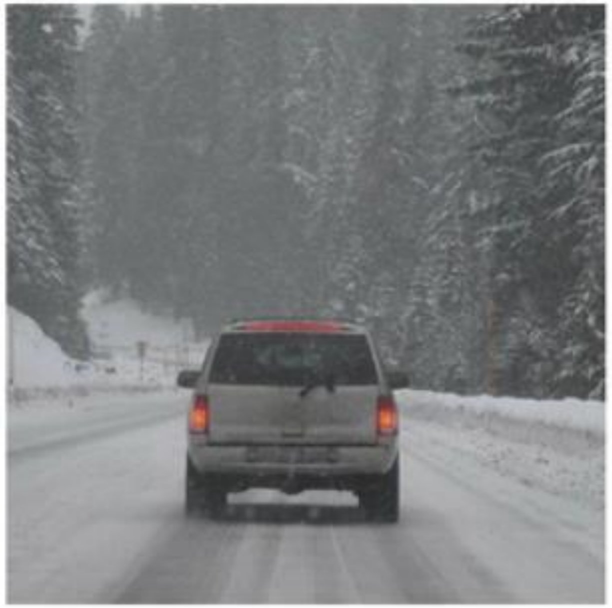
Applications of Convolutional Neural Networks

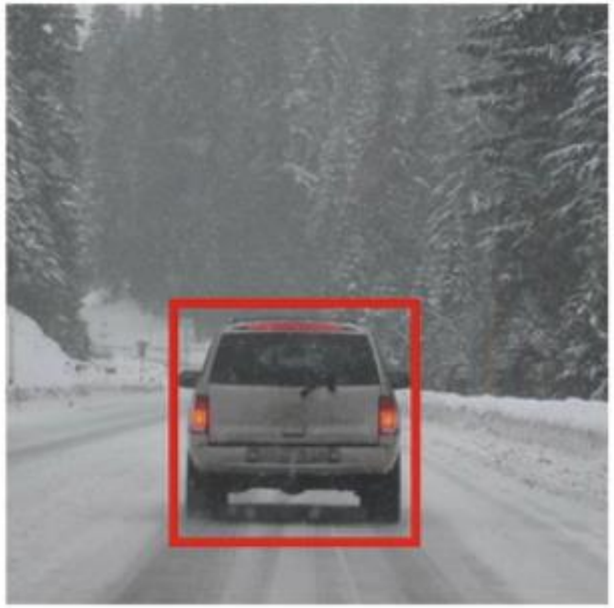
Object Localization and detection

Image classification



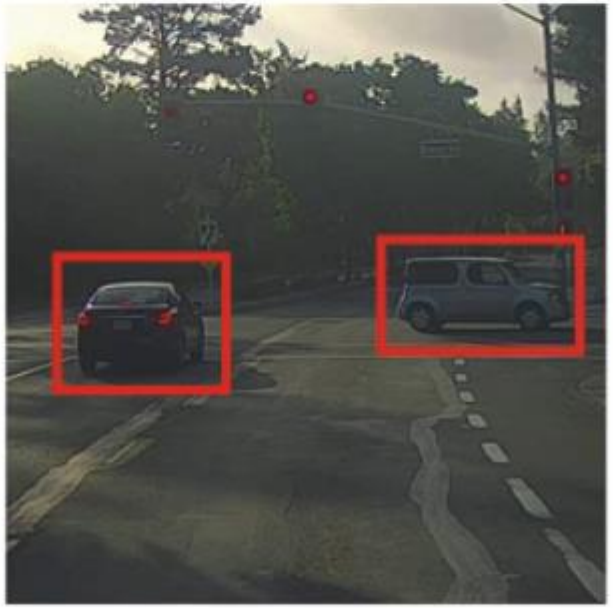
Car or Not

Classification with localization



Usually Single Object

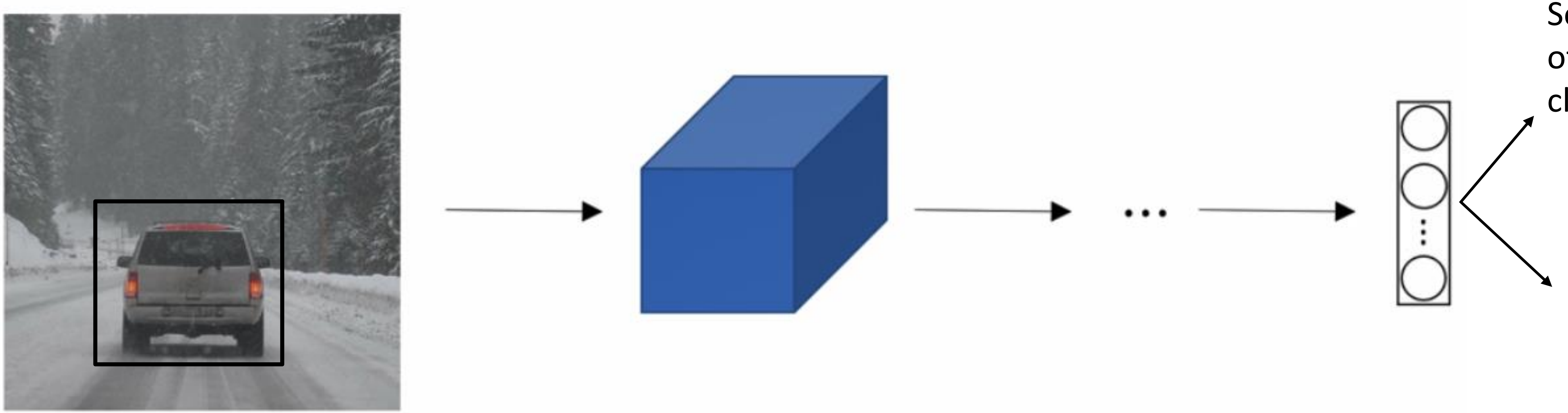
Detection



Multiple Objects

Classification with localization

Pedestrian
Car
Motorcycle
background



Softmax of four classifiers

bx,by, bh,bw

Defining the target label

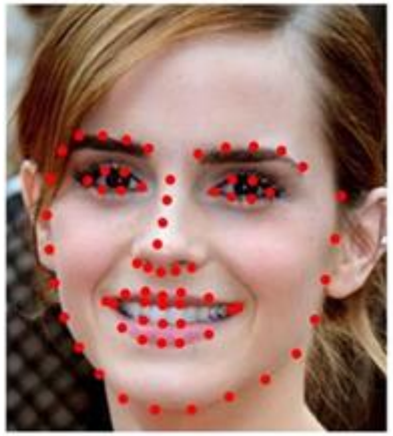
Presence of an object(Y/N)	1	0
b_x	b_x	-
b_y	b_y	-
b_h	b_h	-
b_w	b_w	-
c_1	0	-
c_2	1	-
c_3	0	-

$$L(\hat{y},y)=(\hat{y}_1-y_1)^2+(\hat{y}_2-y_2)^2+(\hat{y}_3-y_3)^2+(\hat{y}_4-y_4)^2+(\hat{y}_5-y_5)^2+(\hat{y}_6-y_6)^2+(\hat{y}_7-y_7)^2+(\hat{y}_8-y_8)^2$$

else

$$(\hat{y}_1-y_1)^2$$

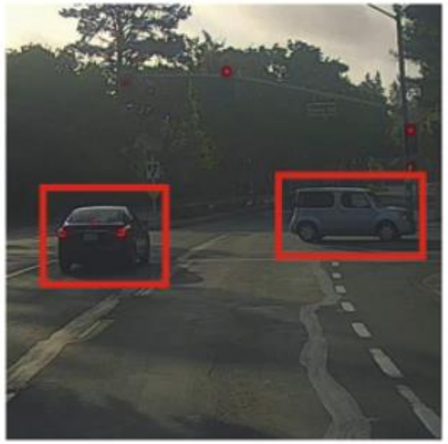
Landmark Detection

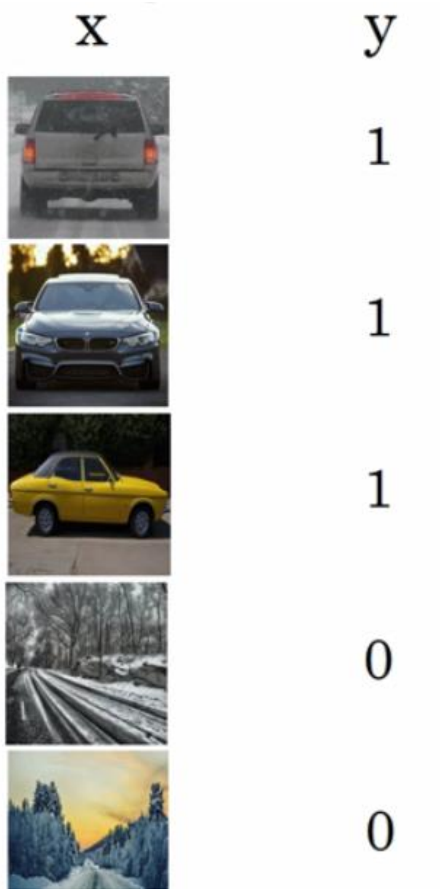


Order/Sequence of the landmarks is predefined

Face? (Y/N)
$L1_{X}, L1_{Y}$
$L2_{X}, L2_{Y}$ $L3_{X}, L3_{Y}$:
:
$L64_{X}, L64_{Y}$

Car Detection

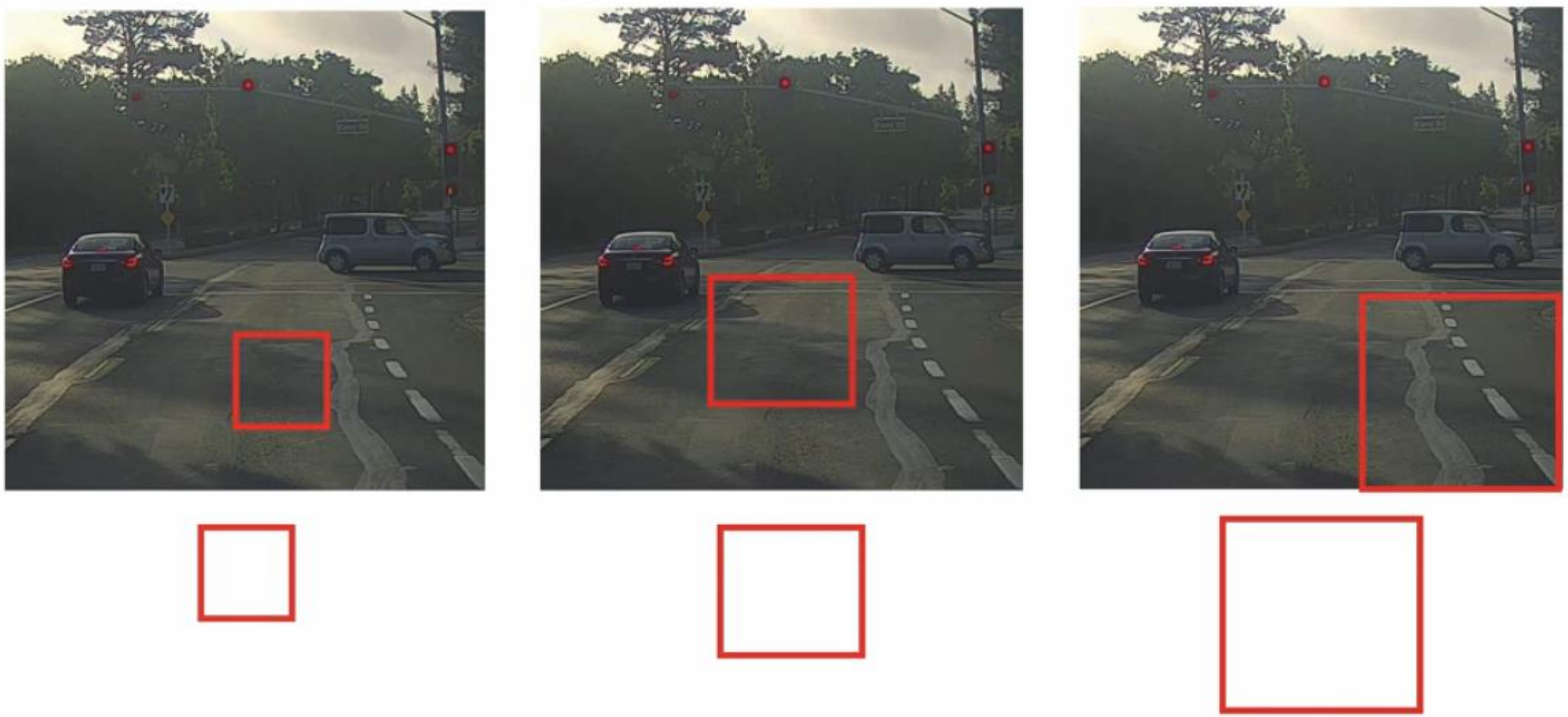




Conv Net → Y

Having Closely Cropped Images to train the System

Sliding Windows Detection



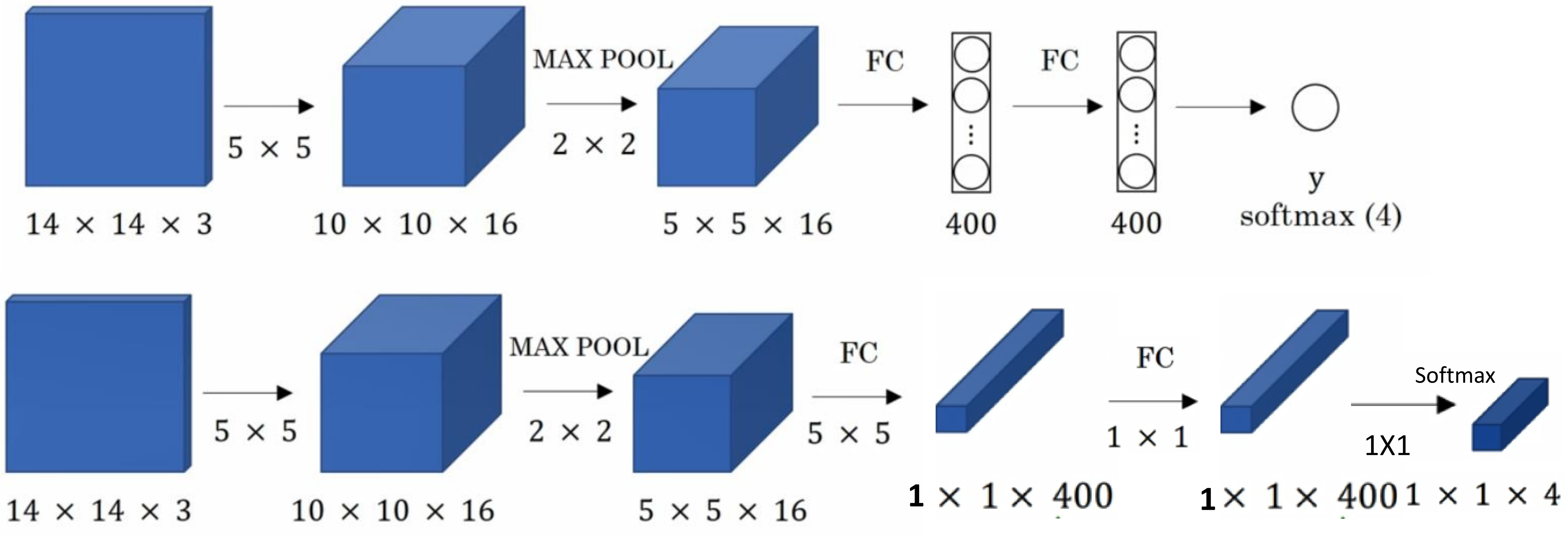
Sliding Window Detection

Each Iteration for each of the Sliding Window Box needs a convnet to classify and localize the object.

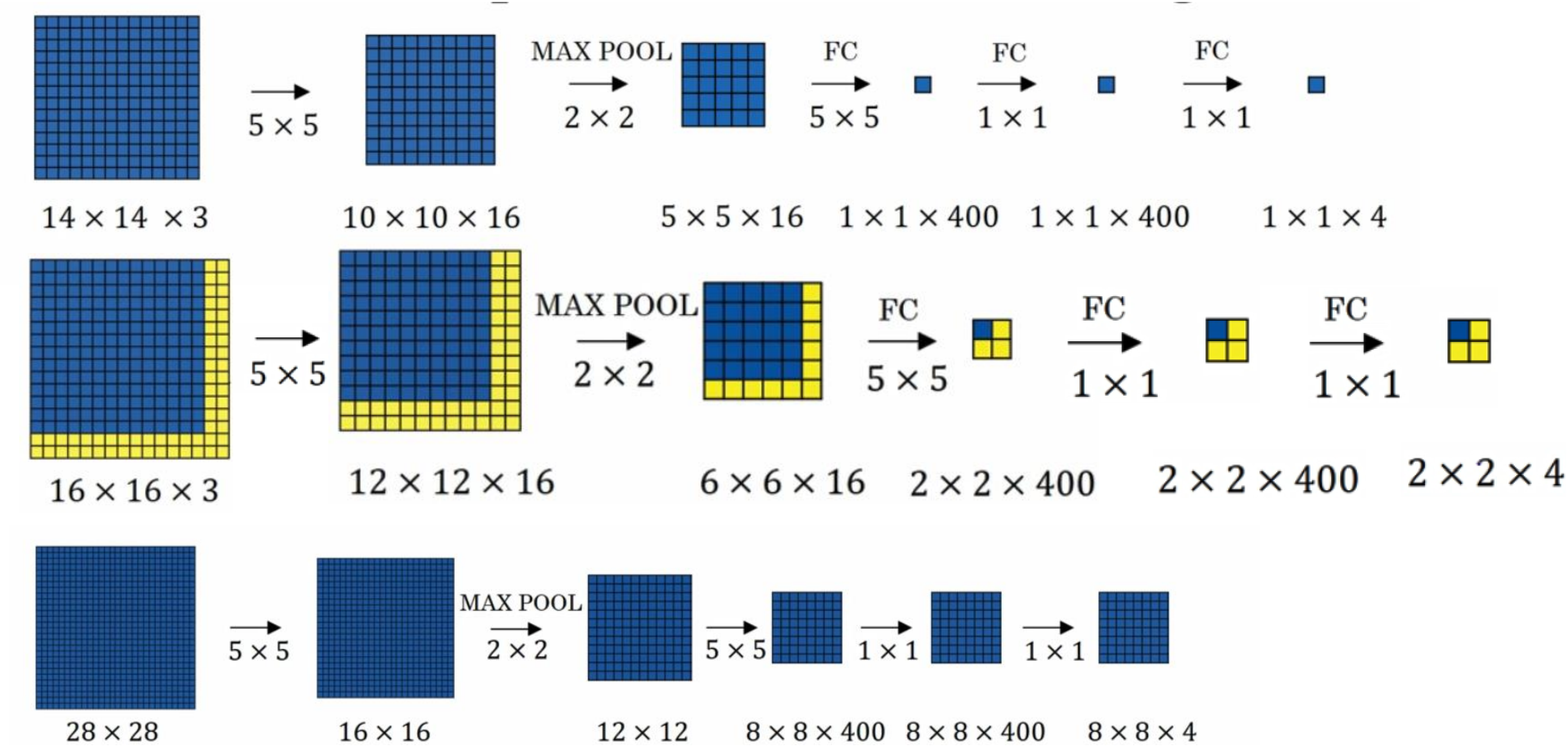
Computation Cost is Huge.

We have a solution

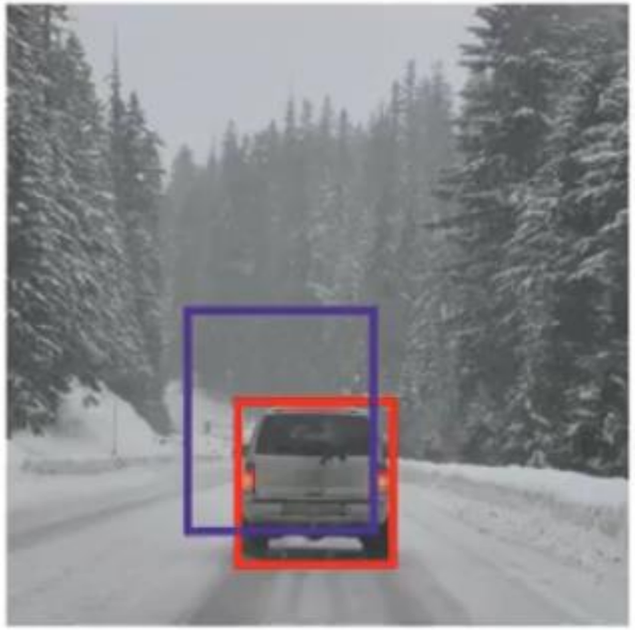
Turning Fully Connected Layer to Convolution Layers

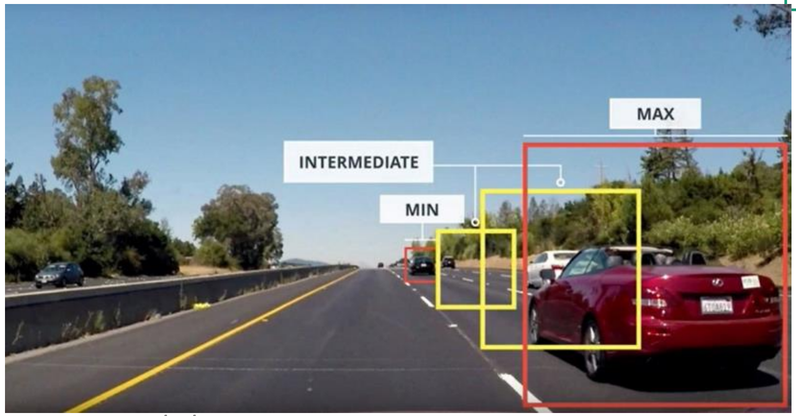


Convolutional Implementation of Sliding Windows



Intersection Over Union



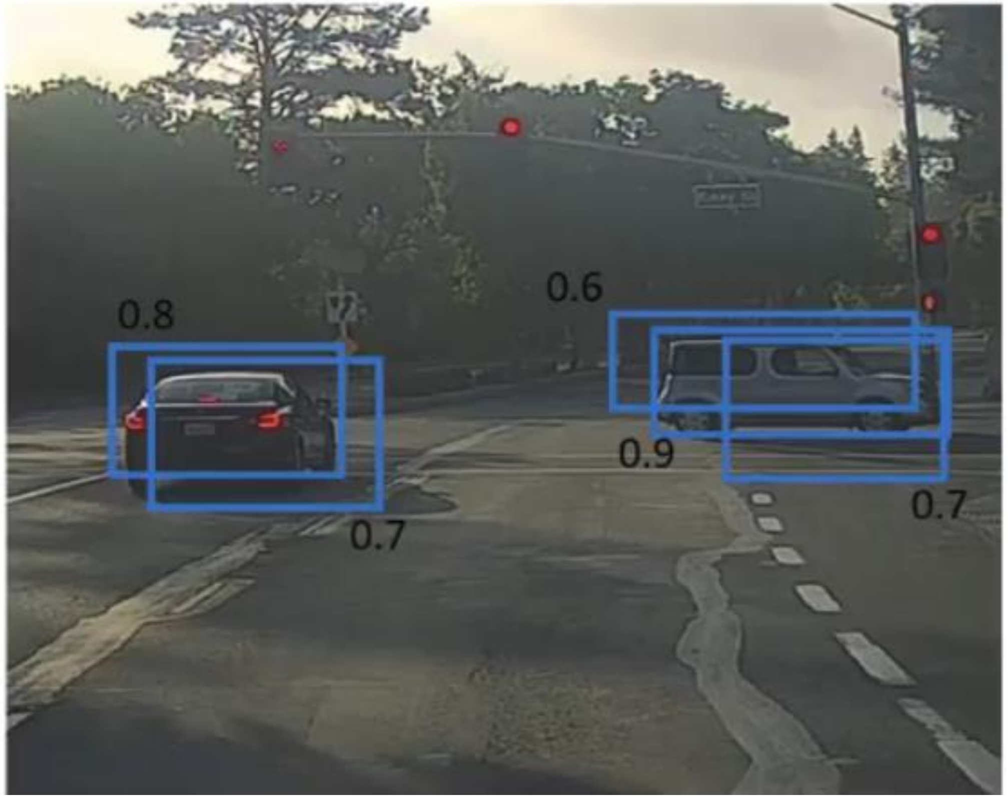


Intersection Over Union

Size of Intersect Area/Size of Union Area
May be Correct if IoU $\geq$ 0.5

So, it is a measure of the overlap between two bounding boxes

Non-Max Suppression



It Cleans up multiple bounding boxes around one object and gives us one bounding box per object.

Each Output (Probabilities associated with multiple bounding boxes is Pc)

Discard all boxes with Pc $\leq$ 0.6

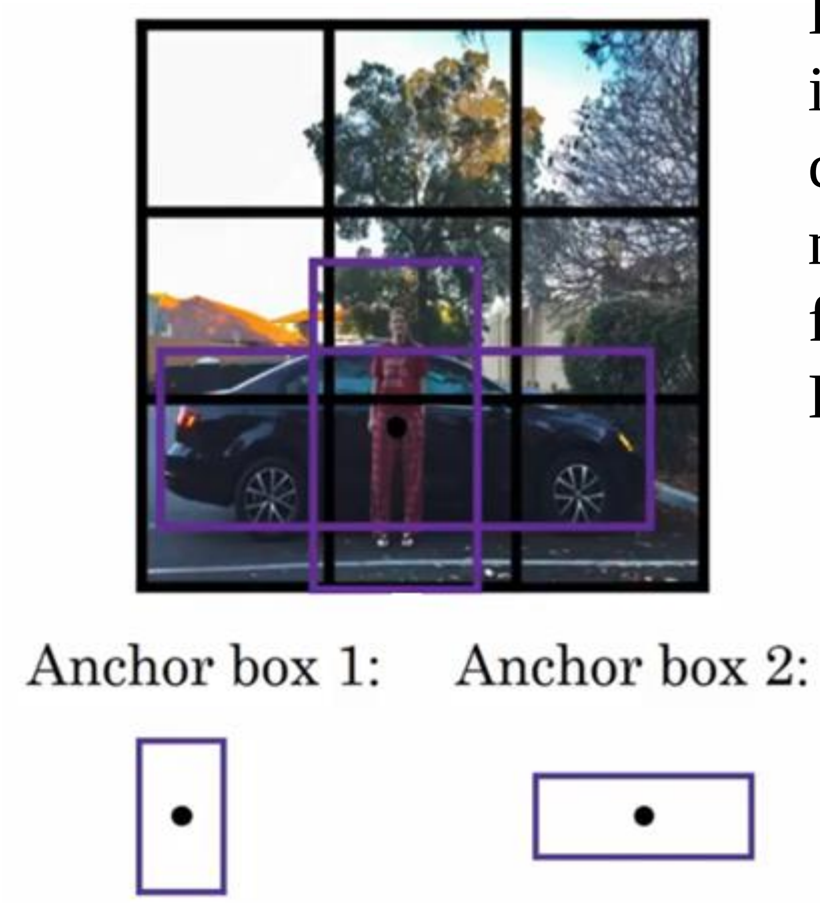
While There are any remaining boxes

  Pick the box with the largest Pc

  Discard any remaining box

For Multiple objects we can repeat Non-max suppression algorithm for each of the object we are trying to find out like pedestrian, motorcycle.

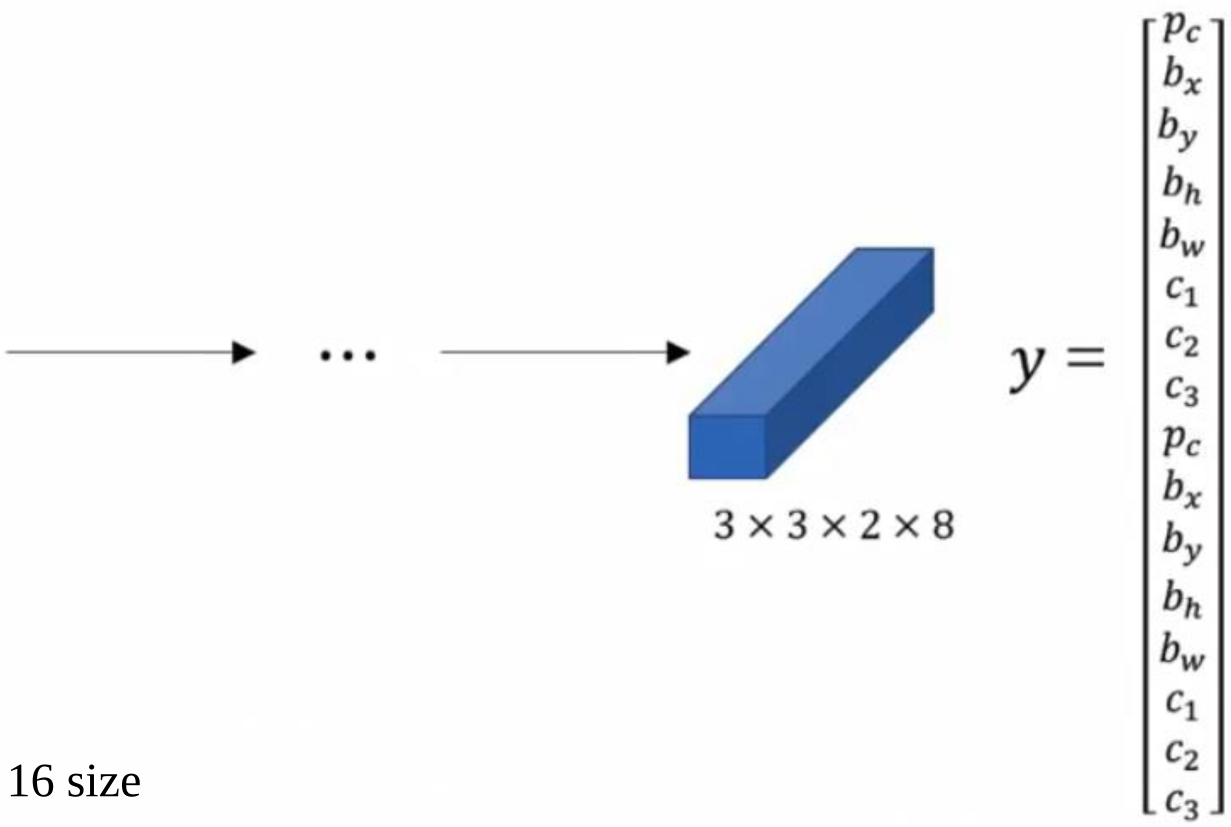
Anchor Box for overlapping objects



Each object in the training image is assigned to grid cell that contains objects midpoint and anchor box for the grid cell with Highest IoU

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

Yolo Algorithm (combines all the pieces together)



Each Grid Cell will produce a 16 size output vector

2 is the Number of Anchor boxes and 8 is the no of output points in the given vector for each object to be detected

Face Recognition Vs. Verification

Verification

Input an image and also the Name/ID of the person

Output whether the input image is of the claimed person

Validation

Input an Image

Output whether the image is any of the K persons in the database

One Shot learning

Where you have only one image for training the system (learning from one example to recognize the person again)

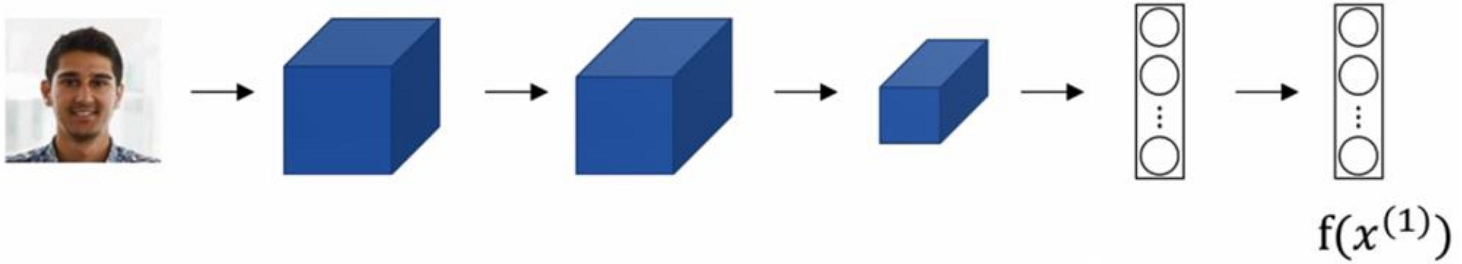
For that You need to learn a similarity function

d(img1, img2) = degree of difference between images

if d(img1,img2) $\leq \tau$ (Match)

Siamese Network for Single Shot learning

Siamese Network means containing two or more identical networks and they are good in finding similarity between two comparable things



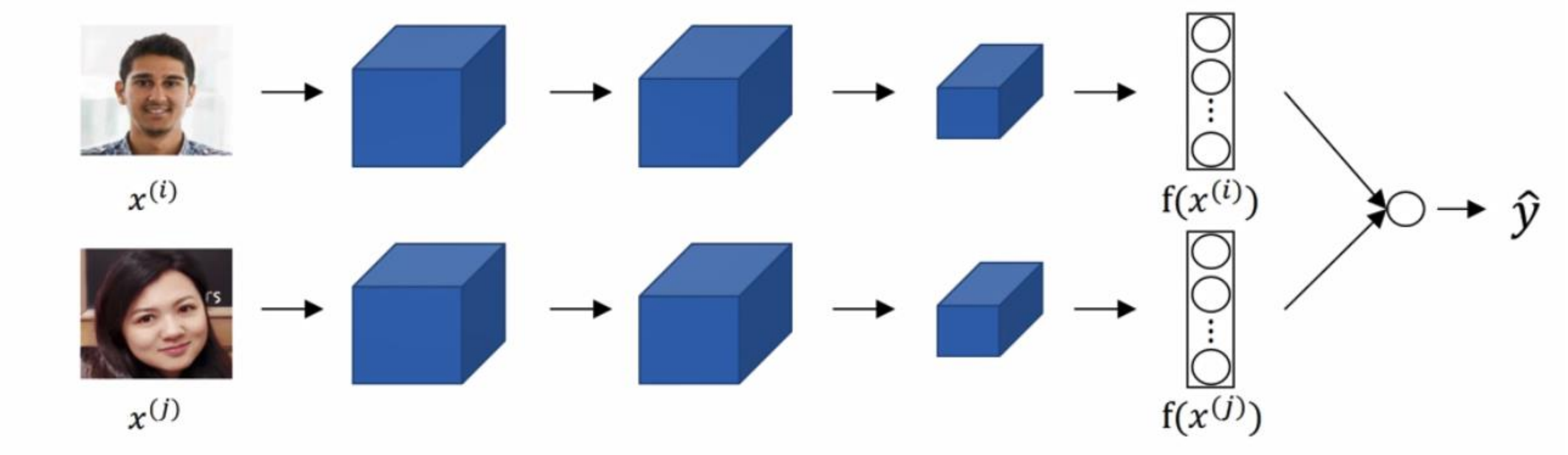
Parameters of NN define an encoding $f(x^{(i)})$

Learn parameters so that:

If $x^{(i)}, x^{(j)}$ are the same person, $\|f(x^{(i)}) - f(x^{(j)})\|^2$ is small.

If $x^{(i)}, x^{(j)}$ are different persons, $\|f(x^{(i)}) - f(x^{(j)})\|^2$ is large.

Face Verification using Siamese Network



To calculate the Sigmoid classification we can take the differences of the $f(x^{(i)})$ and $f(x^{(j)})$ Vectors and can predict whether the two persons are same or not.

Also, to reduce your computations, One of the images which comes from your database can have a precomputed network and may not require to be computed every time.