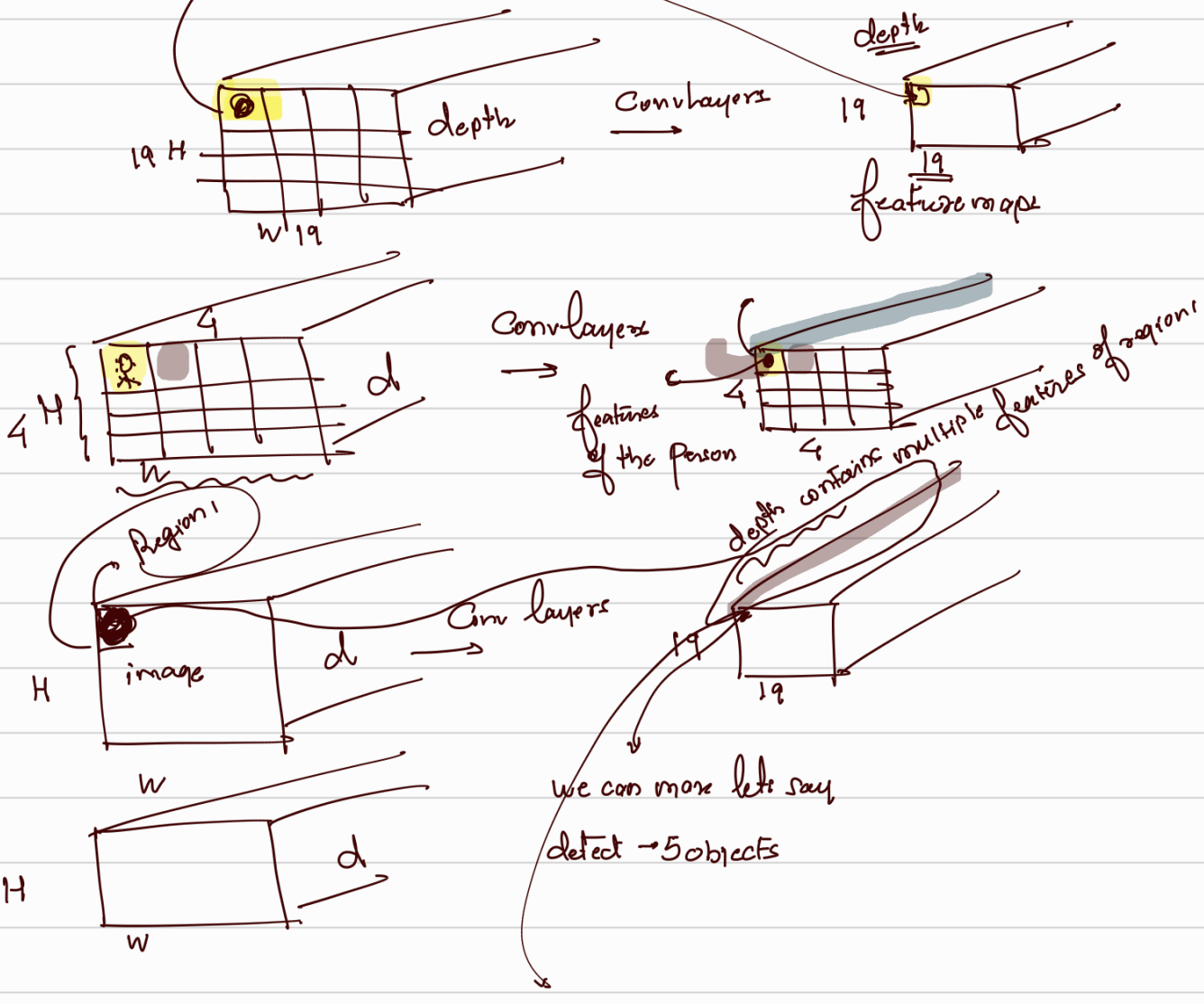
YOLO (You only Look once) (Yolov3)

① Object detection algorithms → RCNN, Fast RCNN, Faster RCNN

SS | | Conv

Region Proposals

within the same network we detect both the region proposals and the bounding box

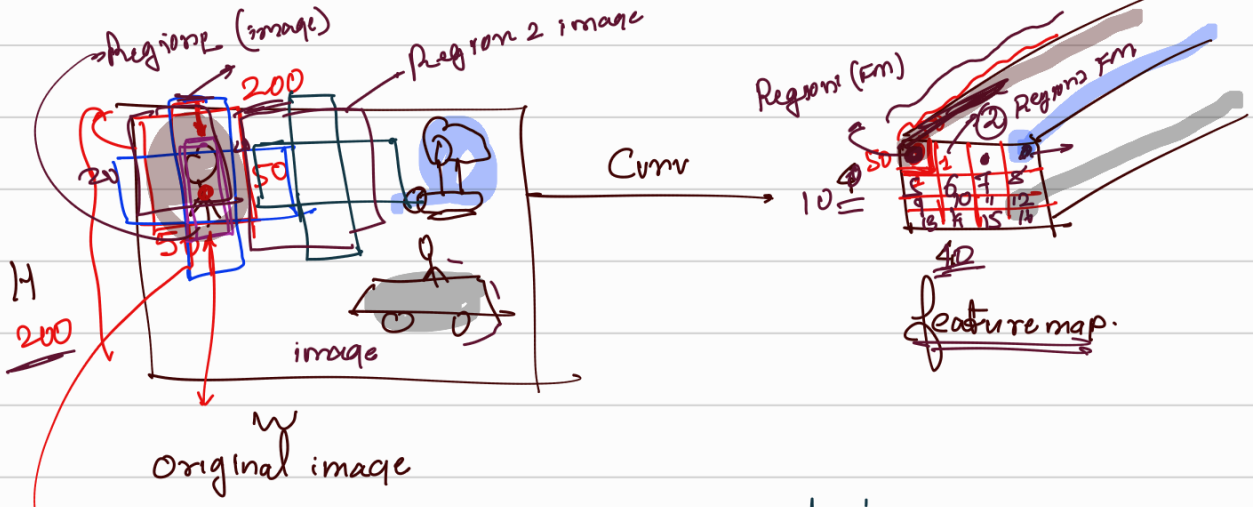


we can detect max 5 objects

→ Anchor Box 1 → Anchor Box 5

→ Anchor Box 2 → Anchor Box 4

→ Anchor Box 3



at every 50x50 Grid, we can actually detect here say

2 objects

→ Horizontal object

→ Vertical object

Which Anchor Box matches has high IoU with the pink Box

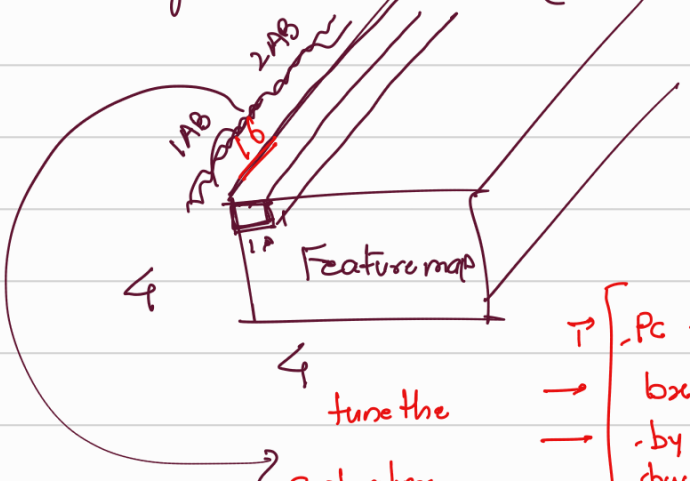
my feature map has to learn that in Region1 FM

→ we have a Person (Vertical object)

→ Region 2 we don't have anything

→ Region 3 does not have anything

→ Region4 has a tree (Vertical object) and a person (Horizontal)



C_1 → Person

C_2 → Tree

C_3 → Car.

tune the anchor box centres

tune the anchor box height and width to match the actual ground truth

Pc → Y/N (1)

bx

by

bw

bh

Cp → 1

C_2 → 0

C_3 → 0

dimensions of the Pink box → Vertical box

Pc → 0 → horizontal

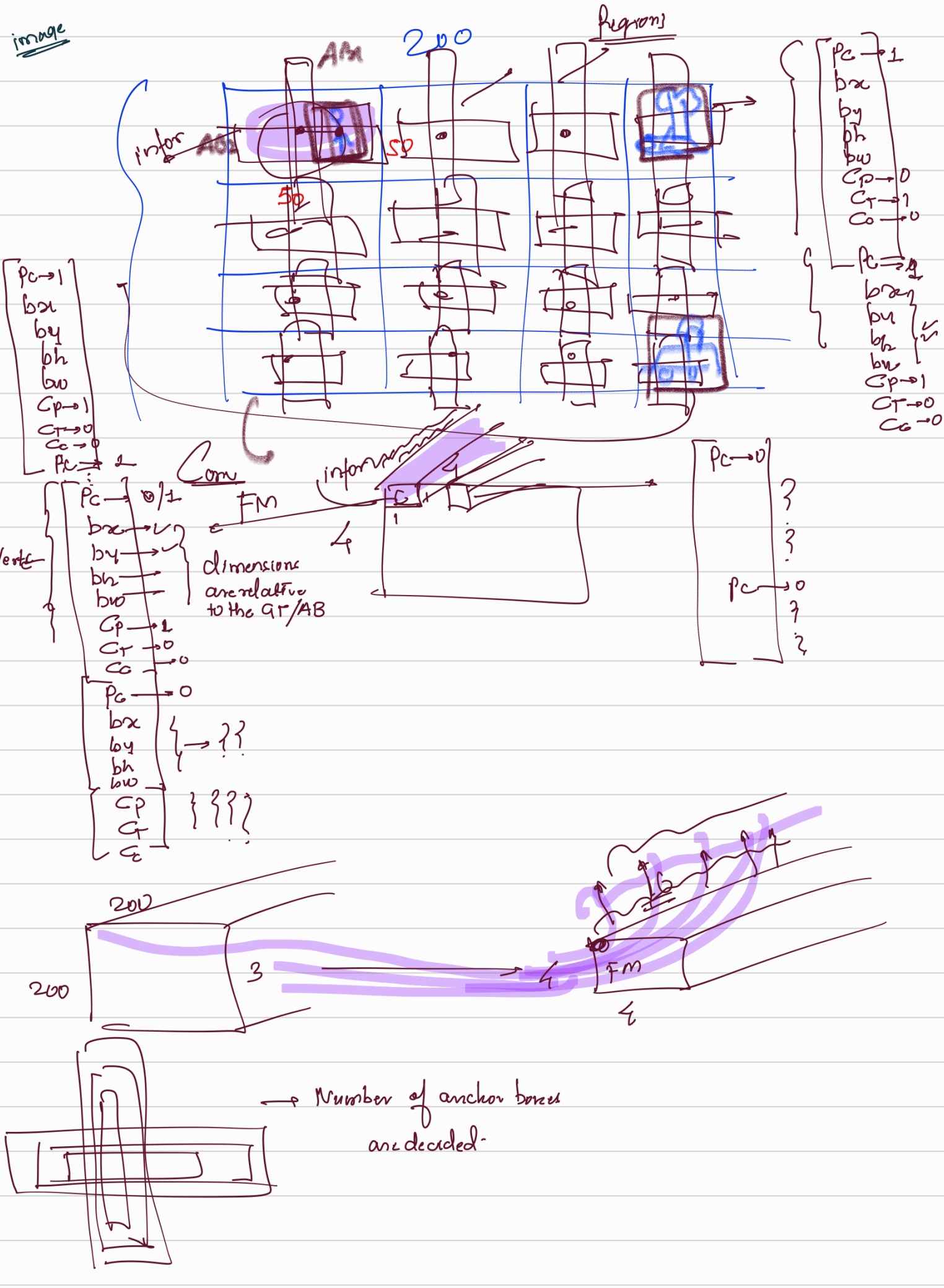
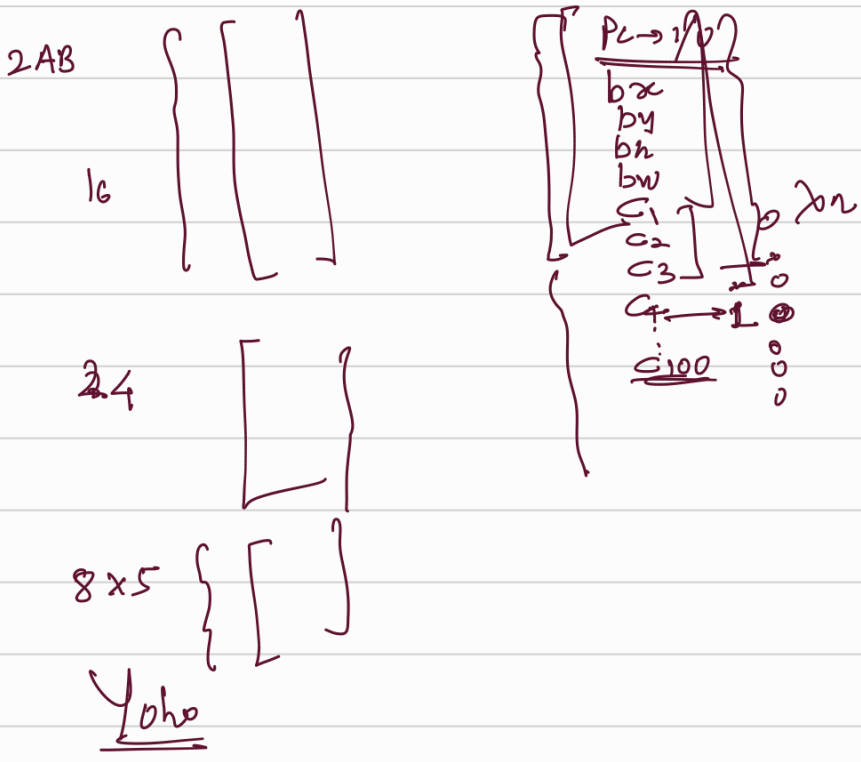
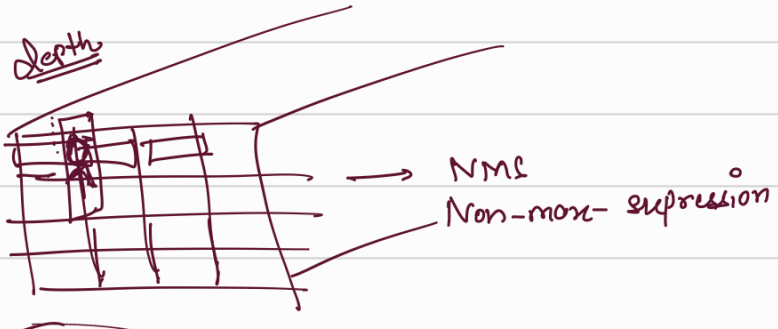


image
AM
200
Regions
infor
50
50
Pc→1
bx
by
bh
bw
Cp→0
CT→1
Co→0
Pc→1
bx
by
bh
bw
Cp→1
CT→0
Co→0
Pc→1
Conv
FM
information
Pc→0/1
bx→
by→
bh
bw
Cp→1
CT→0
Co→0
Vertex
dimensions are relative to the GT/AB
Pc→0
bx
by
bh
bw
Cp
CT
Co
→??
{???
Pc→0
?
?
?
Pc→0
?
?
200
200
3
FM
4
4
→ Number of anchor boxes are decided.



Inference Time

image → Yolo model (Conv, MP, Conv, MP) →



Anchor Boxes → how many classes

Kmeans clustering

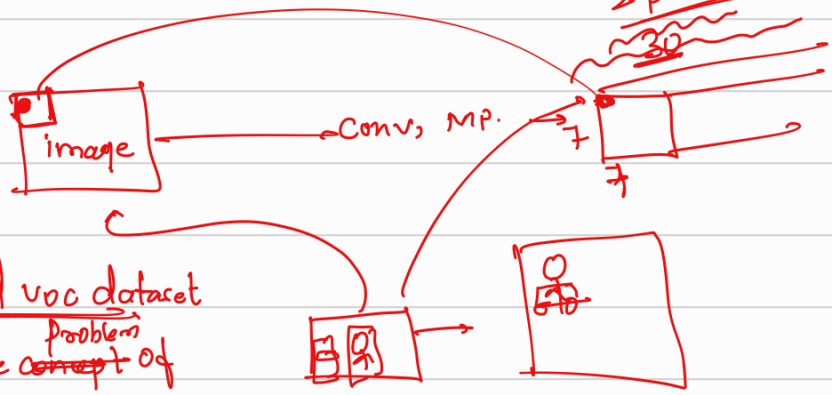
20 Class Probabilities

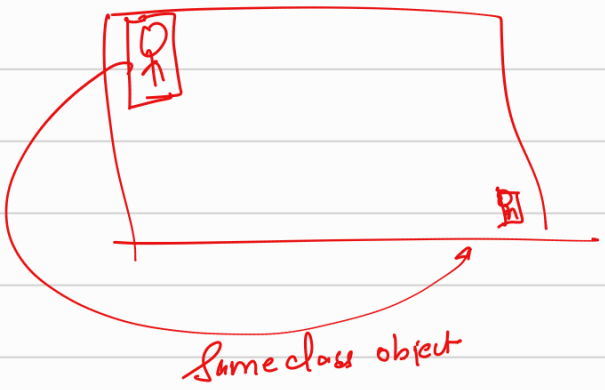
2 bounding boxes

2 Bounding Boxes x,y,bw,bh Pc x,y,bw,bh Pc

5 x 2

30





Yolo-1 could not match the capability to predict :

① overlapping objects

② Same objects different size

Yolo-V2

① Batch Normalization

② High Resolution images → Yolov1 → 224x224
448x448

③ Anchor Boxes → we define them by doing K-means clustering on existing ground truths of my training data

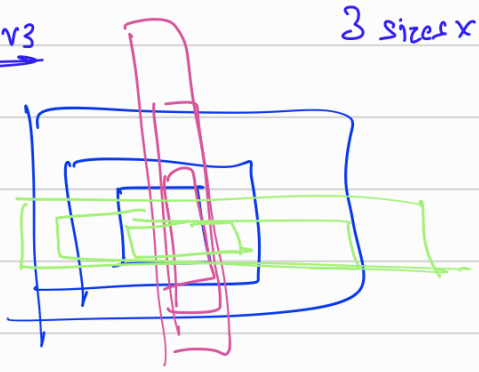
④ Feature map → 13x13

⑤ 19 convolution layers and 5 more Pooling layers

V2 faster, accurate

Yolo-v3 = 3x3

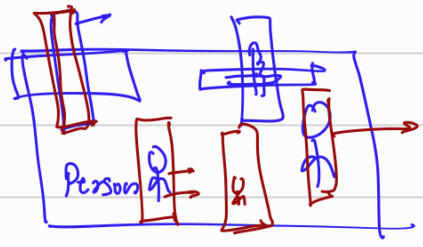
Yolo-v3



3 sizes x 3 aspect ratios

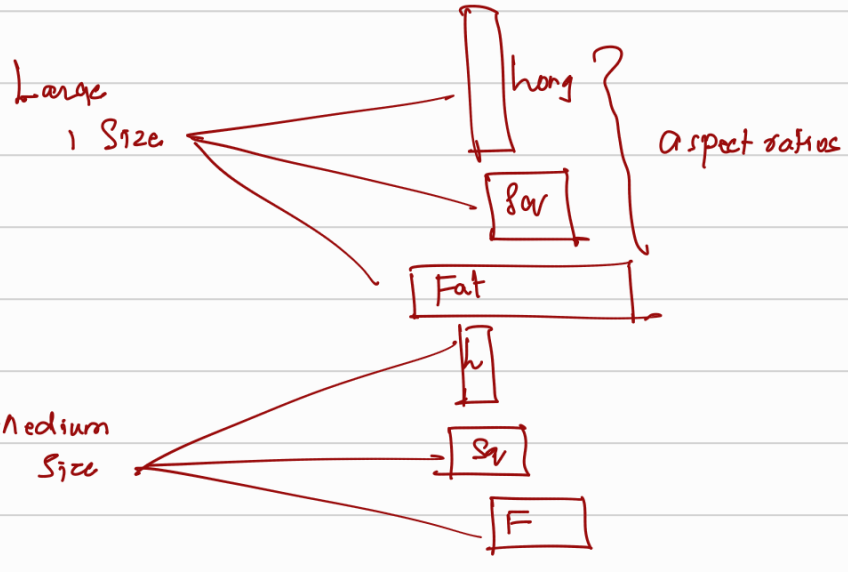
Anchor boxes → Kmeans Clustering

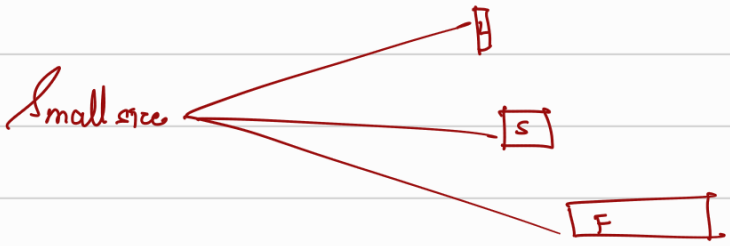
Yolo v3 model

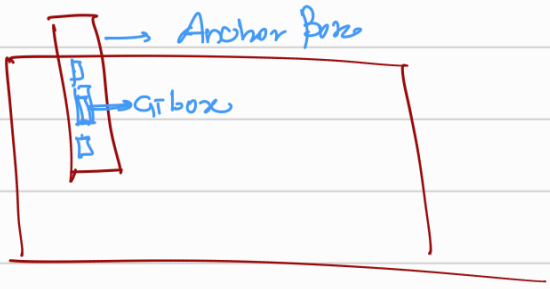


Only one size for anchor boxes →

Yolo-v3 → Sizes and aspects ratios for the anchor box



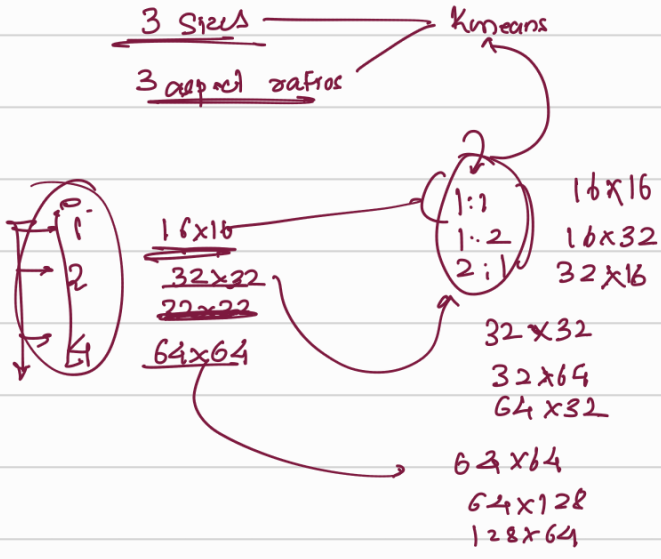


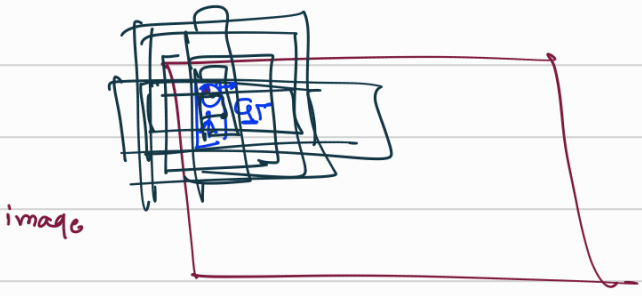


Yolov3 model → 3x3 Anchor Boxes

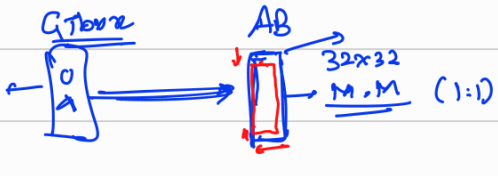
3 sizes , 3 aspect ratios

every grid cell will have 9 anchor boxes to fit in the ground truth.

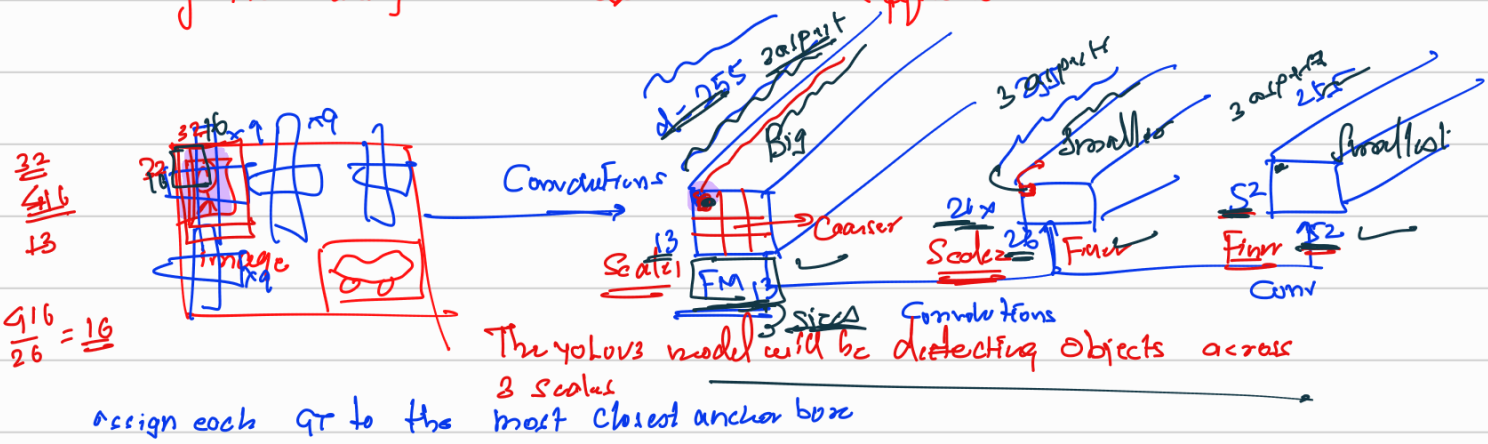




Assign the GT box to the most closest anchor box.



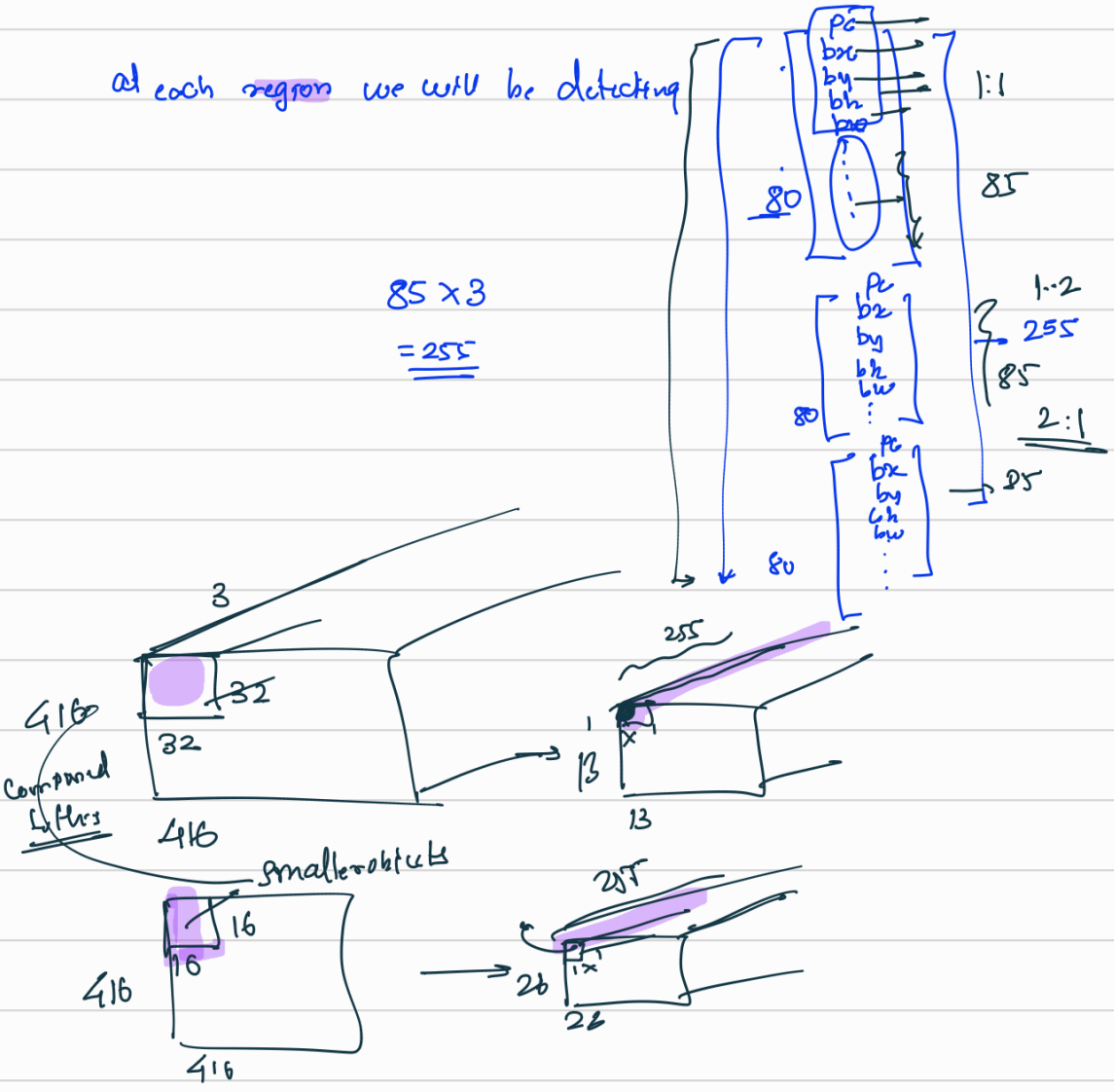
During the training we have to ~~detect~~ train the above difference



The YoLov3 model will be detecting objects across 3 scales

Assign each GT to the most closest anchor box

YoLov3 was trained on a dataset which has 80 classes.



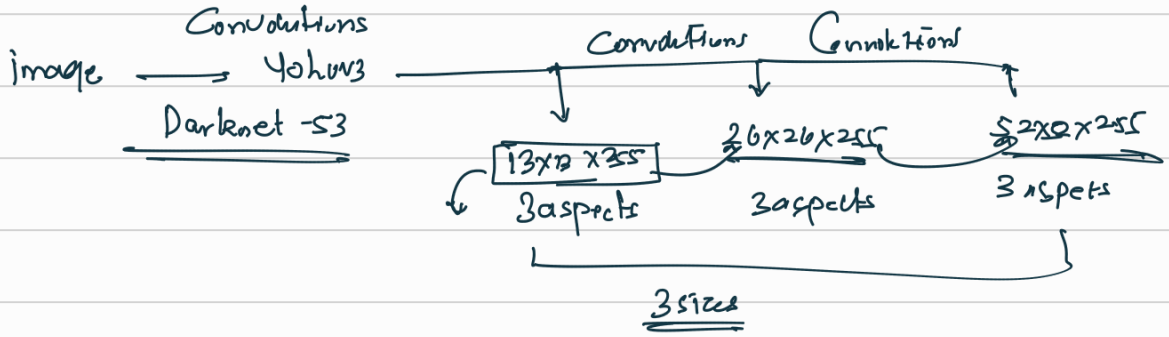
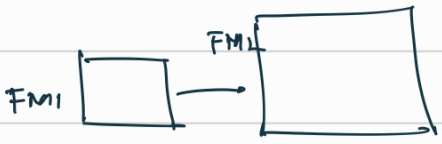


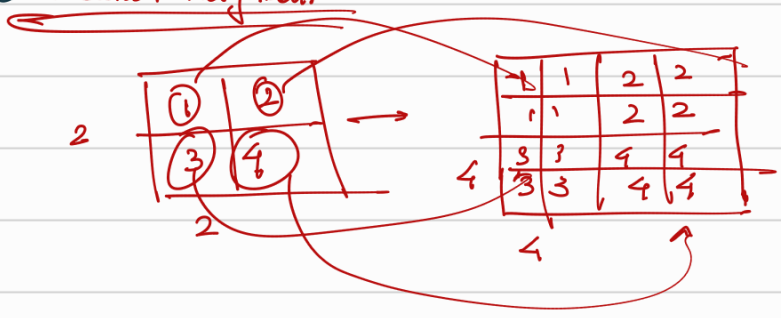
image → Convolutions → Downsample

FM → Convolutions → Bigger FM

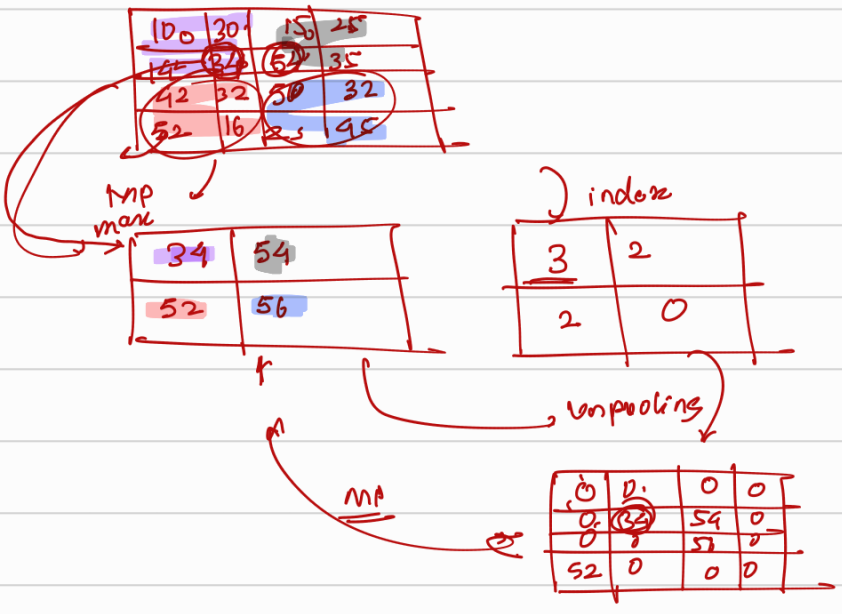
Upsampling (Transpose convolution)



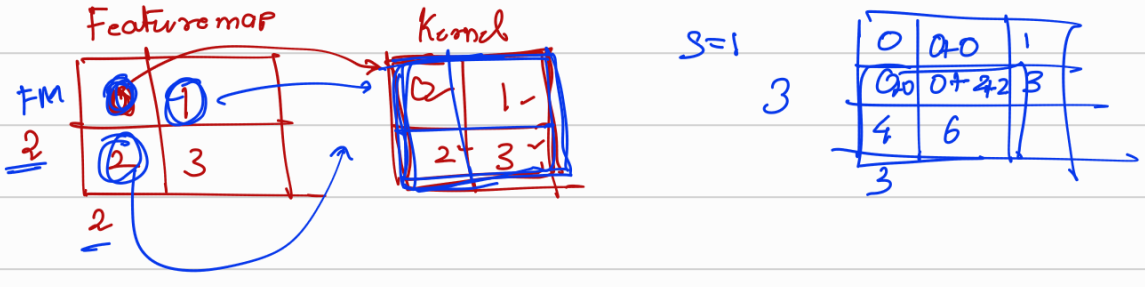
① Nearest neighbour

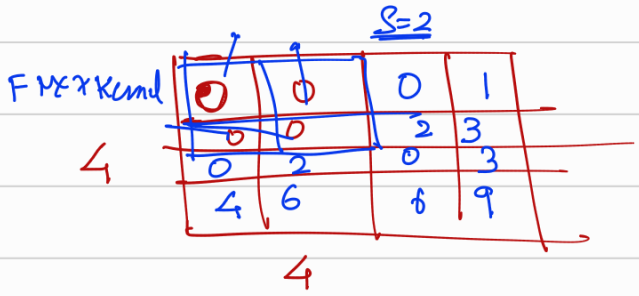


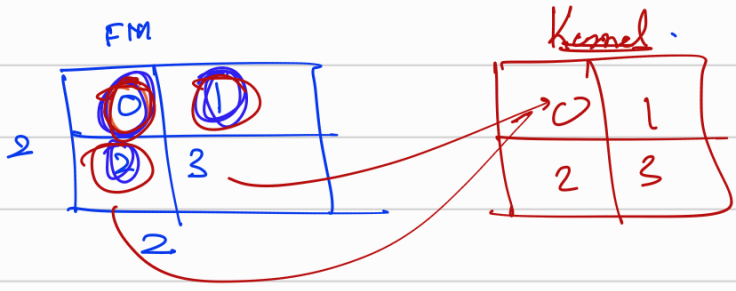
② Unpooling



③ Transpose convolution



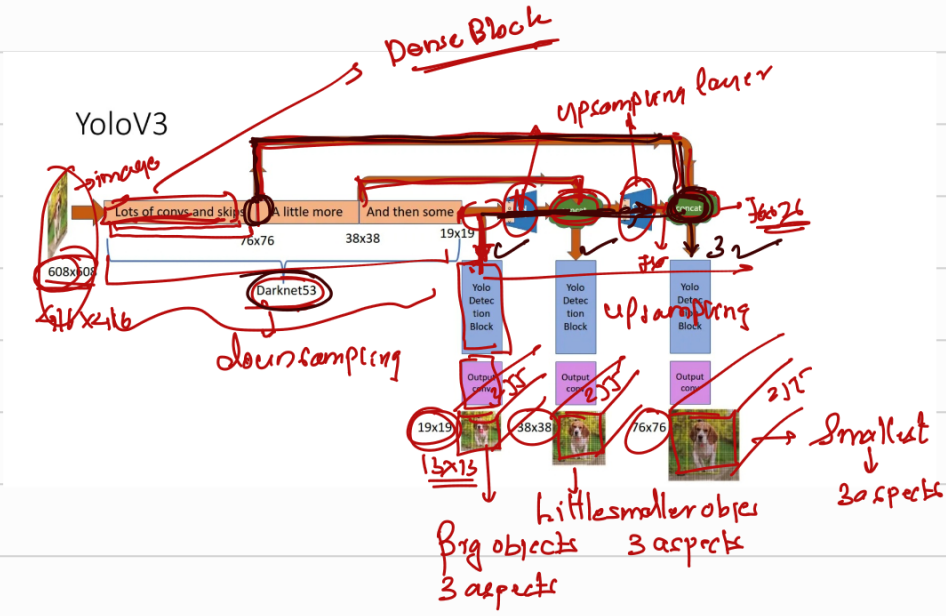


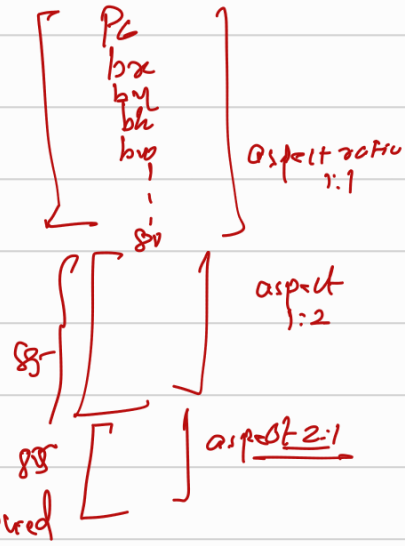


Output map

0	0+1x0	2
0+0	0+2+1+2+0	3+3
4	6+0	9

Transpose convolution is basically for upsampling your feature map





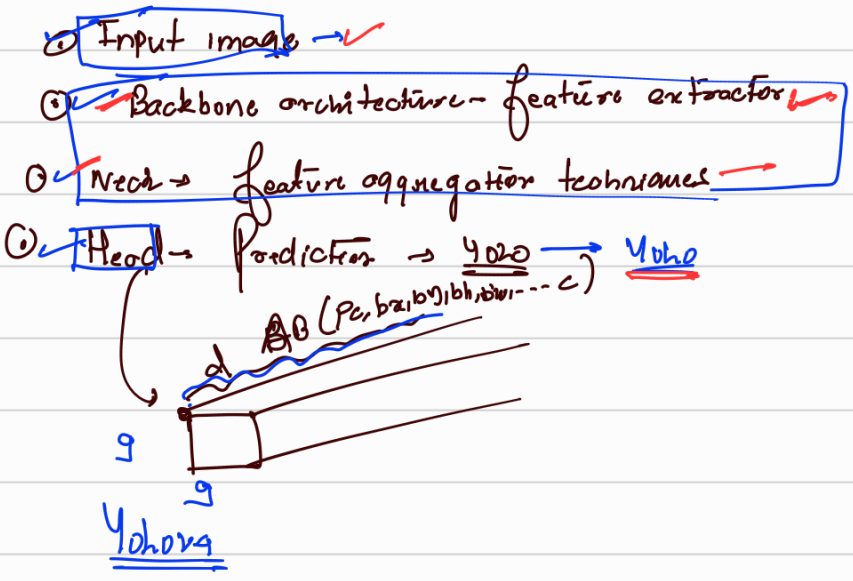
Yolov3 architecture is nothing but a series of convolutions applied along with certain skip connections to generate output at

19x19 (big objects) → upsample → conv → 38x38 → upsample → 76x76

YOLO-v4 .. v7

Yolov4 model

Building Components of an object detection technique:



Backbone architecture:

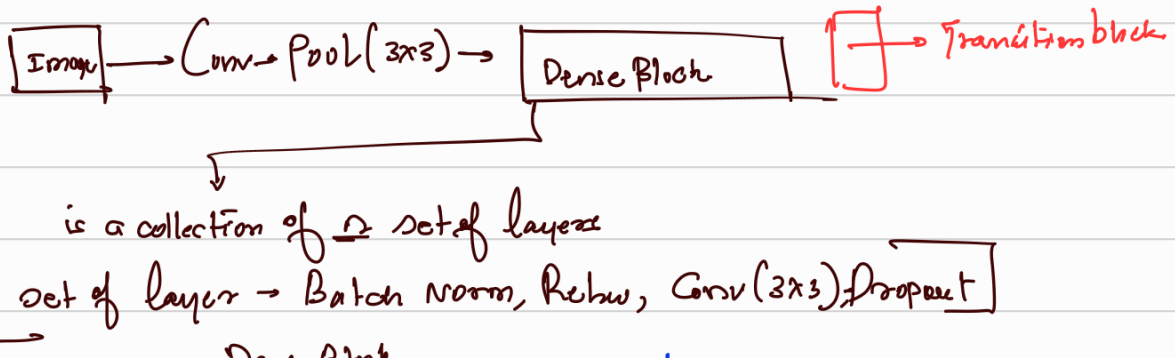
① DenseNet, CSPNet, CSP Darknet-53 → Yolov4

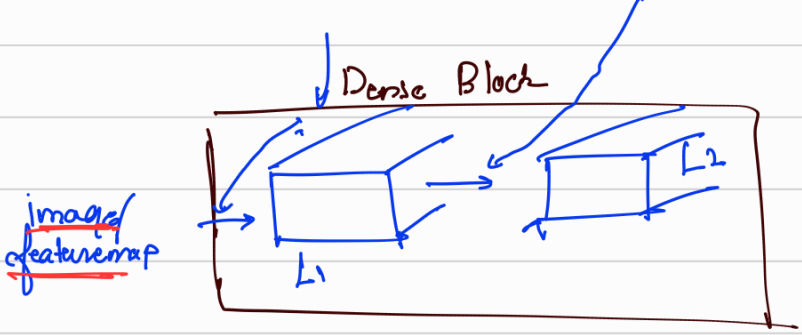
Neck architecture:

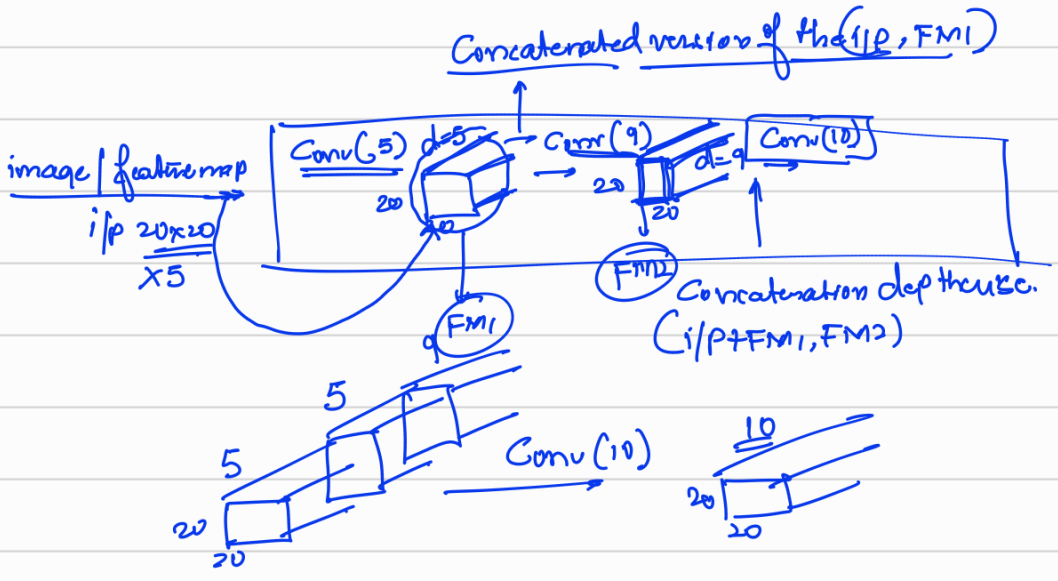
→ FPN → SPP (Spatial Pyramid Pooling) → PAN → Yolov4

Feature Pyramid network

Dense Net (Backbone architecture)





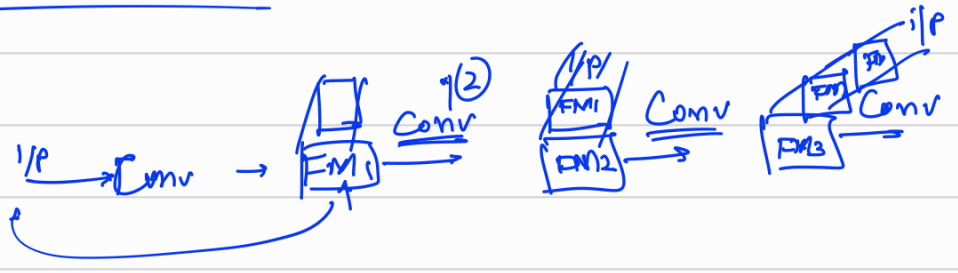


The dense block every func f apply a [Conv.]

we apply always to the concatenated version

→ i^{th}

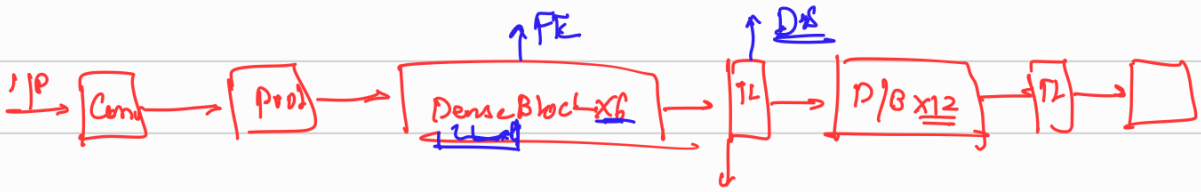
[FM$_{i-1}$, FM$_{i-2}$ .... i/p]



this is a very computationally expensive phenomenon

[Cross stage Partial Network]

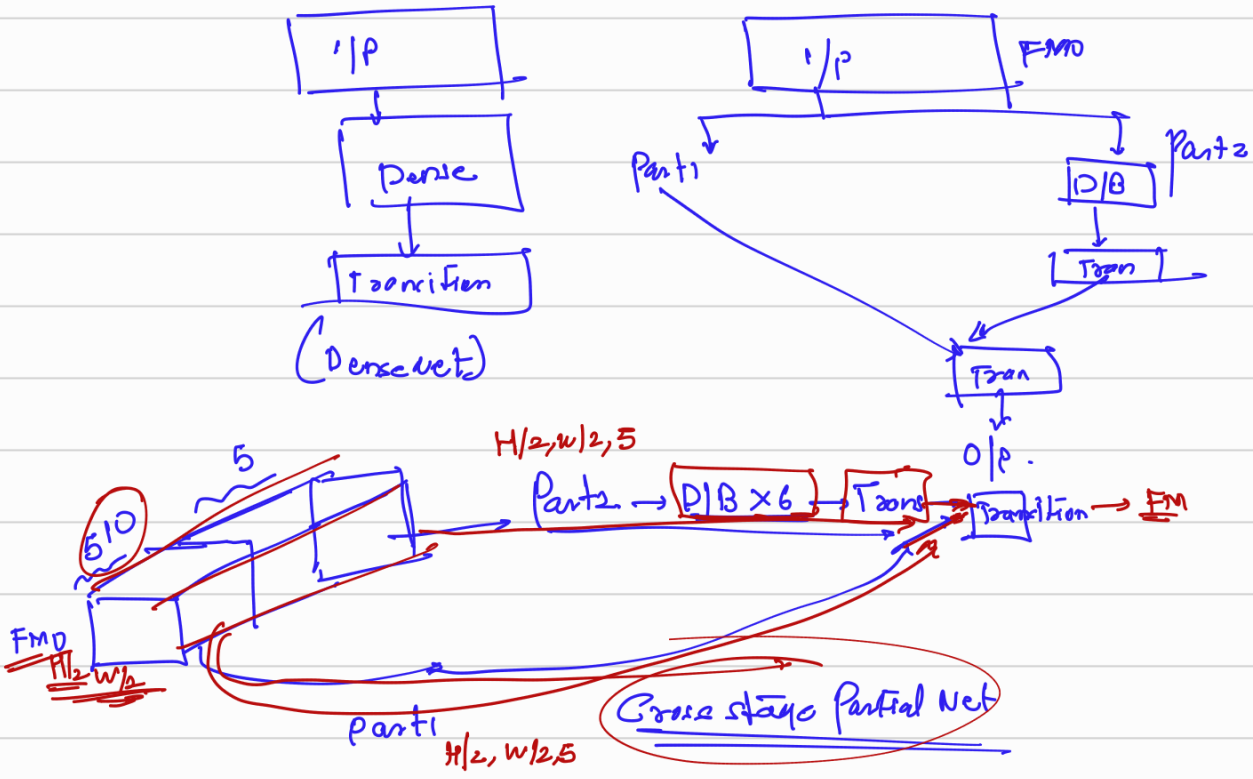
Transition Block



downsampling

TL → Batch norm, Relu, Conv(1x1), D/O, Pool(2x2)
L1

Cross Stage Partial Network



Yolov3

Mish activation → Yolov4

Backbone Yolov3 ——— image ——— ..... D/B — D/S — Yolo

Backbone Yolov4 ——— image ——— D/B CSP → CSP → CSP
↓
Mish activation