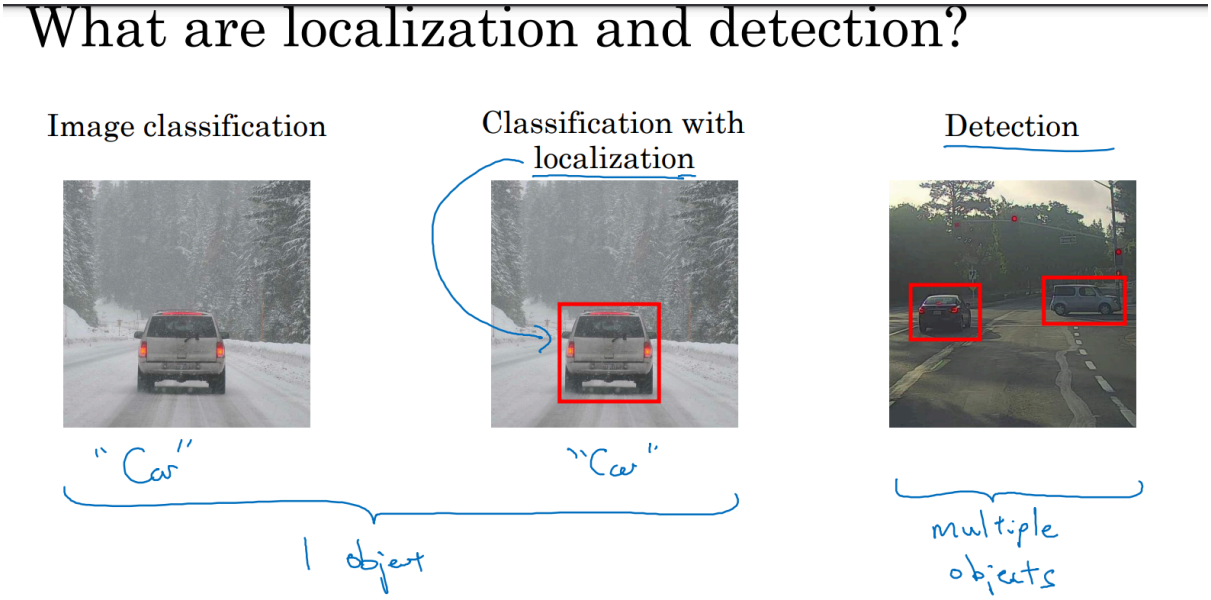
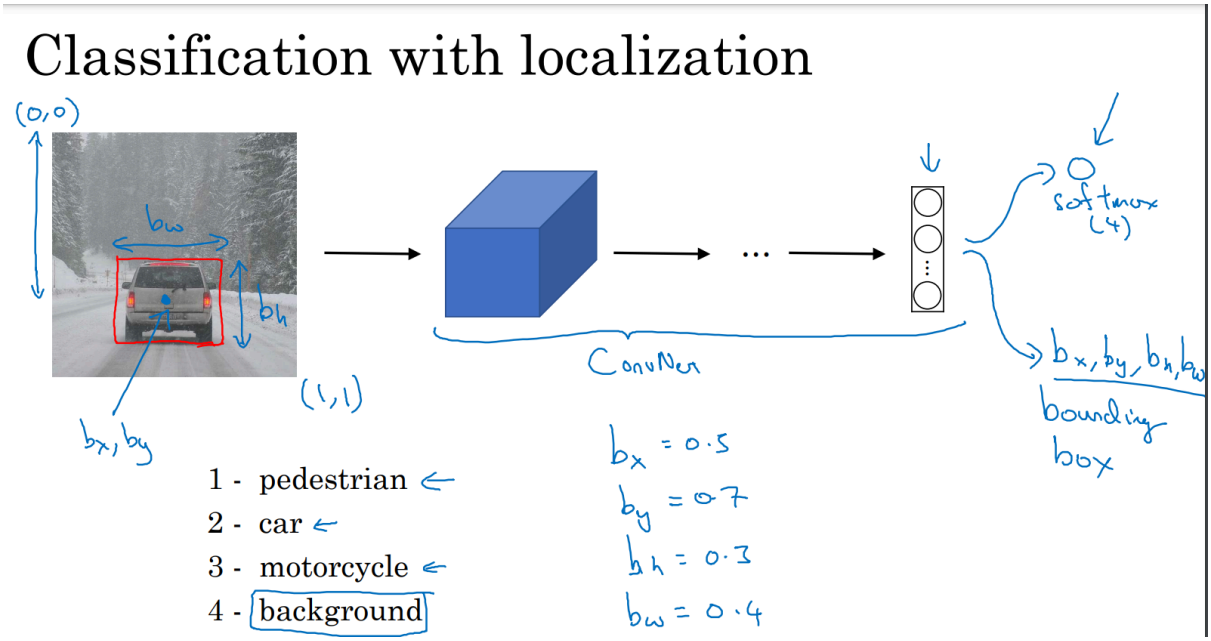
Week 3 course 4



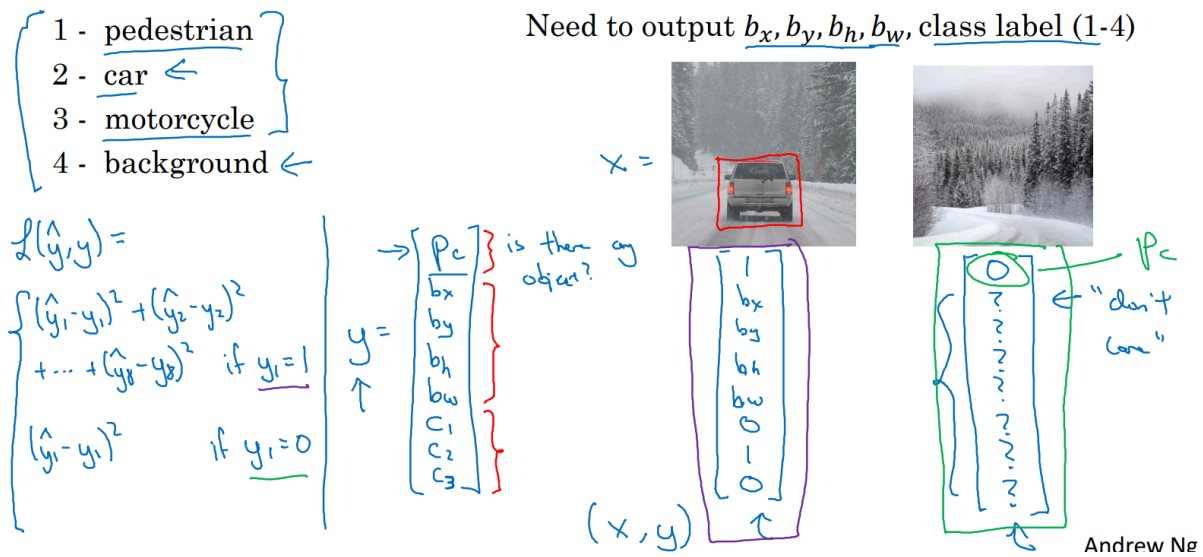
Localisation differs from classification cus of the bounding box - present only in localisation
Detection is to detect the object and perform classification followed by localisation



The softmax outputs bounding box dimensions along with the classification outputs

Bx,by form the midpoint of the object

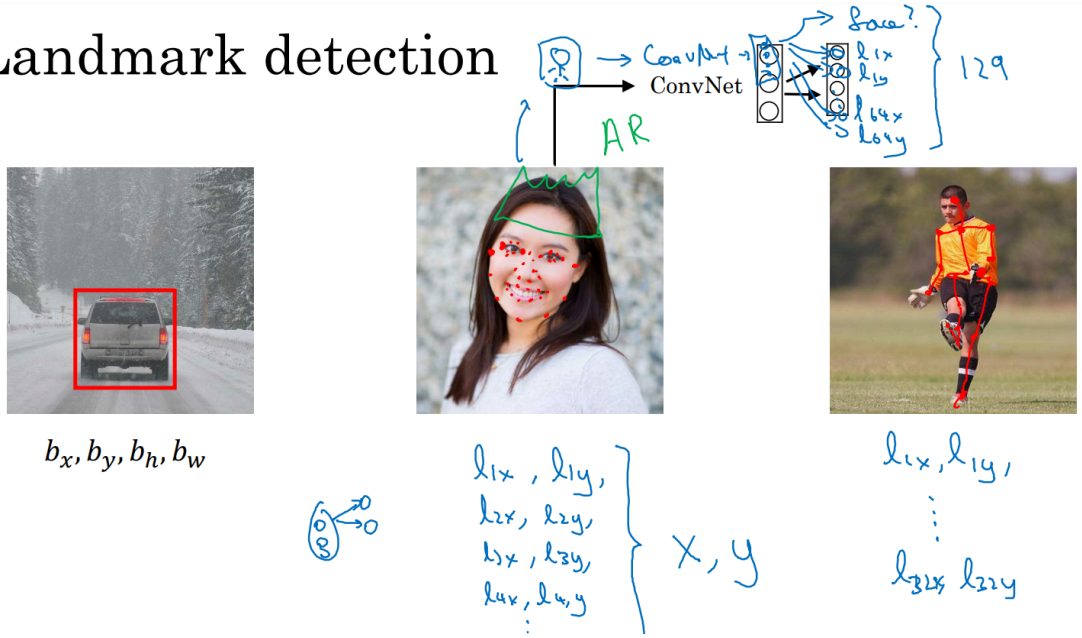
Defining the target label y



The target y hat is a column vector consisting the 1st parameter representing the confirmation of some object
If 0, the rest parameters are set to dont care and the loss function equals (y_1 hat - y_1)^2
The background category of classification denotes this type of detection

The last 3 parameters of output vector y are representing which class does the object identified belong to - pedestrian , car or motorcycle

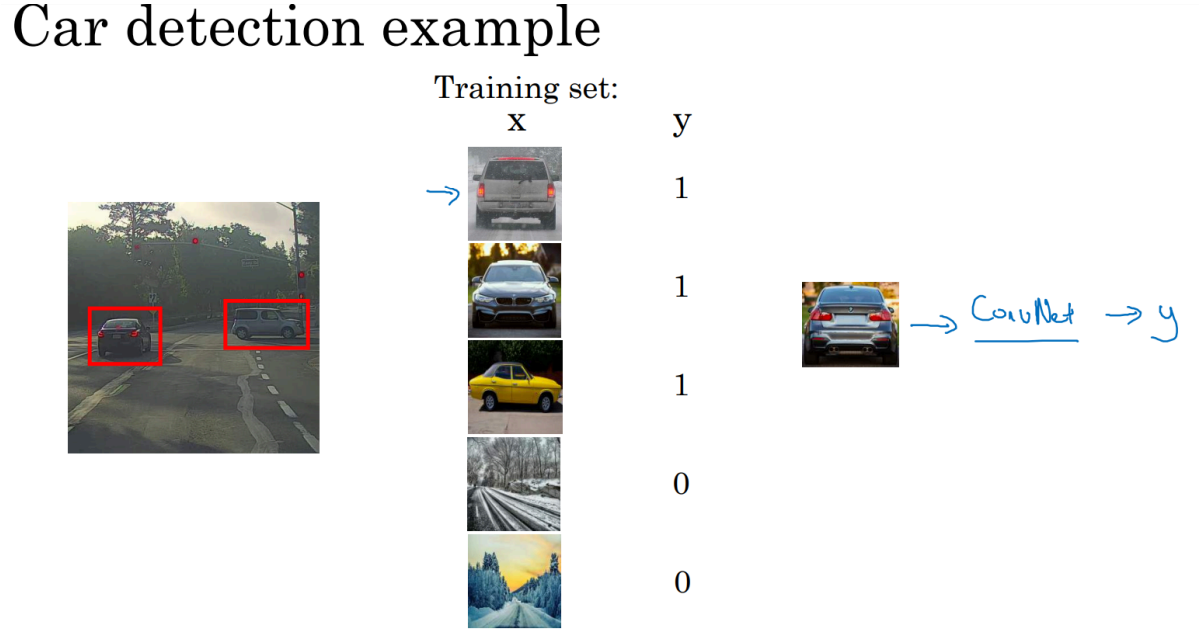
Landmark detection



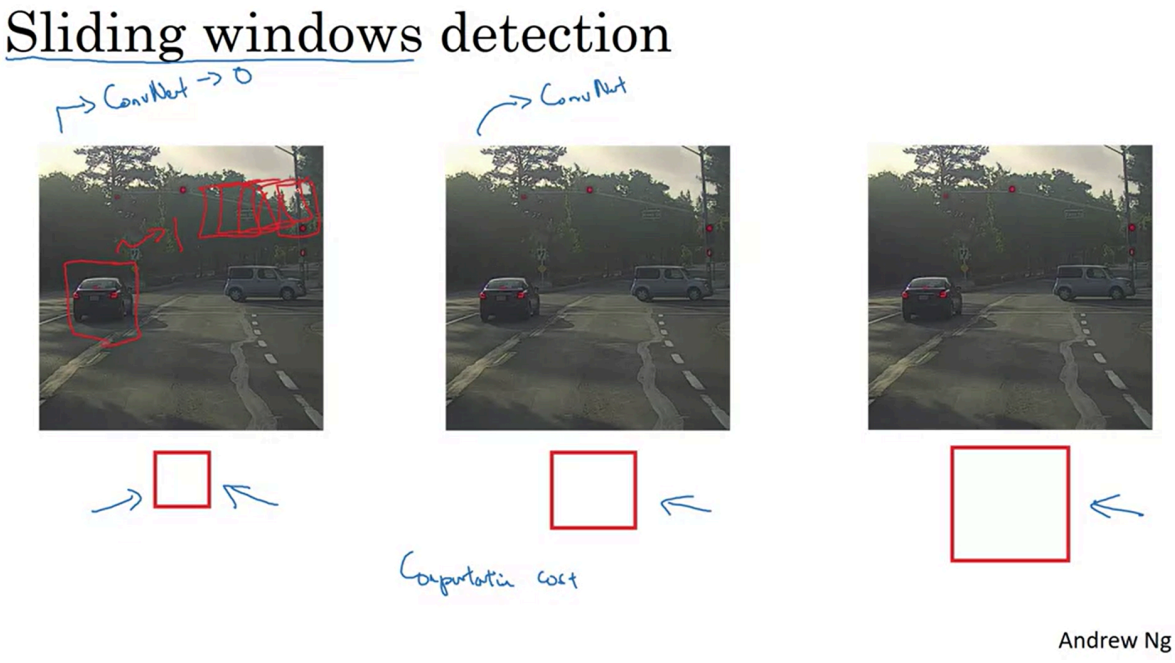
The landmarks denoted by l_1x,etc. Must remain consistent throughout the training process, also, they must be finite no.

Here, too, 1st para in y hat of 2nd image seems to be identifying if the object is a face or not

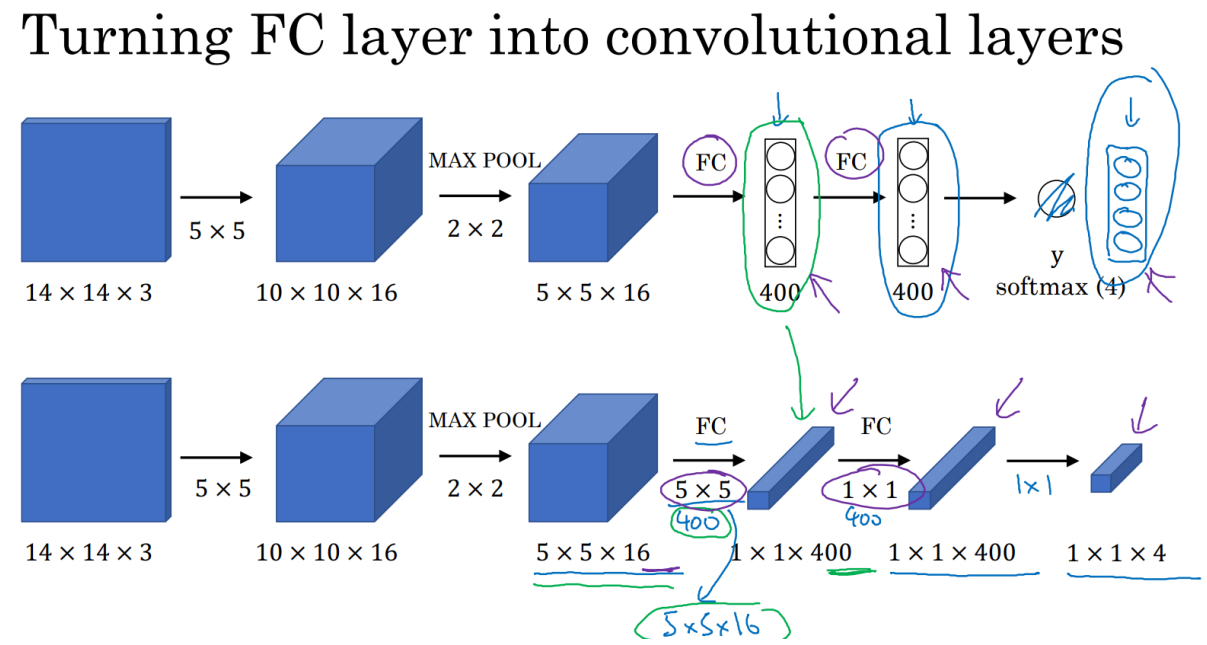
Training eg -



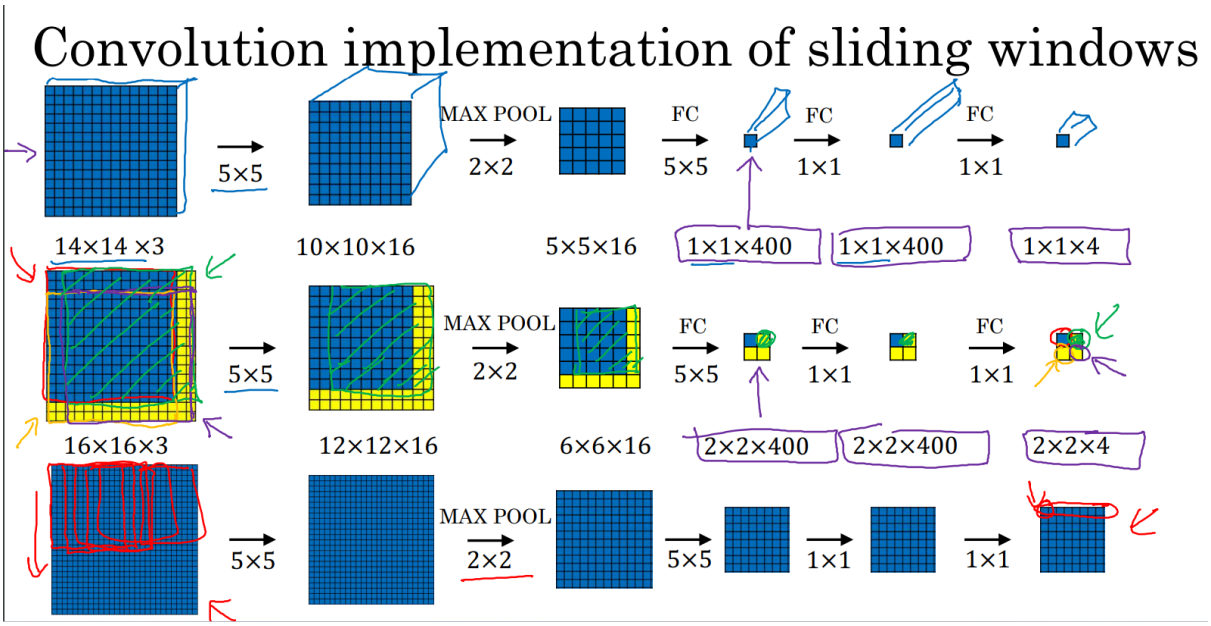
next,



Sliding window outputs 0 or 1 for the classification of wether a car?
Its size increases after each convnet/window has passed the entire image, this is computationally expensive



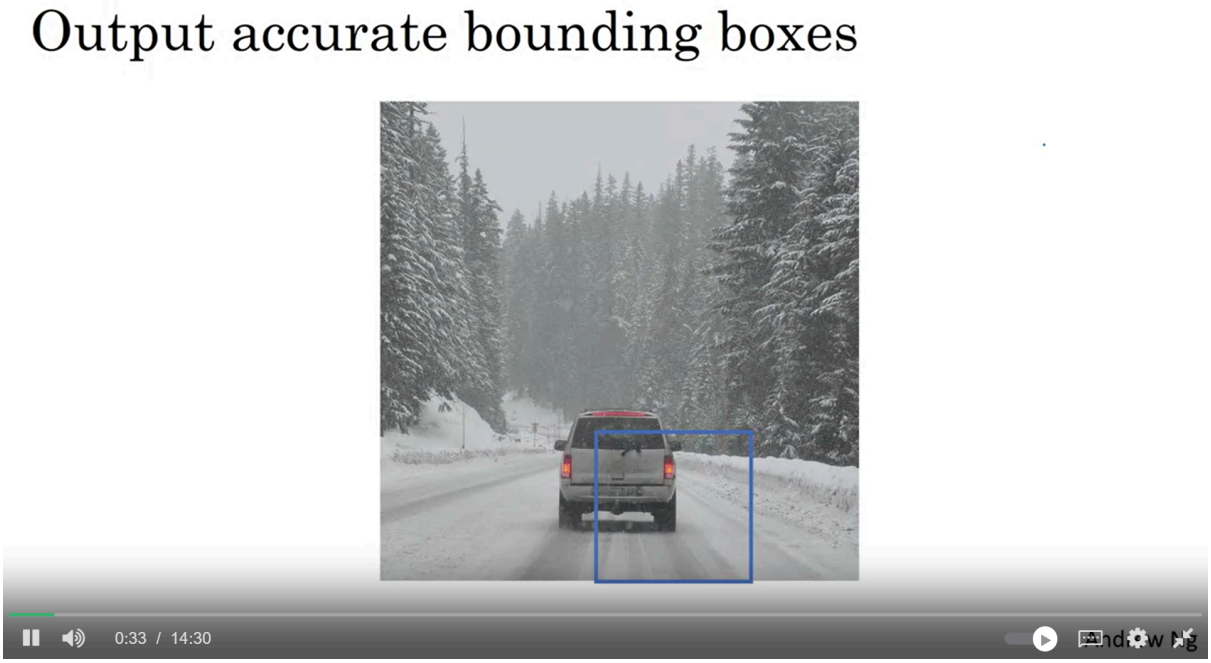
The 1st row shows the normal network, while the bottom row shows the fc turned convo network



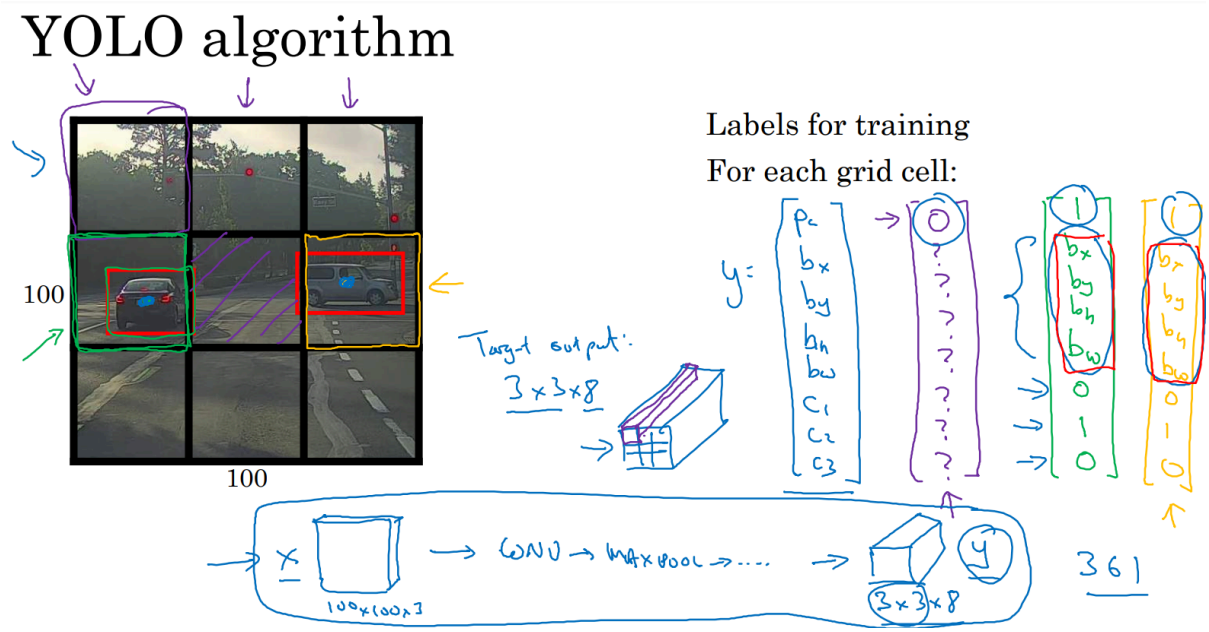
On increasing the image size from 14*14*3 to 16*16*3 shapes of FC also double up,

The result of the 2nd row contains final output of the sliding window after passing whole image
Here, the final matrix has 4 pixels with top left blue and rest yellow, i.e., 1 for top left

No. of pixels in the result equal the no of times the window slided



As seen the bonding box isn't accurate, for solving this problem yolo [you only look once] algorithm was introduced



As per this algo, we form a fine grid of 19*19 but here, used 3*3 for learning purpose

The solution is simple after forming the grid, grid containing Midpoint of the object is considered,

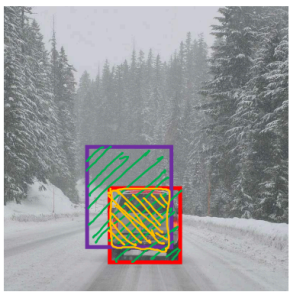
so even if the object is spreaded out to 2 or more grids, only one grid is considered

For the test data set, the algorithm runs only once and not nine times for the 3*3 grid which justifies the name YOLO , algo given at bottom

3*3 *8 is shape of y hat for the test data And train data where,
3*3 is cus of grid while 8 is due to no. of parameters

Note - bx and by - each must be between 0 and 1; while bh and bw can be more or 1

Evaluating object localization



Intersection over Union (IoU)

$$= \frac{\text{size of}}{\text{size of}}$$

"Correct" if IoU $\geq$ 0.5 ←

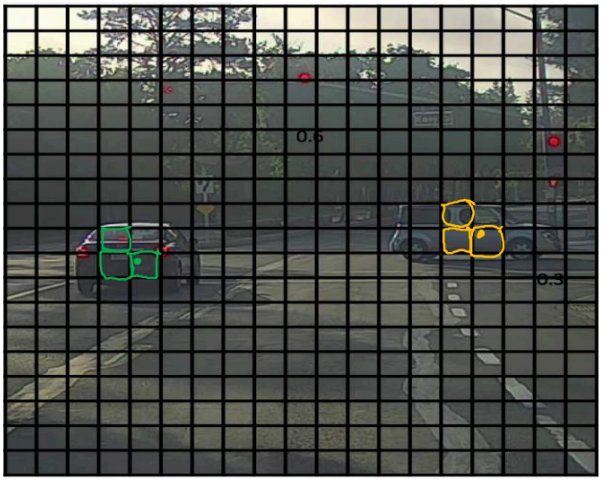
0.6 ←

More generally, IoU is a measure of the overlap between two bounding boxes.

As seen in the photo in the image, the yellow box is the intersection while the green shaded part is the union

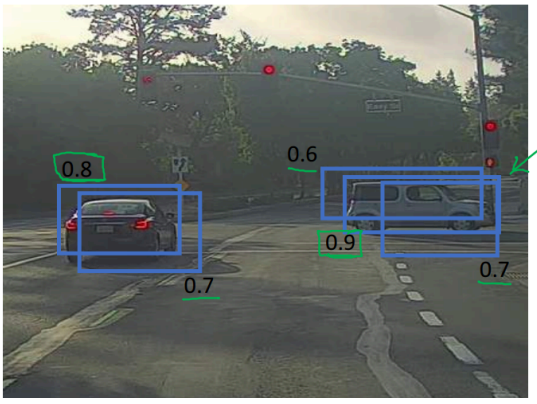
Bigger the ratio **greater overlap between the boxes** and more accurate the object localization algo

Non-max suppression example



19x19

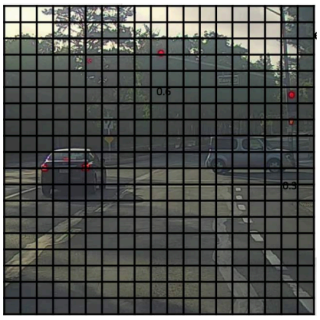
Non-max suppression example



p_c

Multiple grids or sliding windows can claim to have the object
Non-max operation solves this problem by considering the detection with highest probability

here, it is Sliding window with class probability of 0.8 for the left car and 0.9 for the right



19×19

Each output prediction is: p_c, b_x, b_y, b_h, b_w, c_1, c_2, c_3

Discard all boxes with $p_c \leq 0.6$

While there are any remaining boxes:

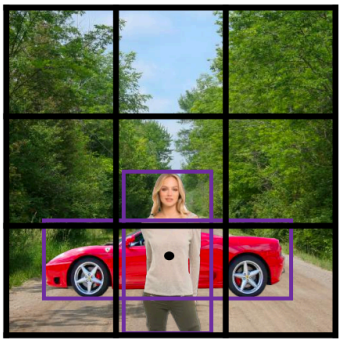
- Pick the box with the largest p_c Output that as a prediction.
- Discard any remaining box with IoU ≥ 0.5 with the box output in the previous step

Andrew Ng

note the output vector shape its due to the objects detected Belong to the same class

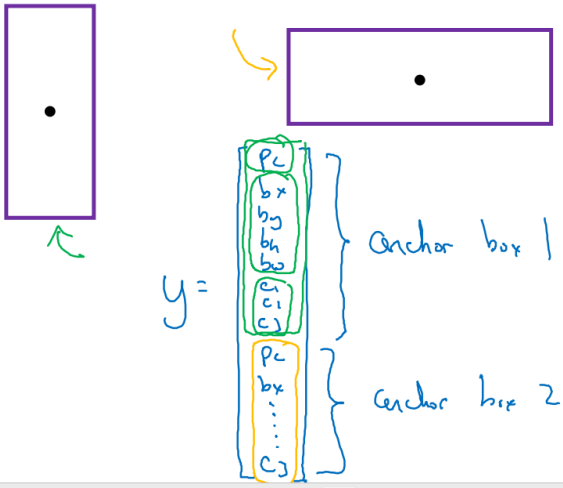
and the second point of discarding is to be noted - remaining iou removed

Overlapping objects:



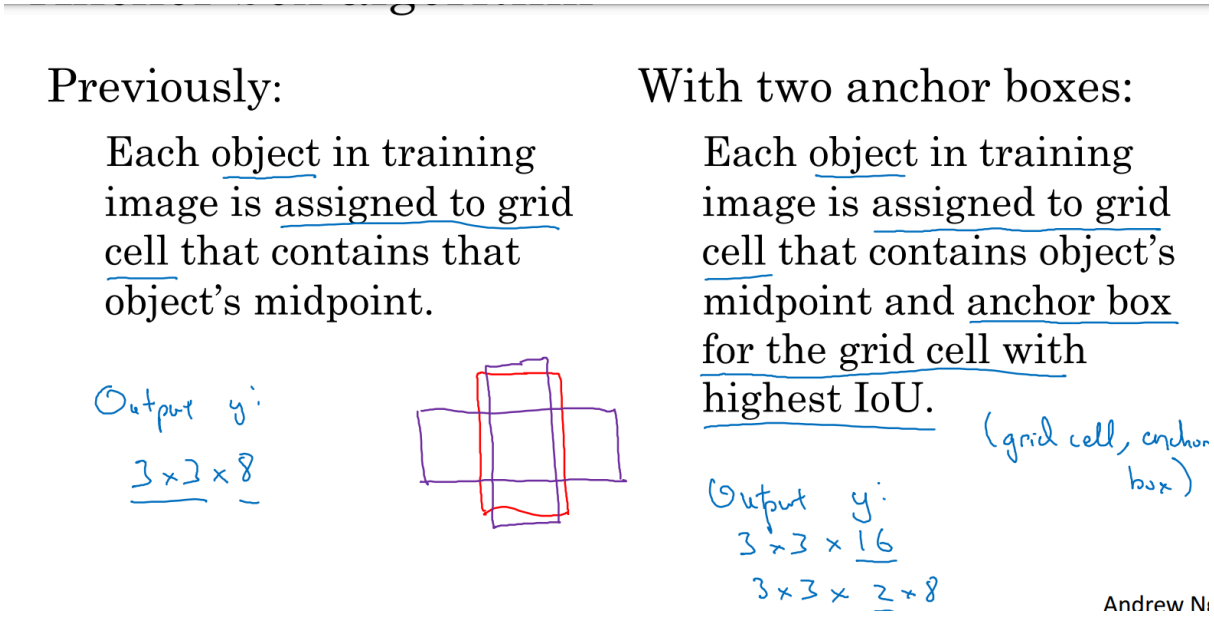
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Anchor box 1: Anchor box 2:

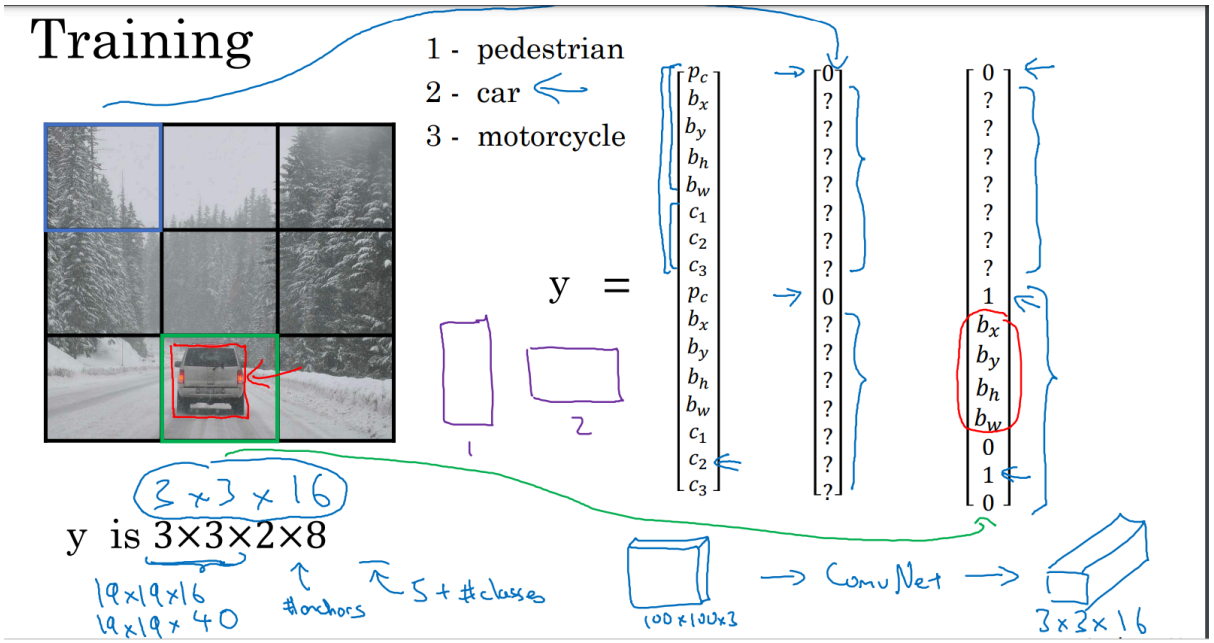


anchor boxes are the solution for object classification when different objects are having same midpoint - the box with highest IoU is the one

different shapes of anchor boxes needed are decided by K algorithm and the output is combination of the outputs of each box



the output shape considers 2 anchors and 8 parameters

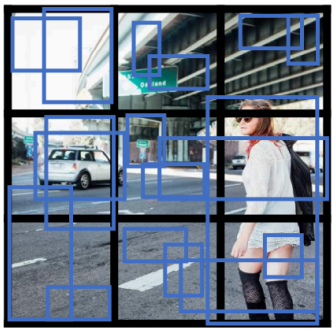


Anchors create some problem when any 1 object is detected to have 0 as pc i.e., 1st para

The above image correctly points output vectors for the respective grid cells

Note The shape of the output vector at bottom

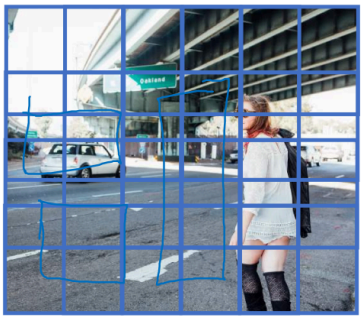
Outputting the non-max supressed outputs

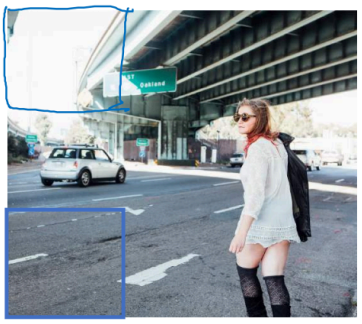


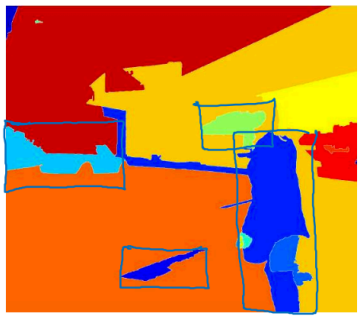
- For each grid call, get 2 predicted bounding boxes.
- Get rid of low probability predictions.
- For each class (pedestrian, car, motorcycle) use non-max suppression to generate final predictions.

For such image with Multiple bounding boxes, the given 3 steps are performed

Region proposal: R-CNN







Segmentation algorithm

~2,000

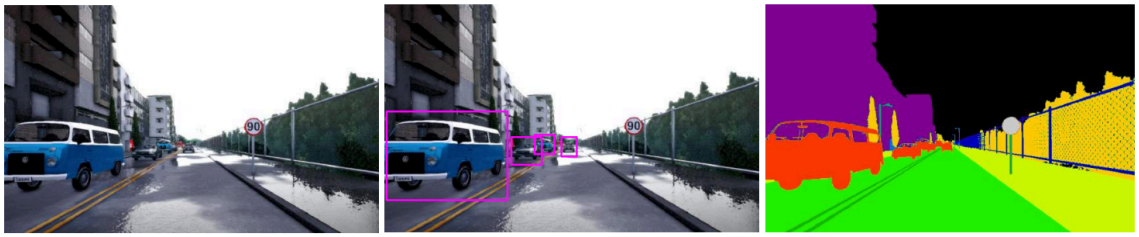
instead of passing the sliding window Throughout the image By segmentation algorithm, we detect the regions where the objects can be found and place a bounding box around this object

So region with CNN also outputs bounding box along with the detection of the region

Aim of More research paper on same topic Developed later -
[NOT imp. As YOLO is better than r-cnn]
Fast R-CNN: Propose regions. Use convolution implementation of sliding windows to classify all the proposed regions.

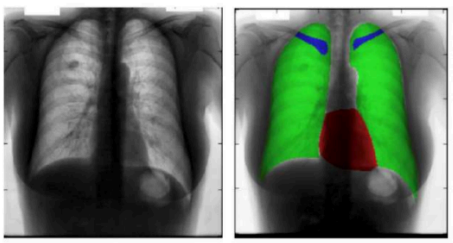
Faster R-CNN: Use convolutional network to propose region

Object Detection vs. Semantic Segmentation

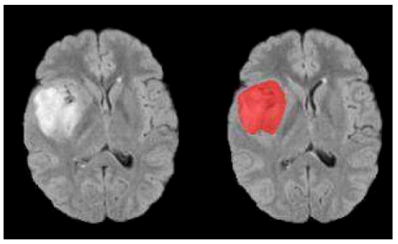


Input image | Object Detection | Semantic Segmentation

Motivation for U-Net

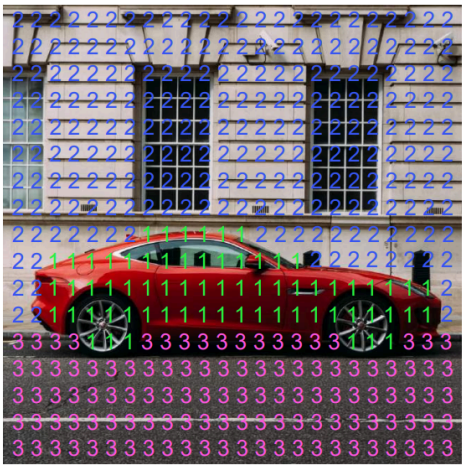


Chest X-Ray



Brain MRI

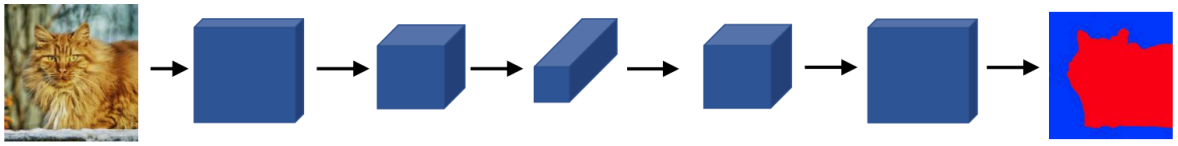
Per-pixel class labels



1. Car
2. Building
3. Road

Segmentation Map

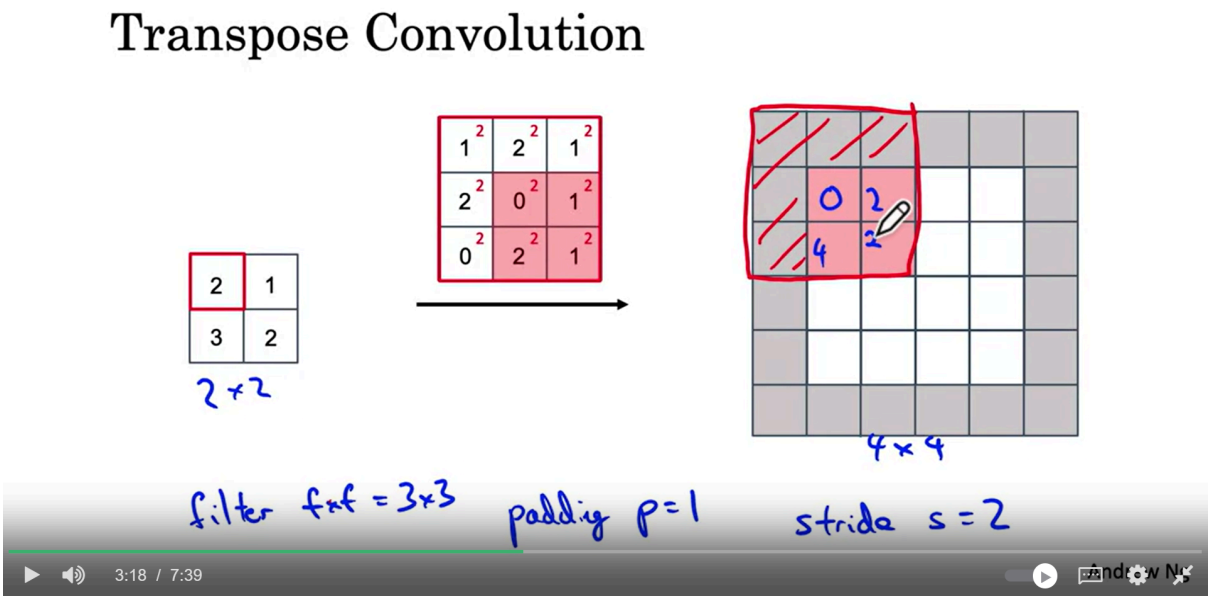
Semantic Segmentation network - The fully connected layers are replaced by convoluational layers



2nd last layer is of same shape as the image so as to get the processed image reverted

Need for increase in size of 3rd last layer *for* the 2nd last layer ,

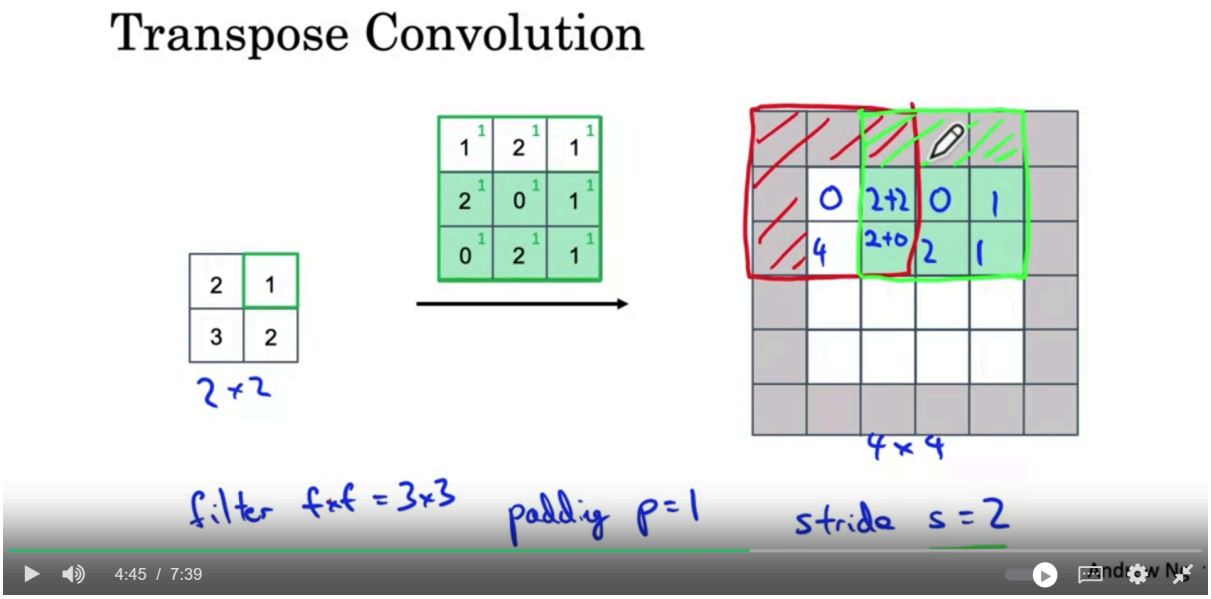
Following is the solution for Expanding the size



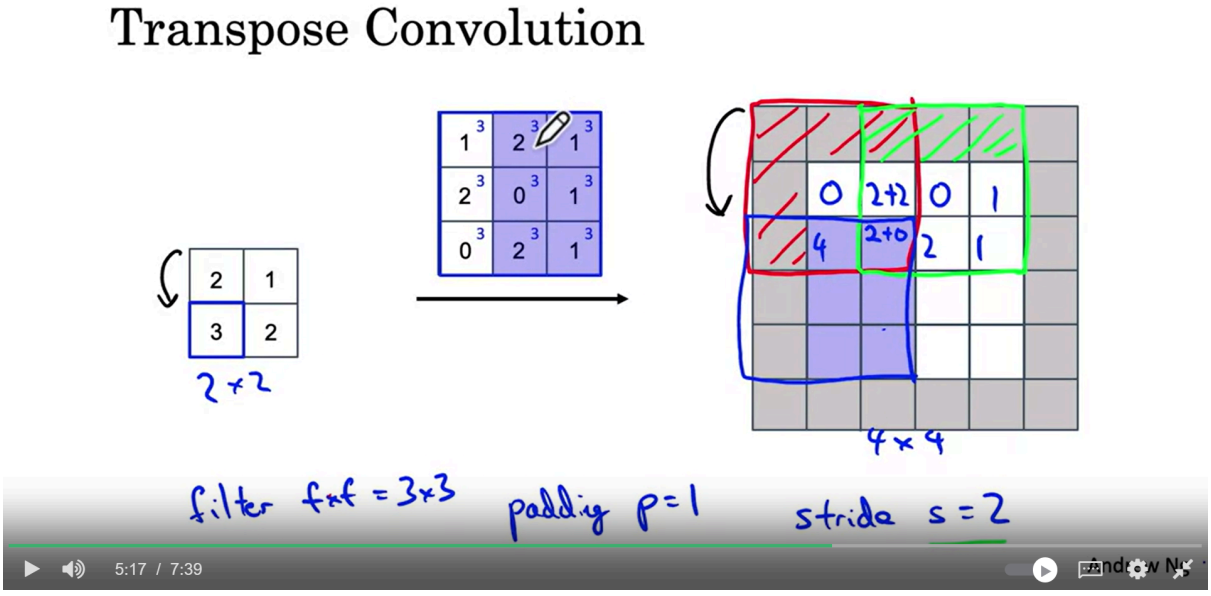
Take 2*2 mx and a filter with f=3

Consider the 1st ele of the input ,2 and multiply it element wise with the filter

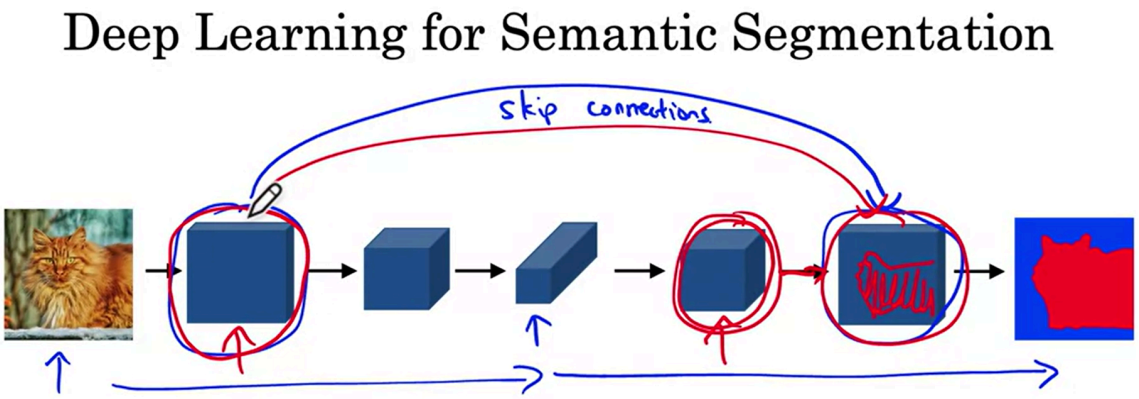
the columns or rows where the output is padded is ignored in the filter during this multiplication



The overlap contains sum of elements from the different multiplications



Jump of 1in the input results in a jump of 2 in the output



Second part of this network denoted by the second horizontal arrow points to the part where transpose convolutions are used

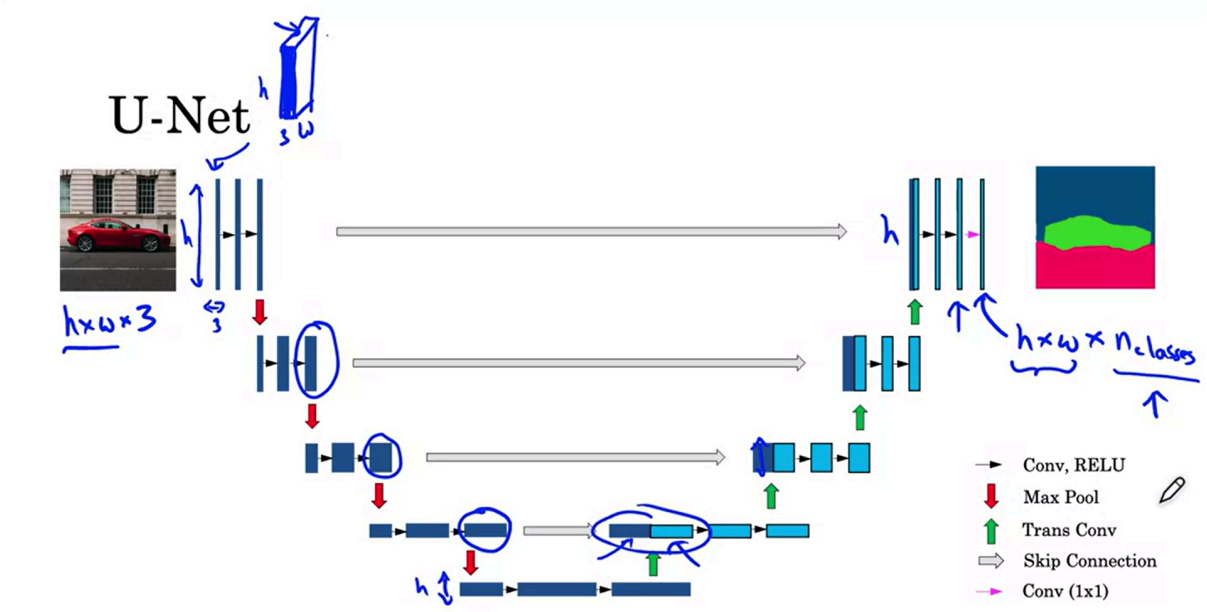
The skip connections in the neural network architecture allow the second last layer to have both -
High contextual information But low spatial resolution from its previous layer and
high spatial resolution information but low contextual information from the second layer

U-net

The most common network in computer version today
given and the following image

Not the type of network Denoted by the various arrows via the information given at the bottom right

Shape of the final output contains number of channels which equal the number of classes



The number of channels doesn't have to match between input image and final output