Convolutional Neural Networks (CNN) for machine vision

Dr.Varodom Toochinda

Dept. of Mechanical Engineering

Kasetsart University

September 2022

$$y[m,n] = x[m,n] * h[m,n] = \sum_j \sum_i x[i,j]h[m-i,n-j]$$

Convolution in image processing

x

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

*

h

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

=

y

0	1	0	1
11	5	-10	-2
13	6	-19	-3
-1	0	-8	-11

2D convolution example

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

*

h

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

=

y

0	1	0	1
11	5	-10	-2
13	6	-19	-3
-1	0	-8	-11

(4)(1)+(2)(0)+(9)(-1)+(3)(1)+(1)(0)+(5)(-1)+(8)(1)+(7)(0)+(1)(-1) =
4+0-9+3+0-5+8+0-1 = 0

x

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

*

h

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

=

y

0	1	0	1
11	5	-10	-2
13	6	-19	-3
-1	0	-8	-11

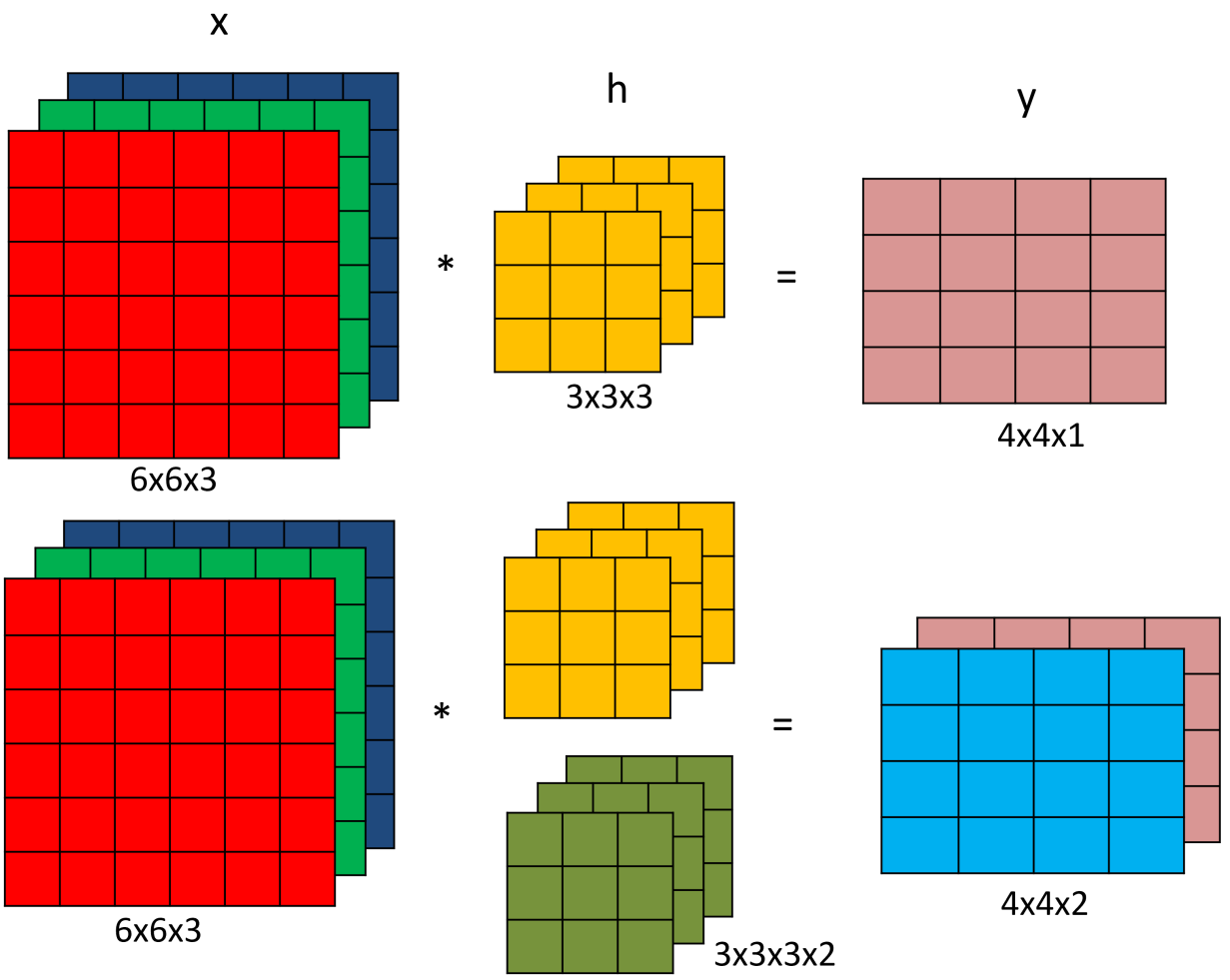
(2)(1)+(6)(0)+(2)(-1)+(1)(1)+(8)(0)+(1)(-1)+(5)(1)+(7)(0)+(8)(-1) =
2+0-2+1+0-1+5+0-8 = -3

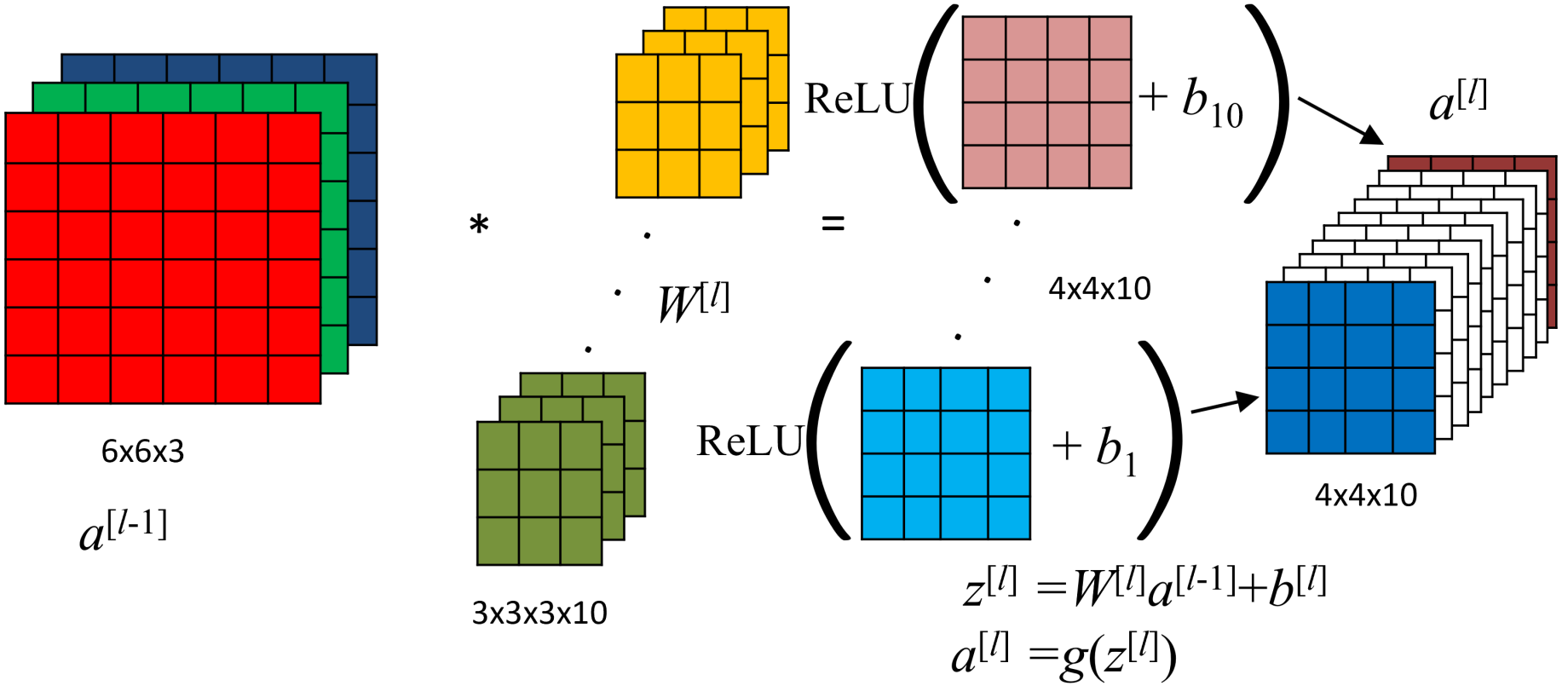
2D convolution example

edge detection with convolution

see convolution.ipynb

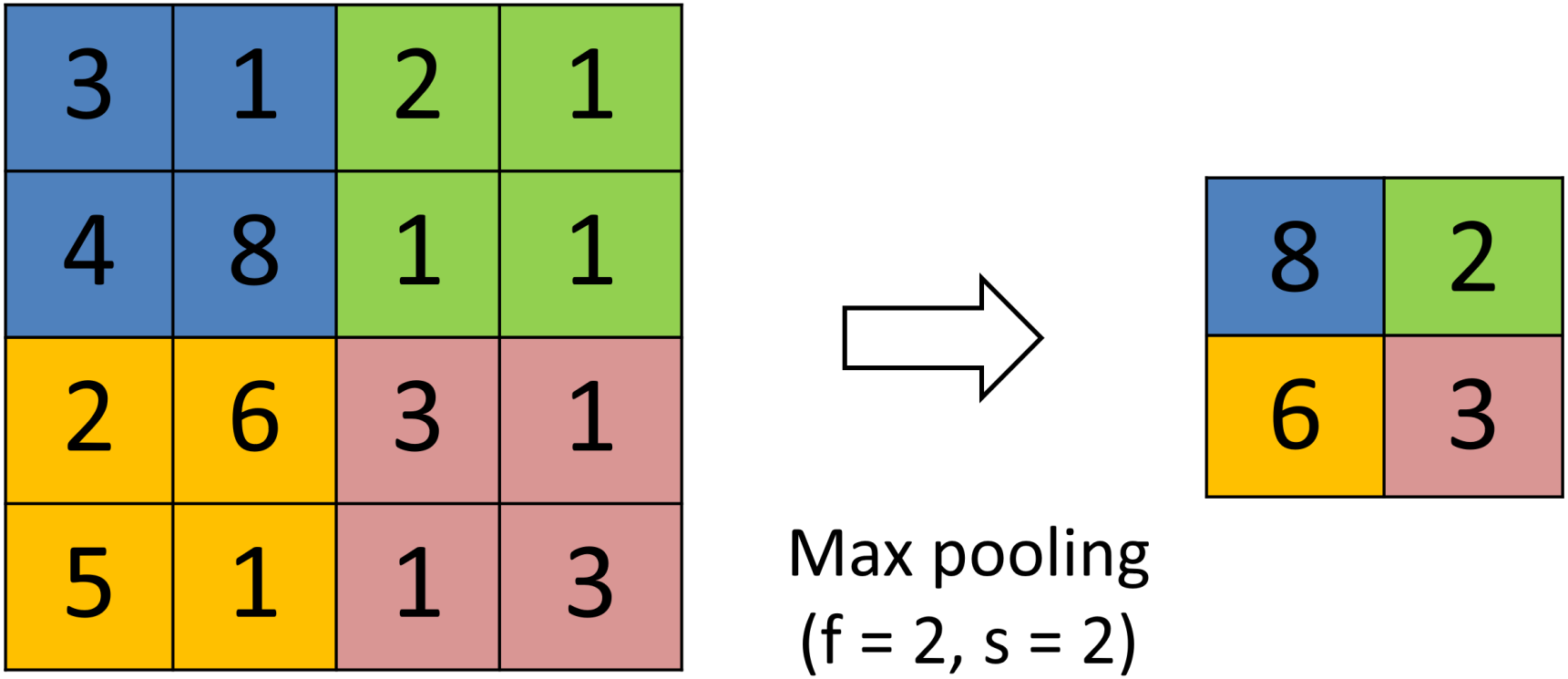
3D convolution



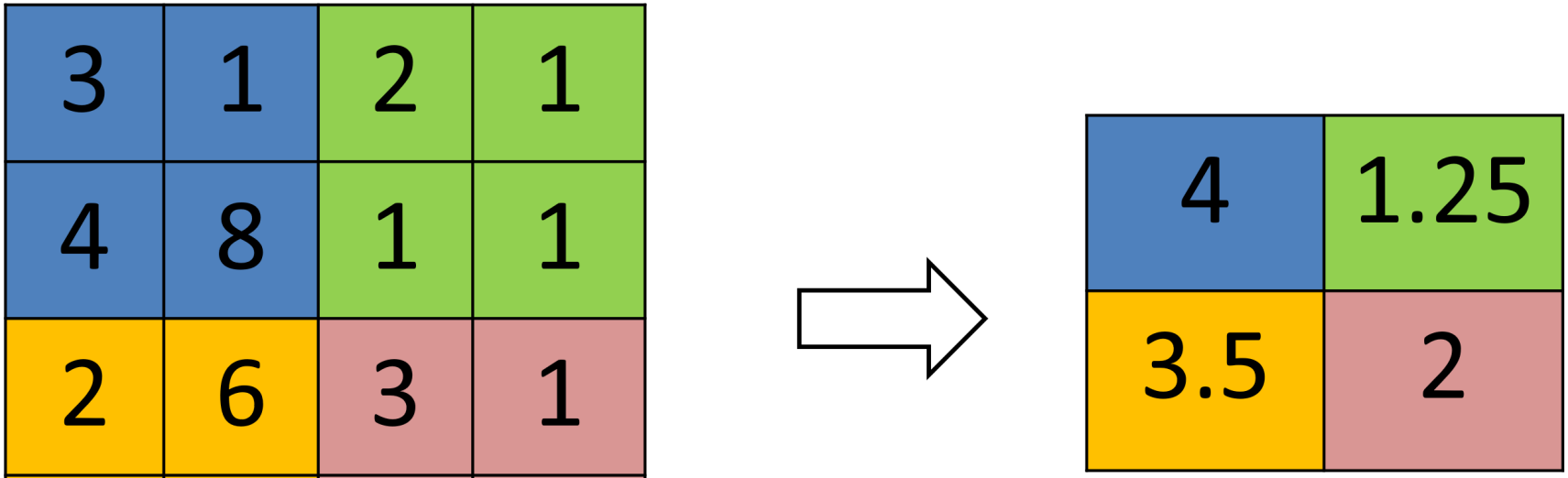


convolution layer in NN

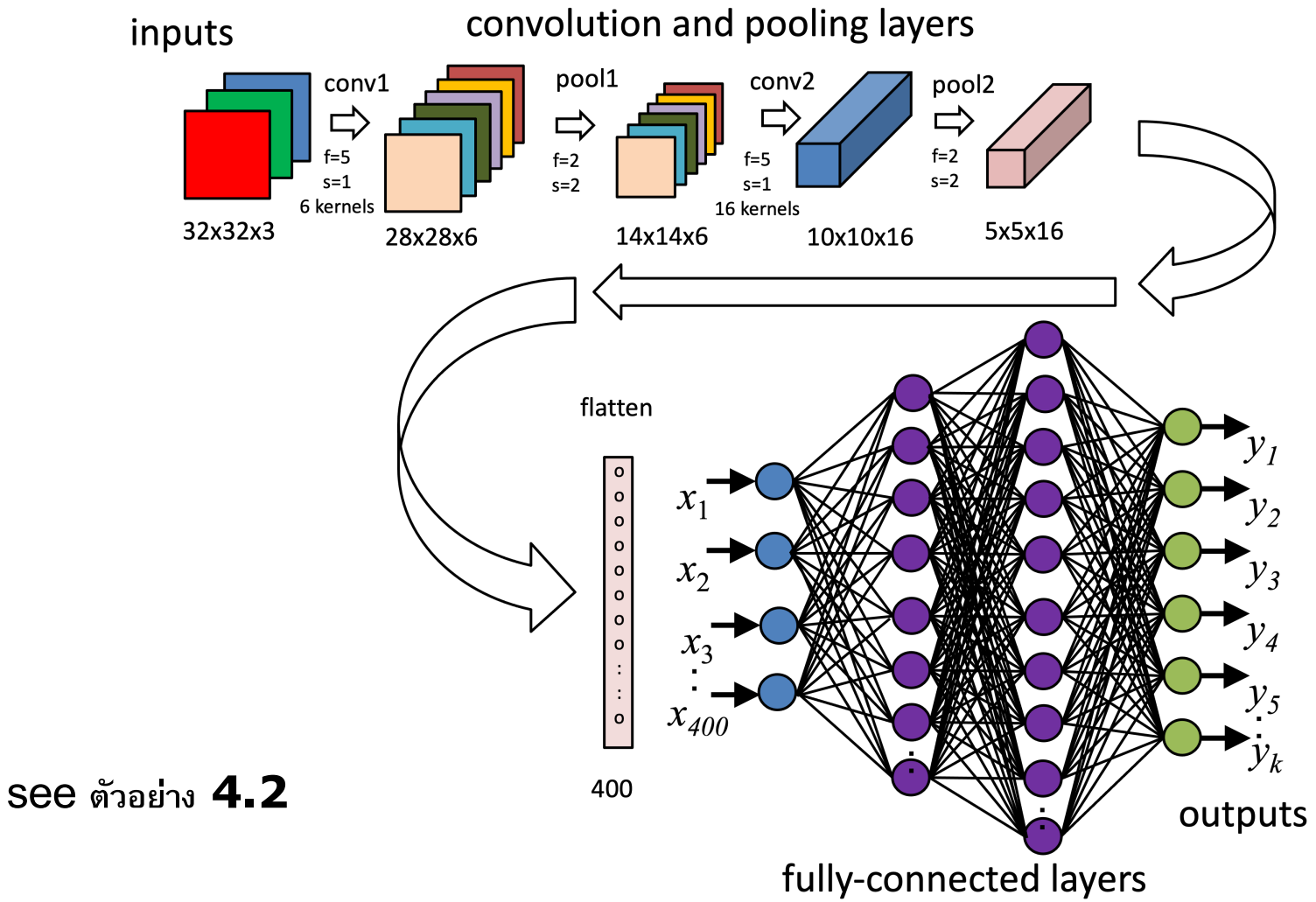
max pooling layer



average pooling layer



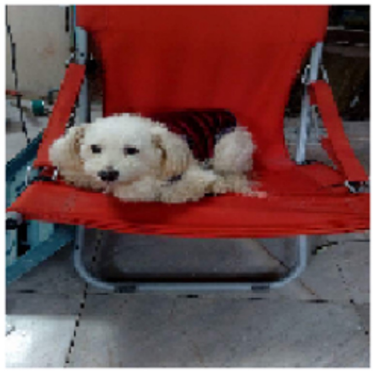
CNN model example



see ตัวอย่าง **4.2**

cat/dog classification

```
[1.]
jackie.jpg is a dog
```

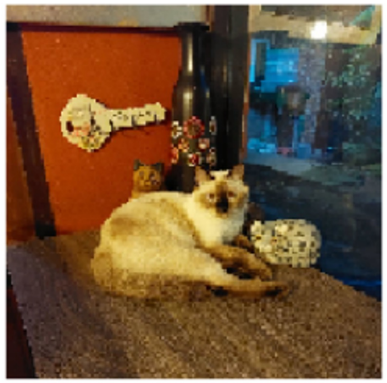


```
[1.]
maam.jpeg is a dog
```

```
[0.]
dollar_scale.jpg is a cat
```

```
[1.]
fongbeer1.jpeg is a dog
```

```
[1.]
bingzoo.jpg is a dog
```



```
[0.]
ninja_bingz.jpg is a cat
```

DNN/CNN comparision (SVHN)

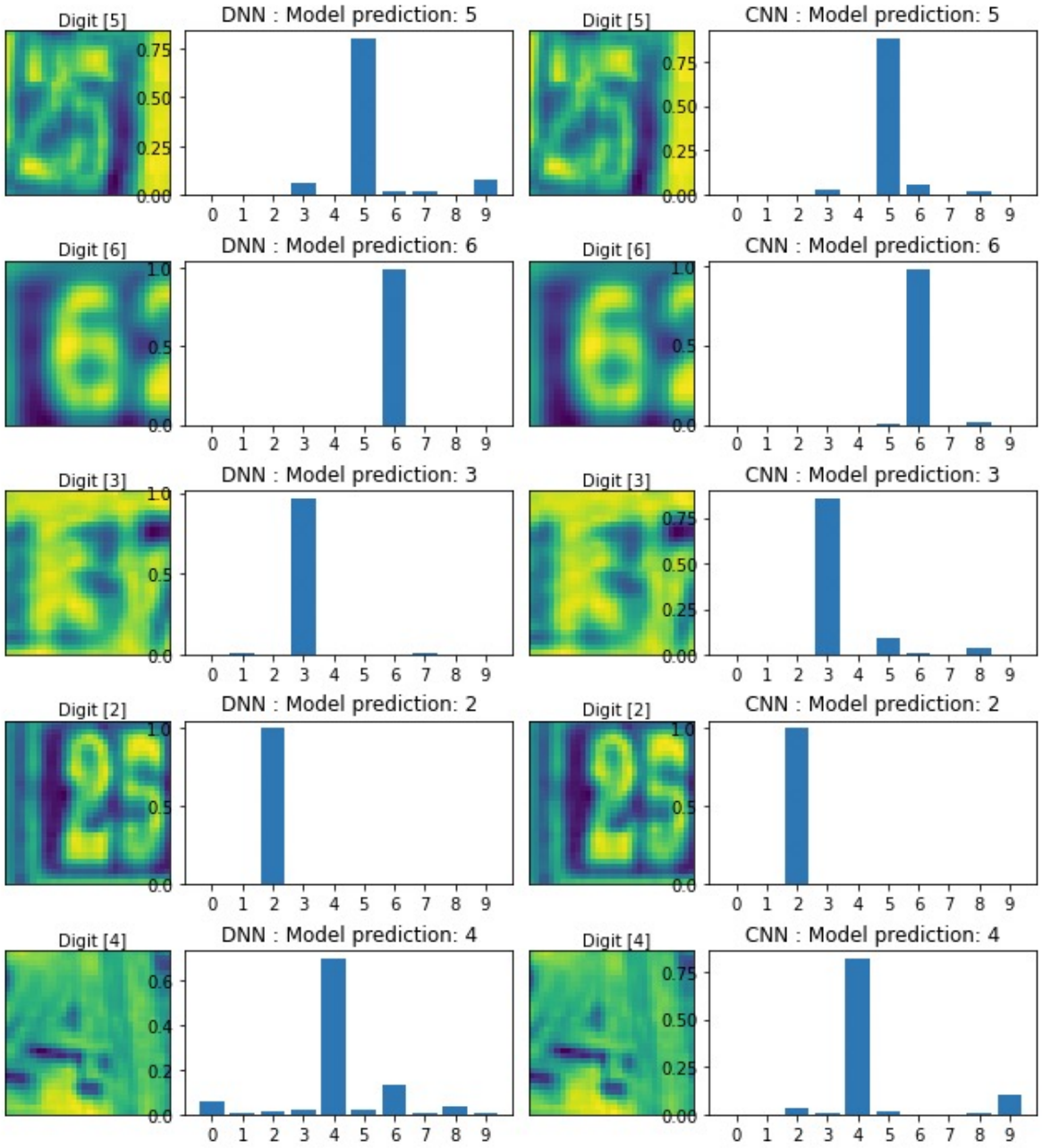
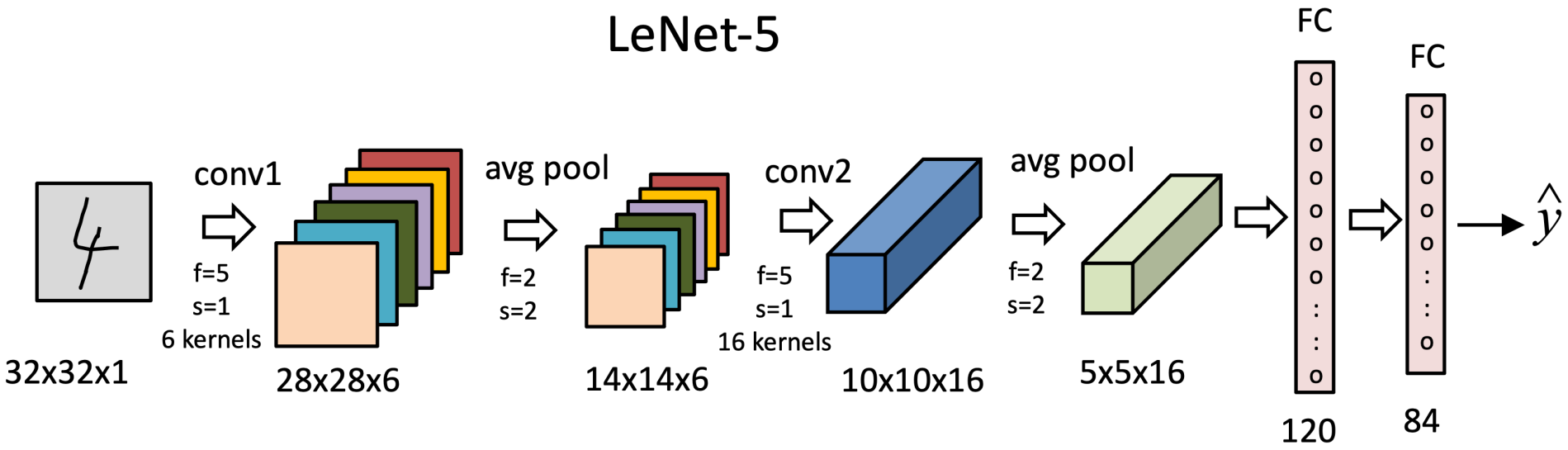


image augmentation

CNN case study

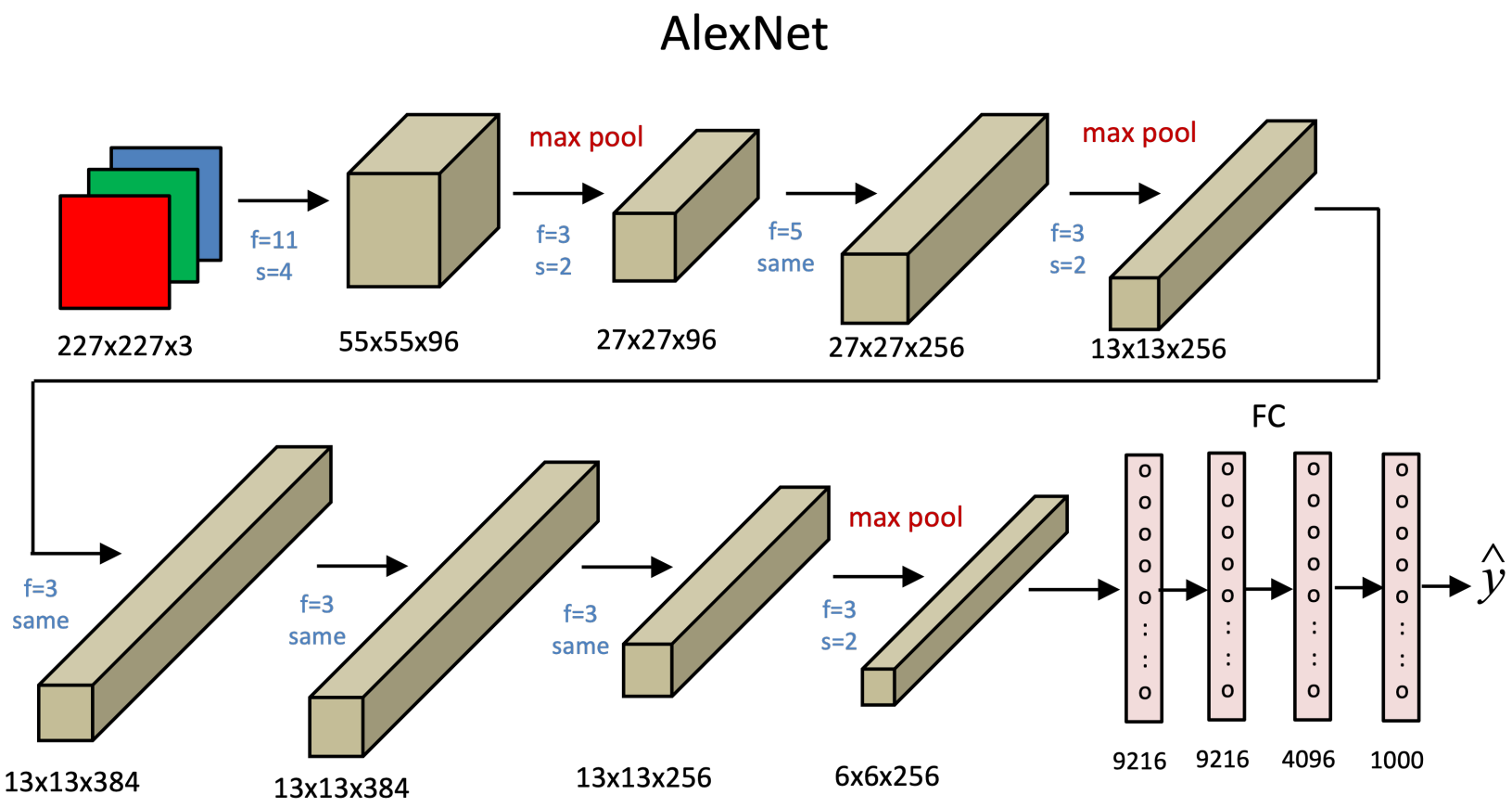
LeNet-5
AlexNet
VGG-16
ResNets
Inception model

LeNet-5



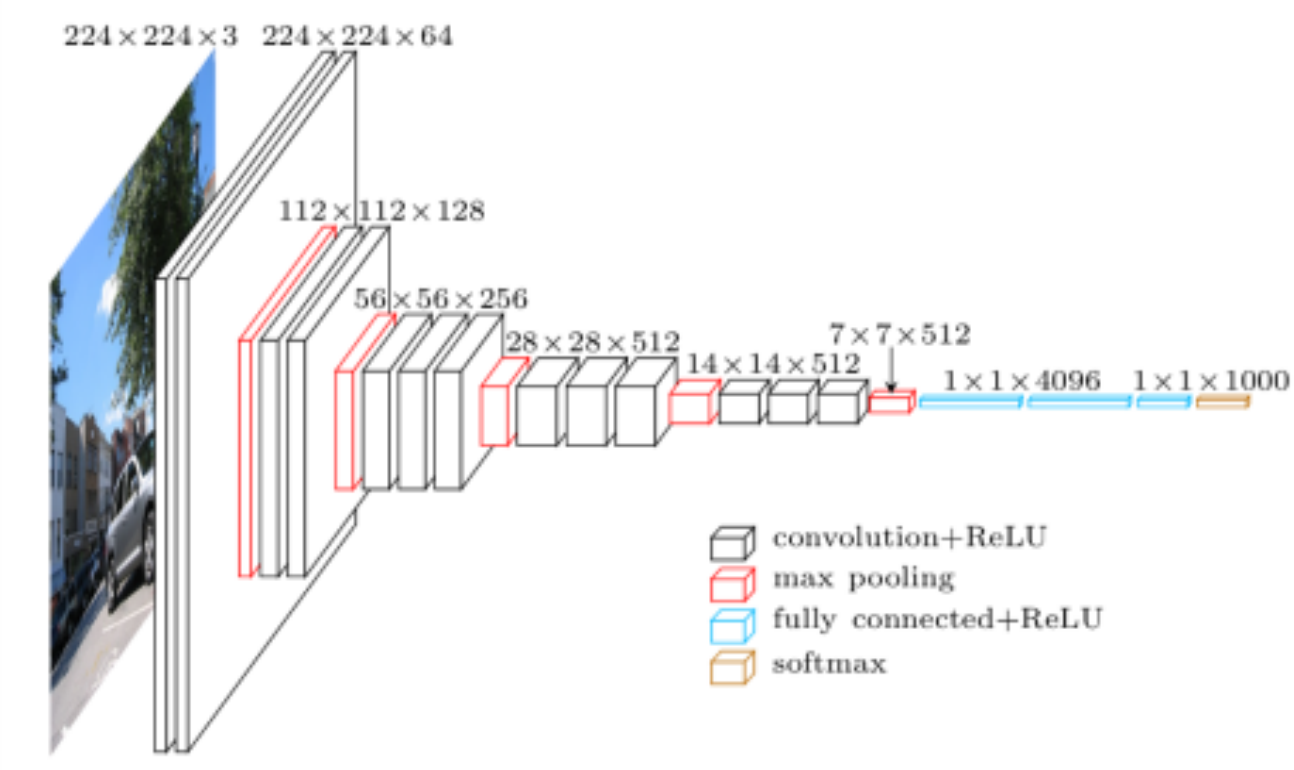
LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541-551.

AlexNet



Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "ImageNet classification with deep convolutional neural networks"

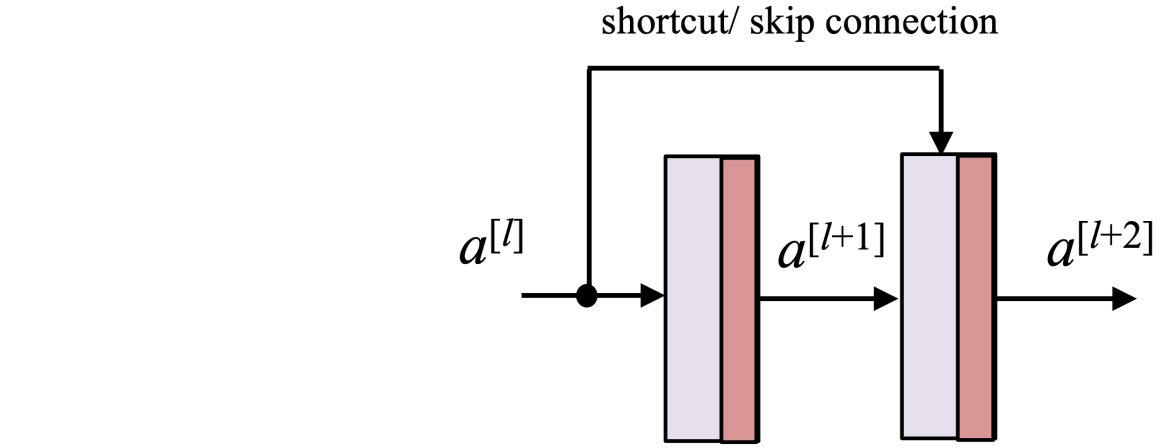
VGG-16

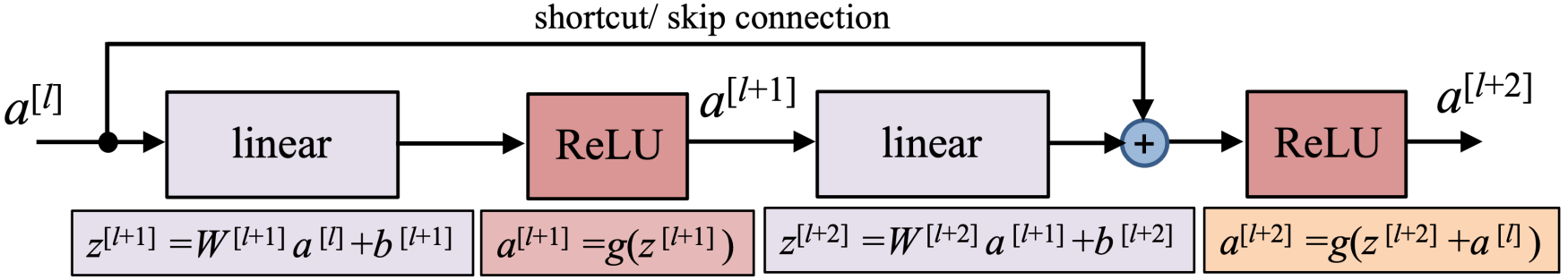


•Simonyan Sisserman. Very deep convolution networks for large-scale image recognition. 2015.

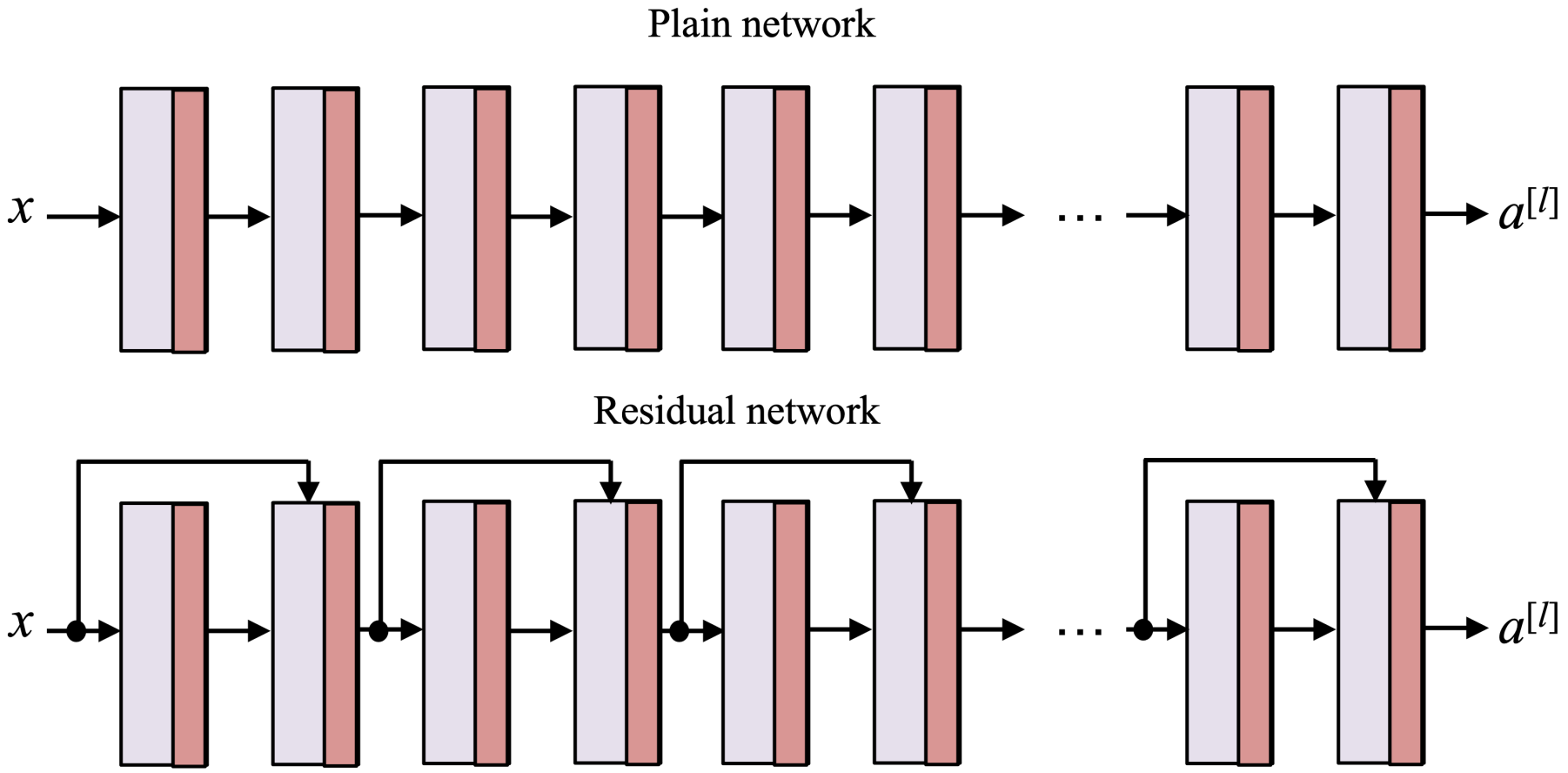
ResNets

He, K; Zhang X; Ren S.; Sun J.; Deep residual learning for image recognition. CVPR. 2015.





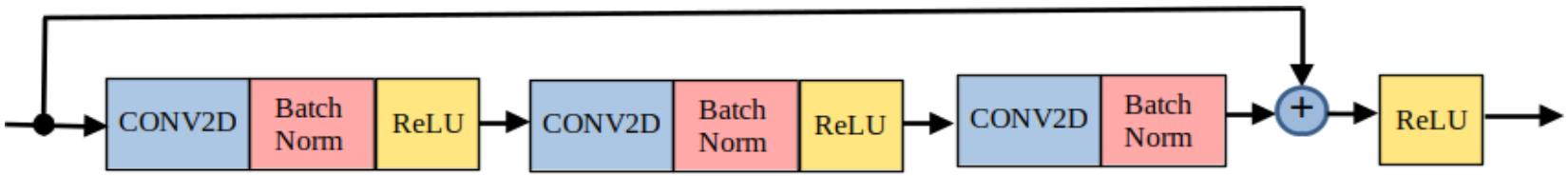
Residual block



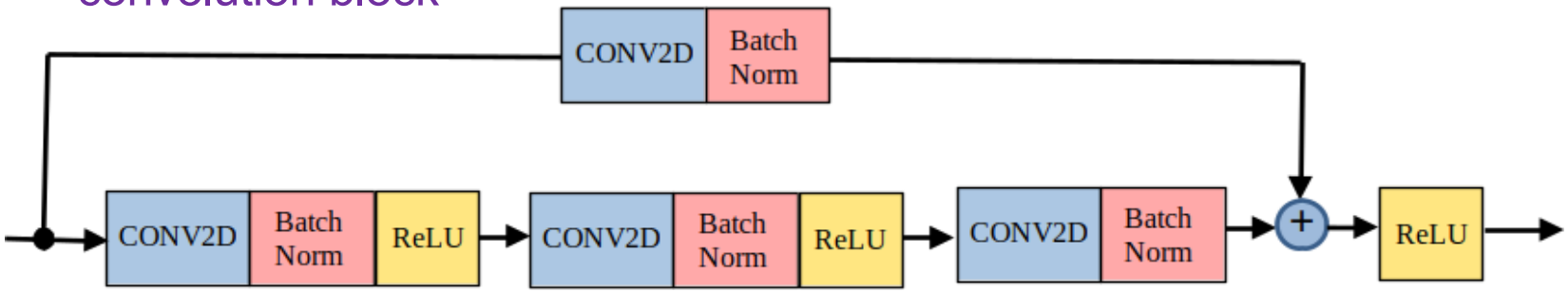
Plain v.s. residual network

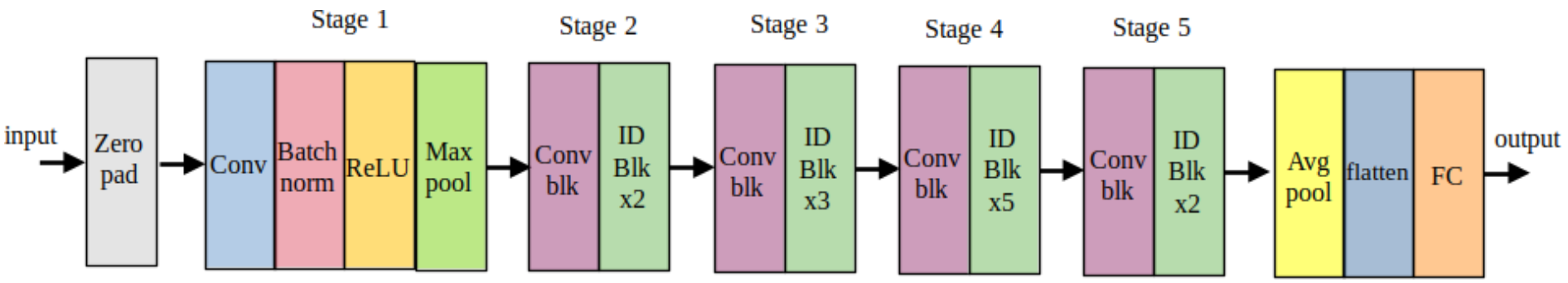
construct ResNets in TensorFlow

identity block



convolution block



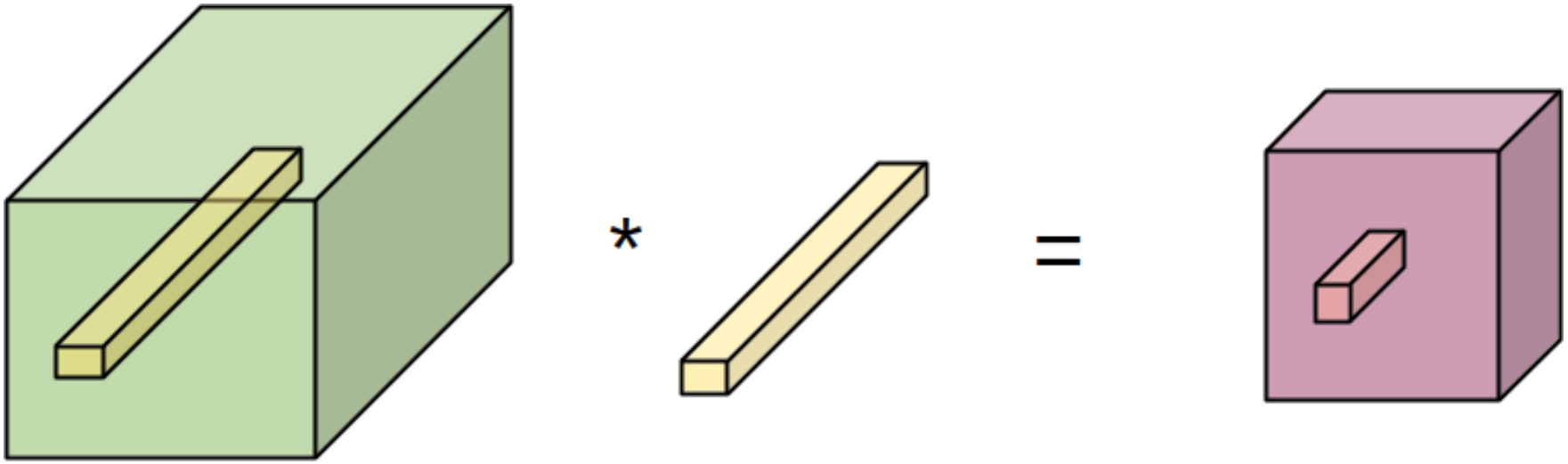


ResNet50 model

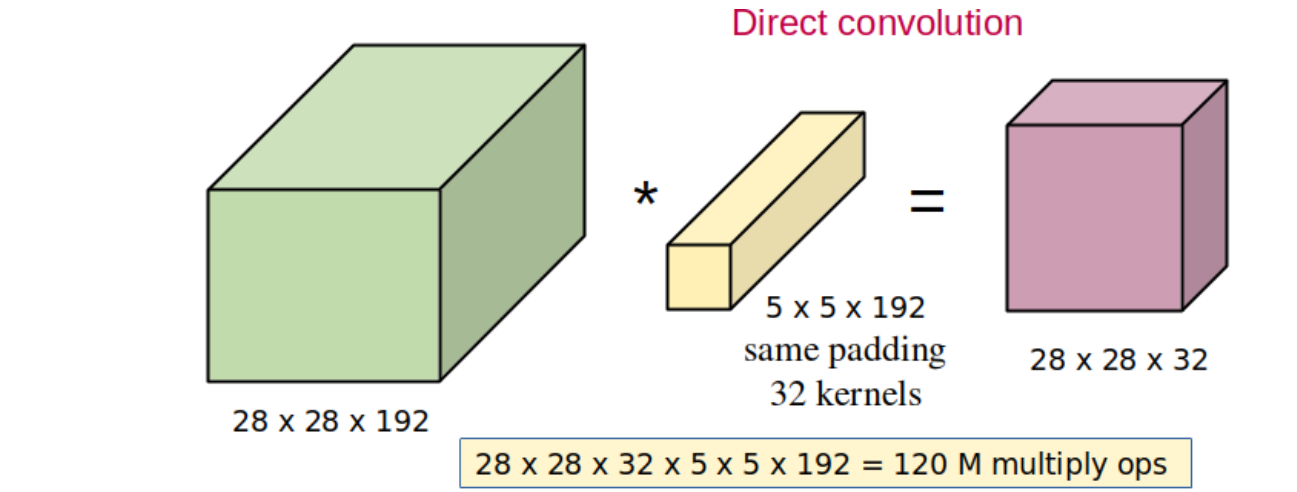
Inception model

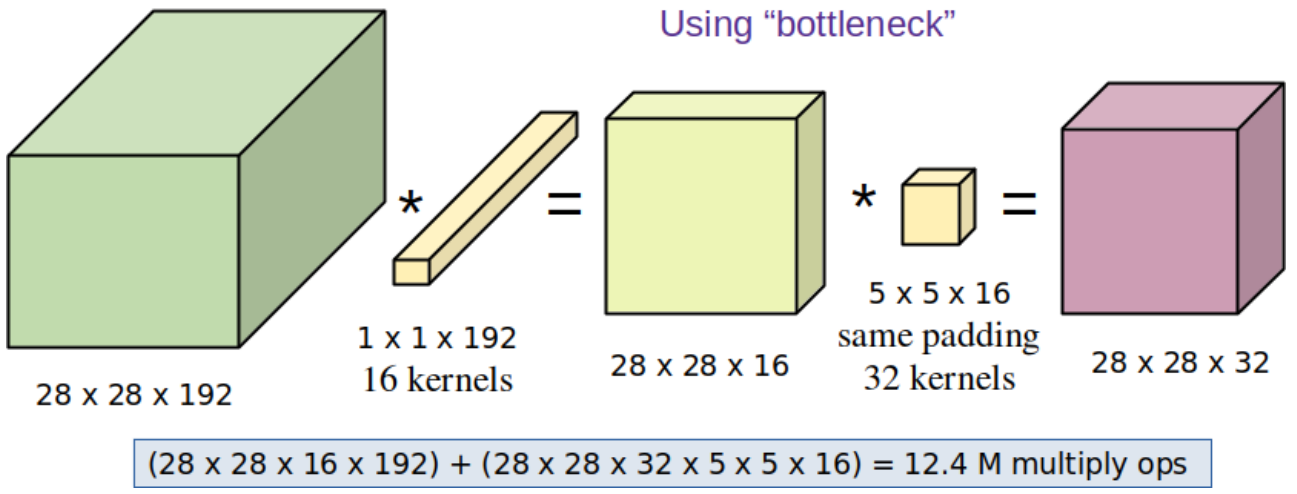
Szegedy et al. Going deeper with convolutions. 2014.

Convolution using 1x1 kernel

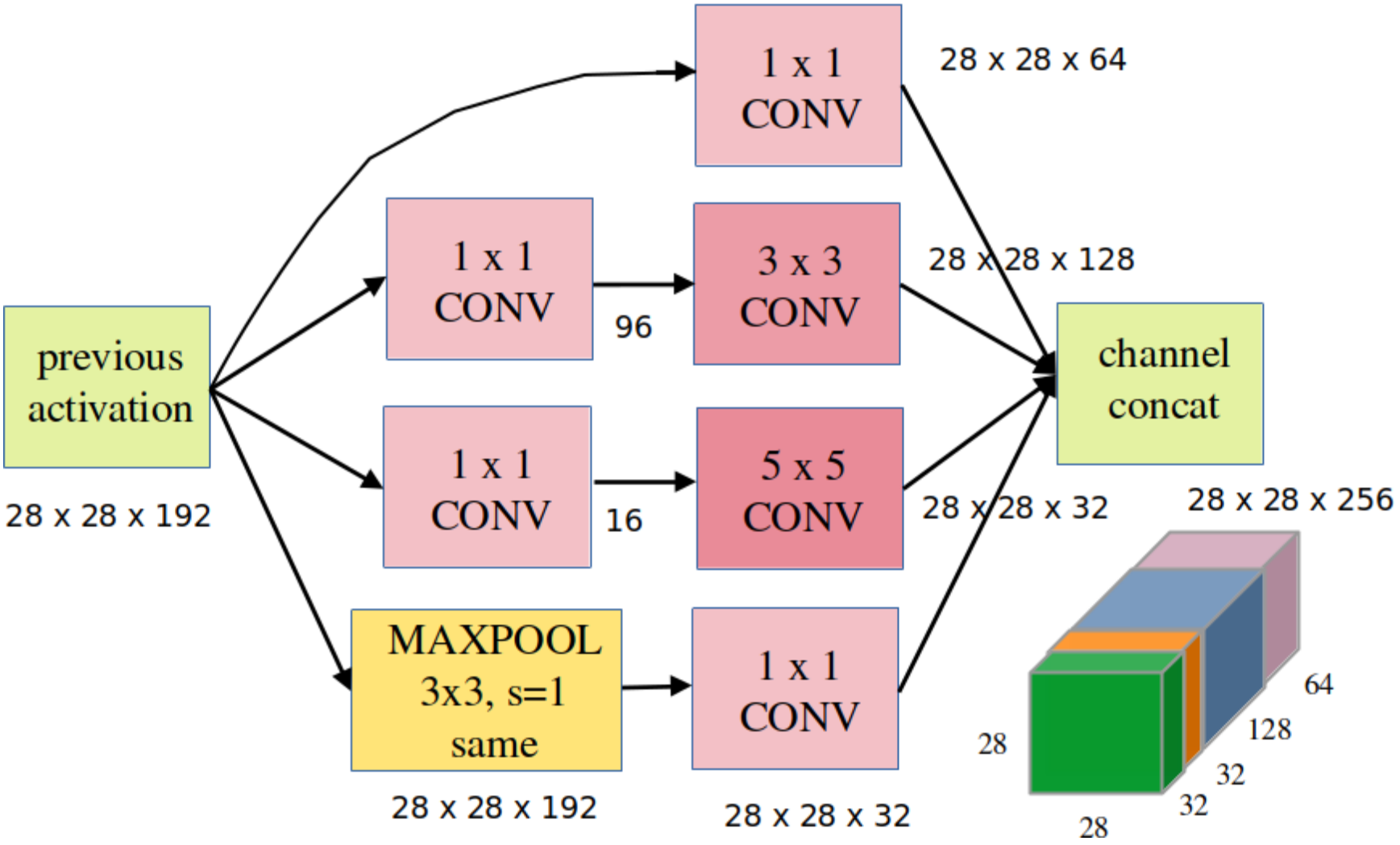


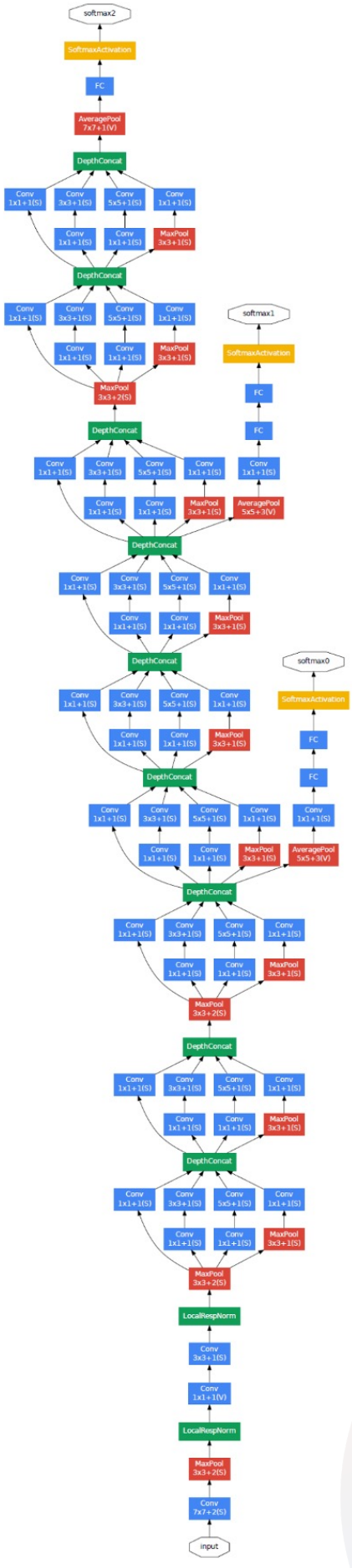
Implement “bottleneck” with 1x1 convolution



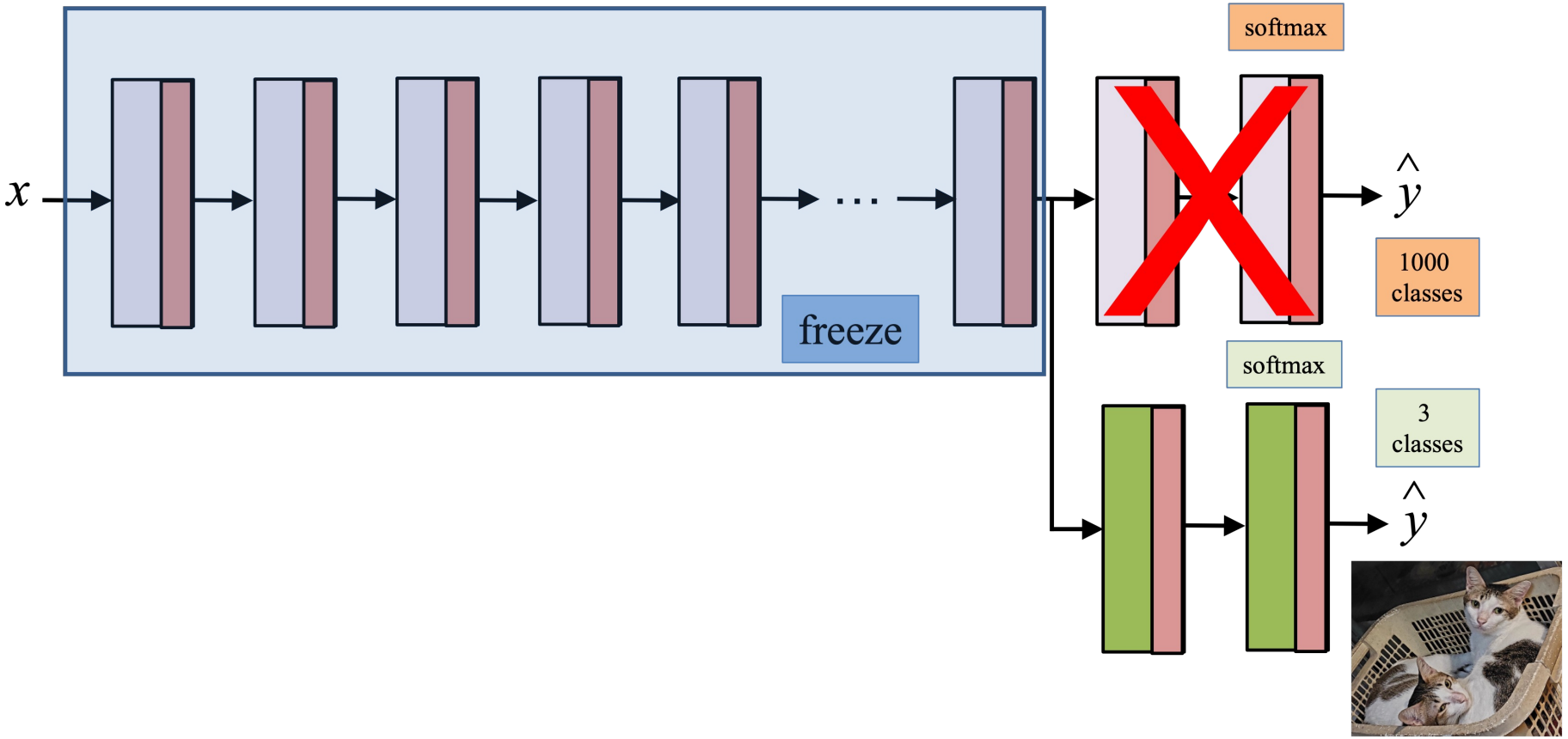


Inception module




softmax2
SoftmaxActivation
FC
AveragePool 7x7+1(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
softmax1
SoftmaxActivation
FC
FC
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
AveragePool 5x5+3(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
softmax0
SoftmaxActivation
FC
FC
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
AveragePool 5x5+3(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
LocalRespNorm
Conv 3x3+1(S)
Conv 1x1+1(V)
LocalRespNorm
MaxPool 3x3+2(S)
Conv 7x7+2(S)
input

googLenet



transfer learning

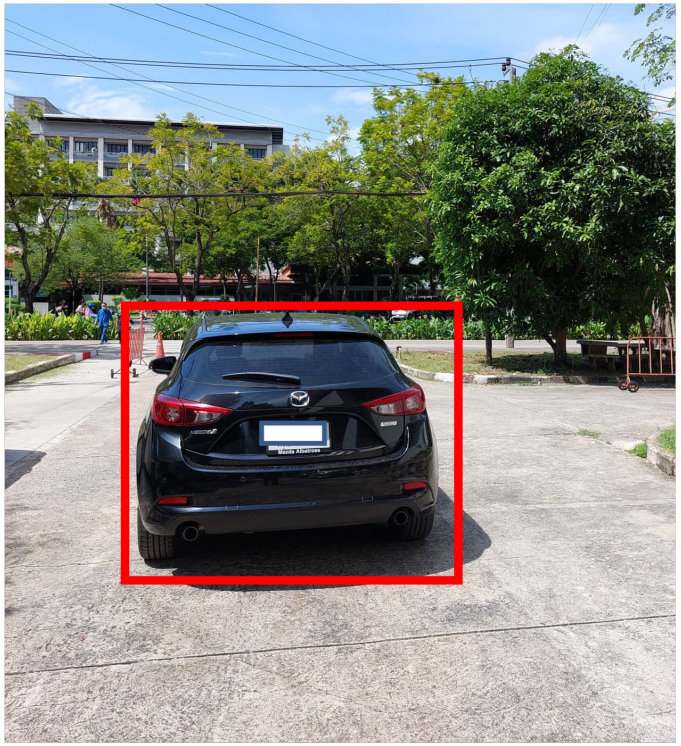
Object detection

Object localization

Classification

Object : car

Classification with localization



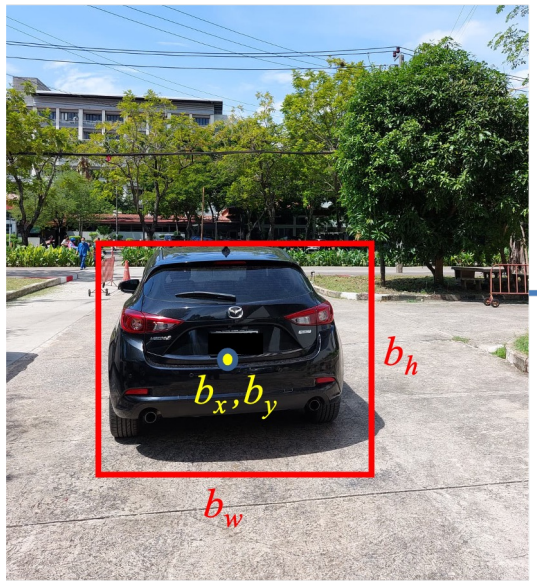
Object : car
Position : b_x , b_y , b_w , b_h

Ex : Classify and localize the following objects

- Pedestrians
- Cars
- Motorcycles
- Background (no object 1-3)

Define output vector (label)

image input



$p_c = 1, c_1 = 0, c_2 = 1, c_3 = 0$

CNN

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

object detected?

box coordiates

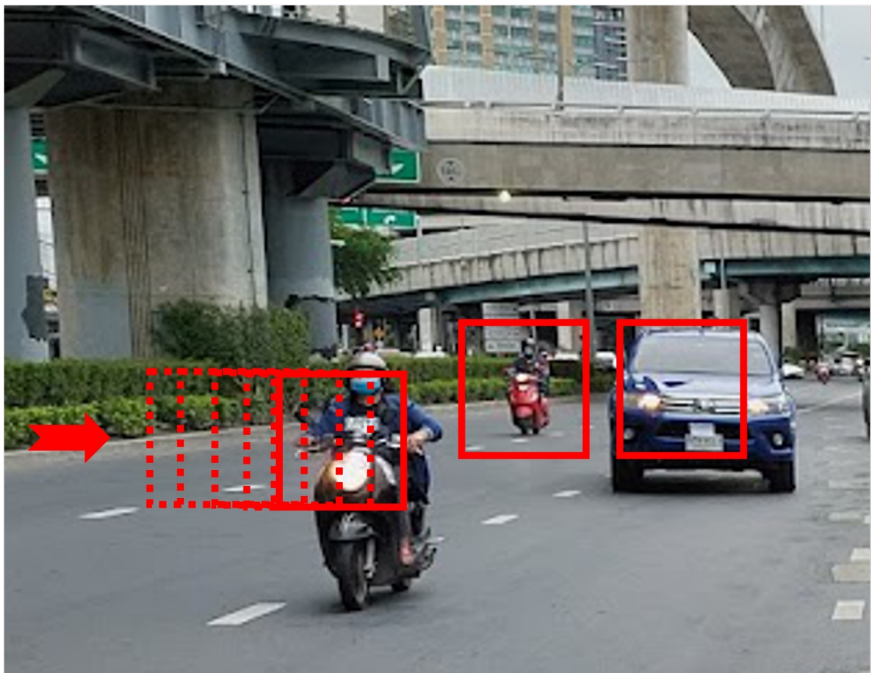
object type

loss function

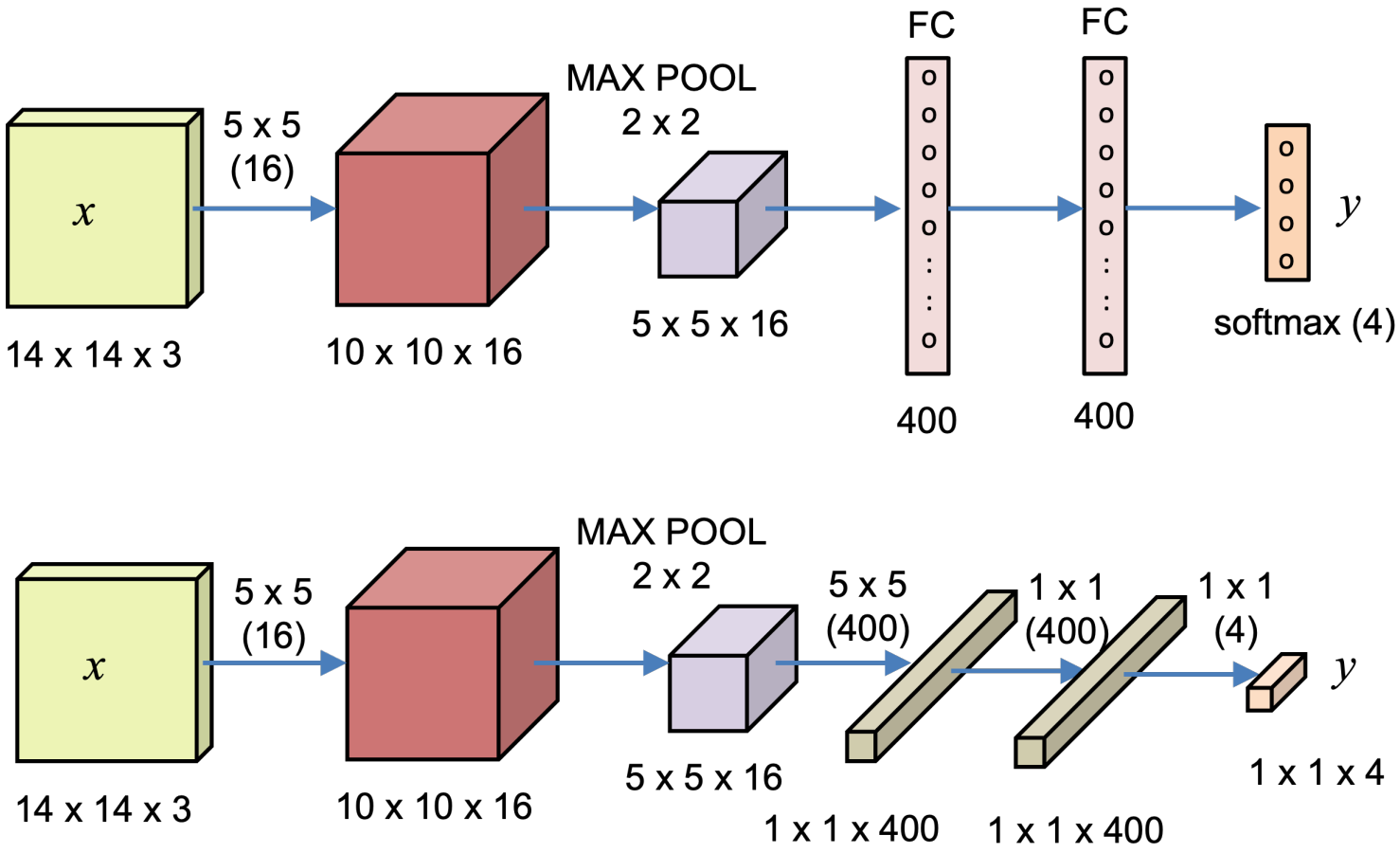
$$\mathcal{L}(\hat{y}, y) = (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \ldots + (\hat{y}_8 - y_8)^2, \quad y_1 = 1$$

$$\mathcal{L}(\hat{y}, y) = (\hat{y}_1 - y_1)^2, \quad y_1 = 0$$

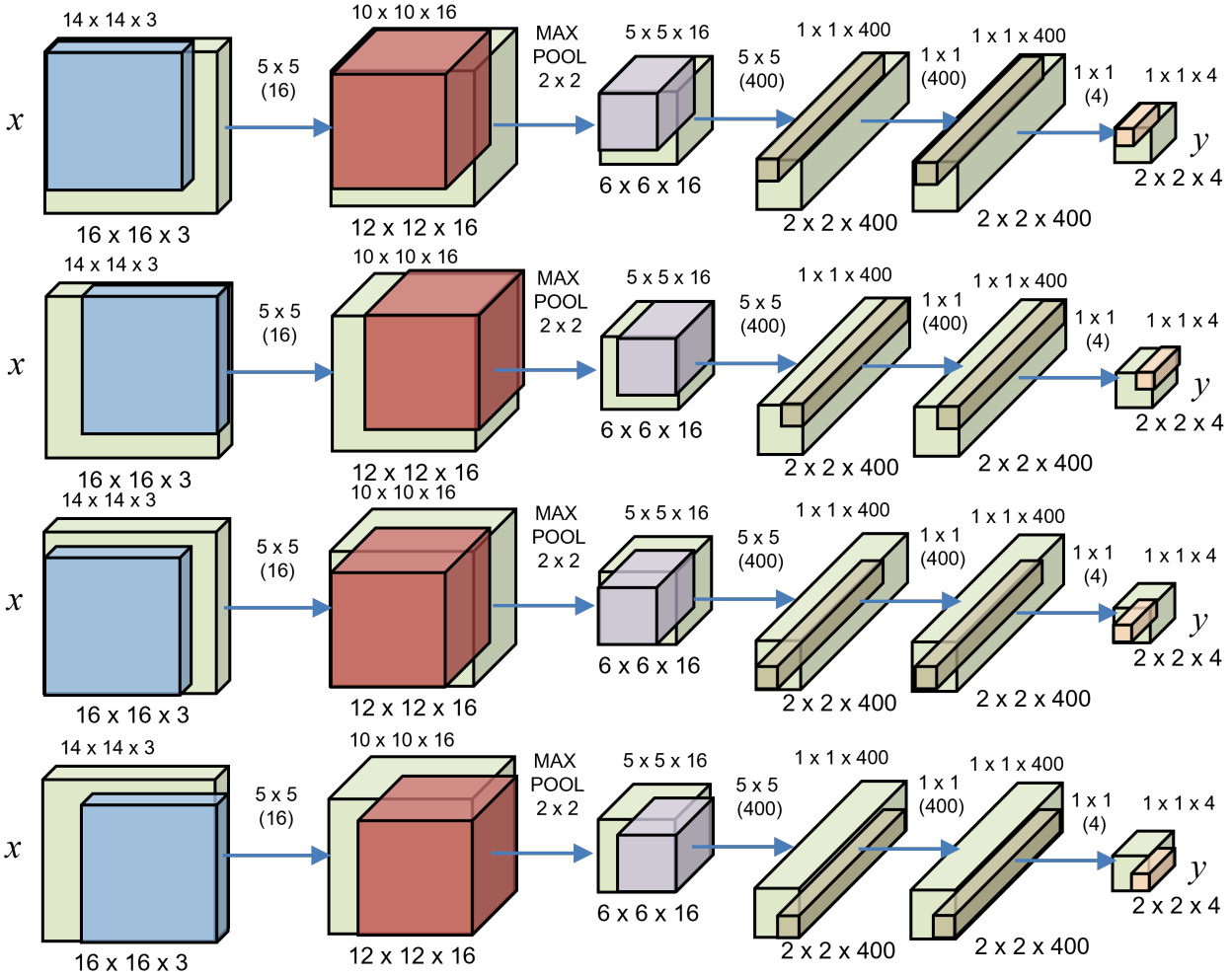
Object detection with sliding window



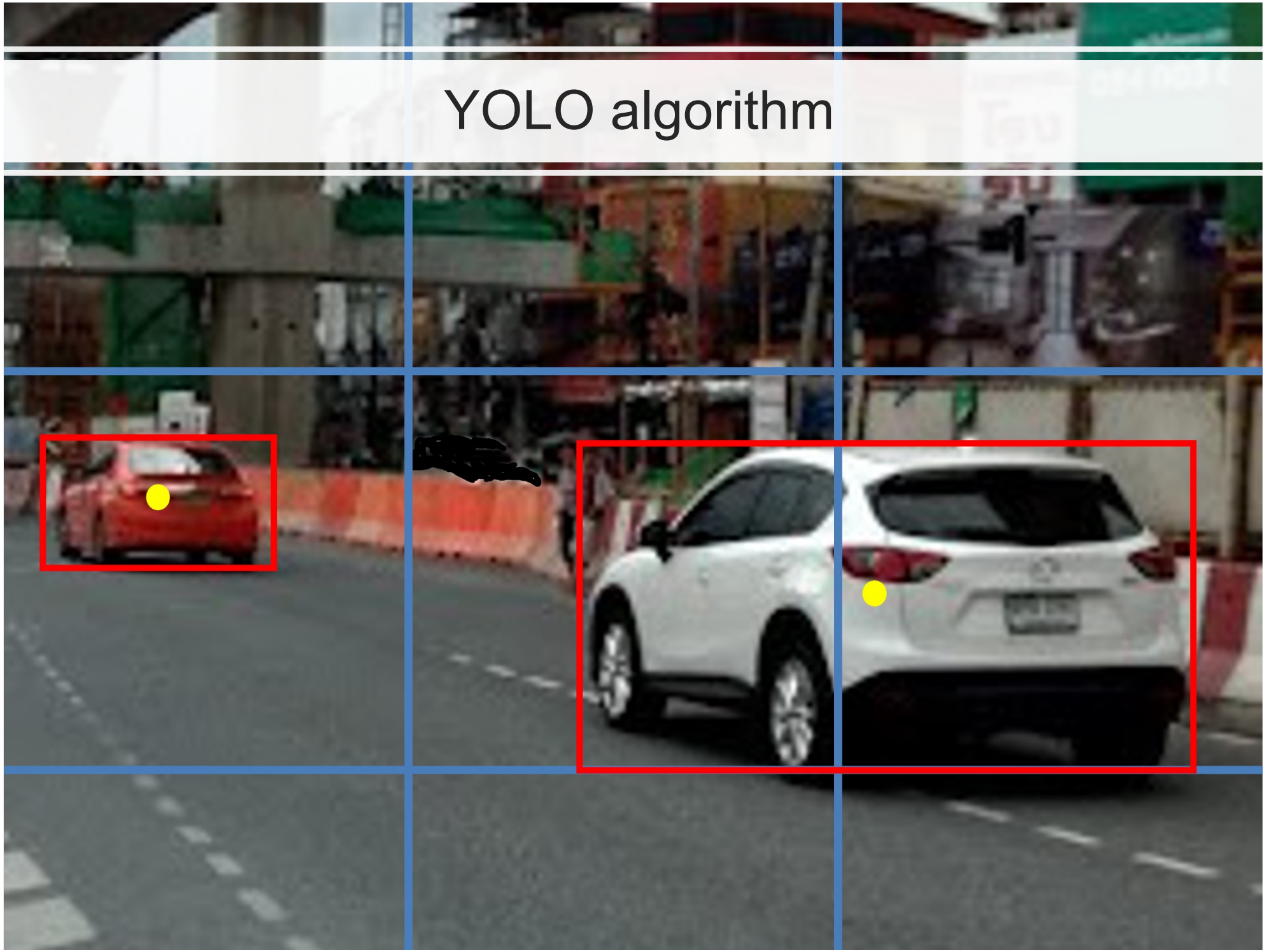
CNN with FC changed to convolutional layers



Sliding window replaced by CNN

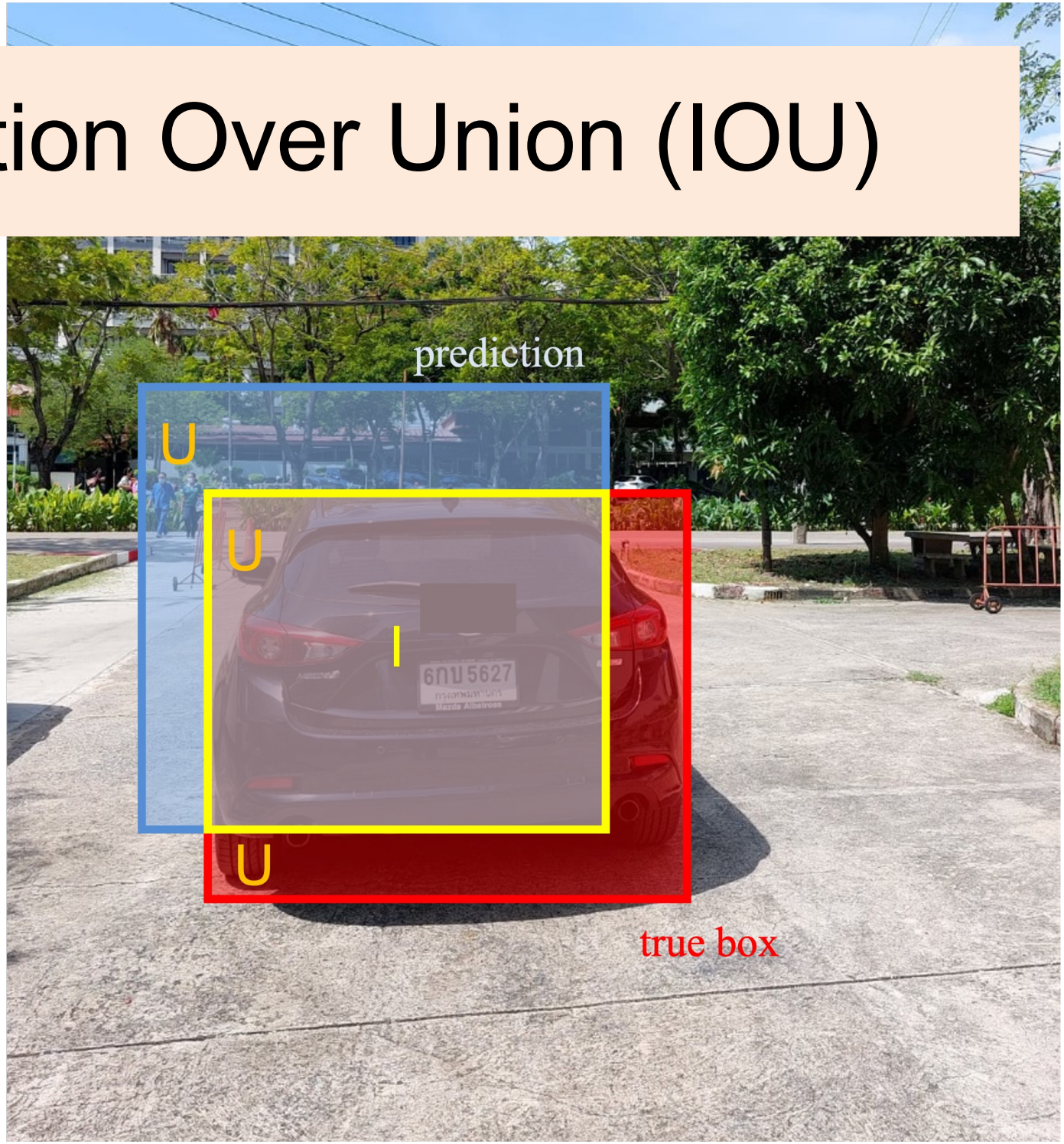


YOLO algorithm



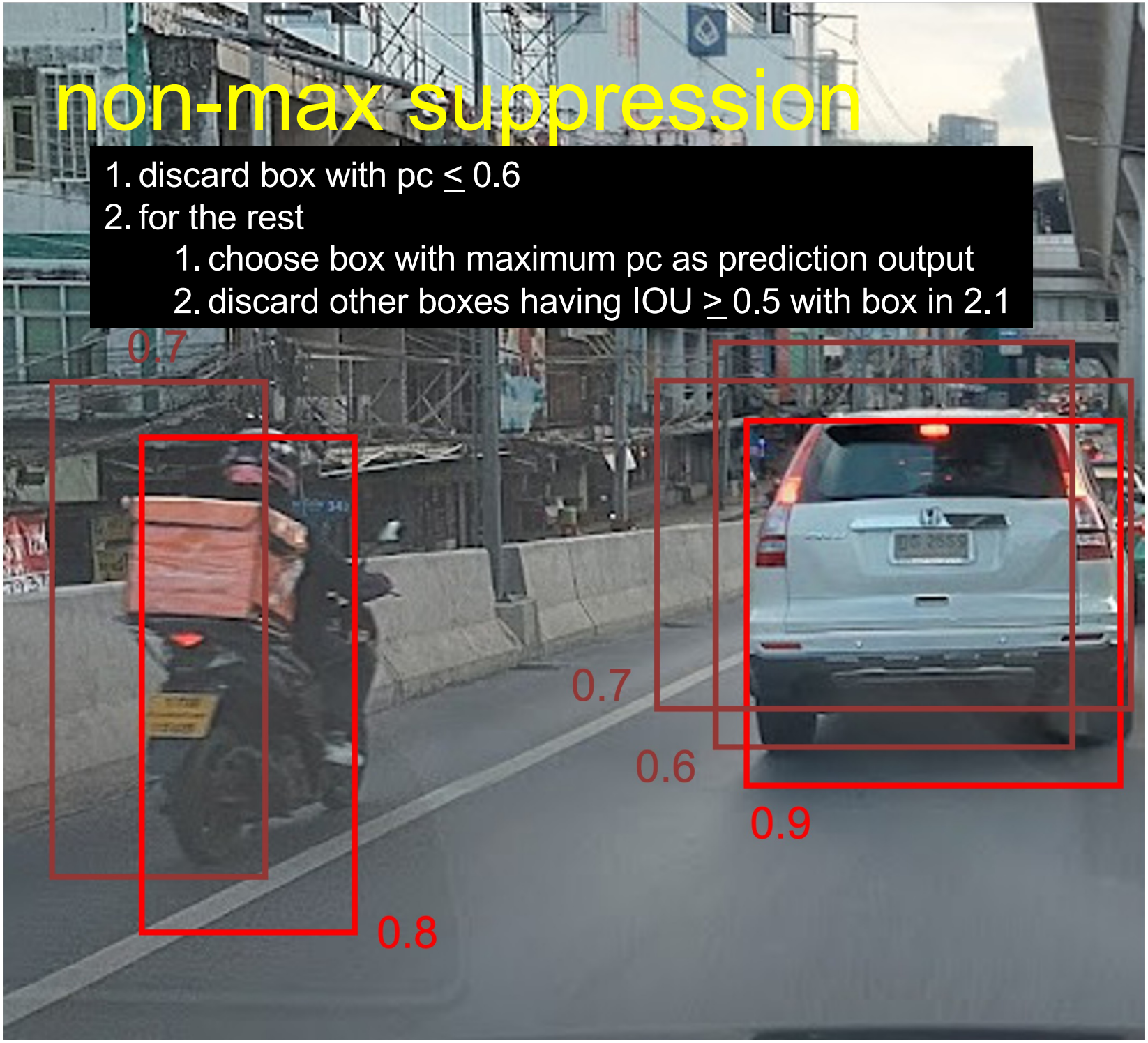
Intersection Over Union (IOU)

$$IOU = \frac{I}{U} \leq 1$$

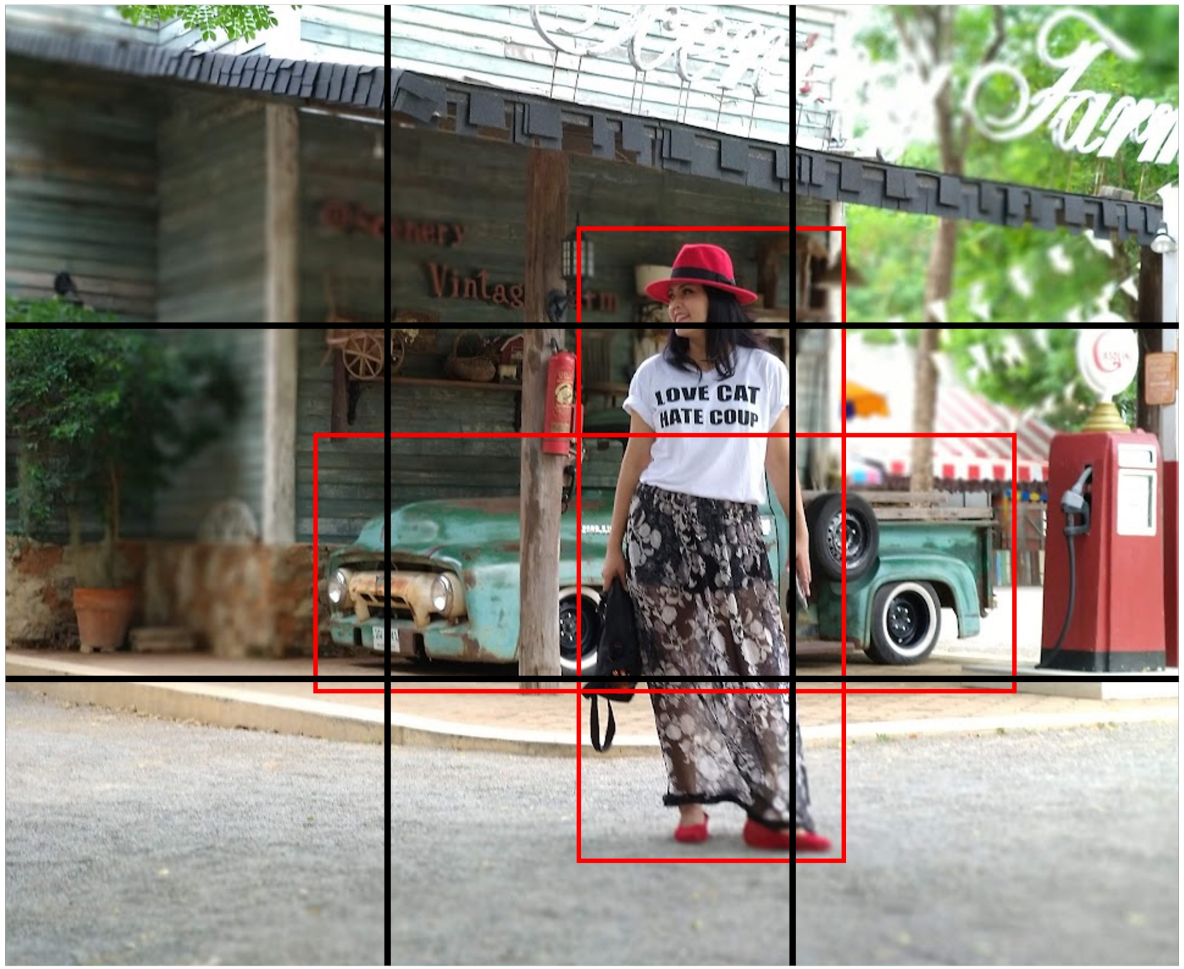


non-max suppression

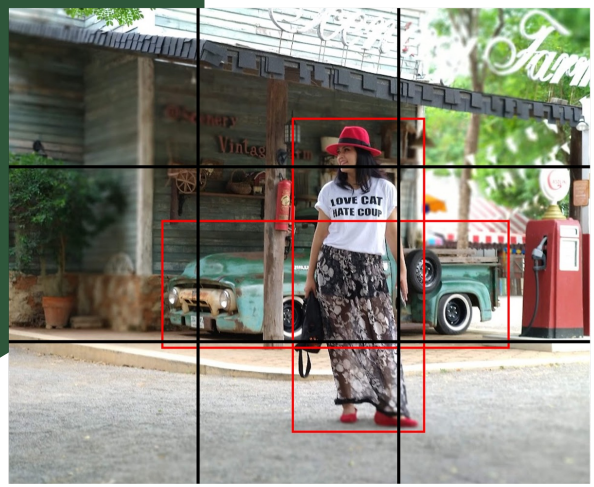
1. discard box with pc $\leq$ 0.6
2. for the rest
    1. choose box with maximum pc as prediction output
    2. discard other boxes having IOU $\geq$ 0.5 with box in 2.1

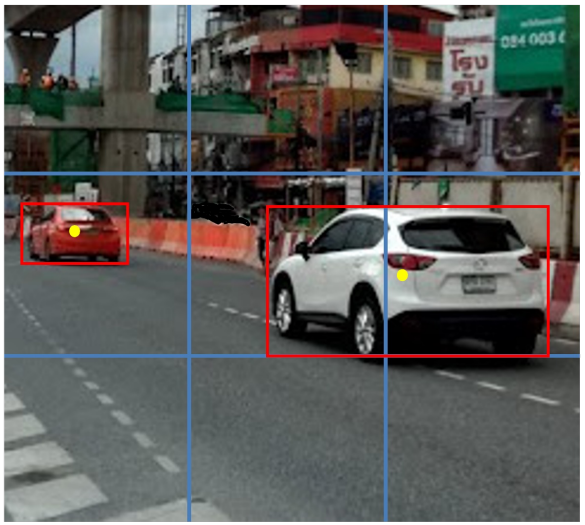


Anchor box



Anchor box output





car and human

$$y = \begin{bmatrix} p_{c1} \\ b_{x1} \\ b_{y1} \\ b_{w1} \\ b_{h1} \\ c_{11} \\ c_{12} \\ c_{13} \\ p_{c2} \\ b_{x2} \\ b_{y2} \\ b_{w2} \\ b_{h2} \\ c_{21} \\ c_{22} \\ c_{23} \end{bmatrix} = \begin{bmatrix} 1 \\ b_{x1} \\ b_{y1} \\ b_{w1} \\ b_{h1} \\ 1 \\ 0 \\ 0 \\ 1 \\ b_{x2} \\ b_{y2} \\ b_{w2} \\ b_{h2} \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

car only

$$y = \begin{bmatrix} p_{c1} \\ b_{x1} \\ b_{y1} \\ b_{w1} \\ b_{h1} \\ c_{11} \\ c_{12} \\ c_{13} \\ p_{c2} \\ b_{x2} \\ b_{y2} \\ b_{w2} \\ b_{h2} \\ c_{21} \\ c_{22} \\ c_{23} \end{bmatrix} = \begin{bmatrix} 0 \\ x \\ x \\ x \\ x \\ x \\ x \\ x \\ 1 \\ b_{x2} \\ b_{y2} \\ b_{w2} \\ b_{h2} \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

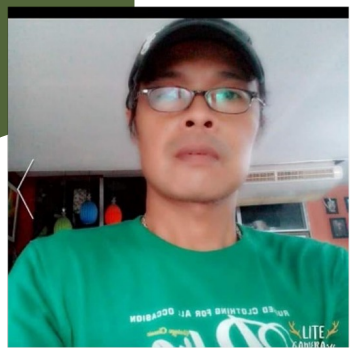
don't care

anchor box 1

anchor box 2

Face recognition with CNN

One-shot problem

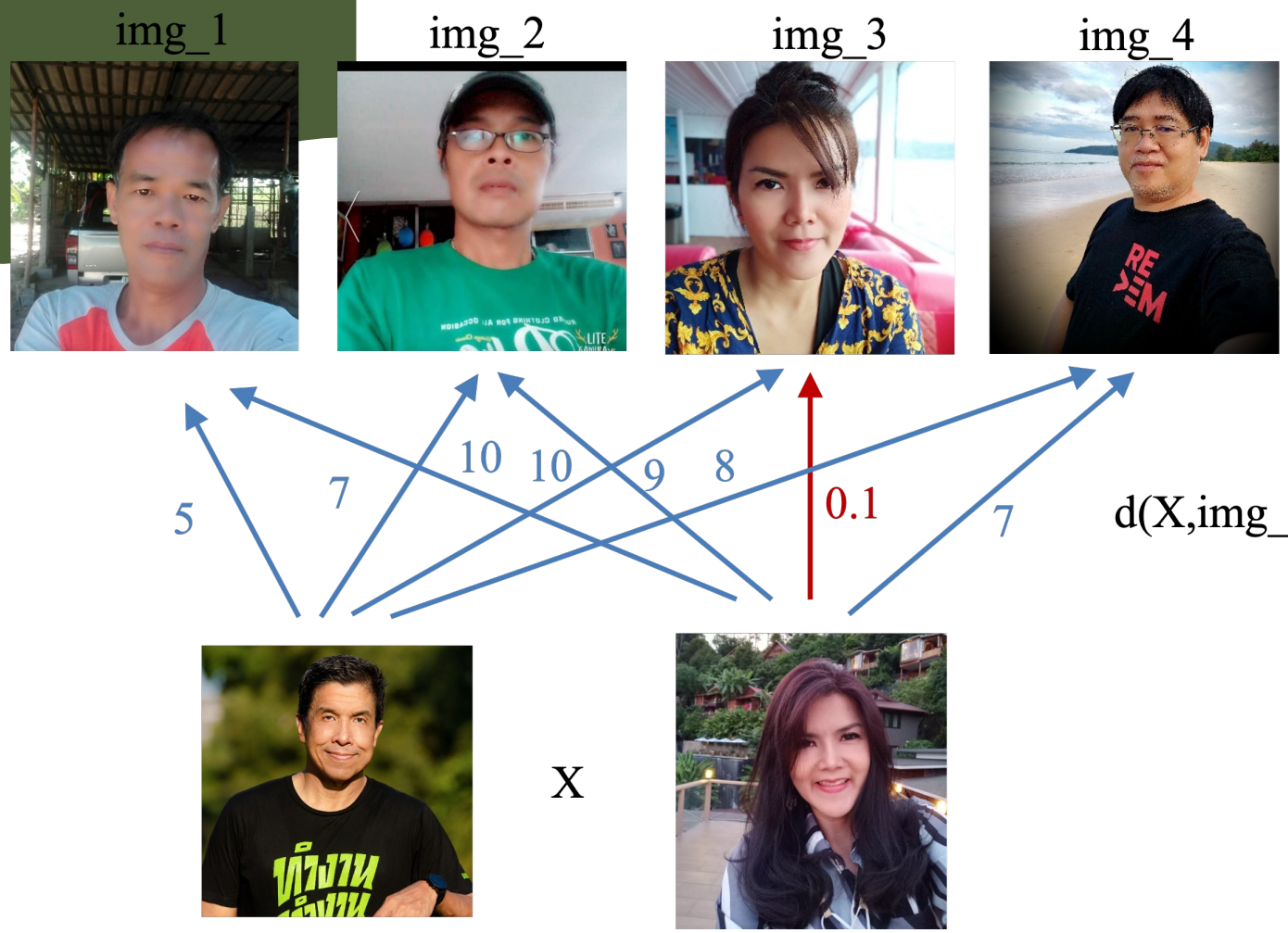
personnel database

match

?

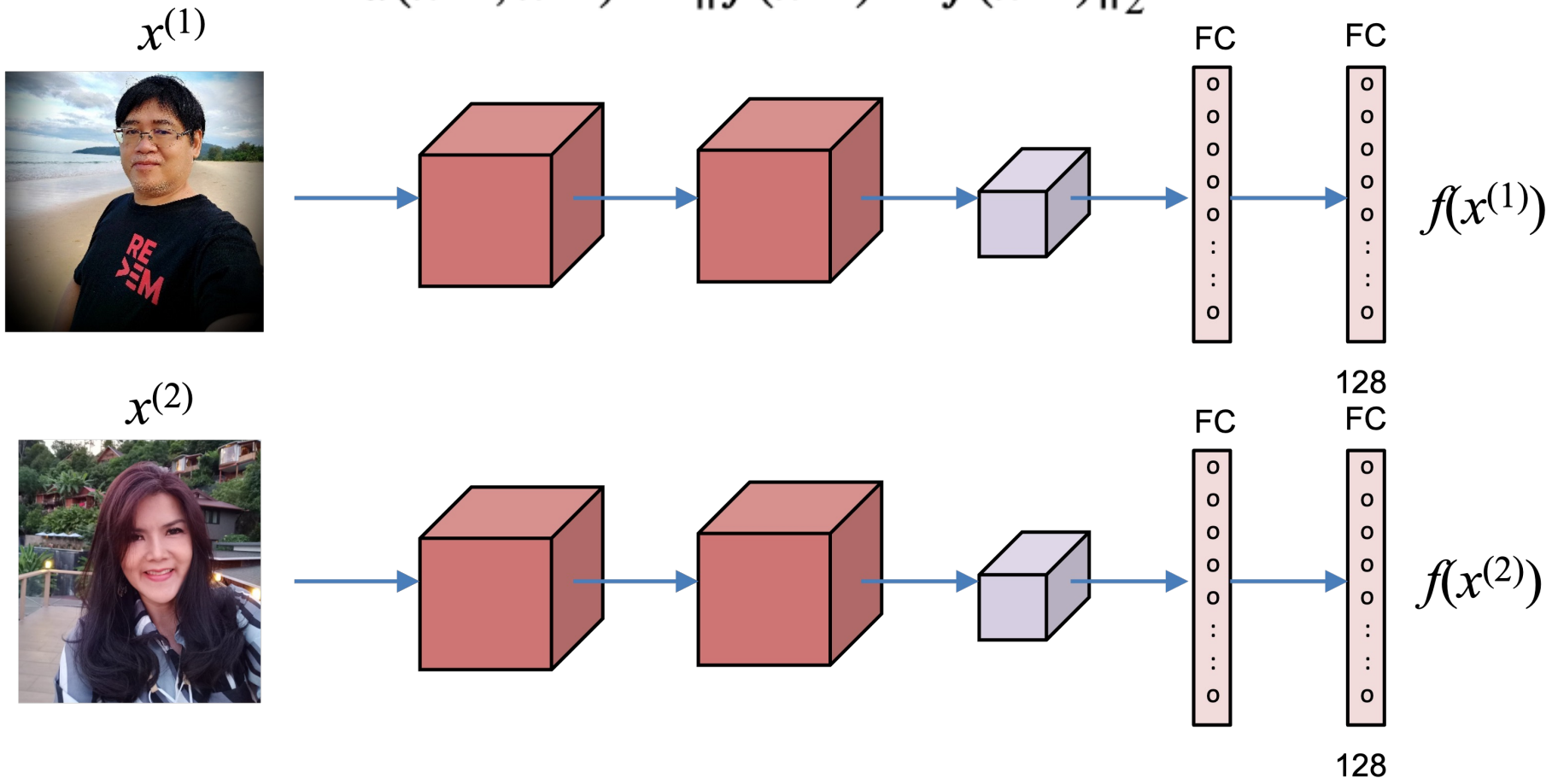

sample images

Similarity function



Siamese network

$$d(x^{(1)}, x^{(2)}) = \|f(x^{(1)}) - f(x^{(2)})\|_2^2$$

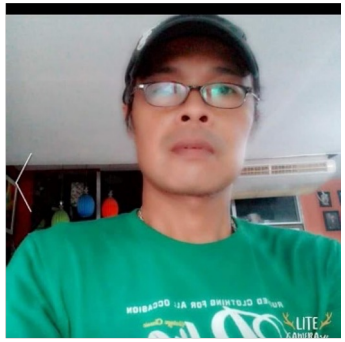


Triplet loss

Anchor
(A)

Positive
(P)

Anchor
(A)



Negative
(N)

$$d(A, P) = \|f(A) - f(P)\|_2^2 \leq d(A, N) = \|f(A) - f(N)\|_2^2$$

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 \leq 0$$

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha \leq 0$$

avoid trivial solution

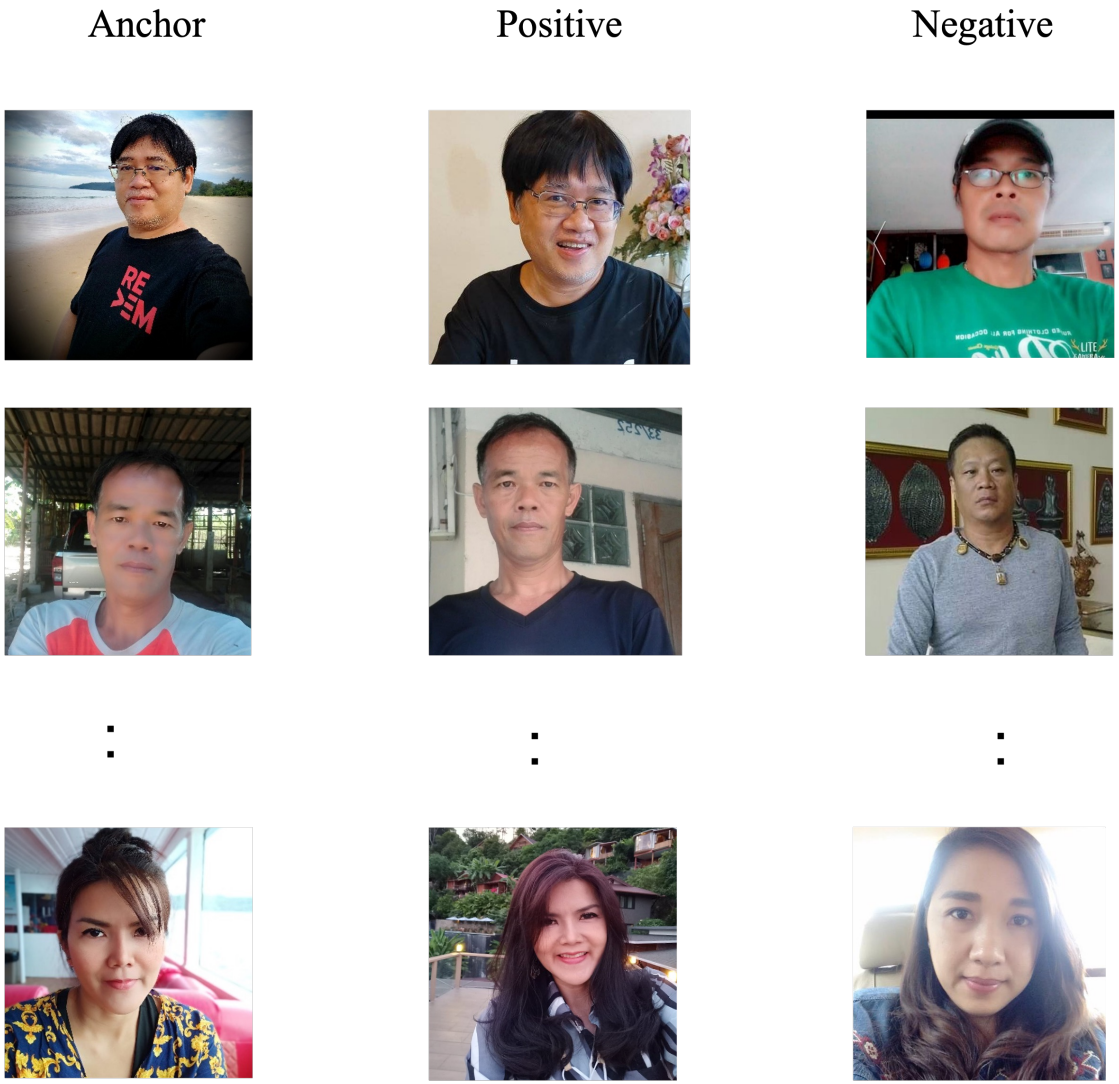
Total cost function

$$J = \sum_{i=1}^{m} \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

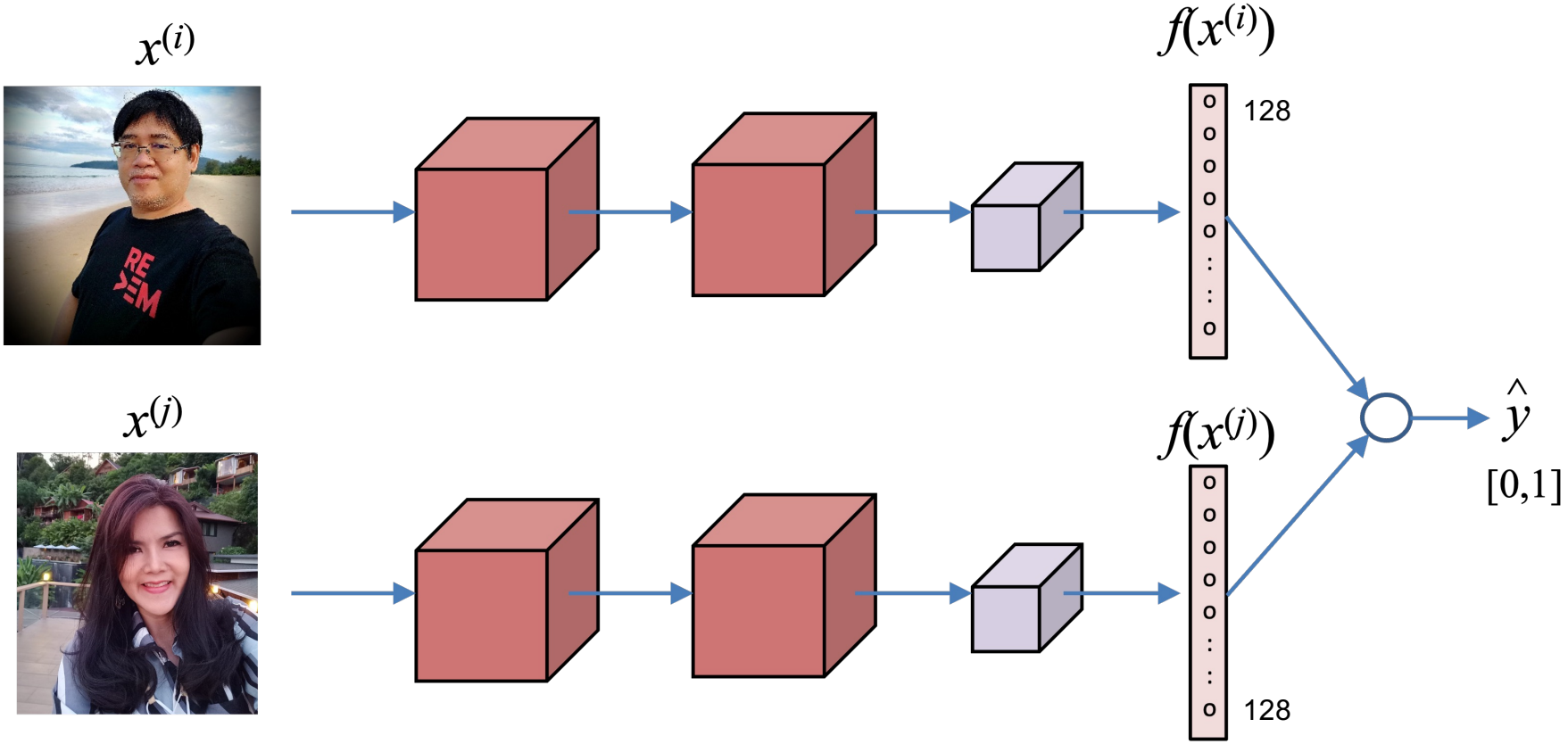
Triplet loss function

$$\mathcal{L}(A, P, N) = max(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha, 0)$$

Train model with triplets



similarity function learning



$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_i |f(x^{(i)})_k - f(x^{(j)})_k| + b_i\right)$$