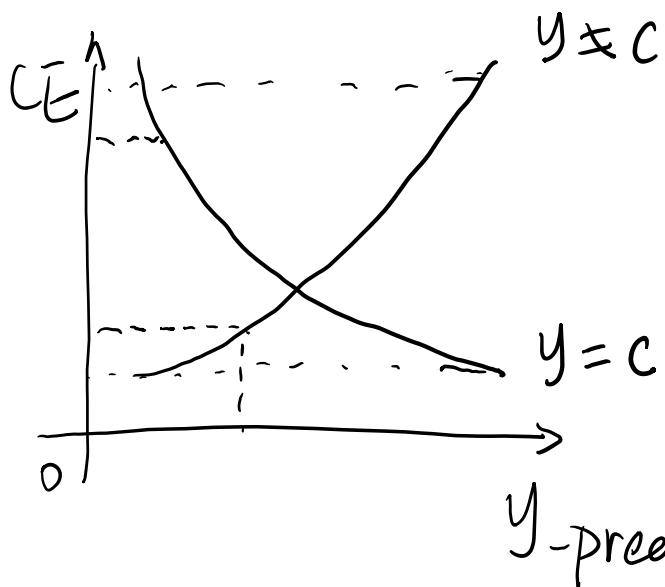
Stochastic Gradient
(SGD)

Descent

Training Fit Learn Optimize



Cross entropy
 $\text{CE}(y_{\text{pred}}, y)$



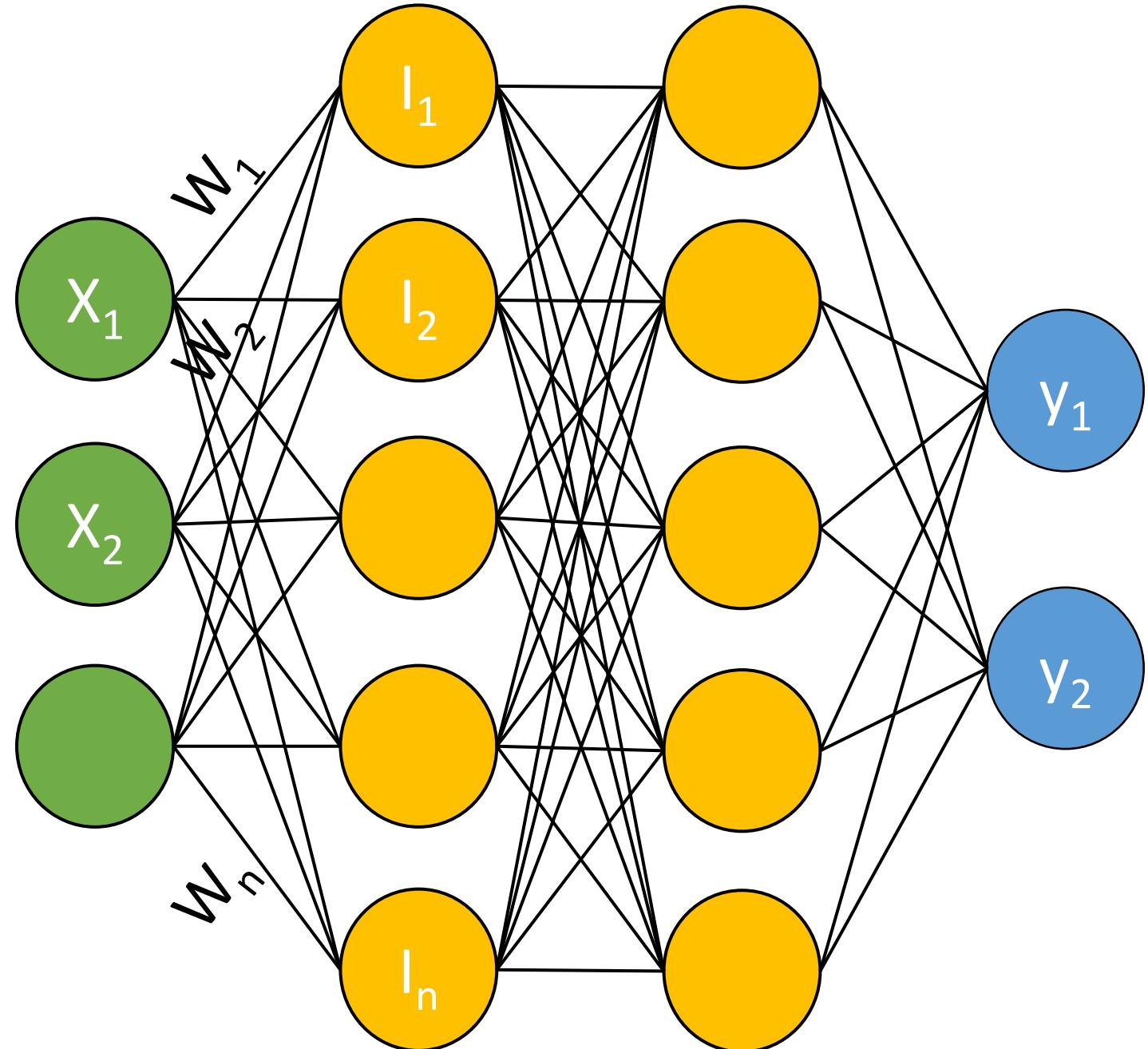
$$\begin{bmatrix} 70 \\ 50 \\ 900 \\ 0 \\ \vdots \\ 10 \\ 26 \end{bmatrix} \quad 26 \times 1$$

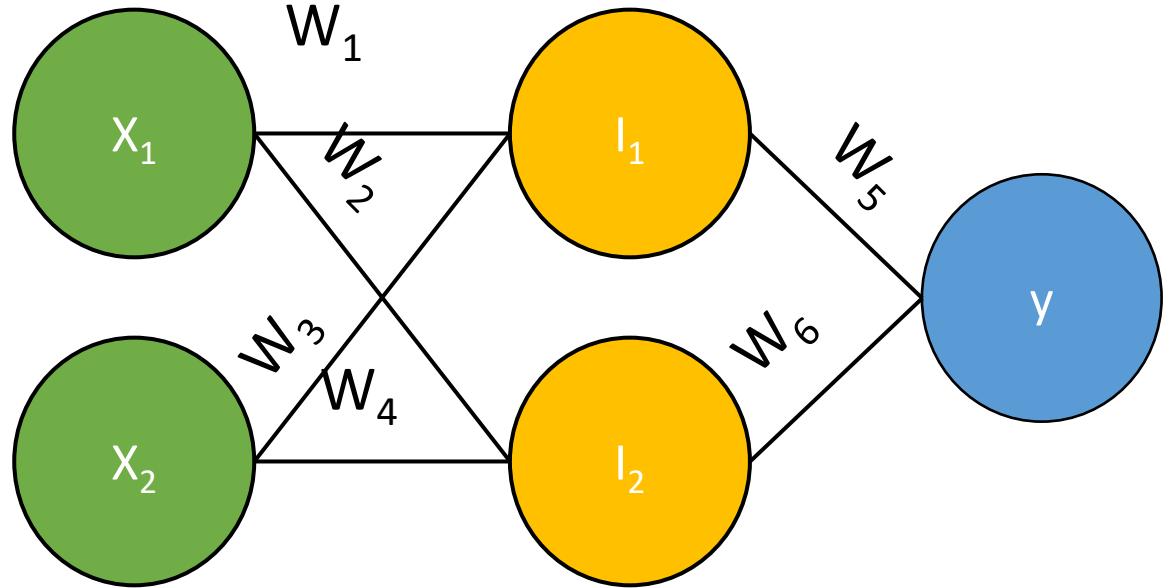
Pred	GT	A
$\frac{\partial \text{CE}}{\partial w}$	$\begin{bmatrix} 0.05 \\ 0.03 \\ 0.9 \\ \vdots \\ 0.01 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$
$\frac{\partial \text{CE}}{\partial w}$	26×1	26×1

For each connection:

$$I_n = f(\sum_n X_n W_n + b)$$

- f is the activation function
- W_n is the weight
- b is the bias.
- A DNN has millions of weights and biases





$$y_{pred} = (X_1 W_1 + X_2 W_3) W_5 + \\ (X_1 W_2 + X_2 W_4) W_6$$

$$Loss = 1/2(y_{pred} - y_{true})^2$$

$$W_n' = W_n - LR \ (\partial Loss / \partial W_n)$$

$$e.g., W_6' = W_6 - LR \ (\partial Loss / \partial W_6)$$

- ❑ Y_{true} are from the dataset labels
- ❑ W_n are randomly initialized
- ❑ Many types of loss function
- ❑ Learning rate (LR) is very small (e.g., 0.0001)
- ❑ Repeat in many epochs

Learning From Error

$$MSE = \frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2 = \frac{1}{n} \sum_{i=1}^n (y - (mx + b))^2$$

$$MSE = \frac{1}{2} ((3 - (m(1) + b))^2 + (5 - (m(2) + b))^2)$$

$$\frac{\partial MSE}{\partial m} = 5m + 3b - 13$$

$$\frac{\partial MSE}{\partial m} = -3$$

$$\frac{\partial MSE}{\partial b} = 3m + 2b - 8$$

$$\frac{\partial MSE}{\partial b} = -1$$

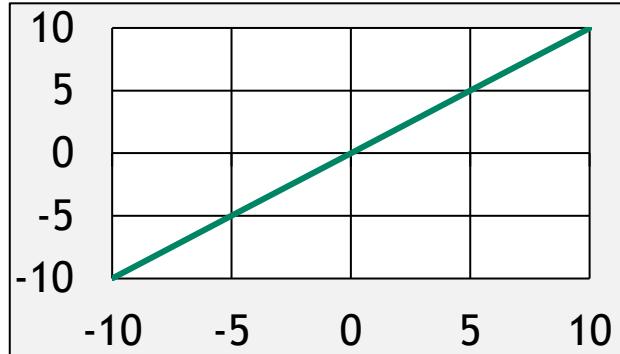
$$\begin{aligned} m &= -1 \\ b &= 5 \end{aligned}$$

ACTIVATION FUNCTIONS

Linear

$$\hat{y} = wx + b$$

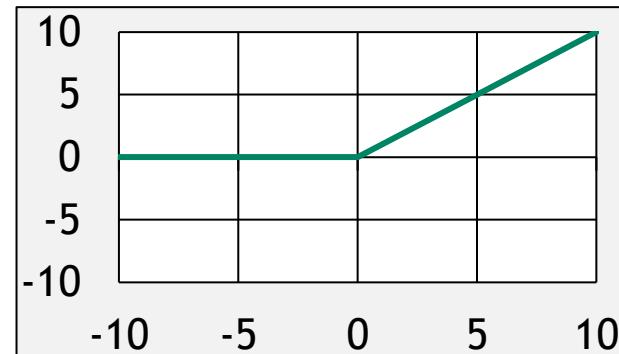
```
1 # Multiply each input  
2 # with a weight (w) and  
3 # add intercept (b)  
4 y_hat = wx+b
```



ReLU

$$\hat{y} = \begin{cases} wx + b & \text{if } wx + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

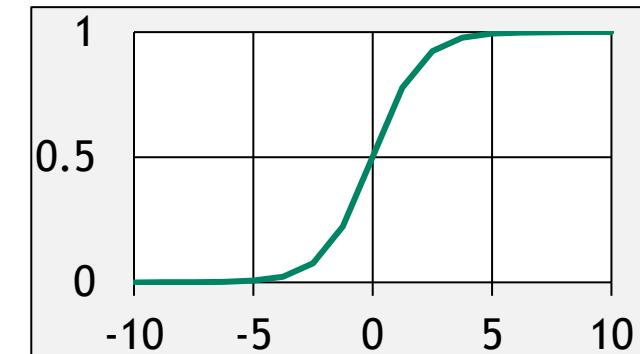
```
1 # Only return result  
2 # if total is positive  
3 linear = wx+b  
4 y_hat = linear * (linear > 0)
```



Sigmoid

$$\hat{y} = \frac{1}{1 + e^{-(wx+b)}}$$

```
1 # Start with line  
2 linear = wx + b  
3 # Warp to - inf to 0  
4 inf_to_zero = np.exp(-1 * linear)  
5 # Squish to -1 to 1  
6 y_hat = 1 / (1 + inf_to_zero)
```



$$y = \max(0, x)$$

Output
layer

Softmax
activation function

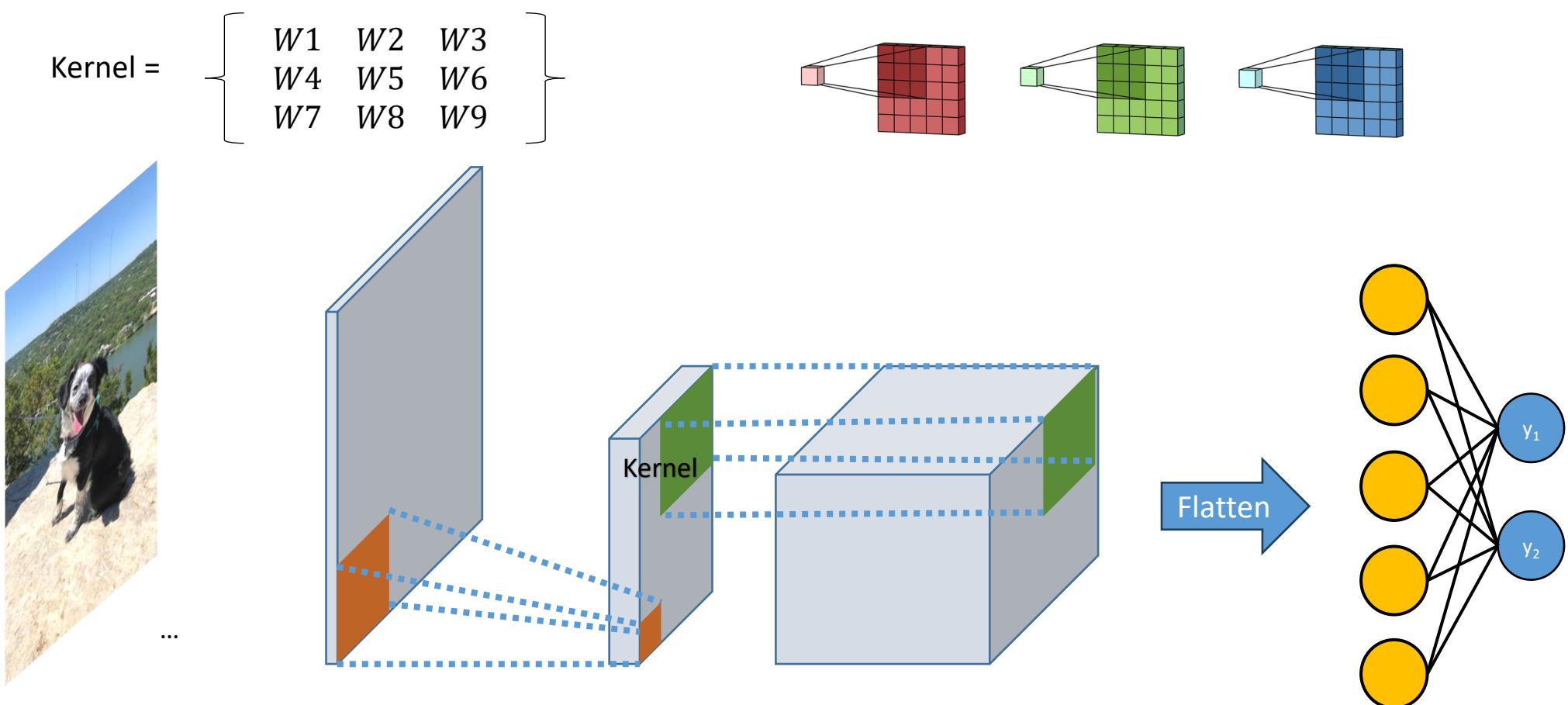
Probabilities

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix}$$

$$\rightarrow \boxed{\frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}}$$

$$\begin{bmatrix} 0.02 \\ 0.90 \\ 0.05 \\ 0.01 \\ 0.02 \end{bmatrix}$$

Convolutional Neural Network (CNN)



Input

Max pooling layer

Convolution layer

Output

Convolution Computation

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



Kernel at position 1



Kernel at position 2



Kernel at position n

7	7	7	7	5
7	7	7	5	5
7	7	5	5	5
7	5	5	5	5
5	5	5	5	5

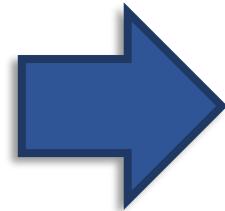
7	7	7	7	5
7	7	7	5	5
7	7	5	5	5
7	5	5	5	5
5	5	5	5	5

7	7	7	7	5
7	7	7	5	5
7	7	5	5	5
7	5	5	5	5
5	5	5	5	5

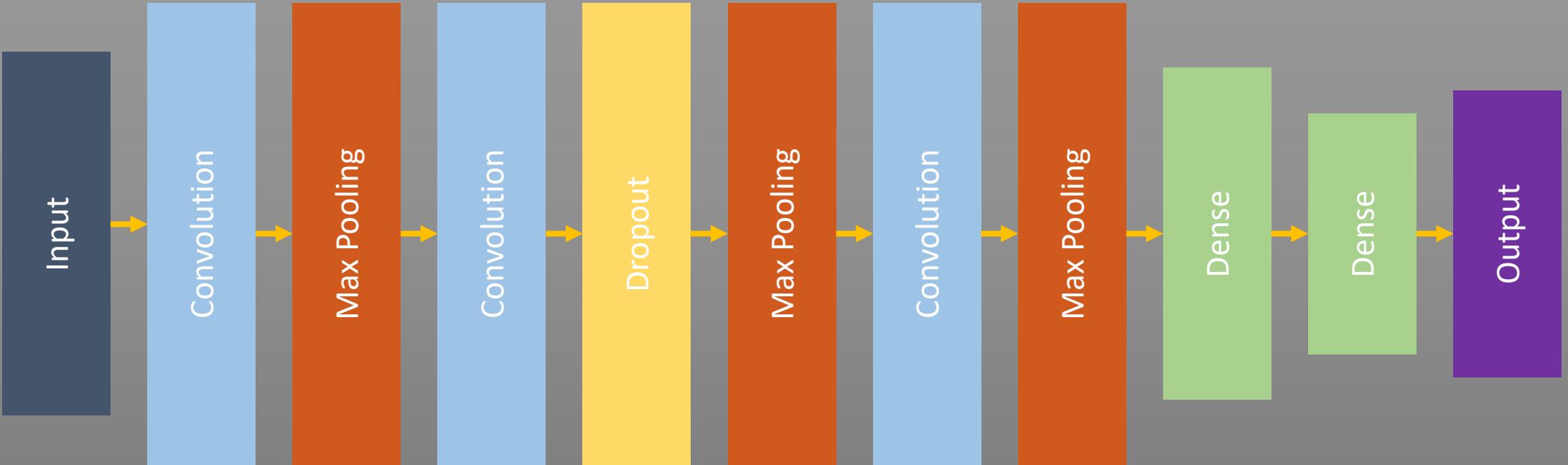
0	0	0	0	0
0	21	19	17	0
0	19	17	15	0
0	17	15	15	0
0	0	0	0	0

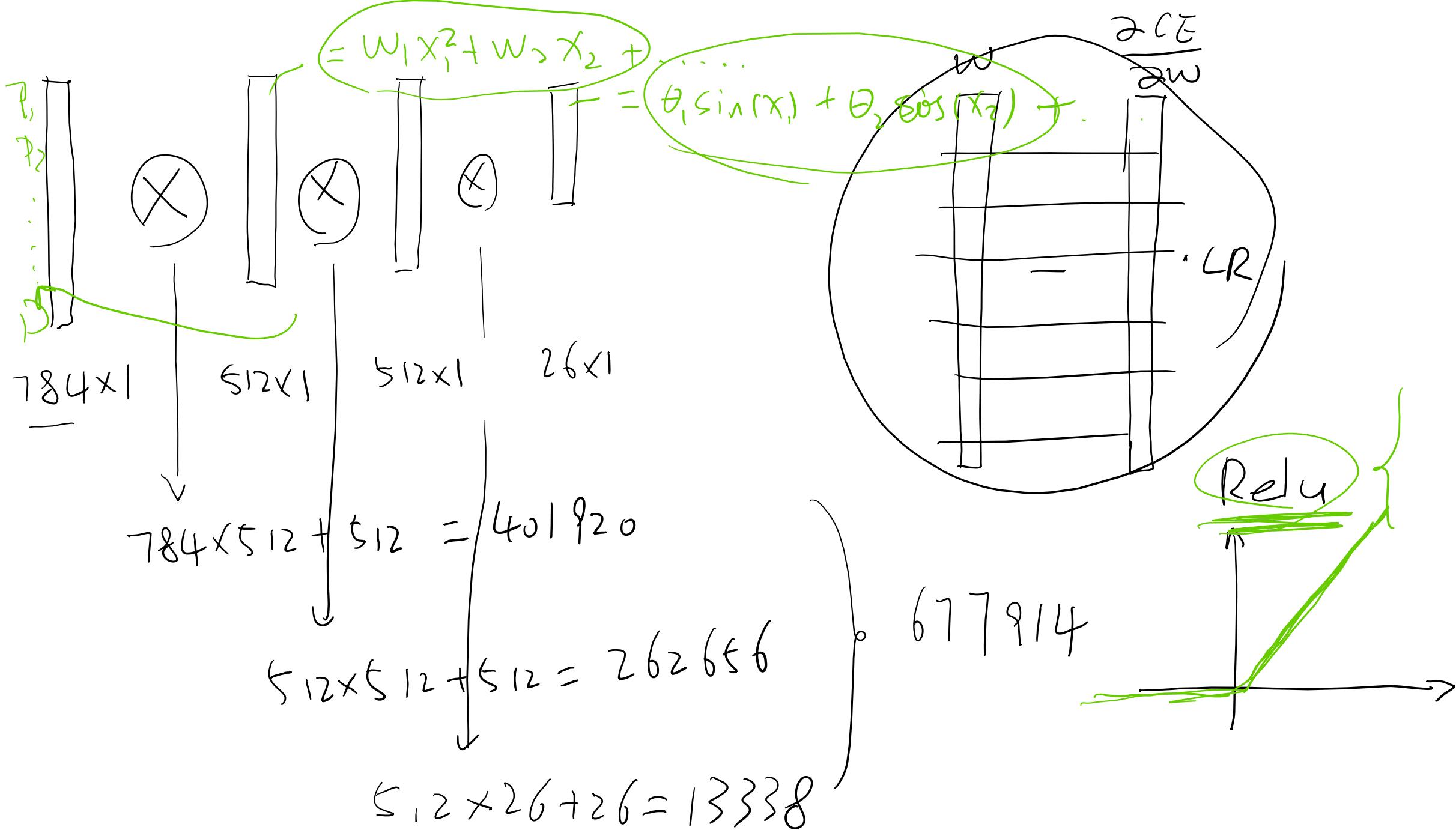
Max Pooling

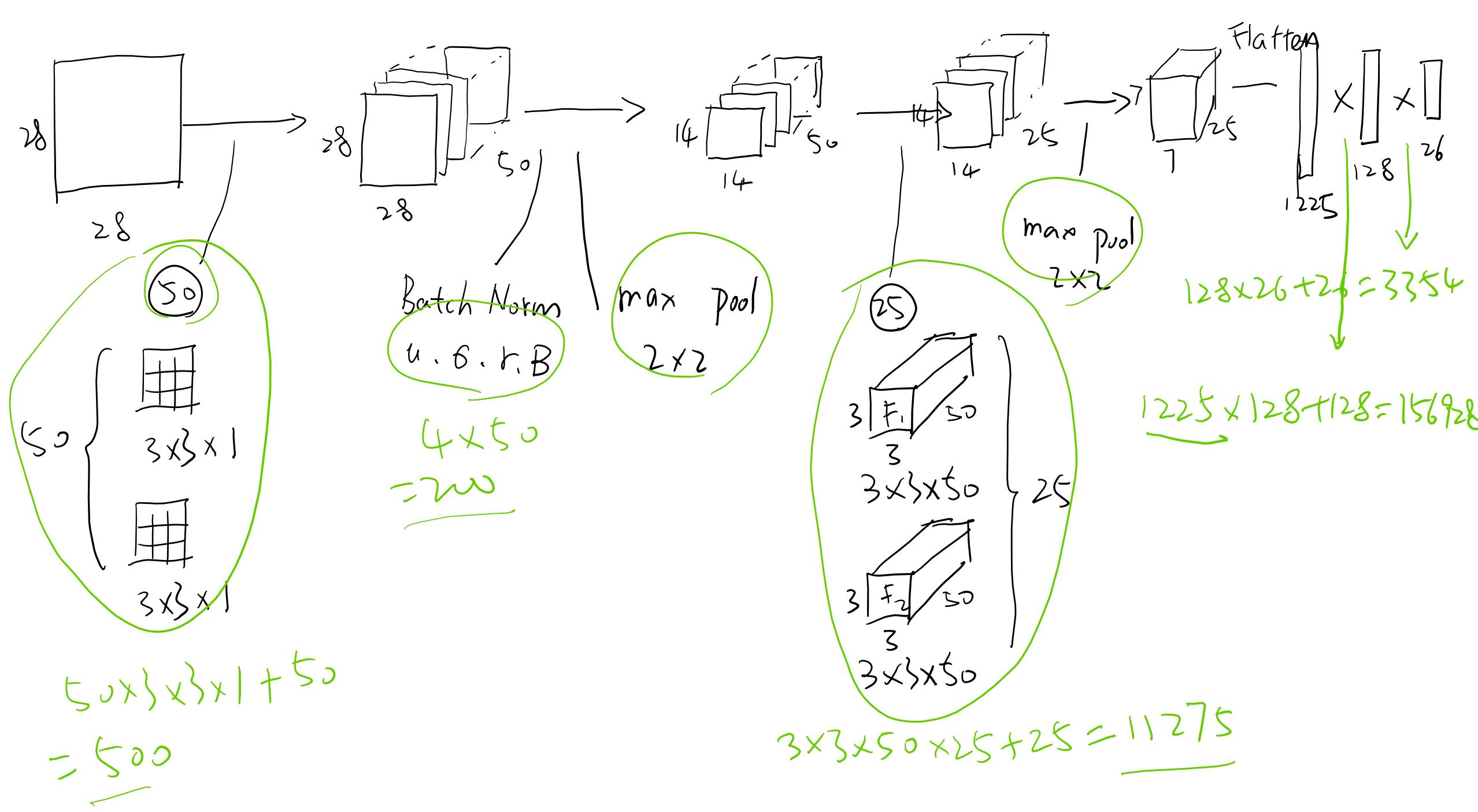
110	255	153	67
12	89	88	43
10	15	50	55
23	9	49	23



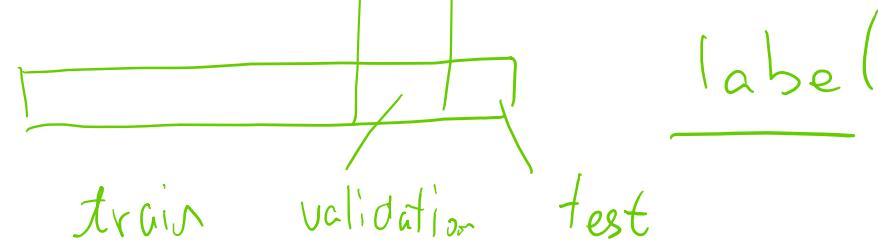
255	153
23	55





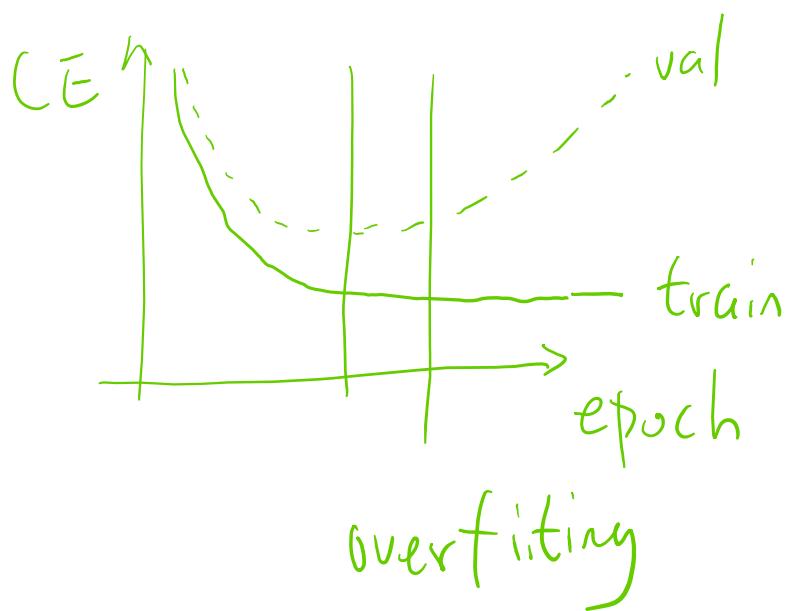


80% | 10% | 10%



Image

5376 × 64 × 64 × 3



early stopping
Augmentation

Image Flipping

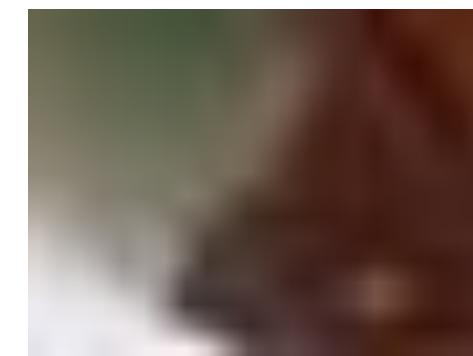
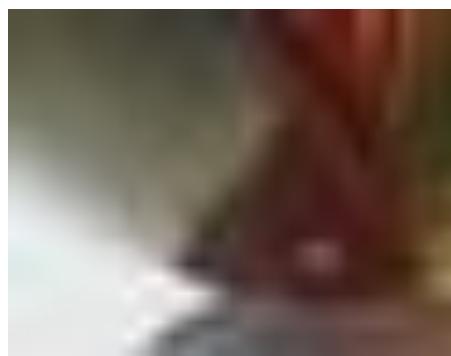
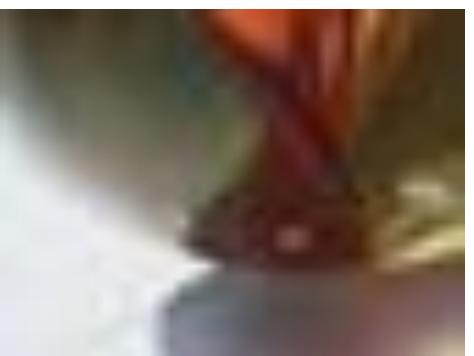
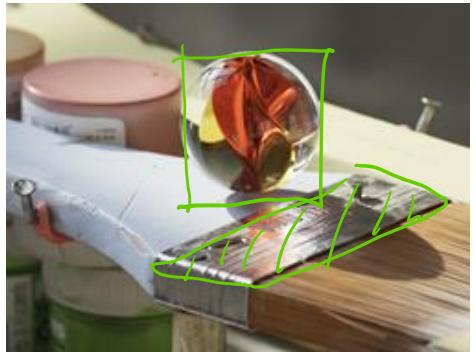
Horizontal Flip



Vertical Flip



Zooming



Brightness



Rotation



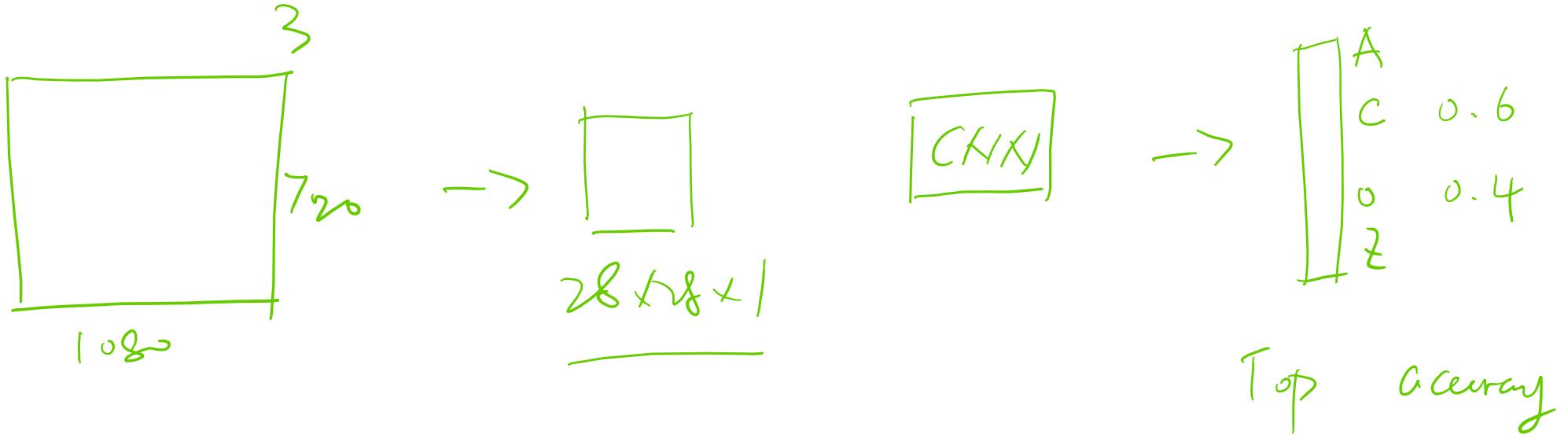
label

Prediction\Ground Truth	Positive C	Negative Not C
Positive	TP	FP
Negative	FN	TN

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$precision = \frac{TP}{TP + FP}$$

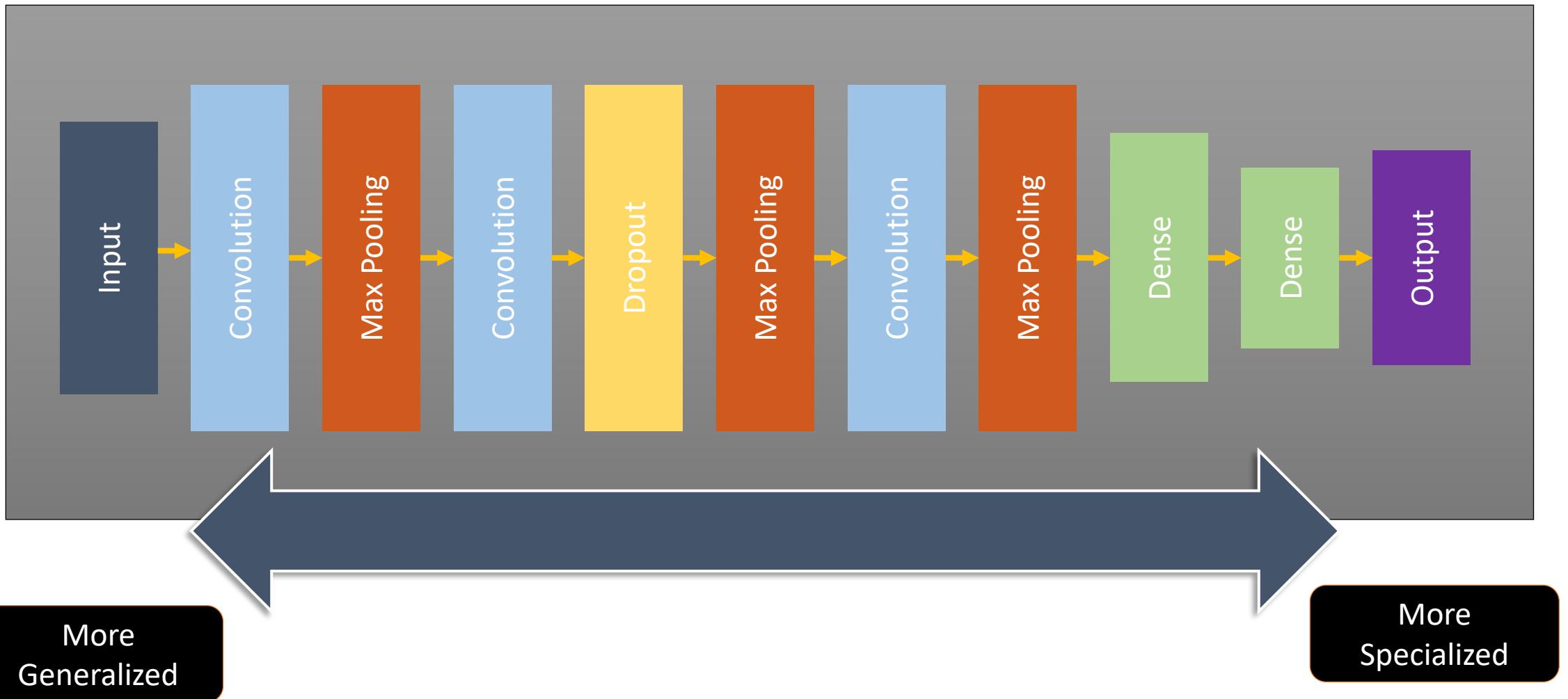
$$recall = \frac{TP}{TP + FN}$$



reshape (1, 28, 28, 1)
 ↴
 ↓

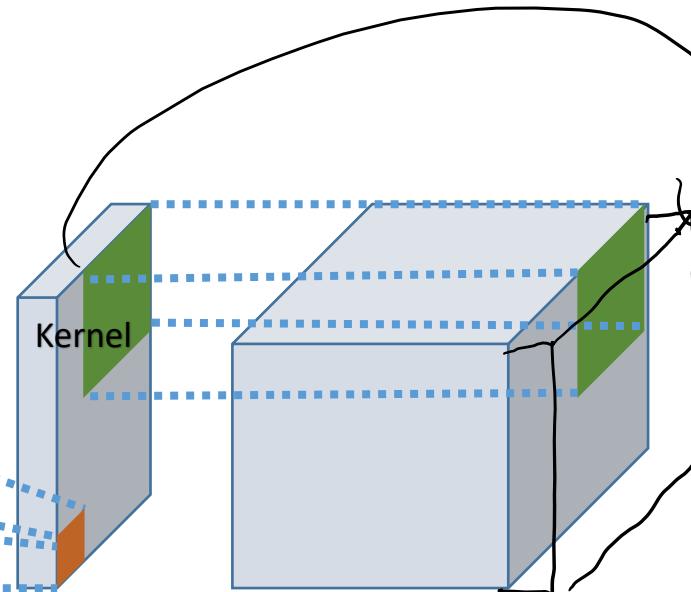
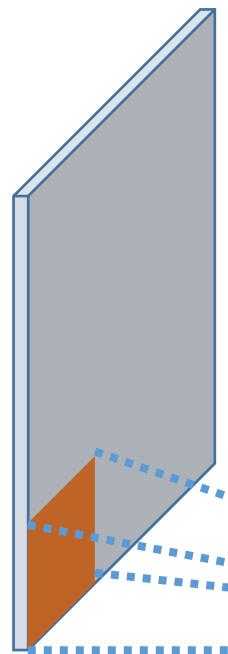
Top Accuracy

Transfer Learning

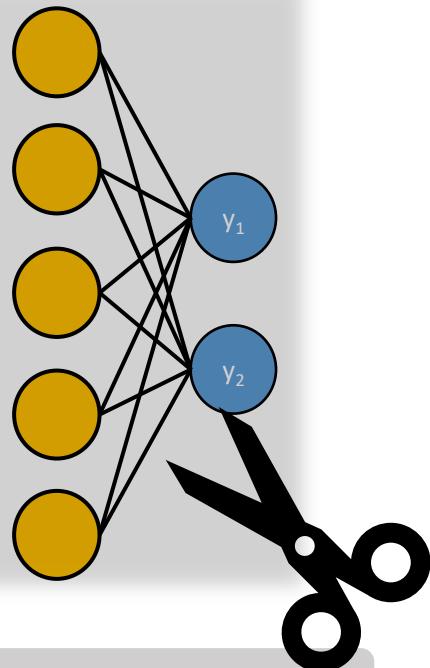


Transfer Learning

$$\text{Kernel} = \begin{bmatrix} W_1 & W_2 & W_3 \\ W_4 & W_5 & W_6 \\ W_7 & W_8 & W_9 \end{bmatrix}$$



Flatten



Input

Max pooling layer

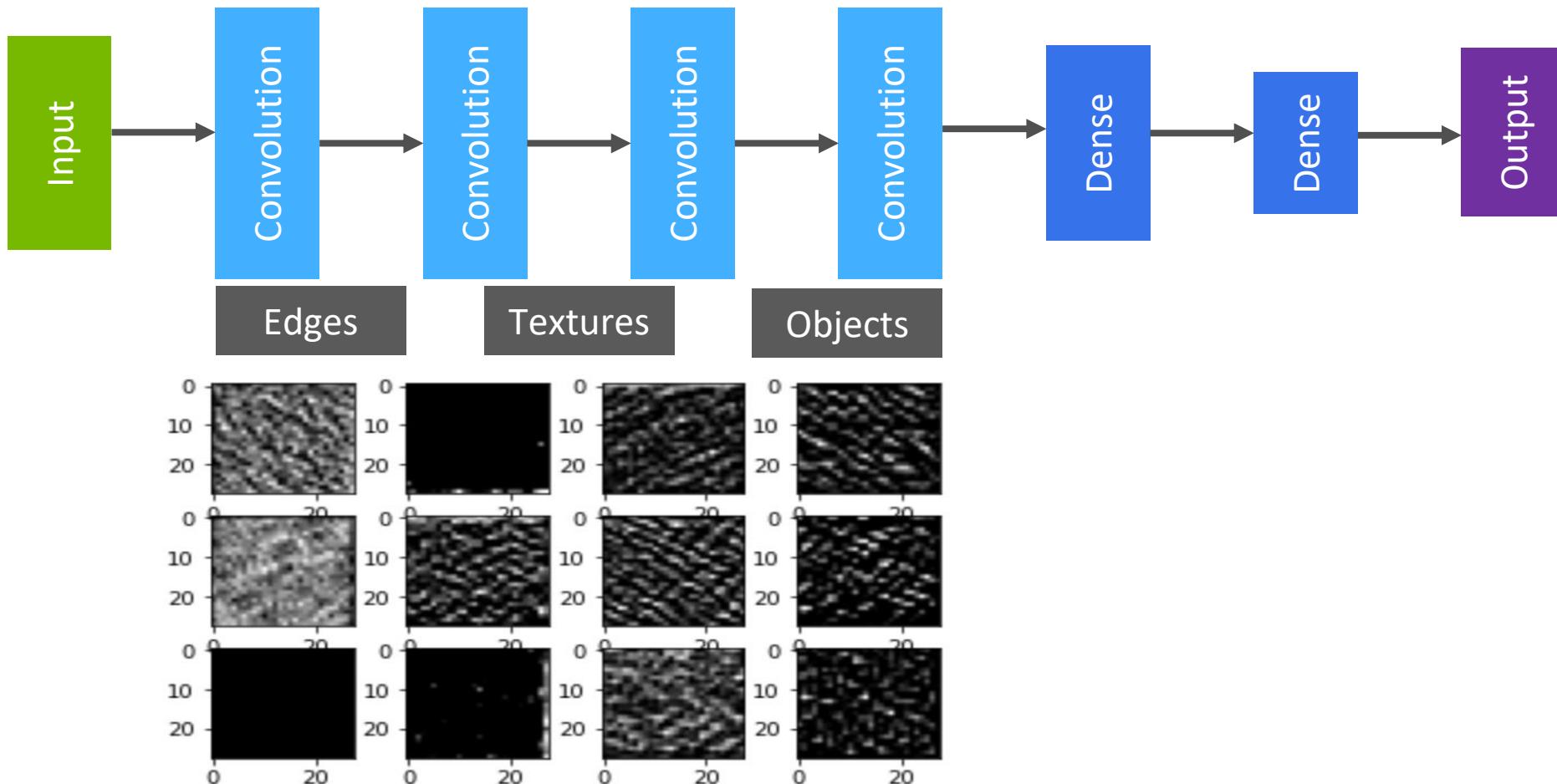
Convolution layer

Output

Residual Net

$$f(\text{conv}(x) + x)$$

NEURAL NETWORK PERCEPTION



Pre-Trained Models

TensorFlow Hub

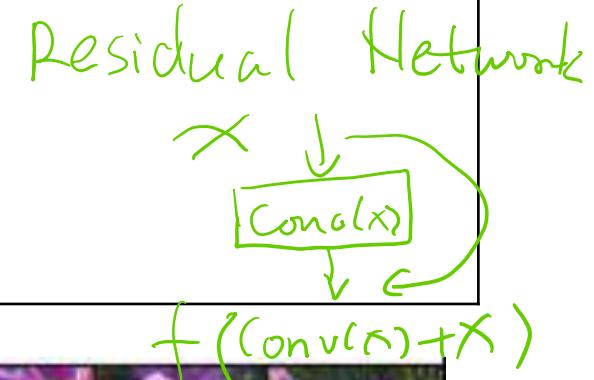
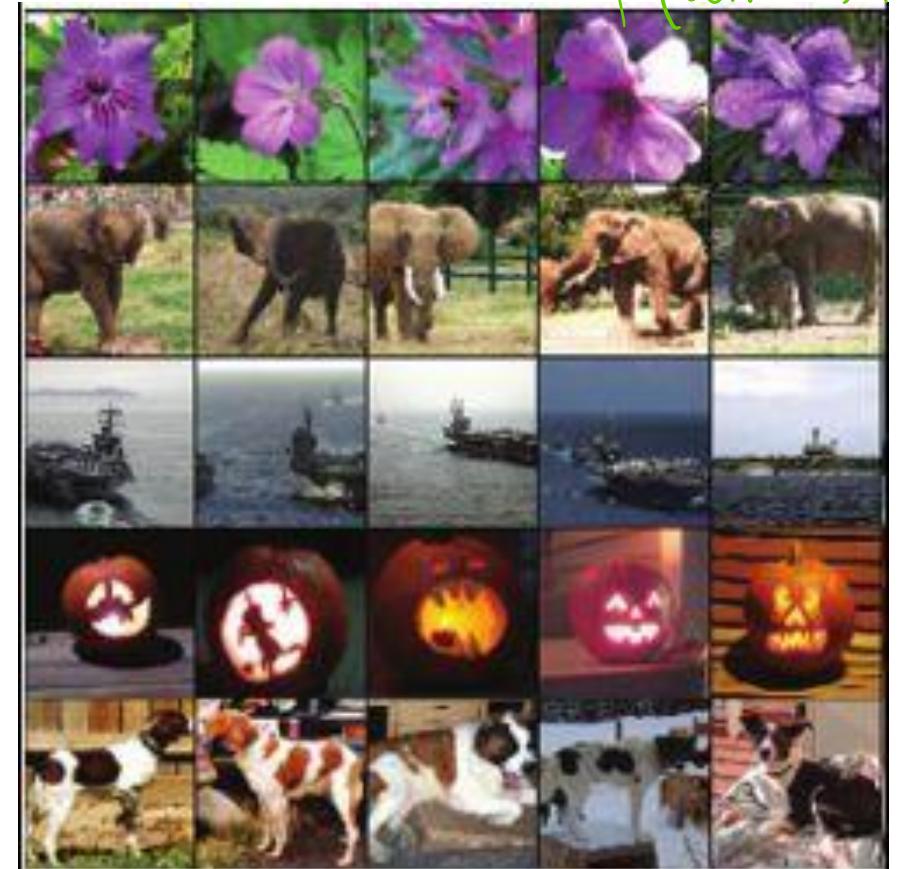


K Keras

PYTORCH
HUB

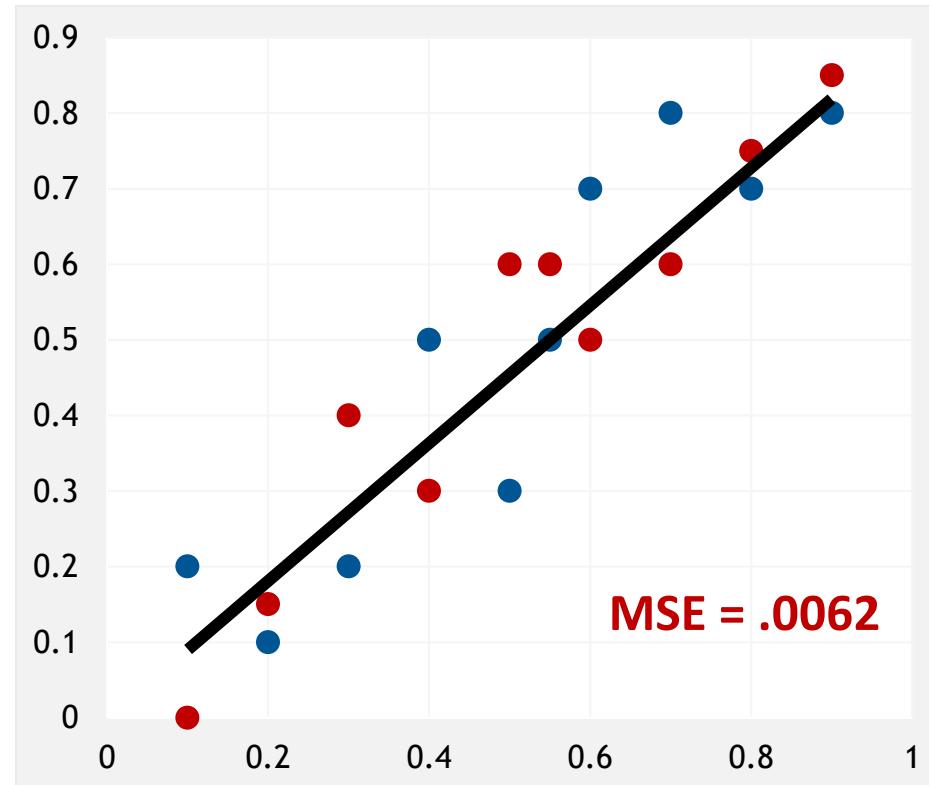
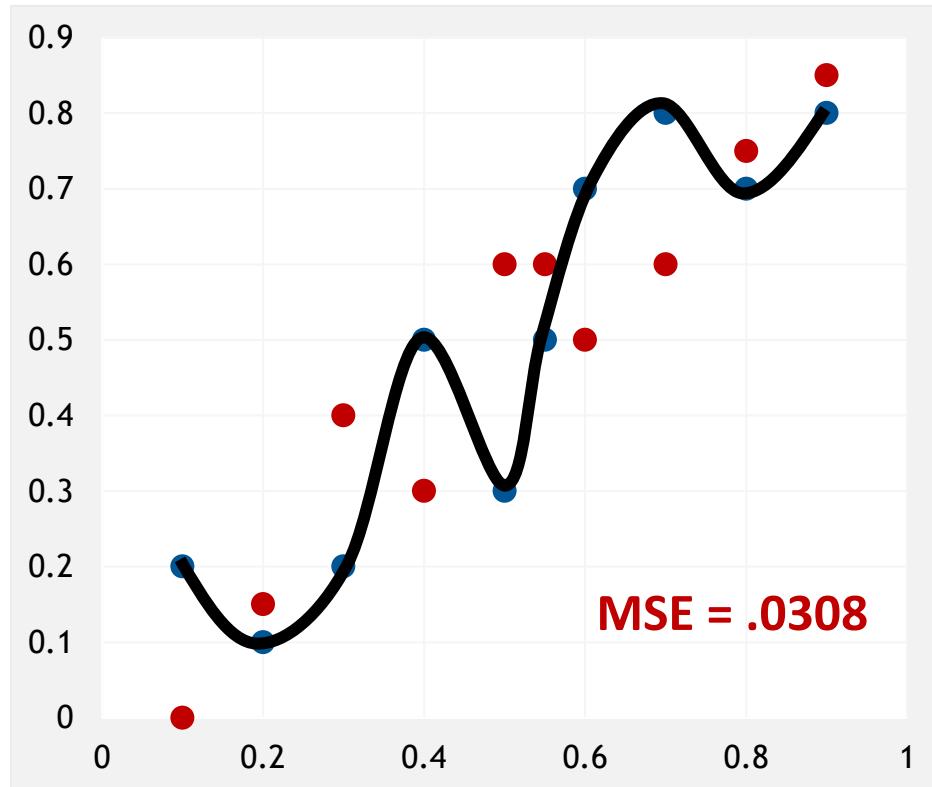
Dataset	Architecture
<input type="checkbox"/> MNIST database (60,000 images) <input type="checkbox"/> ImageNet (1.2 M images for 1000 classes) <input type="checkbox"/> ...	<input type="checkbox"/> VGG 16 (19) series <input type="checkbox"/> ResNet (50, 101, 150) <input type="checkbox"/> ...

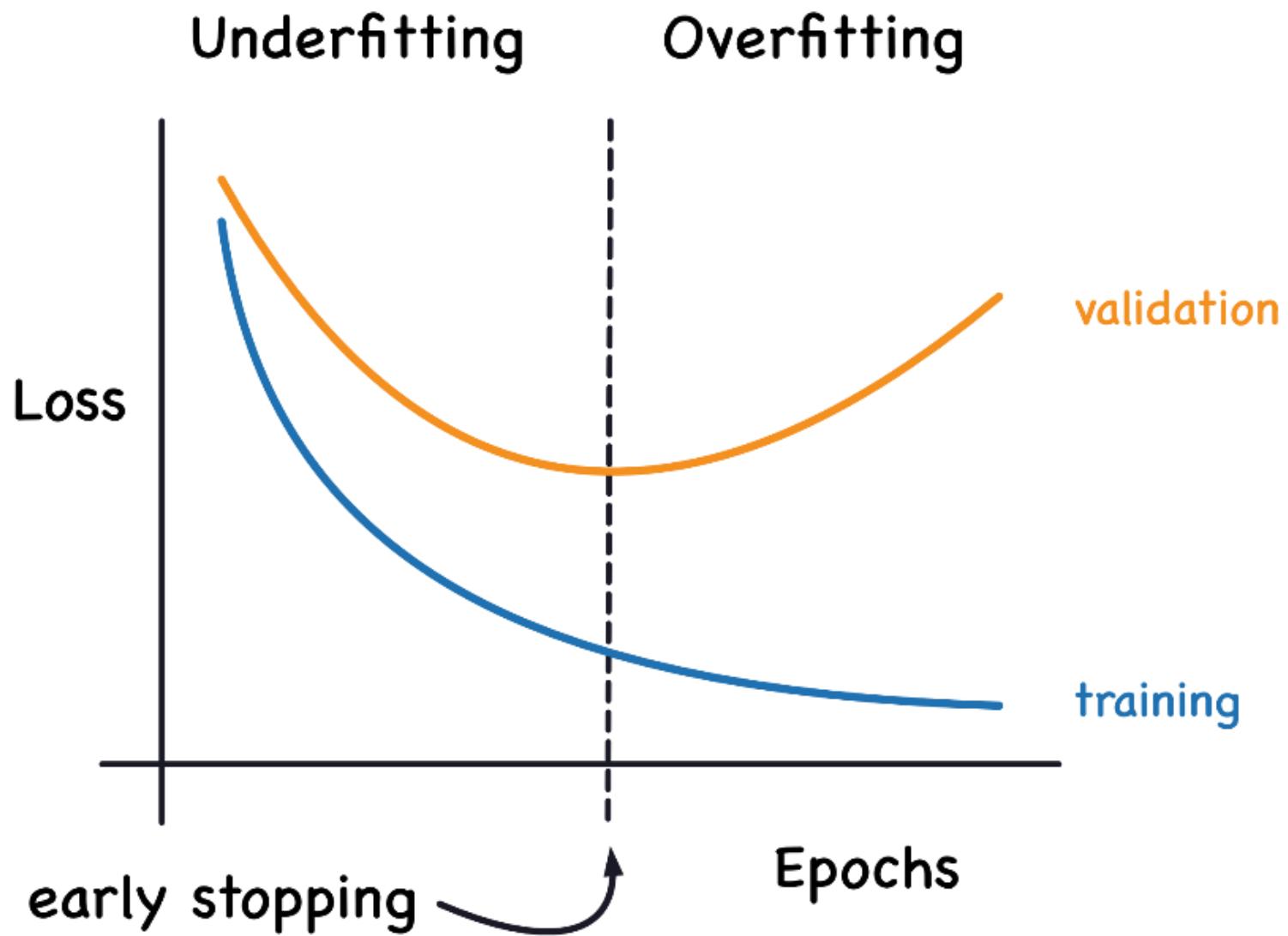
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9



OVERFITTING

Which Trendline is Better?





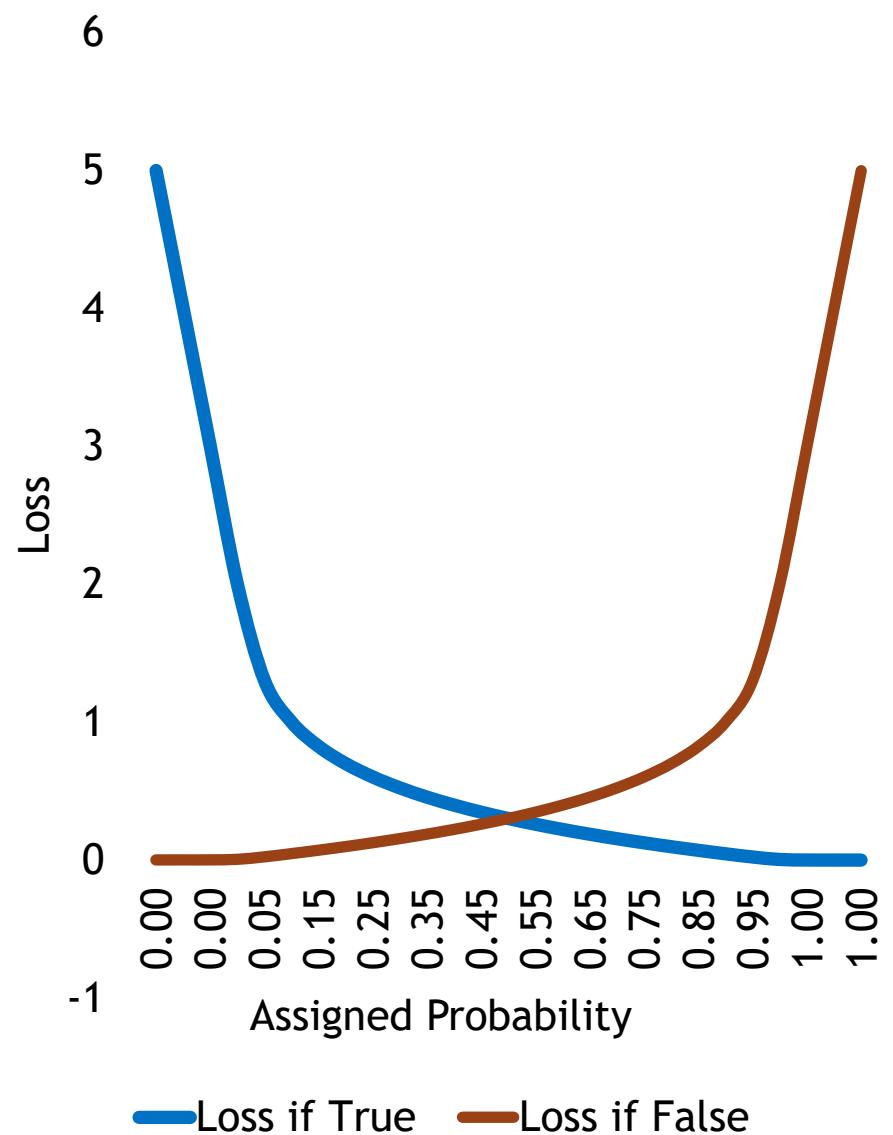
CROSS ENTROPY

```
1 def cross_entropy(y_hat, y_actual):
2     """Infinite error for misplaced confidence."""
3     loss = log(y_hat) if y_actual else log(1-y_hat)
4     return -1*loss
```

$$Loss = -((t(x) \cdot \log(p(x)) + (1 - t(x)) \cdot \log(1 - p(x))))$$

$t(x)$ = target (0 if False, 1 if True)

$p(x)$ = probability prediction of point x

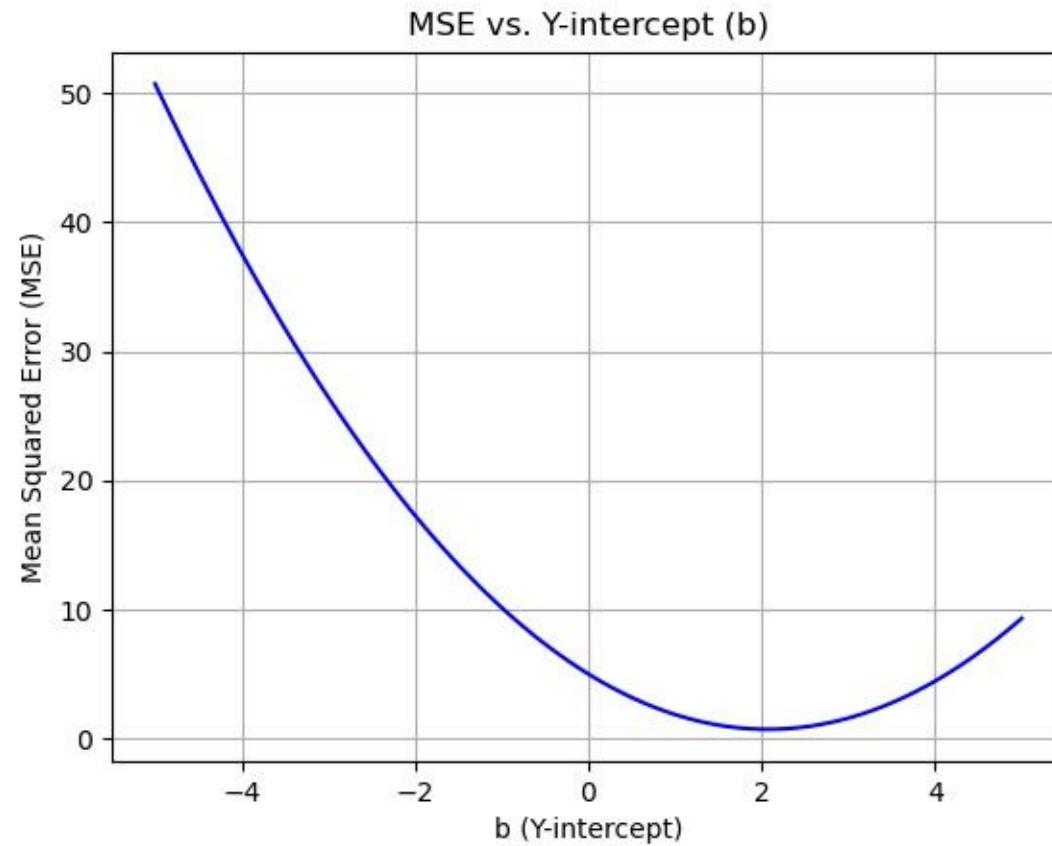


For $y = ax + b$

```
# Calculate MSE for different "b" values
b_values = np.linspace(-5, 5, 100)
mse_values = []

for b in b_values:
    predicted_y = true_slope * x + b
    mse = np.mean((y - predicted_y)**2)
    mse_values.append(mse)
```

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

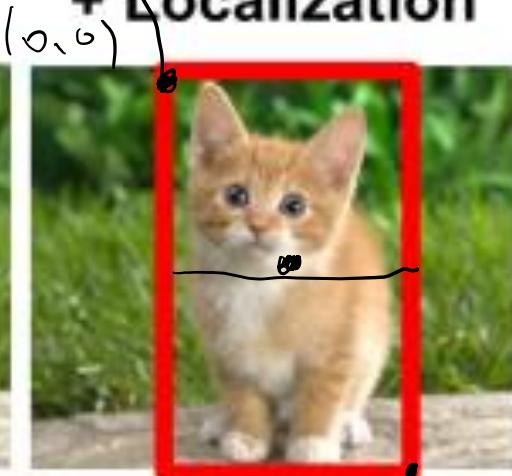


Classification



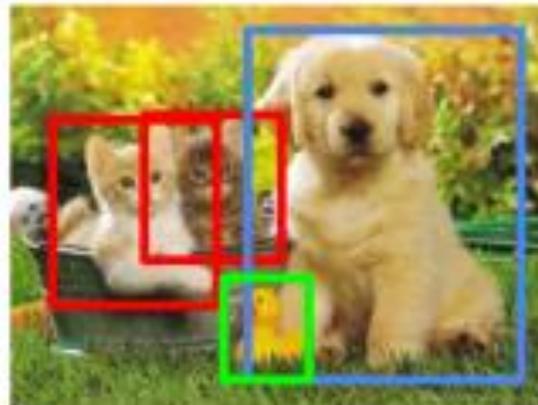
CAT

Classification + Localization



CAT $(x_1, y_1, w, h, C_1, C_2, C_3 \dots)$

Object Detection



CAT, DOG, DUCK $(x_1, y_1, x_2, y_2, C_1, C_2, C_3 \dots)$

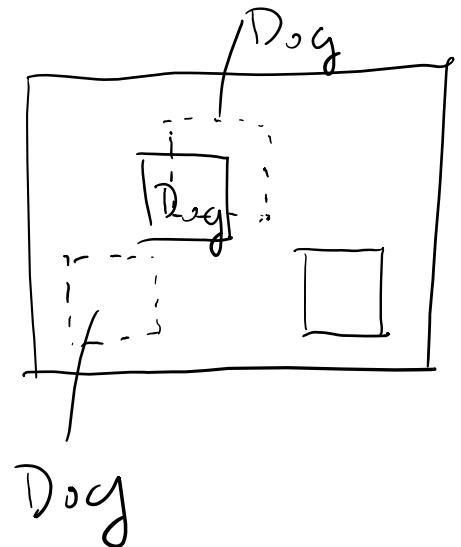
Instance Segmentation



CAT, DOG, DUCK

Single object

Multiple objects



ResNet

object score

$$\frac{\partial \text{loss}}{\partial w}$$

$$\text{boxes} \times (4 + 1 + C)$$

$$1000$$

pred



label



$$\begin{bmatrix} -x \\ -y \\ w \\ h \\ c_1 \\ c_2 \\ c_3 \\ 0 \end{bmatrix} \quad y_o$$

$$\begin{bmatrix} -\hat{x} \\ -\hat{y} \\ \hat{w} \\ \hat{h} \\ \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ 0 \end{bmatrix} \quad (\hat{x} - x)^2$$

Classification

$$\begin{bmatrix} \text{CE1} \\ \text{CE2} \\ \text{CE3} \\ \text{CE} \end{bmatrix}$$

Regression

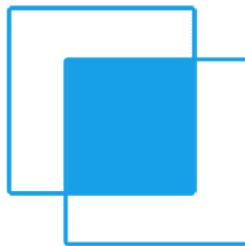
$$\begin{bmatrix} \text{loss} \end{bmatrix}$$

$$\text{loss}_2$$

$$\text{loss}_3$$

Performance metrics

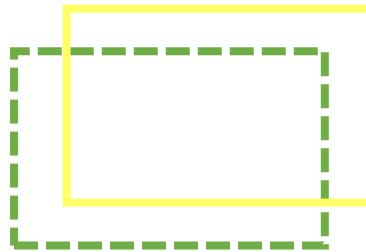
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



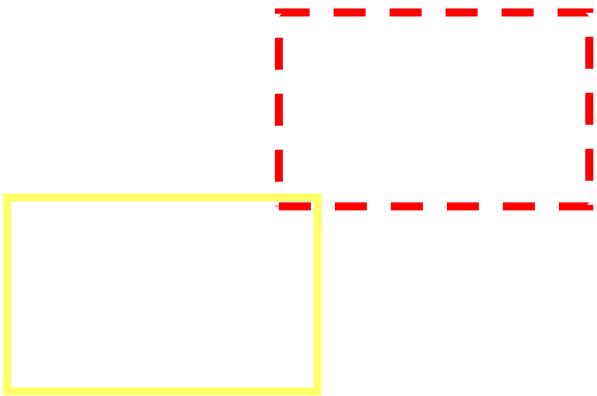
$$\text{precision} = \frac{1}{2} = 50\%$$

$$\text{recall} = \frac{1}{3} = 33\%$$

True Positive (TP): $\text{IoU} \geq 50\%$



False Positive (FP): $\text{IoU} < 50\%$



No True Negative (TN)

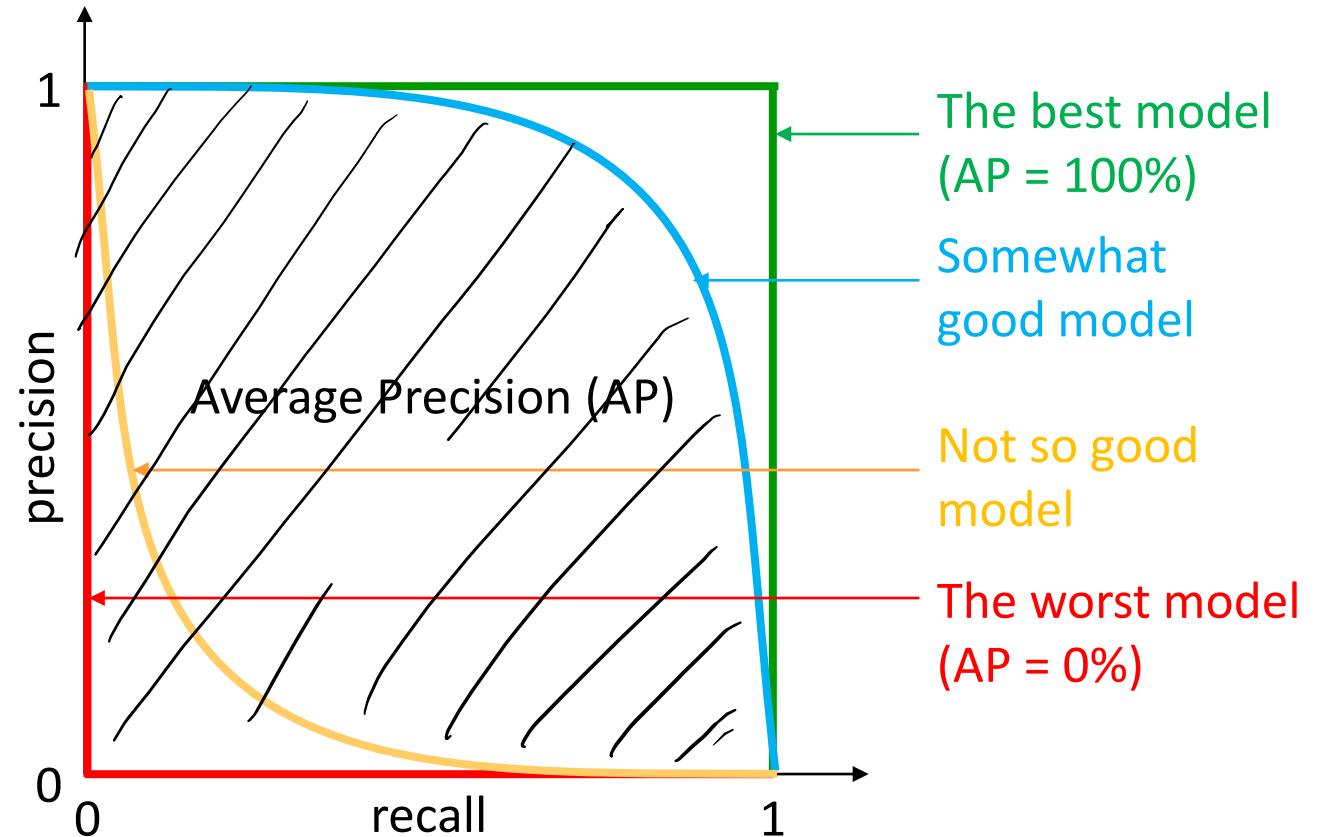


False Negative (FN): $\text{IoU} = 0$

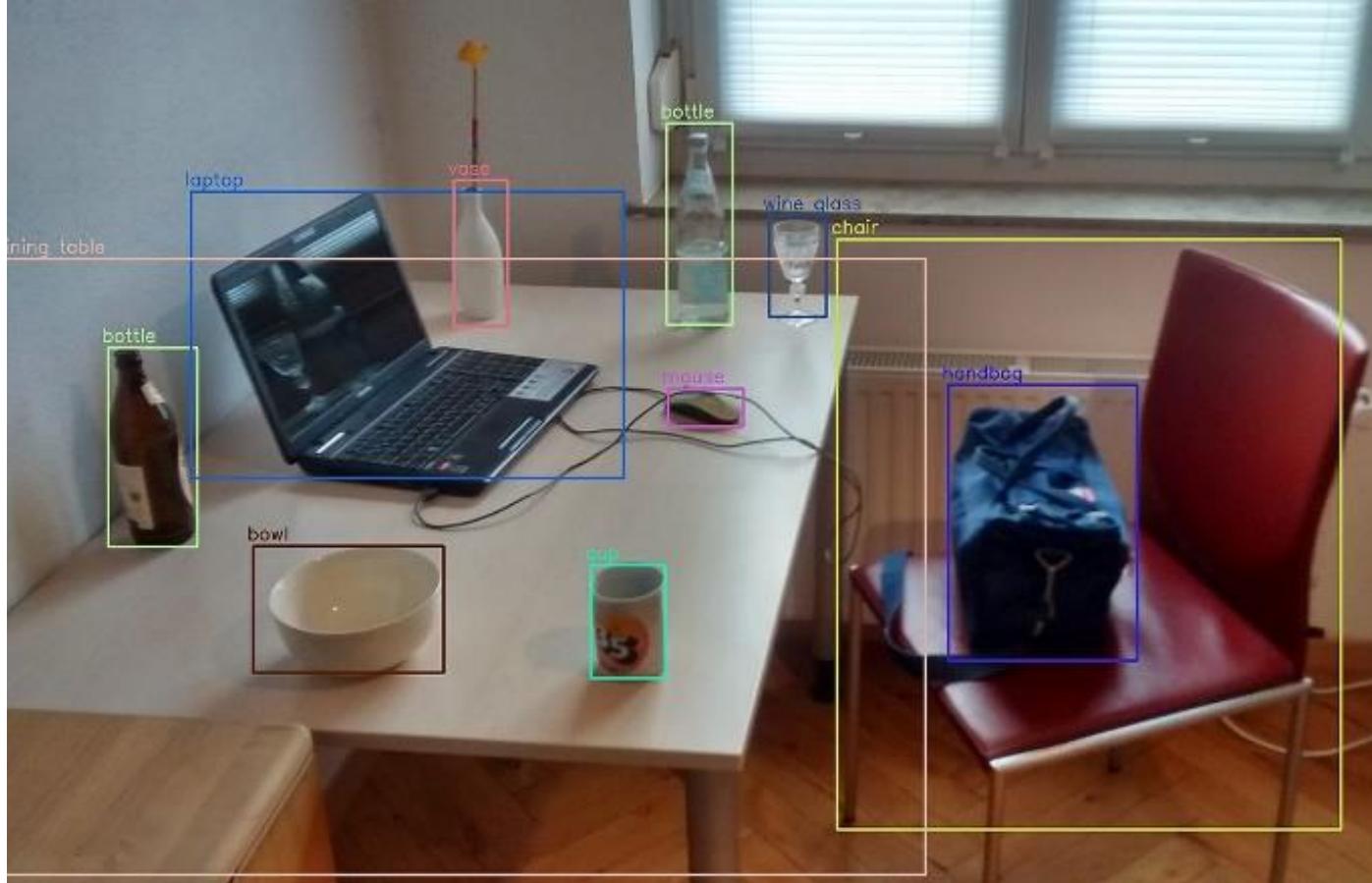
$$precision = \frac{TP}{TP + FP} = \frac{TP}{Predictions}$$

$$recall = \frac{TP}{TP + FN} = \frac{TP}{Ground\ Truth}$$

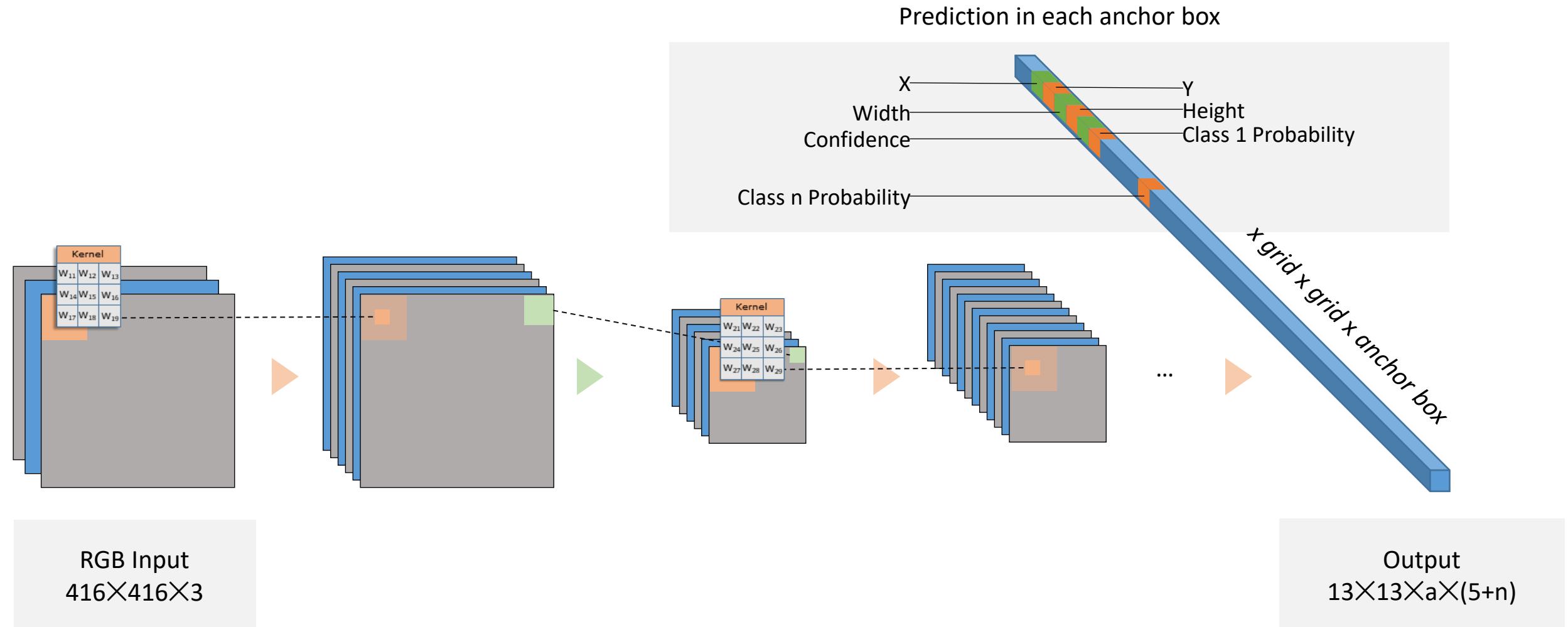
Mean average precision (mAP)



Dataset	Architecture
<input type="checkbox"/> COCO (300K image in 80 classes) <input type="checkbox"/> Open Images (9M images in 600 classes) <input type="checkbox"/> ...	<input type="checkbox"/> YOLO series <input type="checkbox"/> RCNN series <input type="checkbox"/> Retina Net <input type="checkbox"/> ...



You Only Look Once (YOLO) Architecture



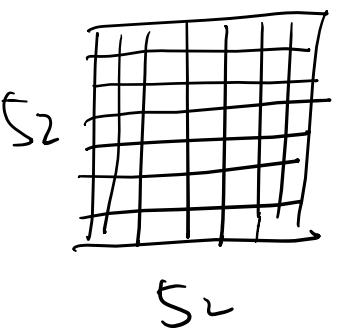
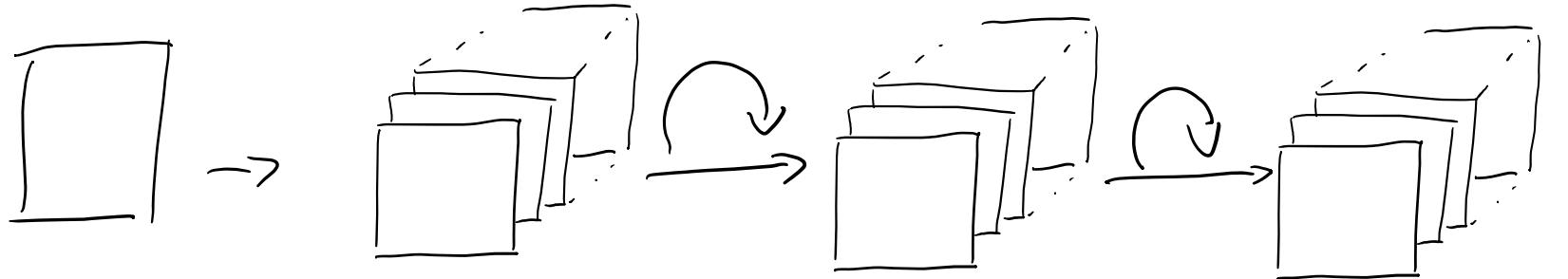
$$\beta \rightarrow [x, y, w, h, c_1, c_2, c_3]$$

$$Yolo \rightarrow$$

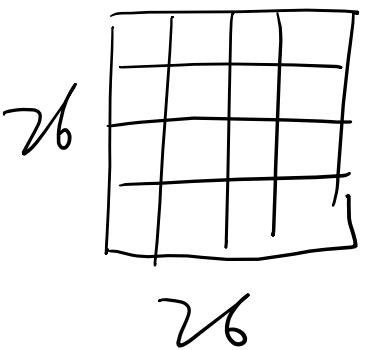
$$\begin{bmatrix} x' \\ y' \\ w' \\ h' \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} + \begin{bmatrix} x_a & y_a & w_a & h_a & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} x & y & w & h & c_1 & c_2 & c_3 \end{bmatrix}$$

offset

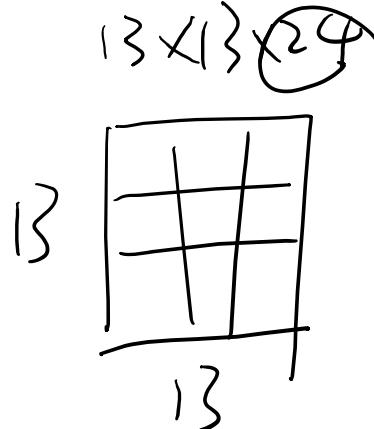
$$\begin{array}{cccccc} x_a & y_a & w_a & h_a \\ | & | & | & | \\ x & y & w & h \end{array}$$



Anchor



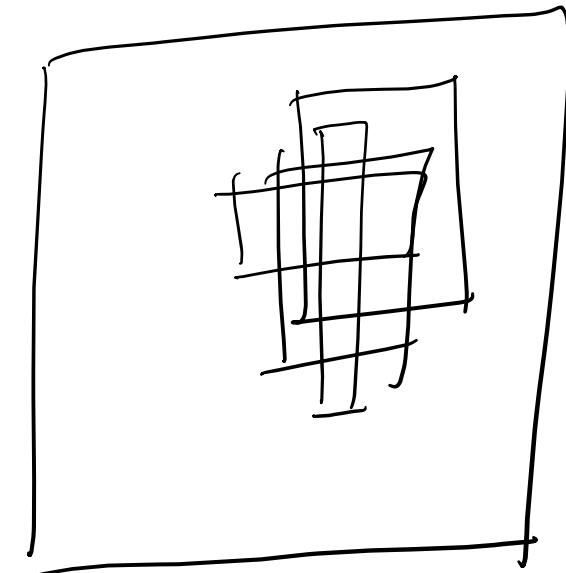
Anchor



Feature Pyrimid Network (FPN)

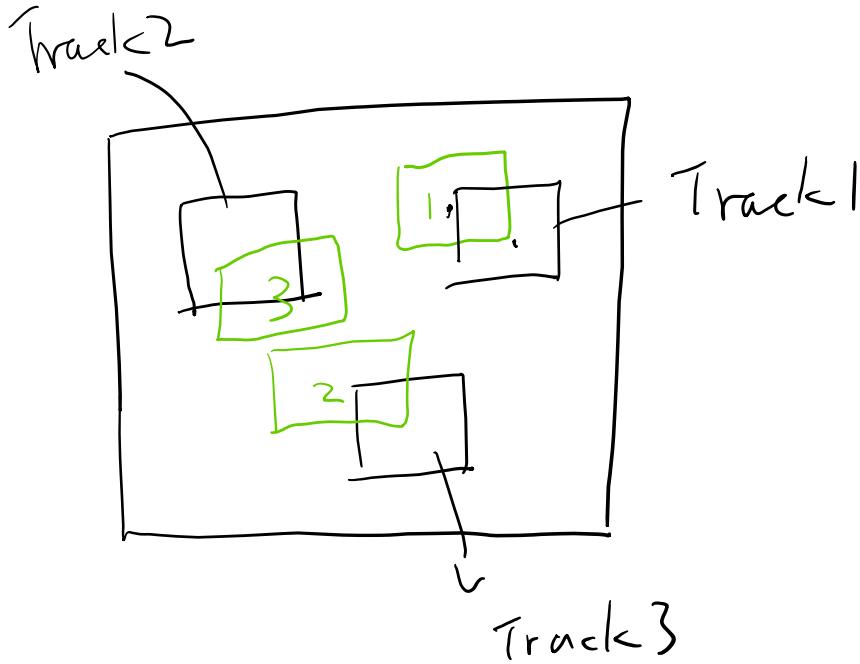
$$24 = 3(4 + 1 + 3)$$

$$13 \times 13 \times 3 = 507$$



Non max suppression

$$\begin{aligned}
& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \quad 1 \text{ when there is object, 0 when there is no object} \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \quad \text{Bounding Box Location (x, y) when there is object} \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \quad \text{Bounding Box size (w, h) when there is object} \\
& + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \quad 1 \text{ when there is no object, 0 when there is object} \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad \text{Confidence when there is no object} \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad \text{Class probabilities when there is object}
\end{aligned}$$

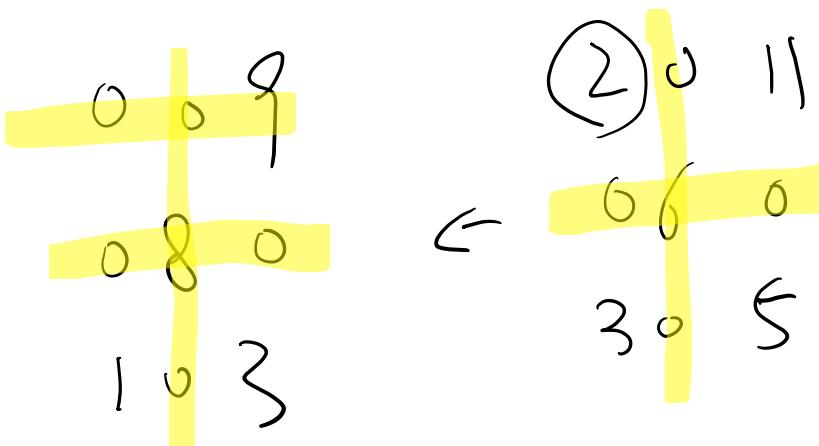


	T1	T2	T3	
D1	10	8	20	(-8)
D2	9	15	16	(-9)
D3	6	3	9	(-3)

D1 - T1 T2 - D3 T3 - D2

frame 1 T
frame 2 D

Hungarian Matching



2	0	12
0	6	1
3	0	6
(-0)	(-0)	(-1)

$$28 \begin{bmatrix} \square \end{bmatrix} \rightarrow \begin{bmatrix} \square \end{bmatrix} \begin{bmatrix} \square \end{bmatrix} \text{ loss 1}$$

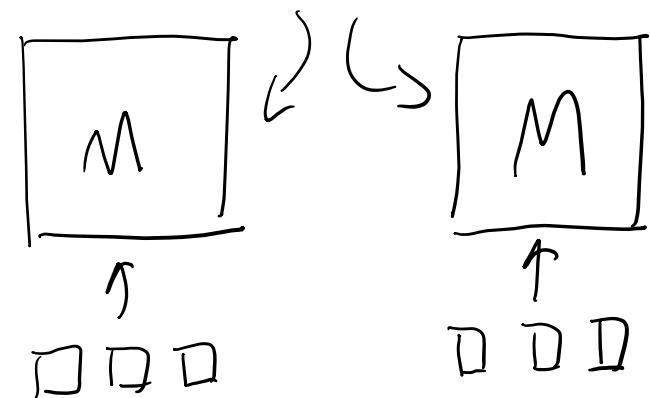
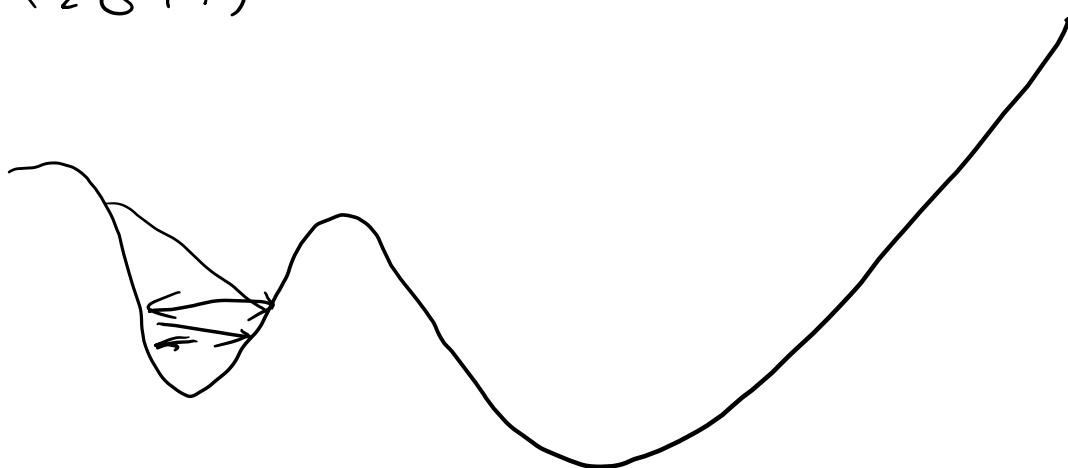
$$\frac{1}{2} \cdot \frac{\partial (\text{loss1} + \text{loss2})}{\partial w} \cdot LR$$

$$28 \begin{bmatrix} \square \end{bmatrix} \rightarrow \begin{bmatrix} \square \end{bmatrix} \begin{bmatrix} \square \end{bmatrix} \text{ loss 2}$$

$$\frac{\frac{\partial \text{loss1}}{\partial w} + \frac{\partial \text{loss2}}{\partial w}}{2}$$

(2, 28, 28, 1)

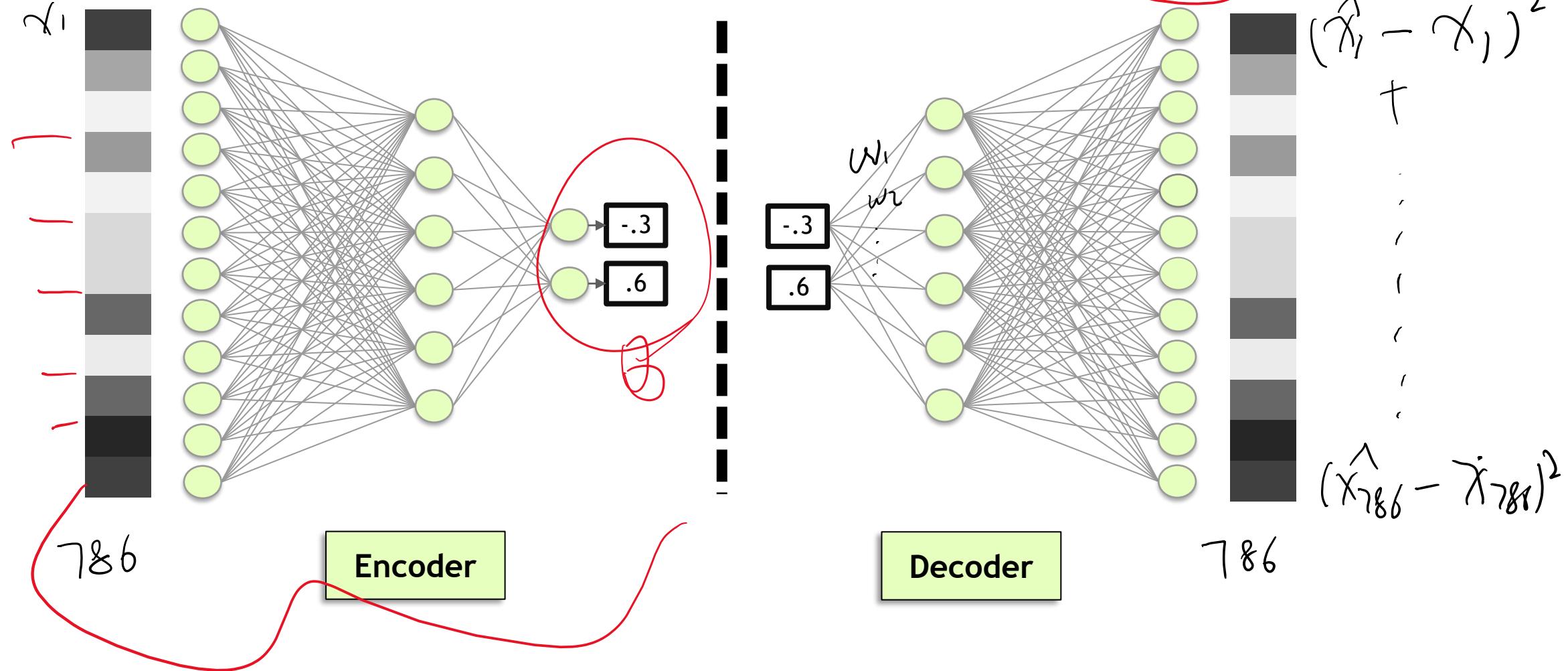
MSE



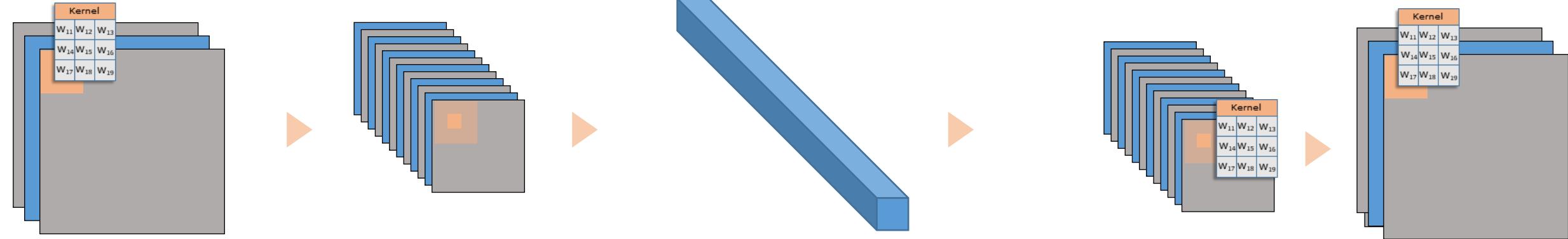
Dataset	Architecture
<input type="checkbox"/> ADE20k (150 classes)	<input type="checkbox"/> UNet
<input type="checkbox"/> COCO	<input type="checkbox"/> SegNet
<input type="checkbox"/> ...	<input type="checkbox"/> ...



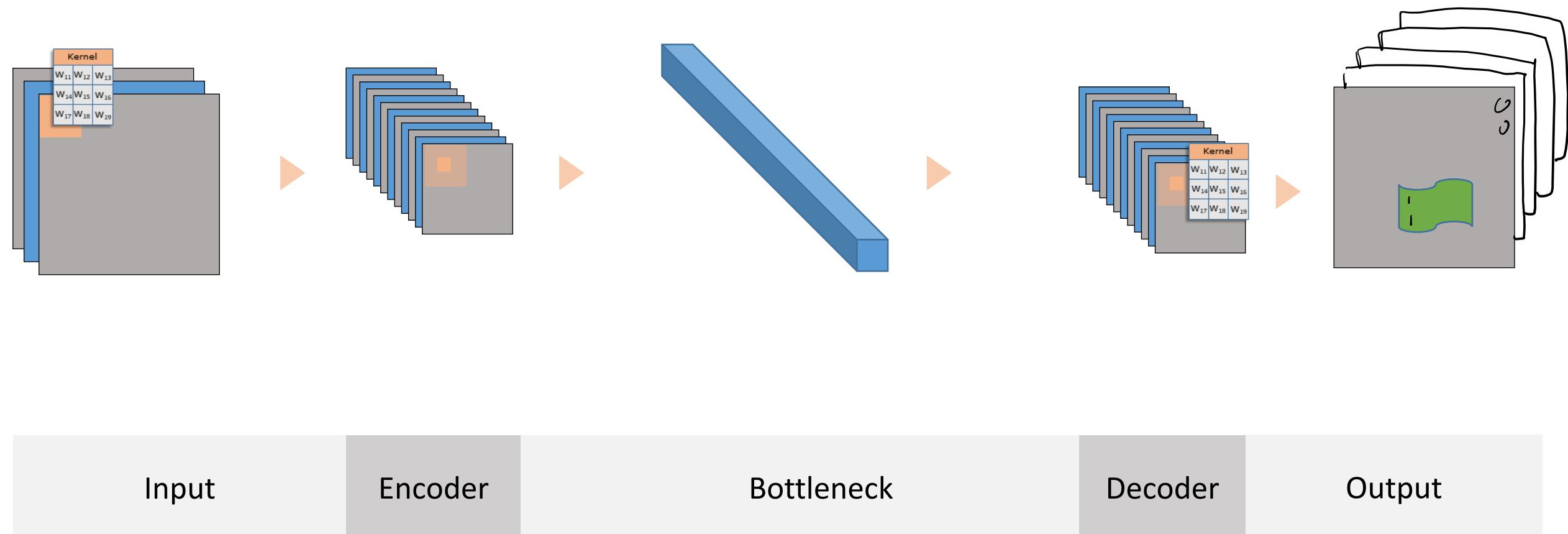
AUTOENCODERS

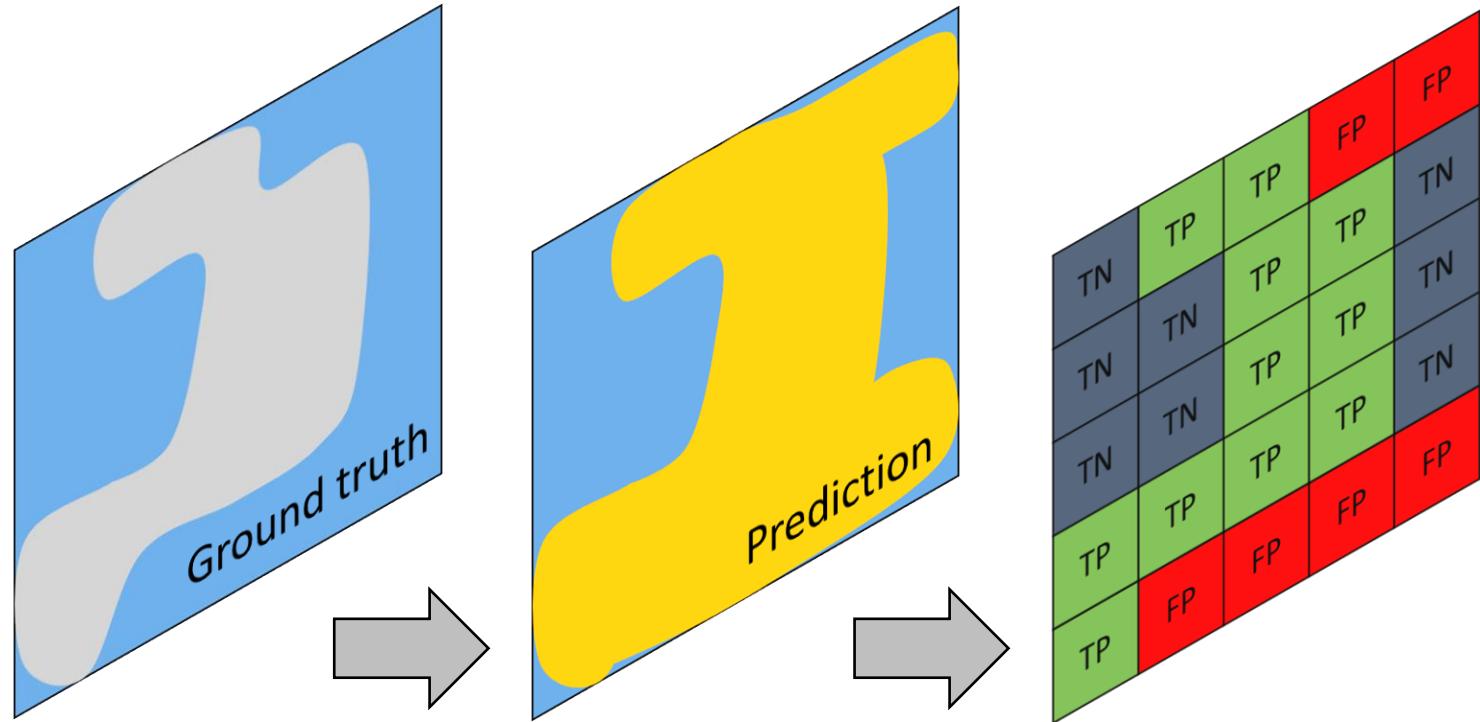


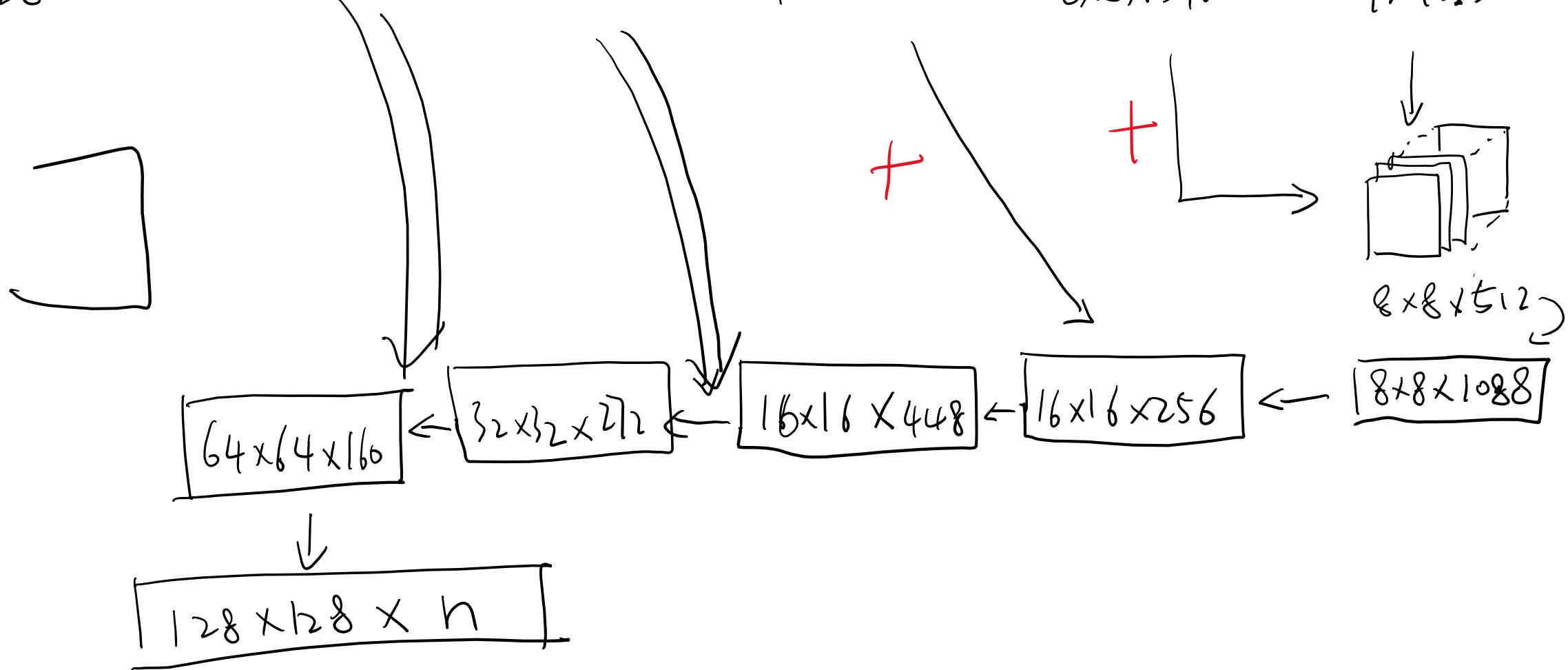
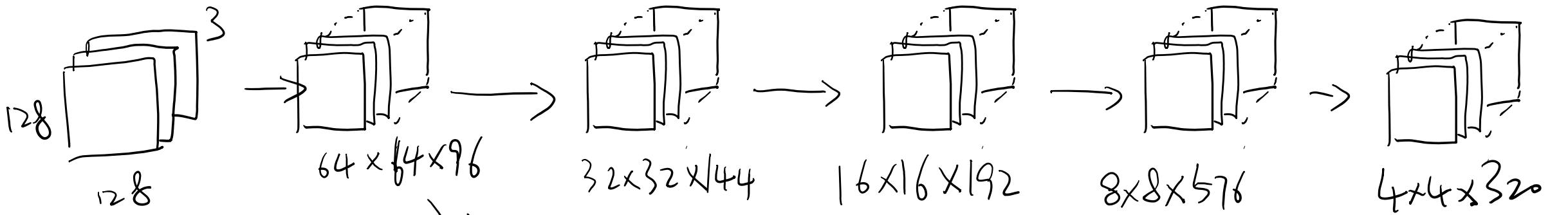
CNN Autocoder

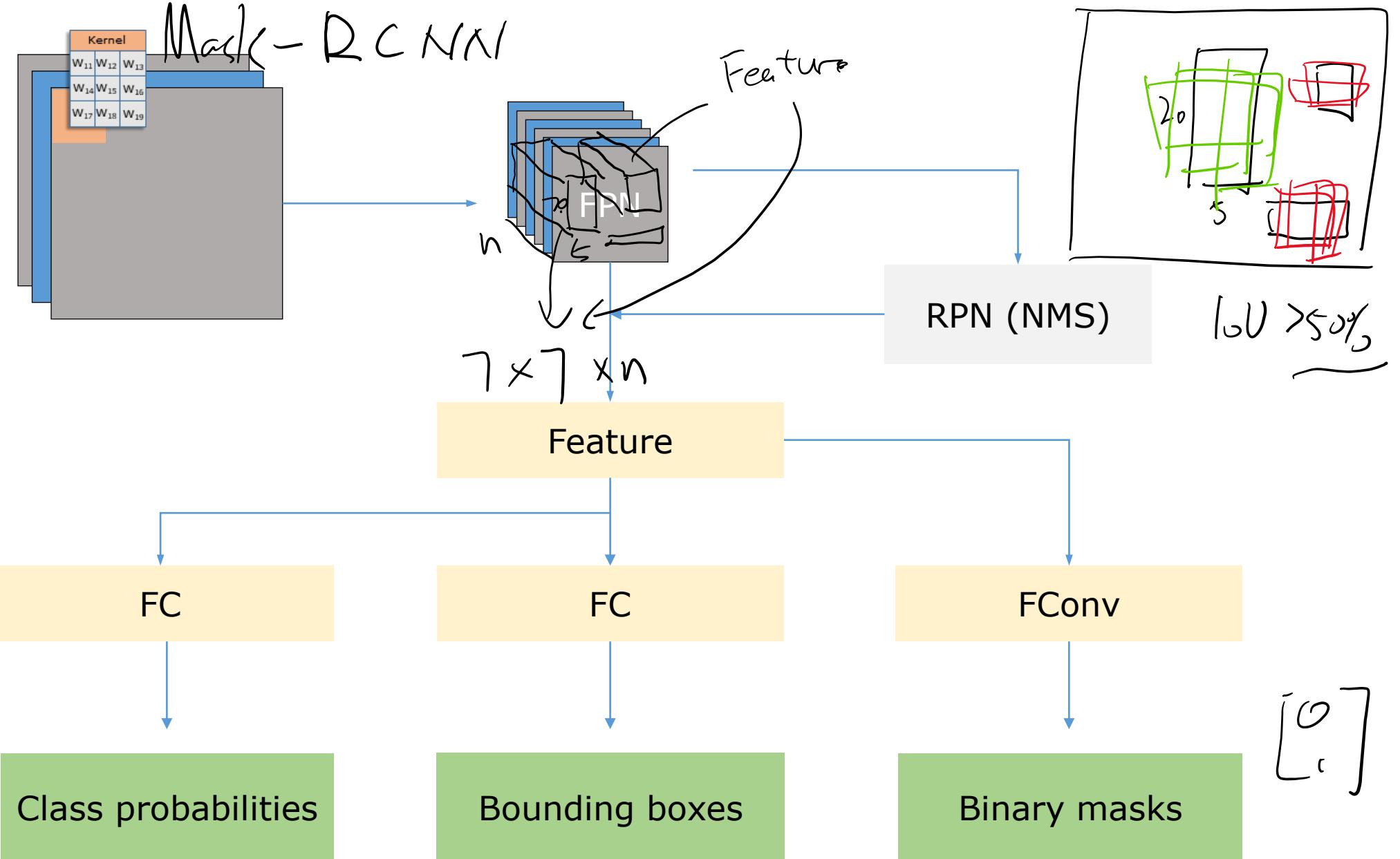


CNN segmentation









Conv2D Transpose

$$\begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix}$$

2×2

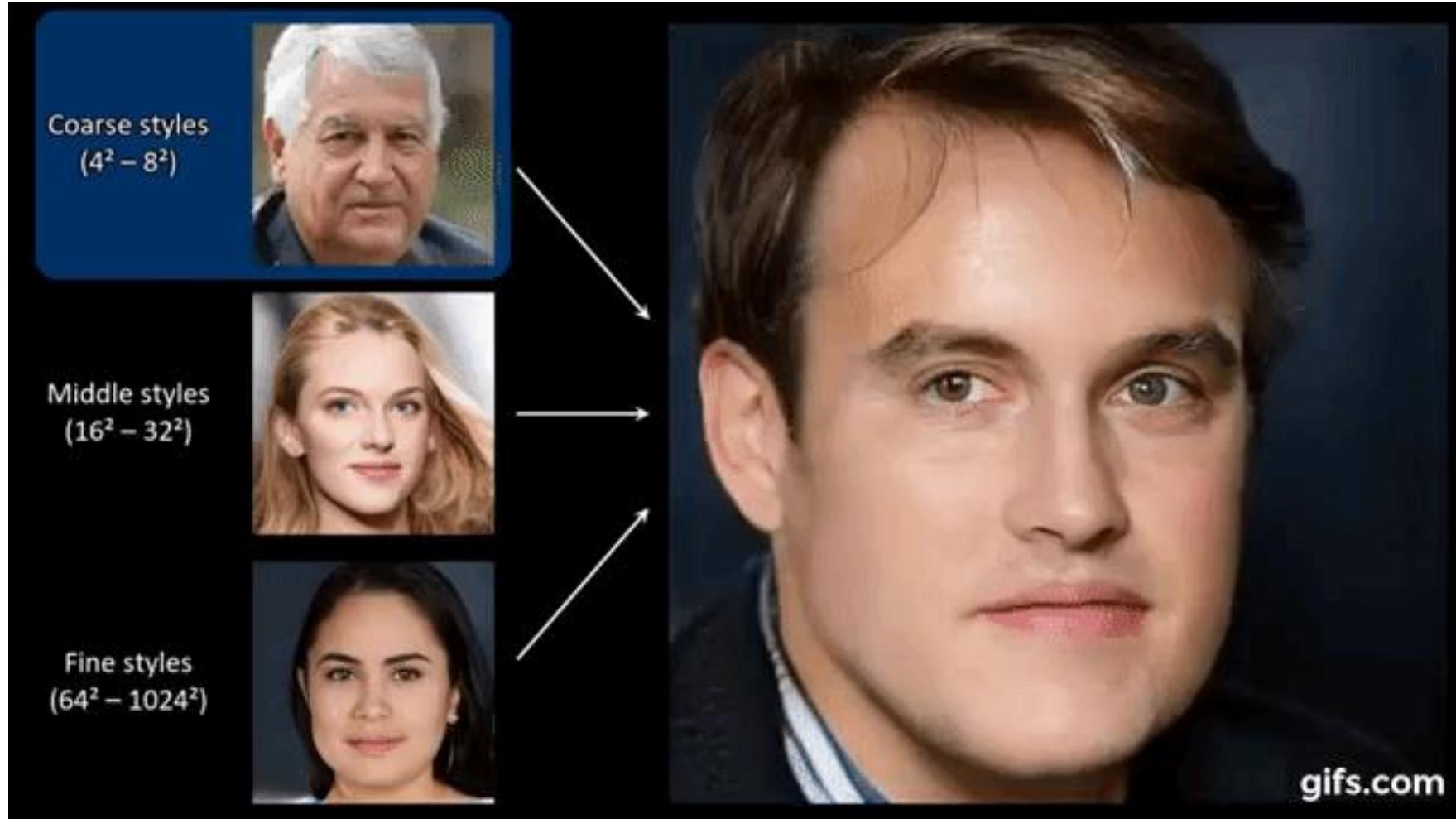
$$\begin{bmatrix} W_1 & W_2 \\ & \vdots \\ & W_8 \end{bmatrix}$$

3×3

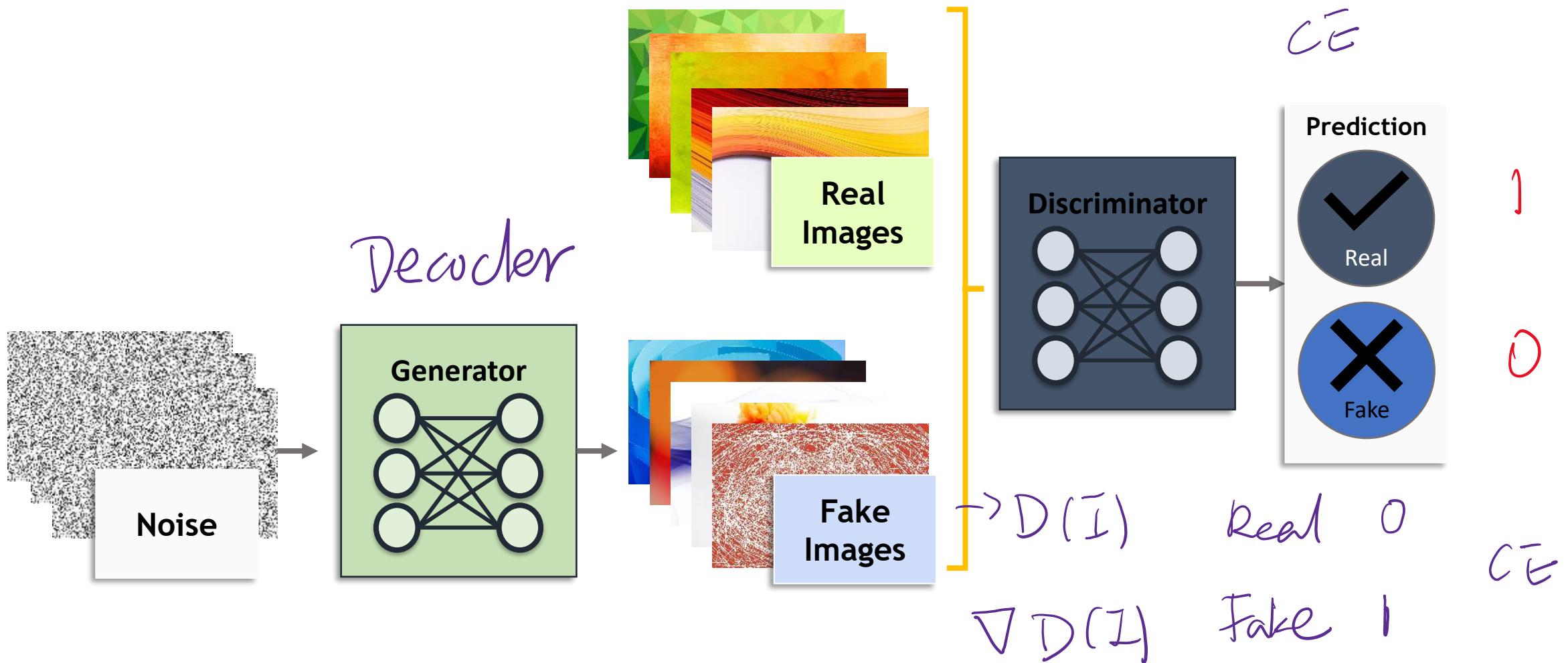
$$\begin{bmatrix} W_1V_2 & W_2V_2 \\ W_2U_1 & \vdots \\ W_8U_2 & W_9V_2 \\ & NqV_2 \end{bmatrix}$$

4×4

Generative Adversarial Network (GAN)



Generative Adversarial Networks (GANs)

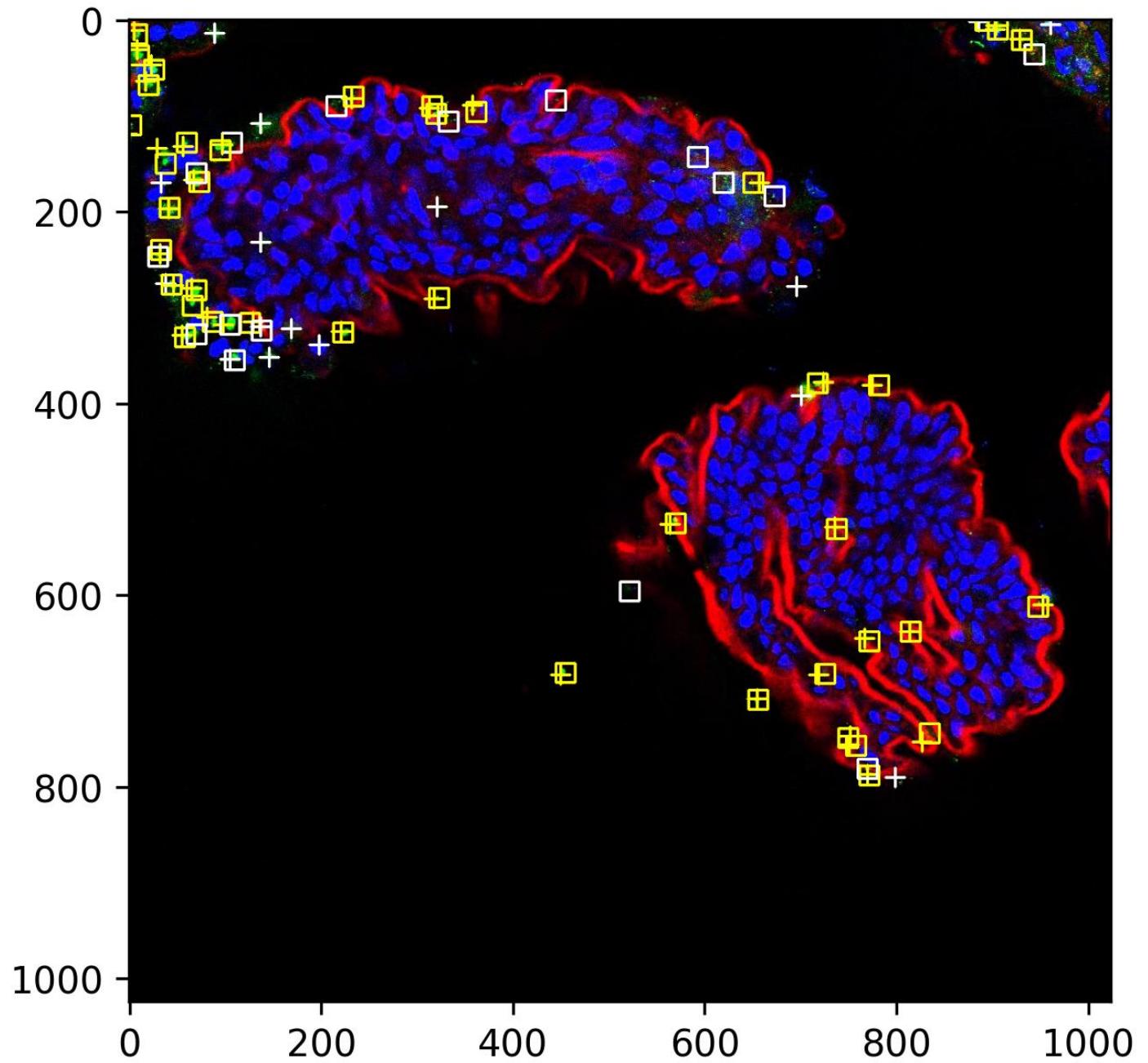


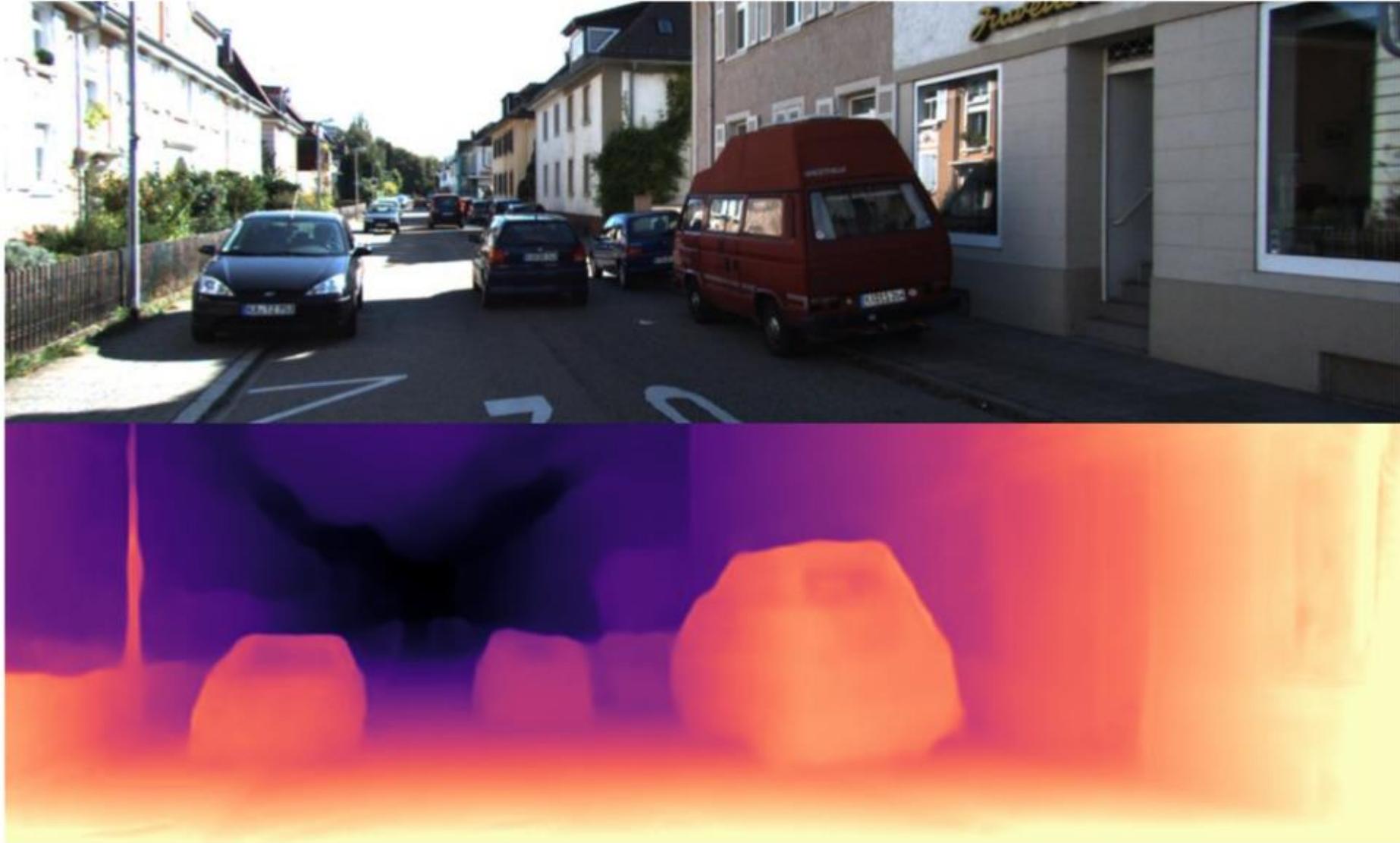


Source: <https://www.youtube.com/watch?v=2DjQUK11YeY>

Source: <https://www.youtube.com/watch?v=pWcXeWIGM>







Human captions from the training set



A cute little dog **sitting** in a heart drawn on a sandy **beach**.



A **dog** walking **next to** a little **dog** on top of a **beach**.



A large brown **dog** **next to** a small **dog** looking out a window.

Automatically captioned



A **dog** is **sitting** on the **beach** **next to** a **dog**.