Ahmad Hussameldin Hamed Hassan

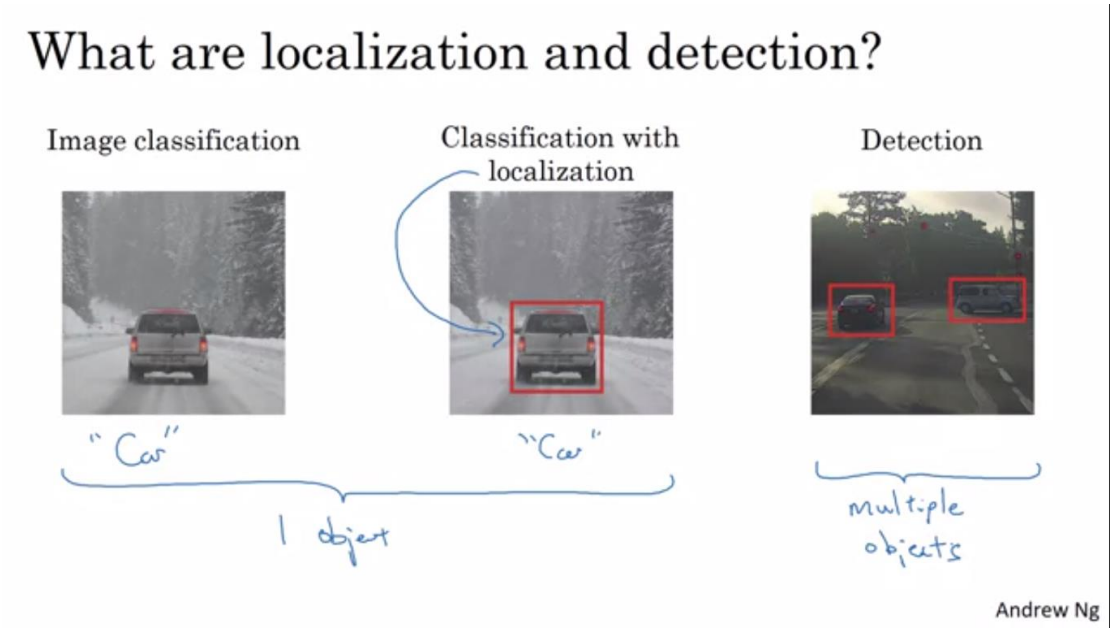
Shared Git-hub link: https://github.com/ahmadhassan1993/sharing-github

Convolutional Neural Network

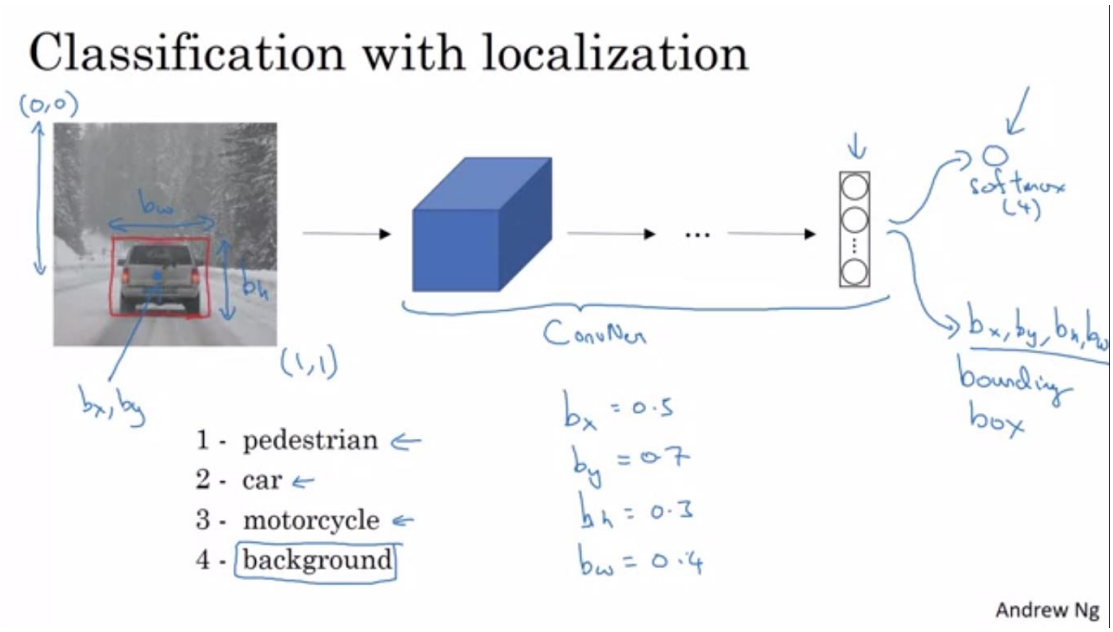
Week 3 Summary

Detection Algorithms

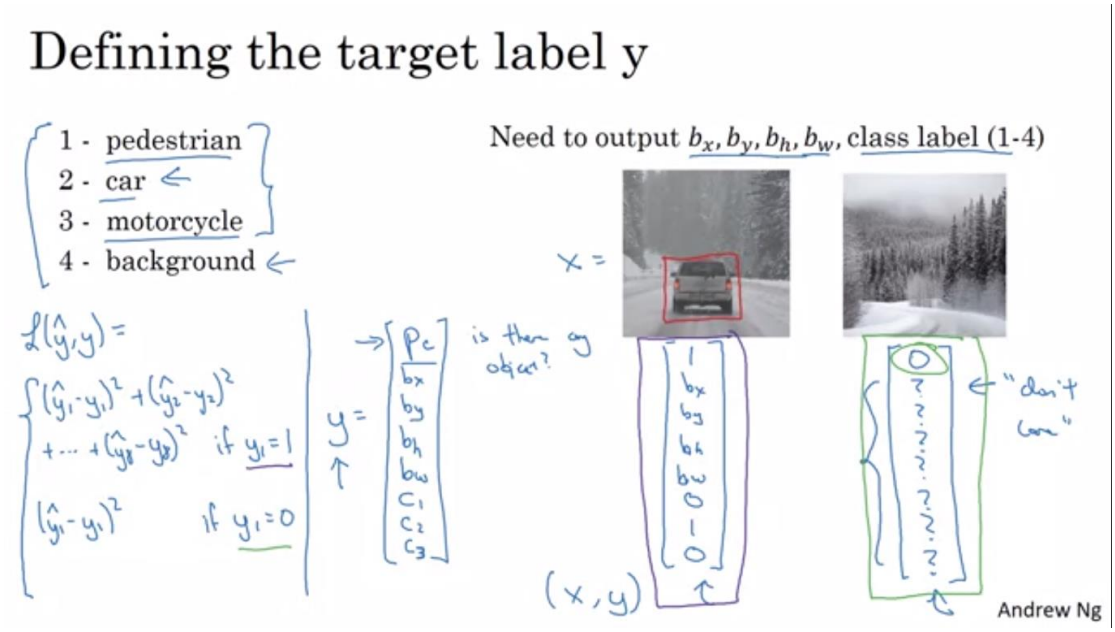
Object Localization vs. Multi-Objects Detection:



1- Object Localization: by supervised learning



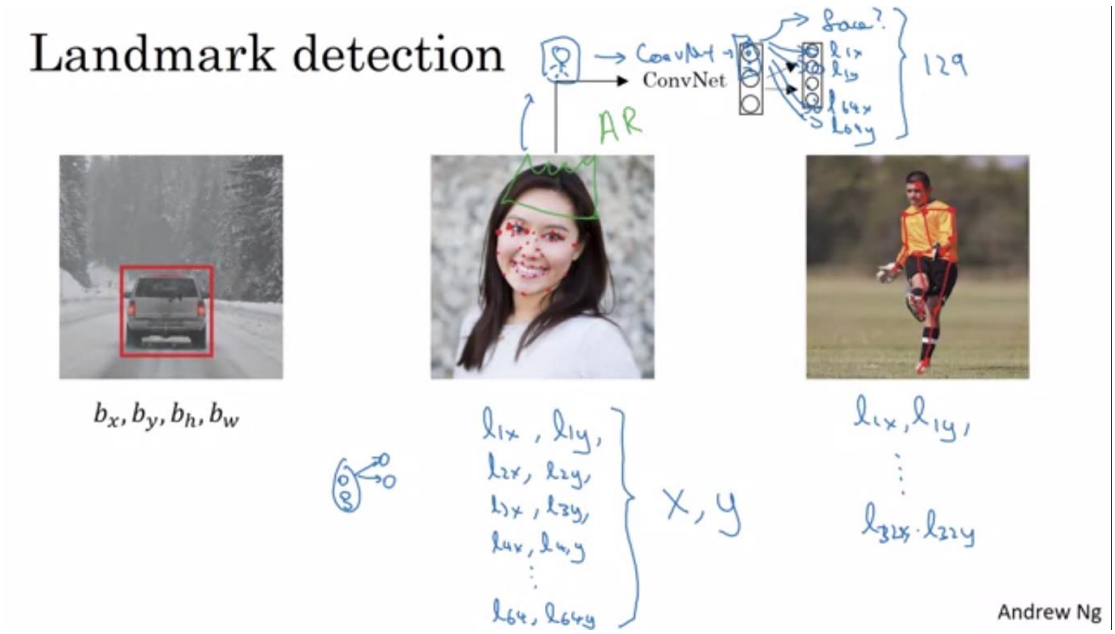
Two examples with their outputs and loss functions:



Square error was used here for simplicity. In practice, we can use log error for classes C1, 2 and 3. Logistic regression for Pc and square error for bounding box.

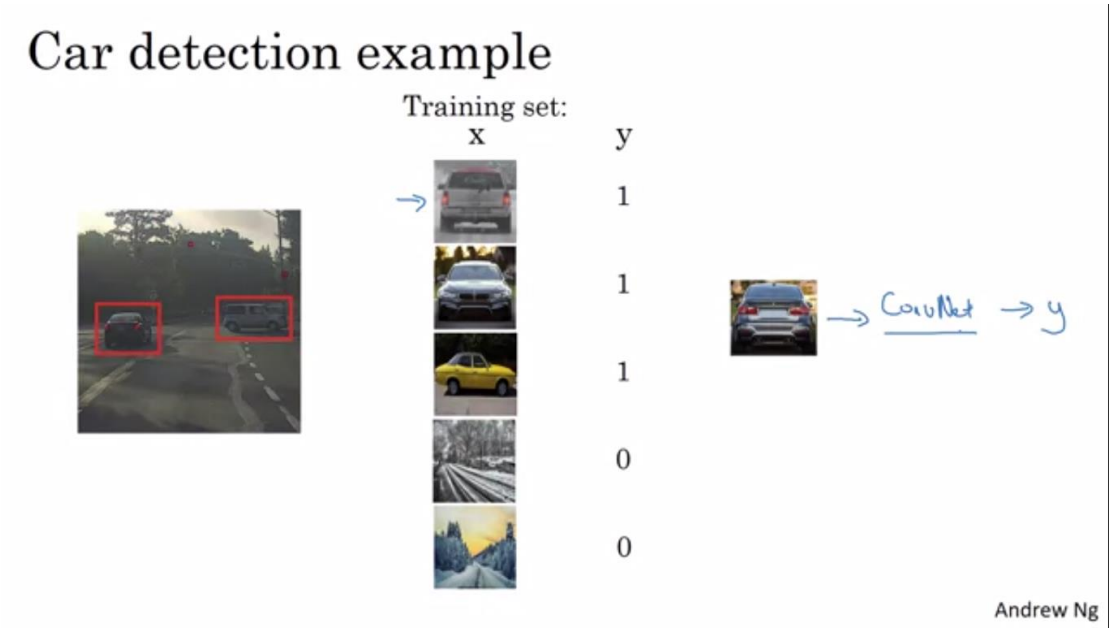
2- Landmark Detection (specific point (x,y) in the image):

For example, defining specific part of a face or the capture pos (or emotion) of a picture:

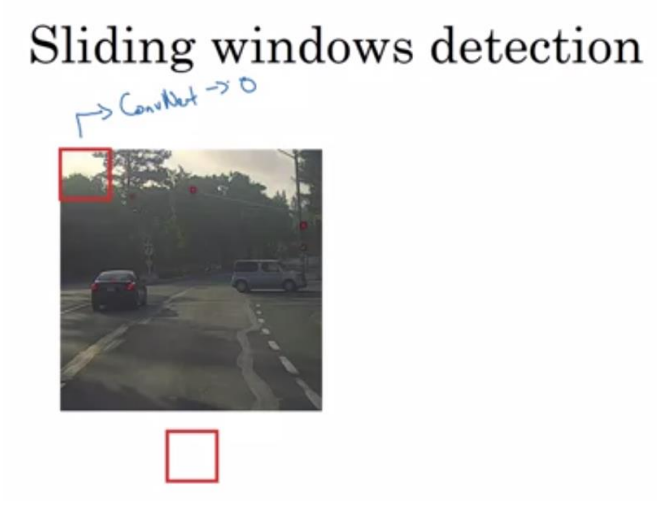


3- Object Detection:

1) Training:



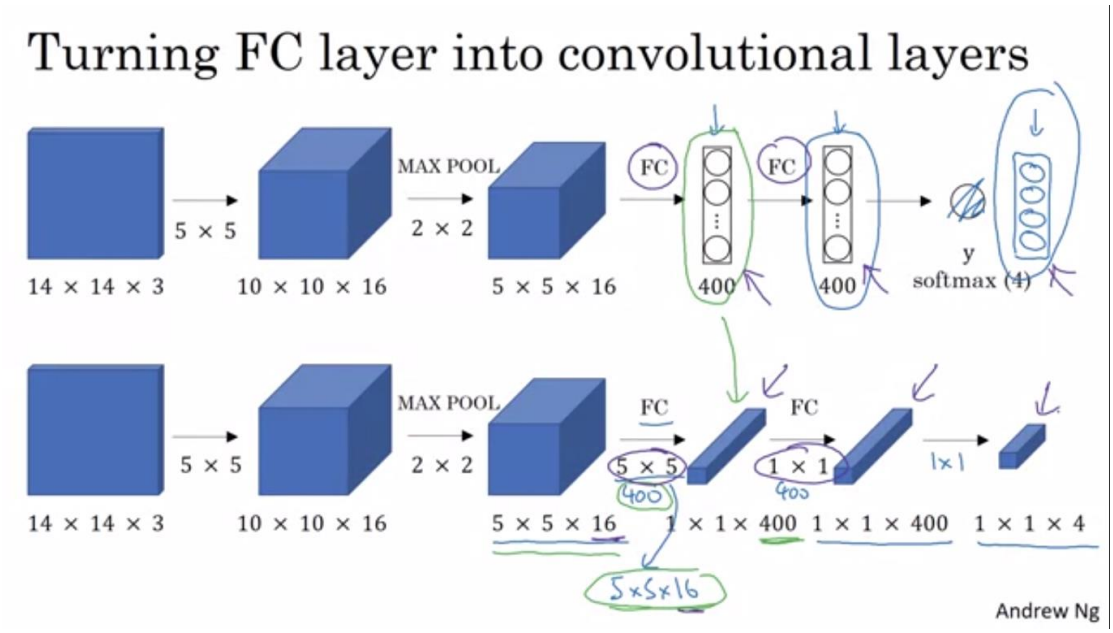
2) Sliding window algorithm of CNN:

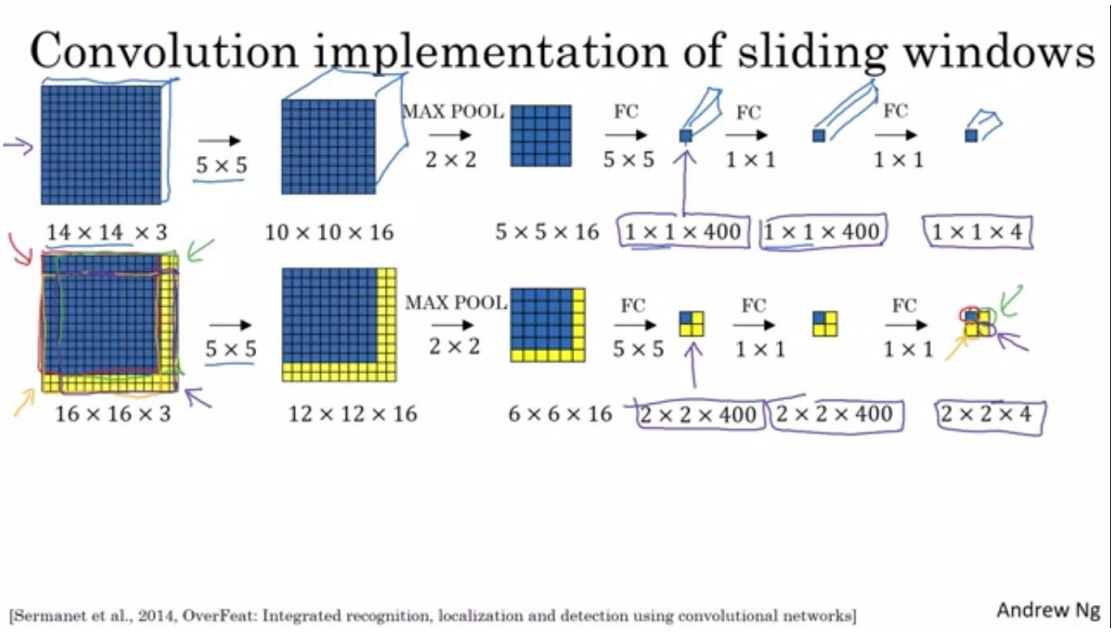


Then large the window size and repeat:

This was used before CNN era and there was a tradeoff between computation cost and performance depending on the window size choose.

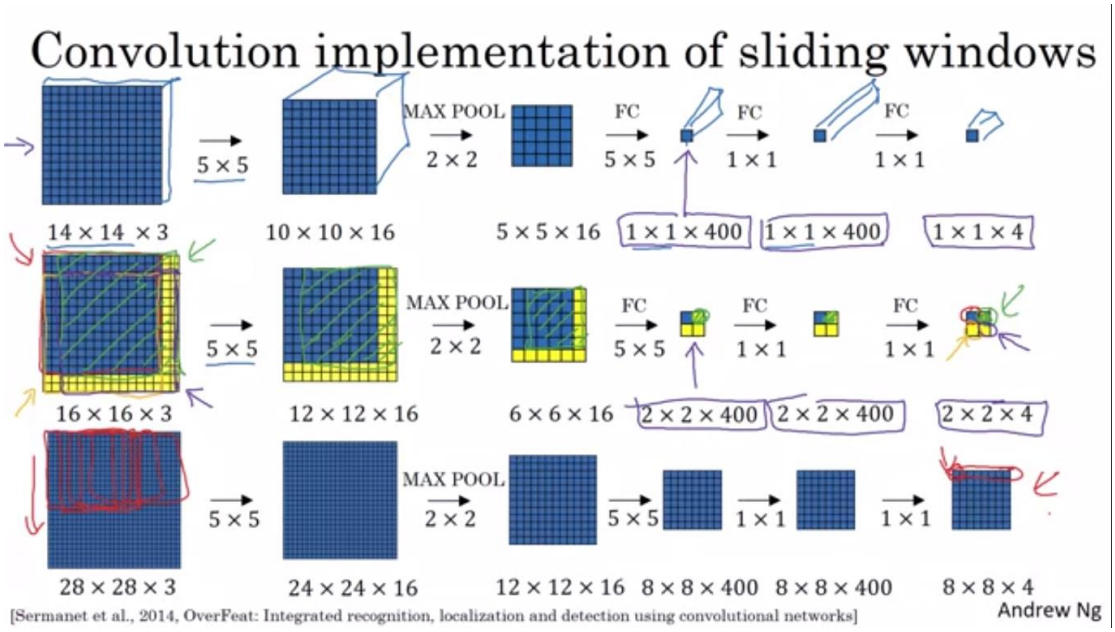
3) Convolution implementation of sliding window:

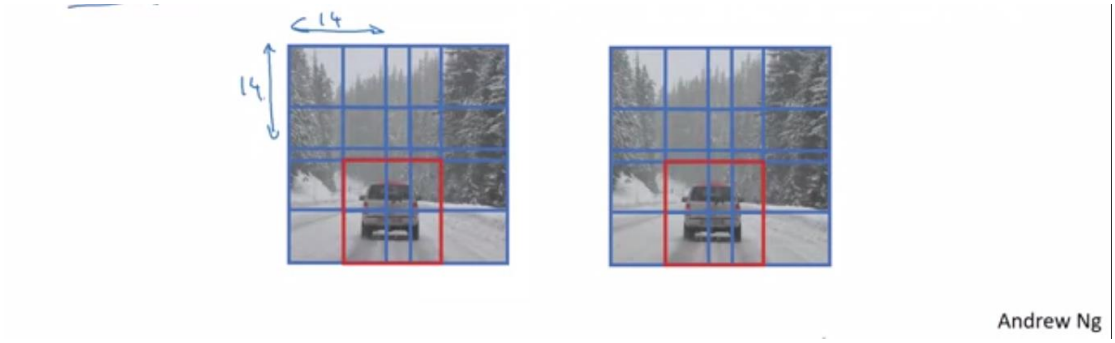




So, convolution made the computation cost of sliding window less because of sharing made by convolution.

Another example:

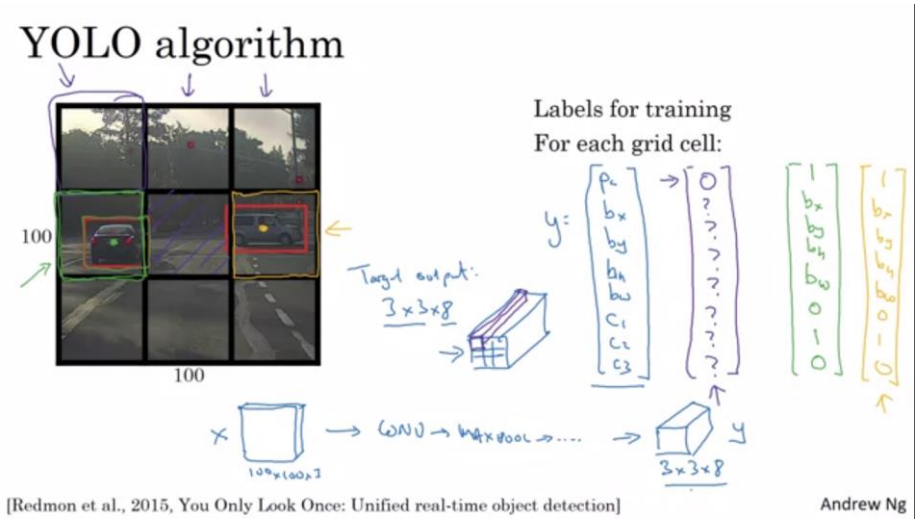




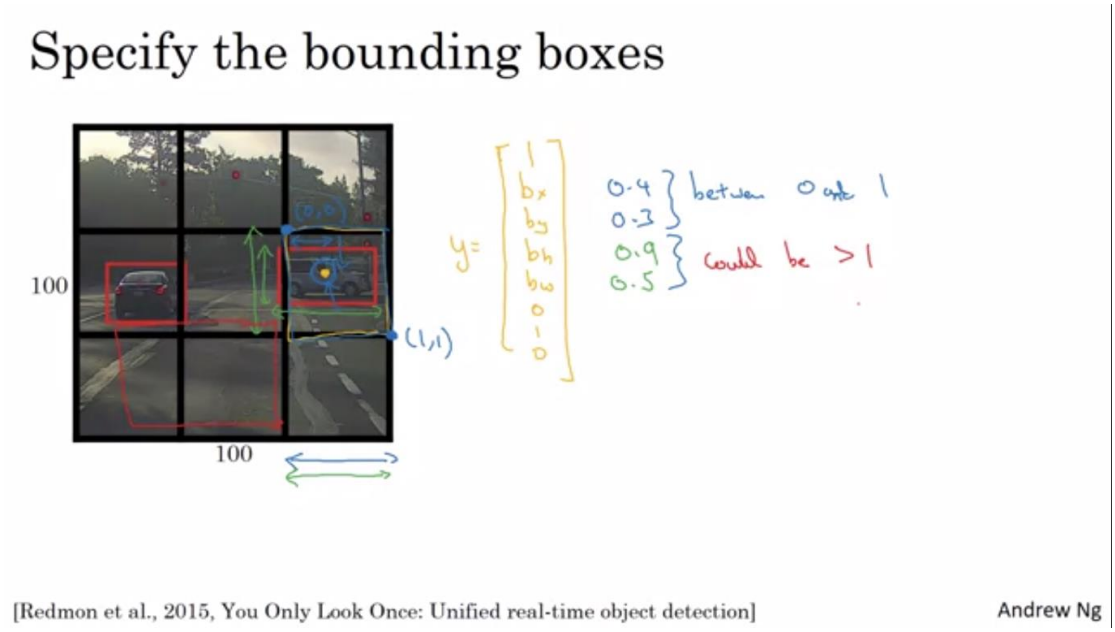
4) Output more accurate bounding boxes (YOLO (you only look once) Algorithm):

(1) Apply grid cells on the image (more cell numbers, more accuracy)

(2) Apply object localization for each cell (we must be sure that each cell has single object (in labeling phase before training))

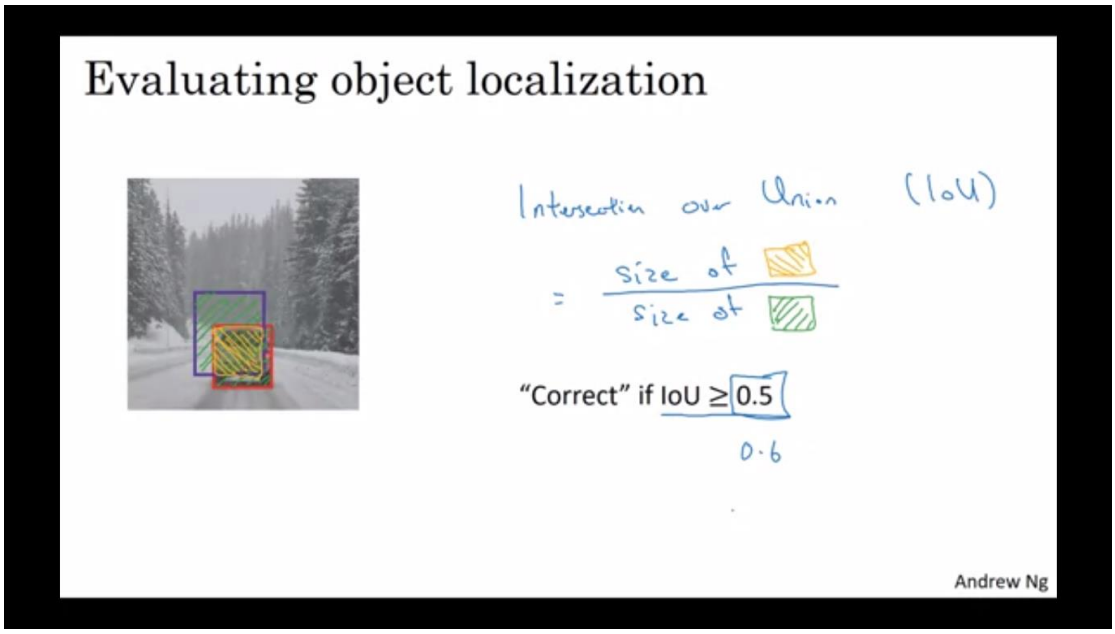


We should specify the bounds with respect to the cell dimensions:

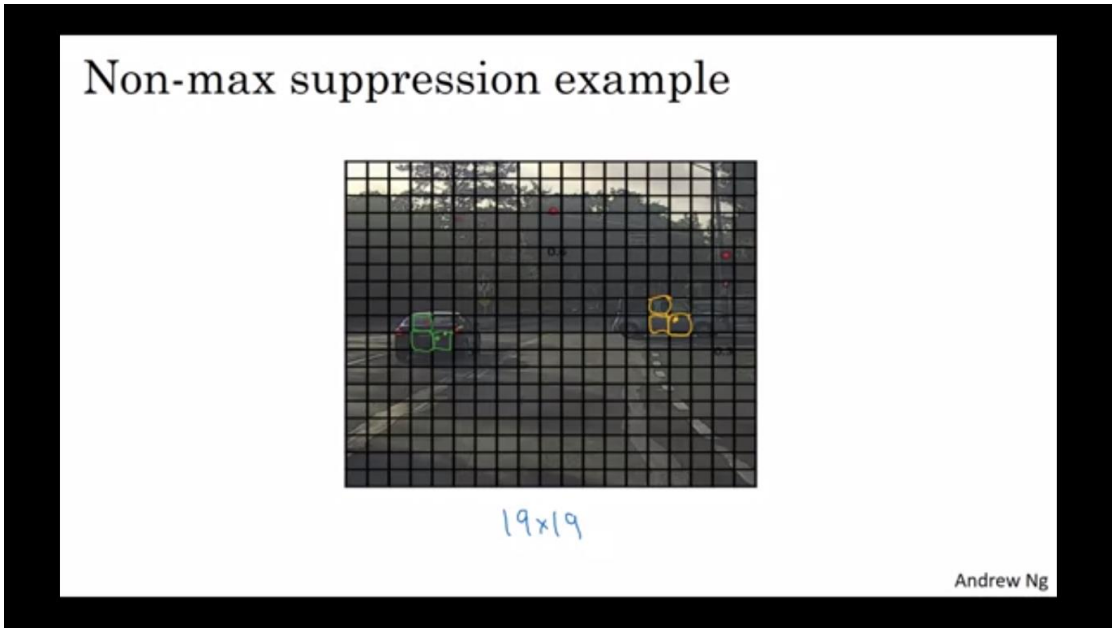


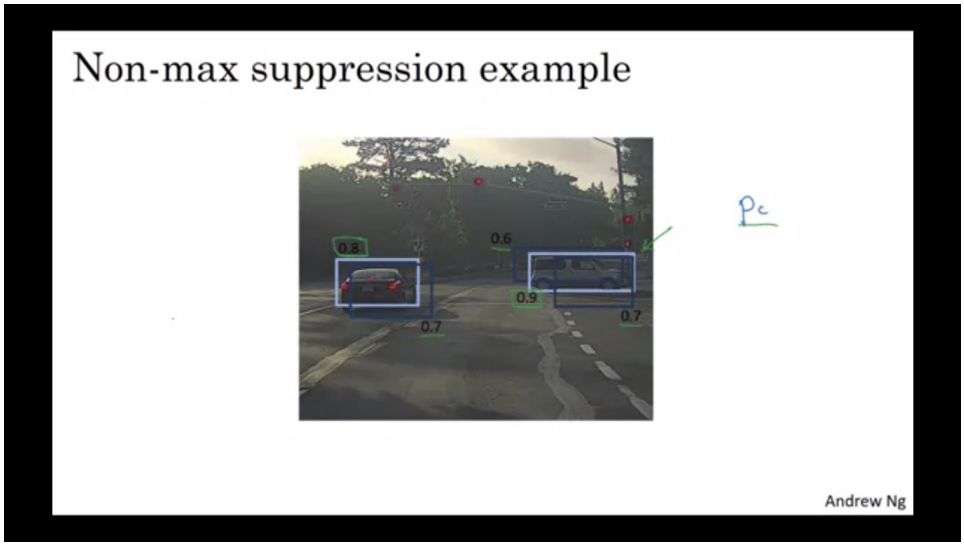
5) Evaluating the output bond box accuracy (Intersection over union, IoU):

The higher the IoU, the more accurate is the bound box from the algorithm:

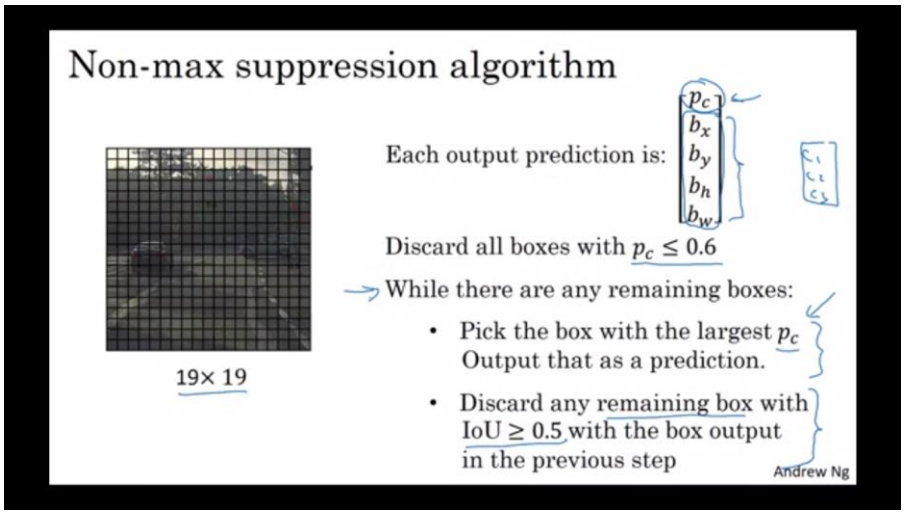


6) Detecting each object only once (Non-Max Suppression):

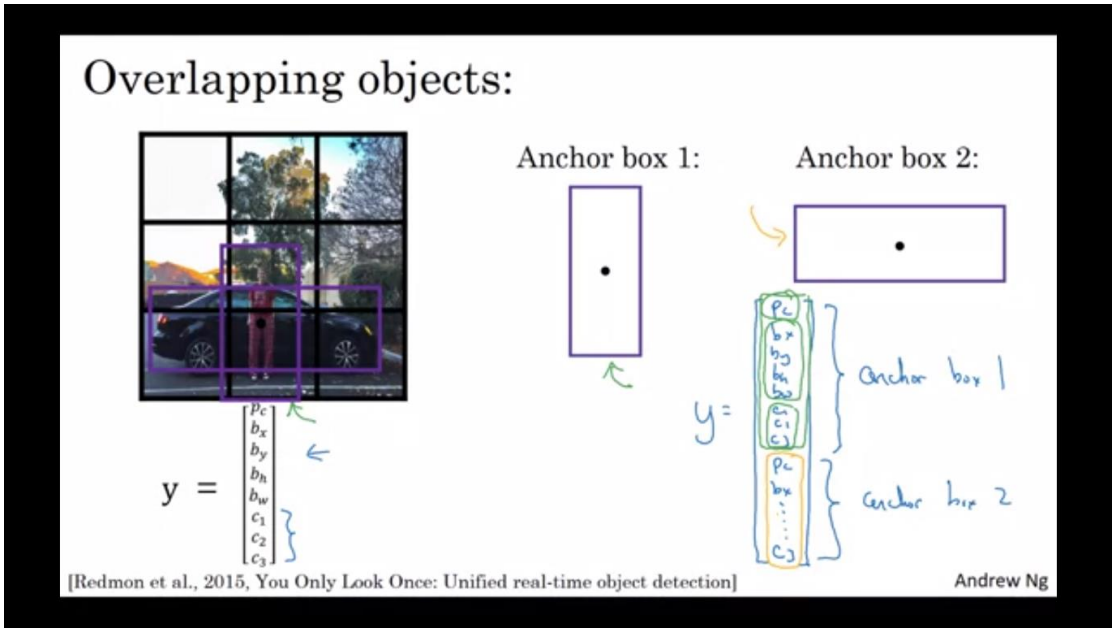


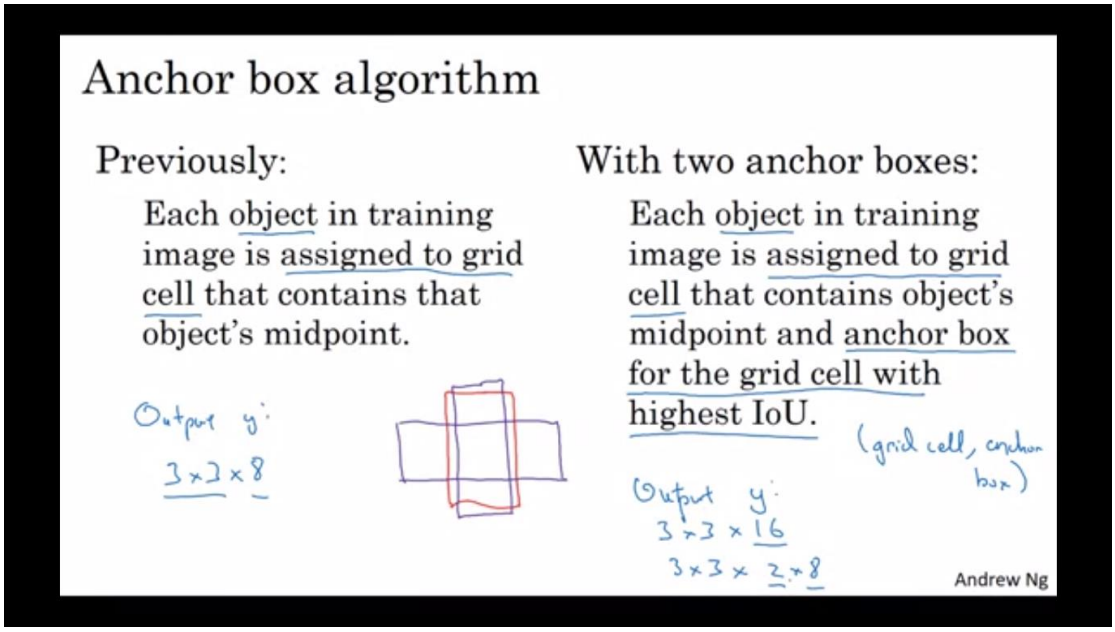


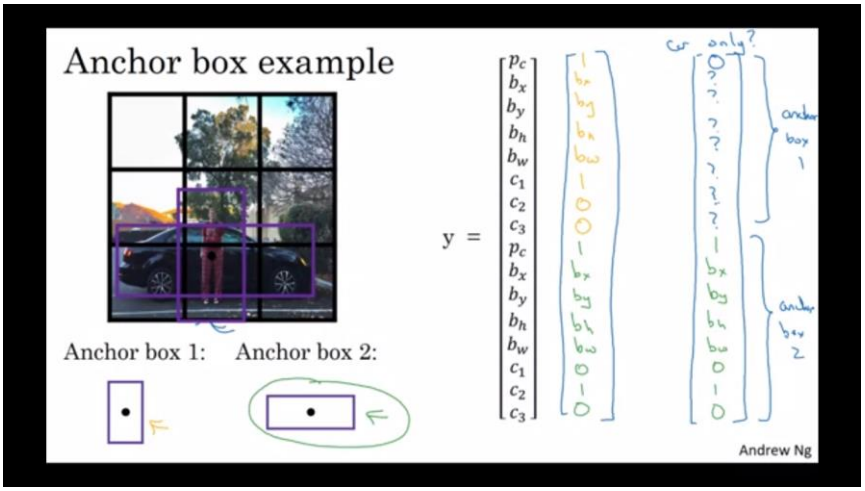
The algorithm:



7) Two Objects detection in single cell (Anchor boxes Algorithm):



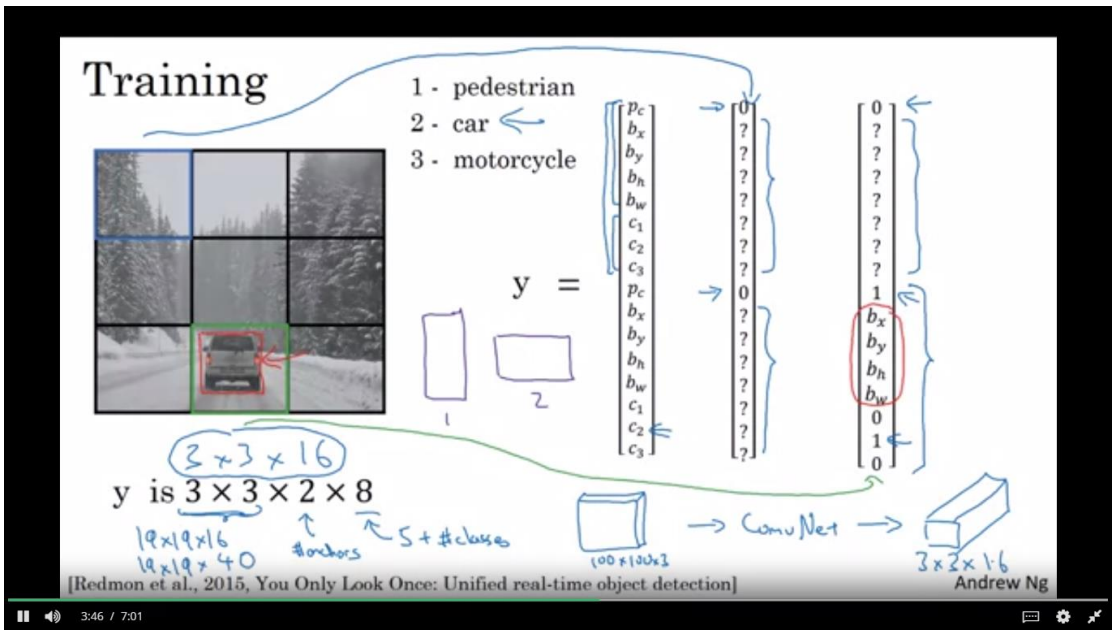


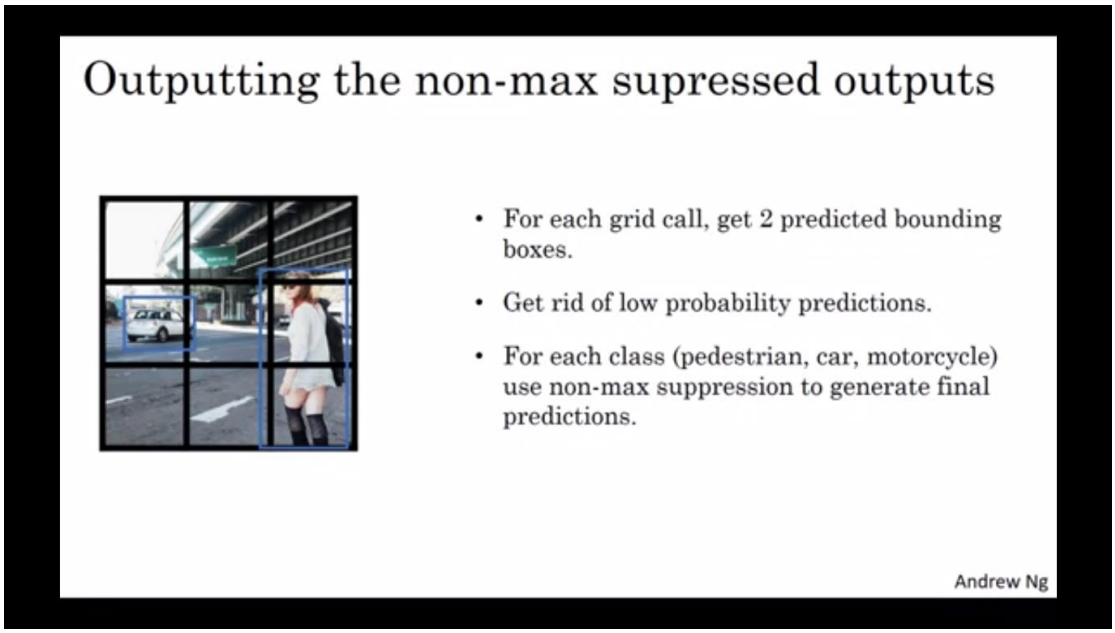


This algorithm will not work:

- If we have more than two objects in a single cell
- If two objects but with same anchor box

Summary:

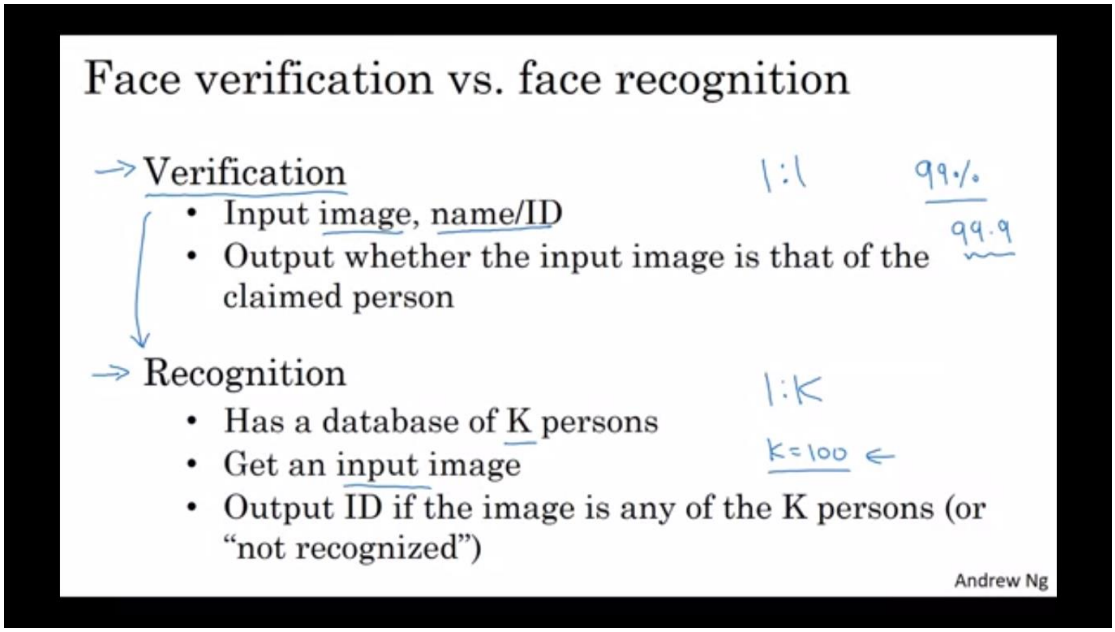




Shared git-hub: https://github.com/ahmadhassan1993/sharing-github/blob/master/Autonomous_driving_application_Car_detection_v3a%20(1).ipynb

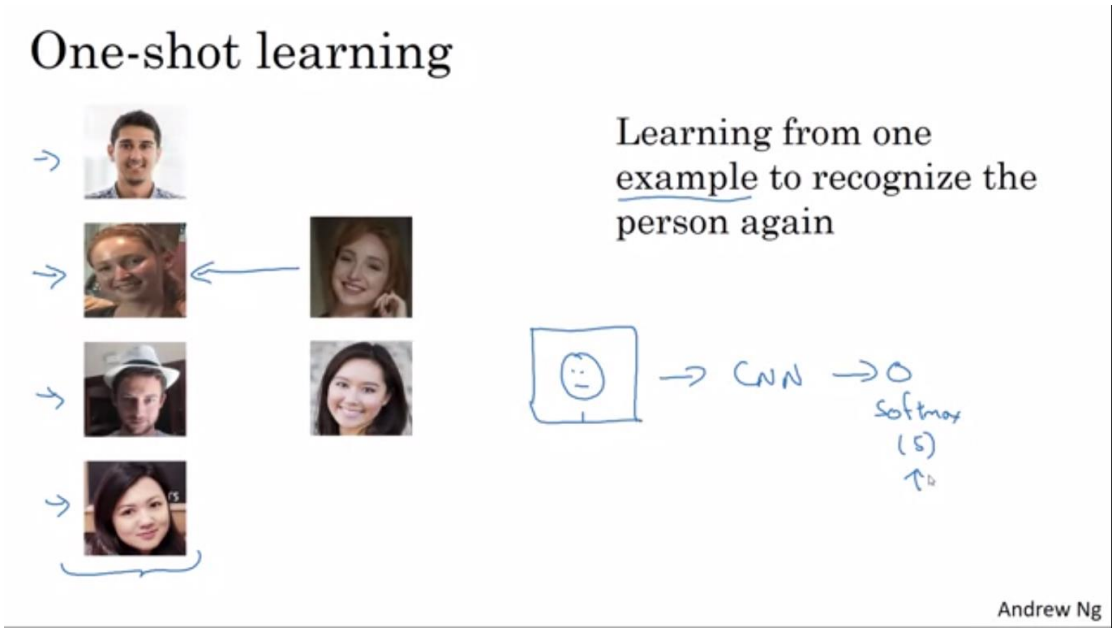
Week 4 Summary

Face Recognition

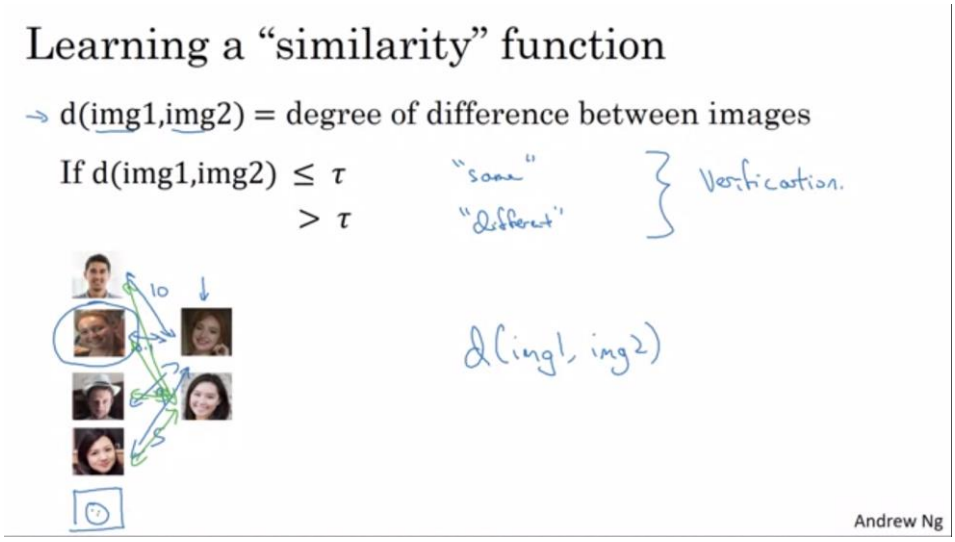


We need to make the accuracy of face verification as high as possible to be able to implement face recognition.

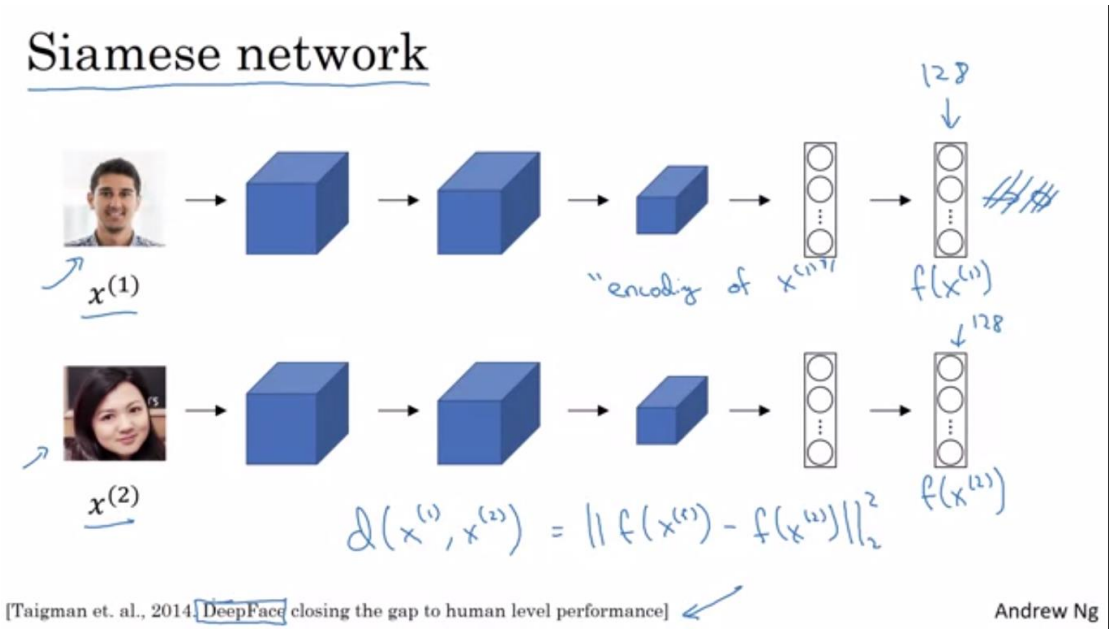
Training problem (one-shot learning or single data set):

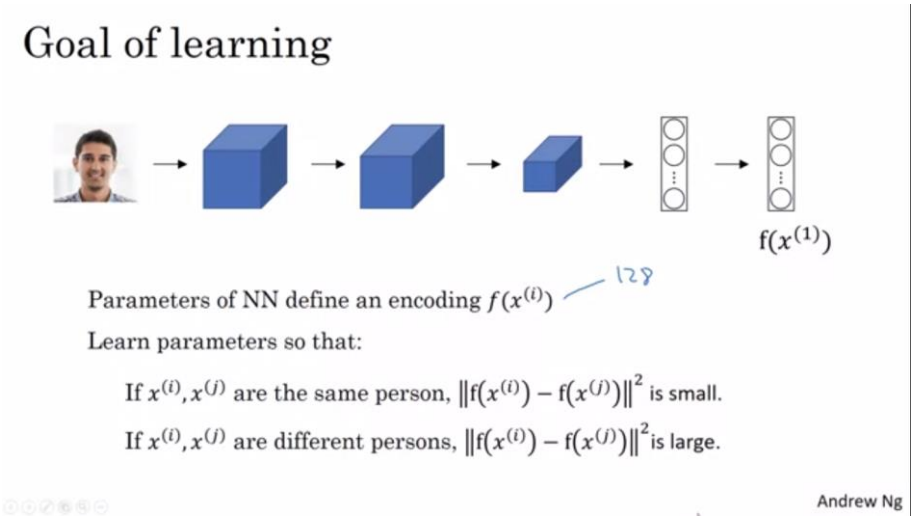


To solve this, we make the Similarity learning function:



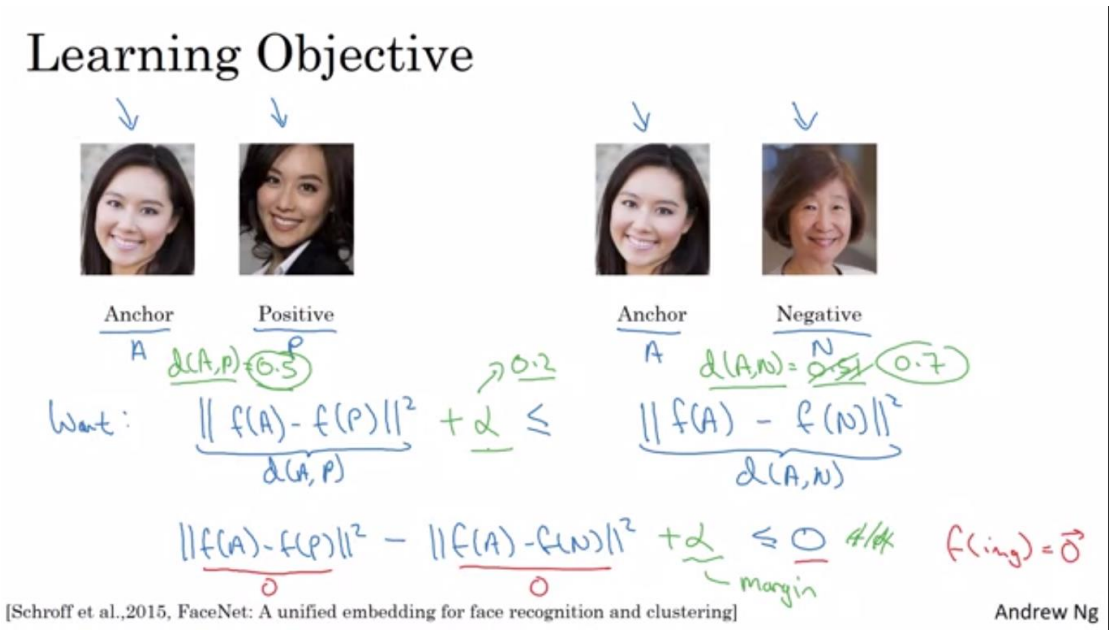
Siamese Network (implementation of similarity learning function):





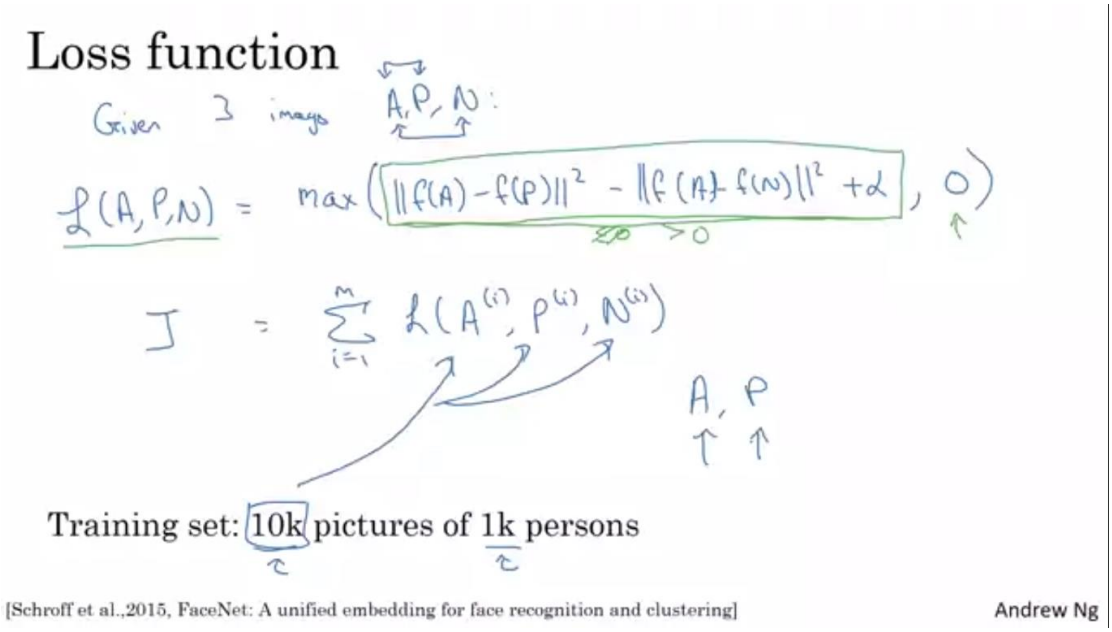
A) Triplet Loss Function (learning NN's parameters to get good encoding):

1- We use margin α to make positive difference and negative difference functions away in values from each other to prevent trivial solution (each of them equals zero):



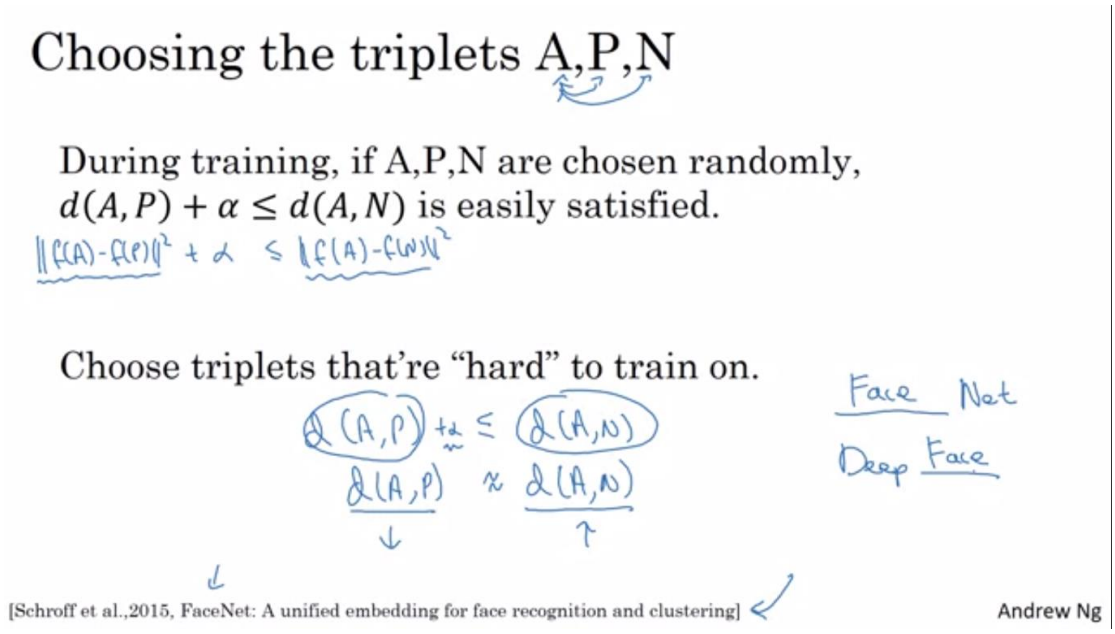
Positive is same example and negative is vice versa.

2- Loss and cost functions:

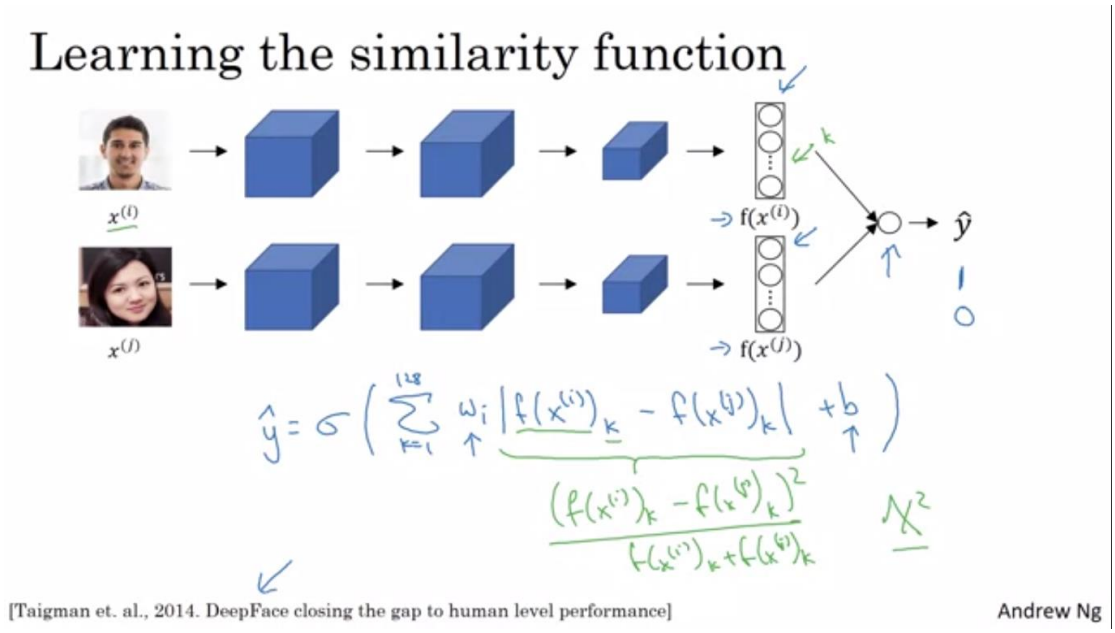


We should have many photos of single person (positive examples) to train NN.

We should take hard examples to train the NN in order to make the gradient descent very efficient:

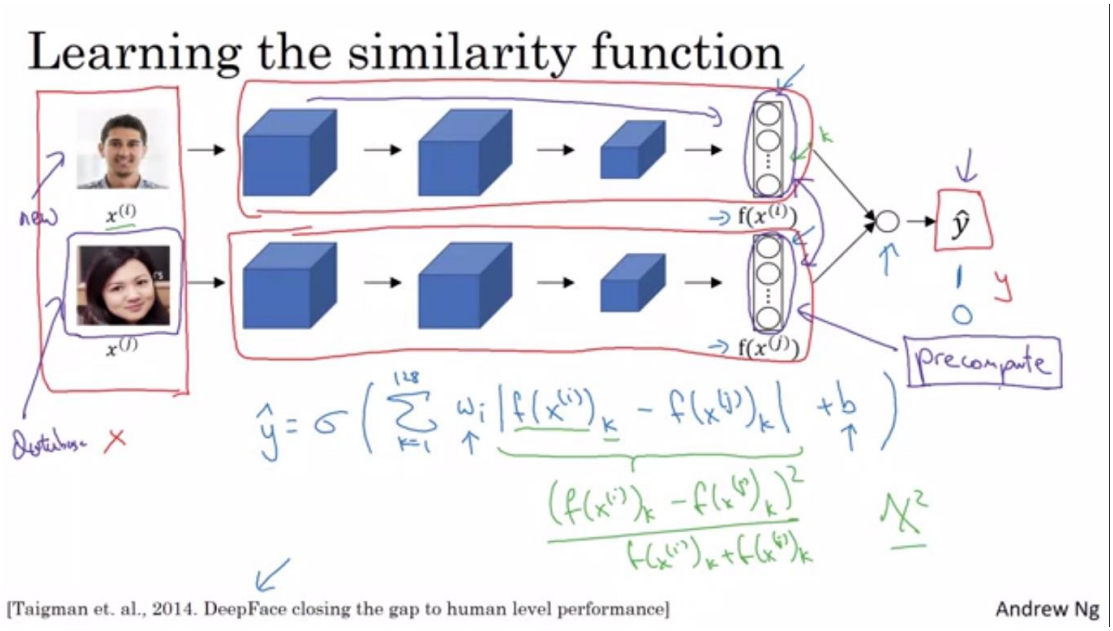


B) Binary classification (1 for same persons, 0 for different persons):



Note that both networks have same parameters.

We can precompute one branch if we already know the image before, and this will reduce memory overhead for the system (when employees for example are very large):



Note that this can also be used in Triple loss function.

Shared git-hub: https://github.com/ahmadhassan1993/sharing-github/blob/master/Face_Recognition_v3a.ipynb