YOLO [You only look once]

[used for advance object Detection]

- It is mostly used in real time object detection.

Ex:- Traffic cameras.

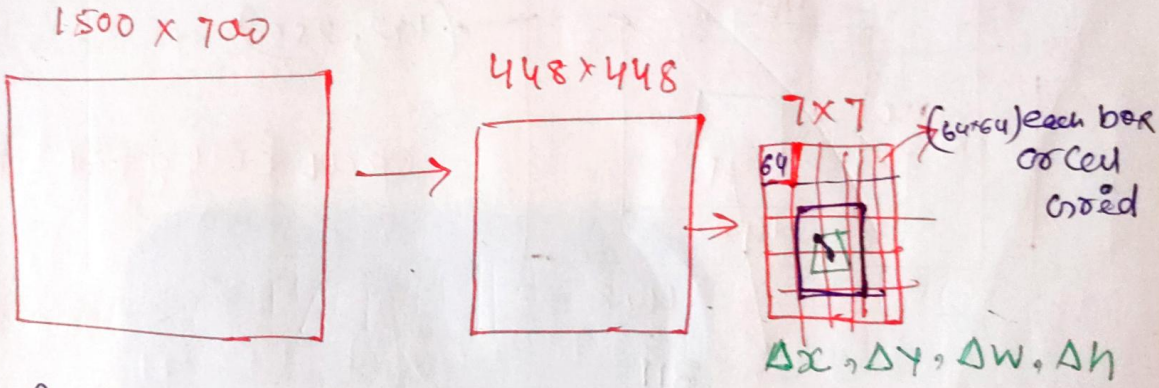
Use of YOLO → ① Different way of coding

② Real time object Detection

③ One shot Training by using Regression

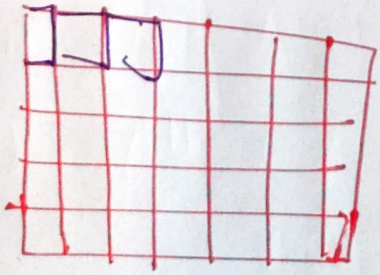
④ Losses used (BB, object , classes)

YOLO V1 Architecture ~~Algorithm~~



① Resize the image to (448x448) and $\frac{448}{7} = 64$

② divide in 8x8 as (7x7).



7x7 BB

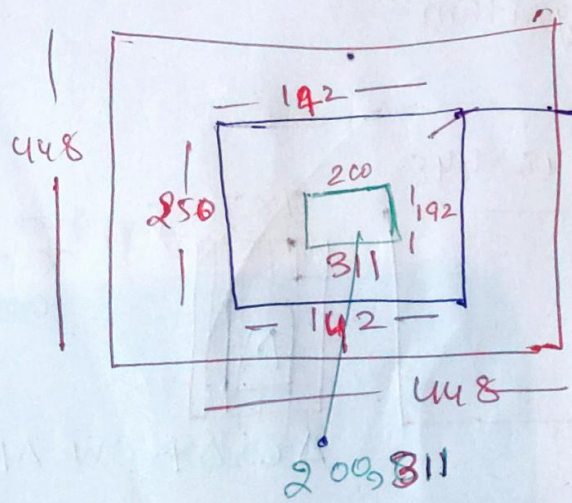
→ ~~for~~ each cell grid → [Δx, Δy, Δw Δh ci.] + P(20 class) ← one hot encode

5 + 20 ← (Pascal data)

= 25 parameters.

	Δx	Δy	Δw	Δh	ci	f	p (20 classes)
A_1	0	0	0	0	0		(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
A_2	0	0	0	0	0		0
A_3							
A_{31}	0.31	0.7	0.1	0.5	1		[0 0 0 1 0 0 0]
A_{40}	0.4	0.8	0.6	0.7	1		[0 0 0 0 0 0 1]
A_{49}	0	0	0	0	0		
A							

• Ground Truth.



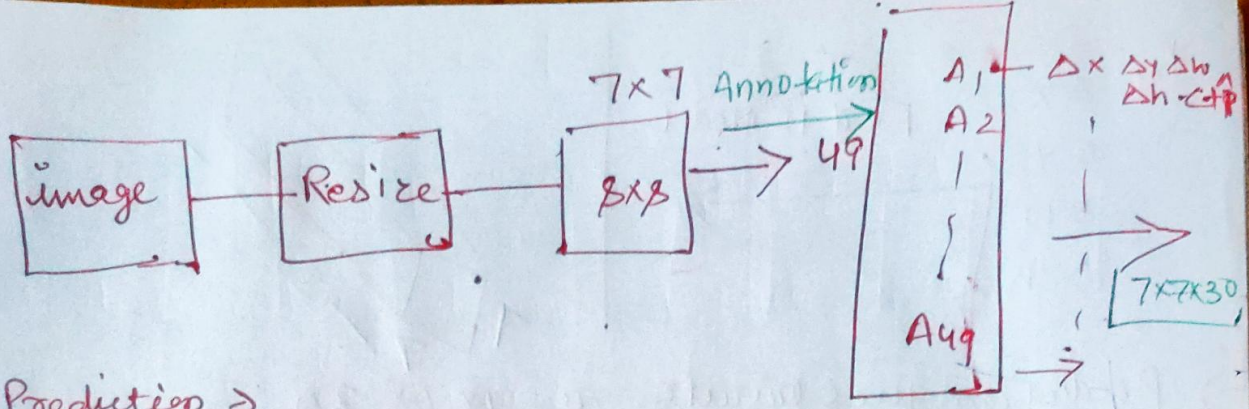
• each cell grid = 64

$$\Delta x = \frac{200 - 192}{64} = x_1$$

$$\Delta y = \frac{311 - 256}{64} = y_1$$

$$\Delta w = \frac{142}{448}$$

$$\Delta h = \frac{250}{448}$$



Prediction →

Take 2 class / output

$\Delta x\ \Delta y\ \Delta w\ \Delta h\ .\ \Delta C_1\ \Delta C_2 + P_1 P_2 P_3$

C = confidence score
P = Probability

Training

1. Take image as input
2. Resize image into (448×448)
3. Convert image into 7×7 format as (8×8)
4. Make grid cell as

20 + 5(Δx Δy Δw Δh ΔC)

(P)

Prediction

1. Image is in 7×7×30 format
2. 5 + 20

(0.1 0.2 -- 0.8 ---) probability

Δx, Δy, Δw, Δh, ΔC } helps to choose Bounding Box

Δx, Δy, Δw, Δh → x, y, w, h

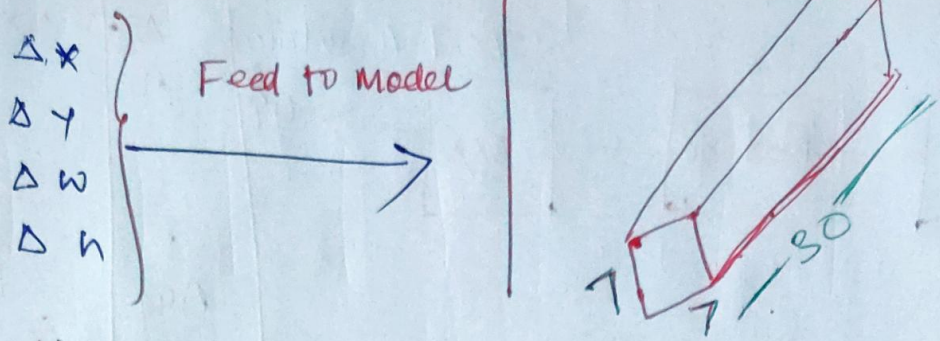
$x = \Delta x * 64 + x_a$

$y = \Delta y * 64 + y_a$

(x_a, y_a) ⇒ Coordinates of Grid cell of top left Corner

$w = \Delta w * 448$

$h = \Delta h * 448$



→ Prediction gives results as 7×7×30

Grid cell full Image

Δx Δy Δw Δh C | Δx Δy Δw Δh C | P_1 P_2 ... P_{20}

5 BB_1+C | 5 BB_2+C | 30 class probability

→ Bounding Box finded acc to confidence value (C)

→ which confidence value is large that Bounding Box is taken.

→ then Loss is find for accuracy of model.

YOLO (V_1) → 24 conv layers + 2 fully connected layers

Mean Avg. Precision (MAP) of FasterRCNN = 70%.

MAP of YOLO V_1 = 68%.

MAP of YOLO V_2 = 78% high accuracy

* Disadvantages of YOLO (V_1) =>

(1) It can only detect 49 objects.

(2) Less accuracy as 68% than YOLO(V_2) & FasterRCNN

(3) Fully connected layer is not present at the end.

⇒ Modification or improvement in YOLO V_2 over V_1

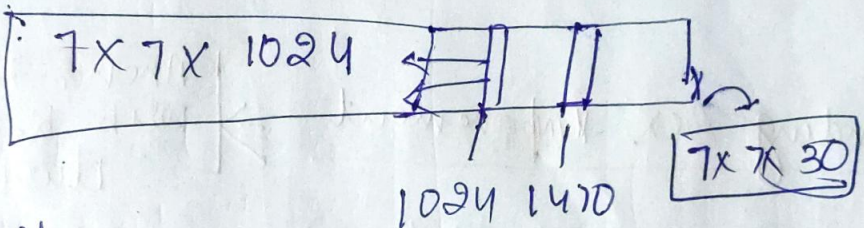
(1) MAP increases from 63 to 78.

(2) Apply Batch normalization to all CNN layers.
- By BN, improve map by 2%.
- overfitting Problem solved
- Regularization effect came.

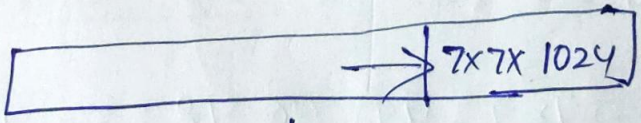
(3) High Resolution classifier used. due to this MAP increases by 4%.

High Resolution Feature MAP →

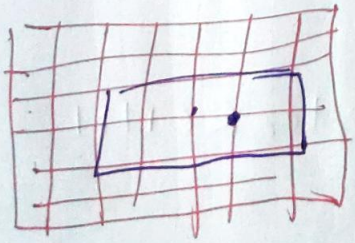
In YOLO V_1



YOLO V_2



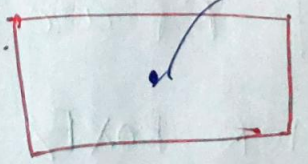
7X7



Centre will be at one Particular Point

odd x odd

14X14



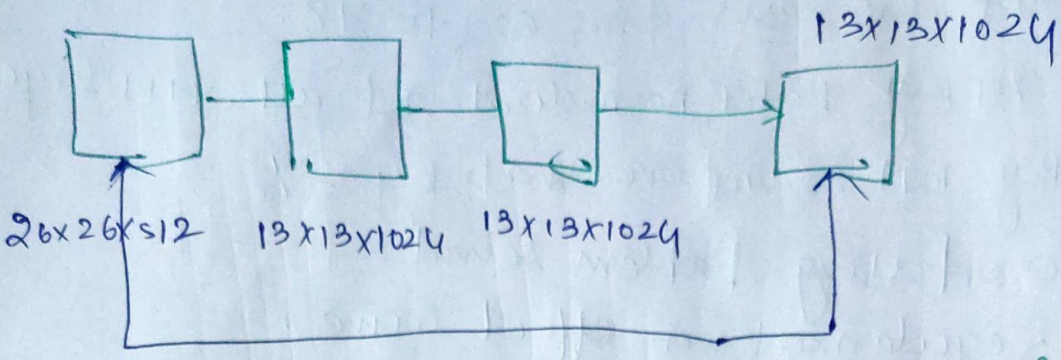
finding centre is not easy

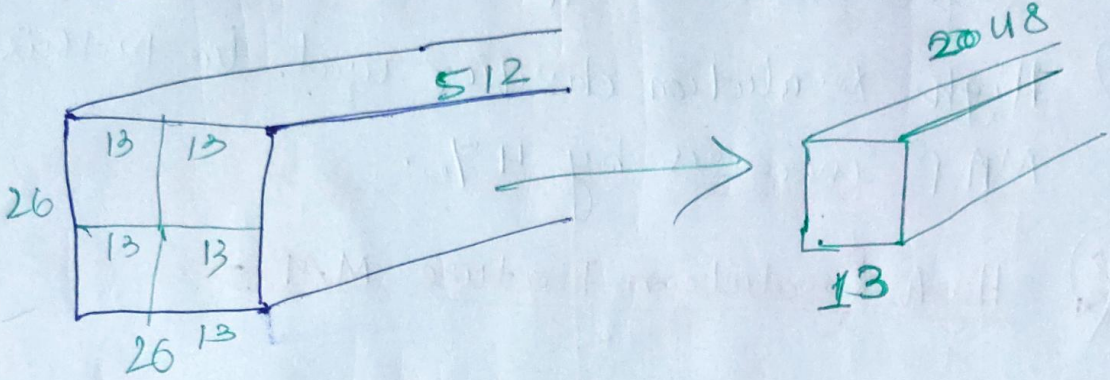
So, shift or resize as

even x even

13X13X1024

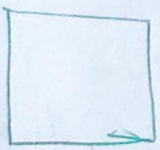
(4) Pass through Layer ⇒





(5) Bounding Box improvement ⇒ [Anchor Boxes with K-Means Clustering]

→ here Region Proposal (RP) are approx 98 or 49



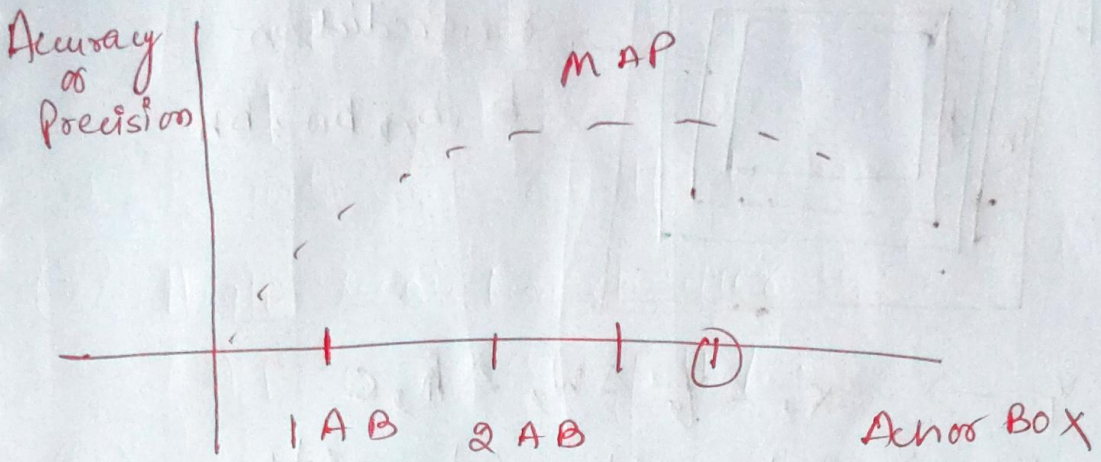
→ 7x7 = 49 or 98 or 169 or 338

so, no. of RP are less here.

- ANCHOR BOXES used in faster RCNN
  - 3 aspect ratio
- ANCHOR BOXES → Every Grid cell can have more than 1 object.

→ 9 anchor boxes present

so, K-Means clustering used on 5 Anchor Boxes for better result.



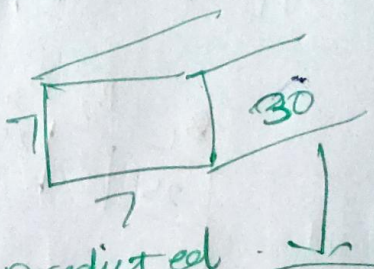
these 5 modifications which improved the model drastically from map of 68% to 78%

* ~~Predictions~~ →

(6) Different activation function used

* Predictions →

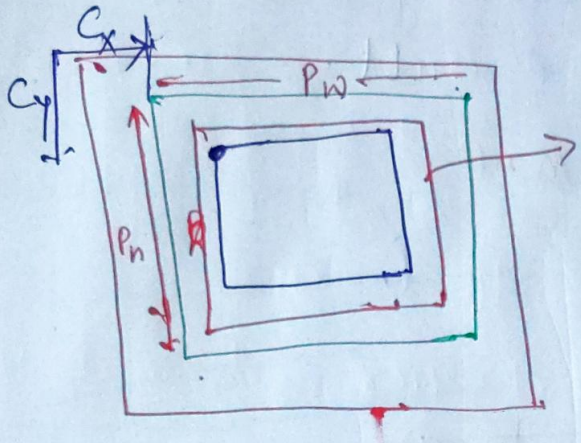
In YOLO V_1 → offsets



2 BB predicted and 1 BB chosen As per confidence value.

→ YOLO V_1 → Linear Activation fu^n used and range of o/p of (x, y) → $[-\infty, \infty]$.

YOLO V2 ⇒ Activation function + [Constraints]



→ for Prediction
b_x, b_y, b_w, b_h

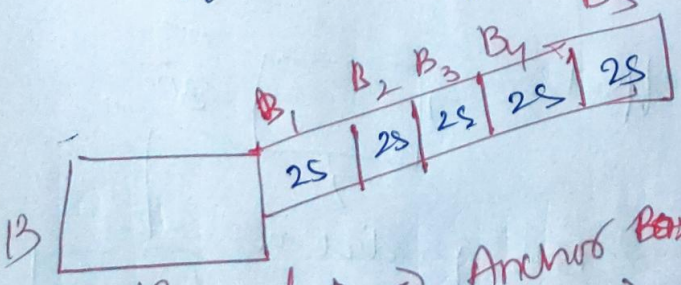
YOLO V_1 → $\Delta x, \Delta y, \Delta w, \Delta h, C$

YOLO V_2 → $\Delta t_x, \Delta t_y, \Delta t_w, \Delta t_h, C$

$$b_x = \sigma(\Delta t_x) + C_x$$
$$b_y = \sigma(\Delta t_y) + C_y$$
$$b_w = p_w \cdot e^{t_w}$$
$$b_h = b_h \cdot e^{t_h}$$

b_x, b_y, b_w, b_h

BB target value



Δt

(B → Anchor Box (AB))

[5 - Anchor Boxes used]

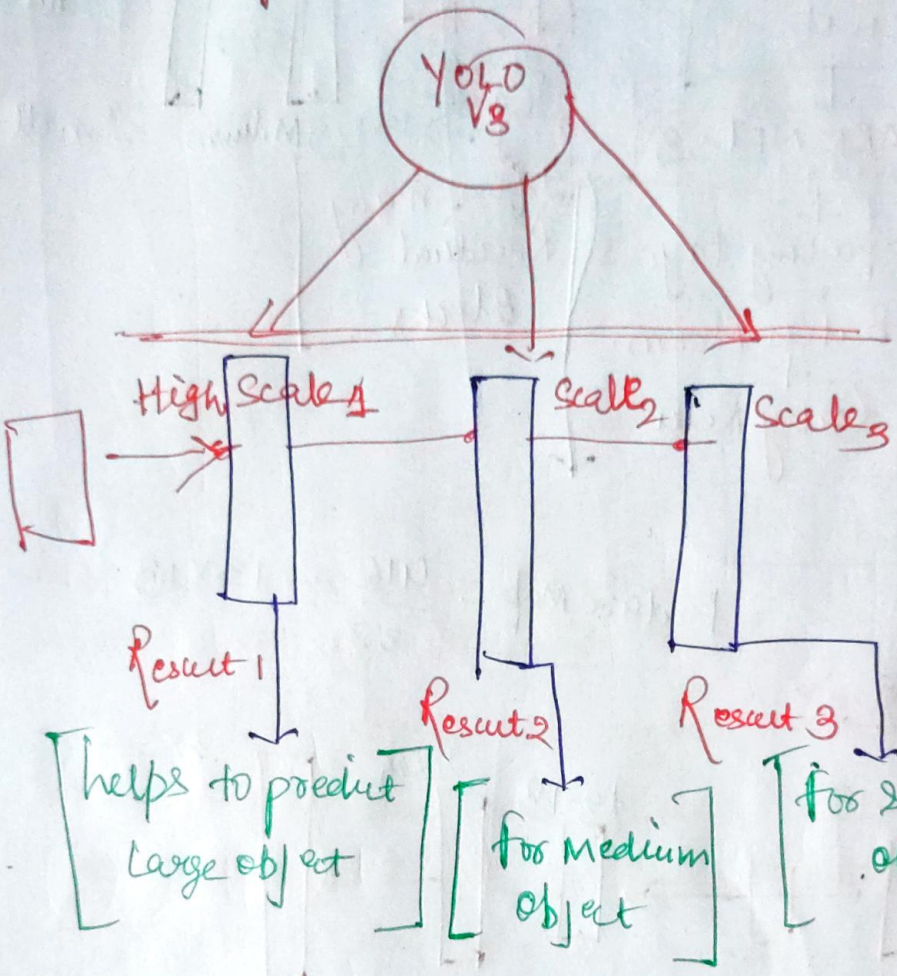
$\Delta t_x, \Delta t_y, \Delta t_w, \Delta t_h, C$, 20

125

[13 × 13 × 125]

YOLO V_3

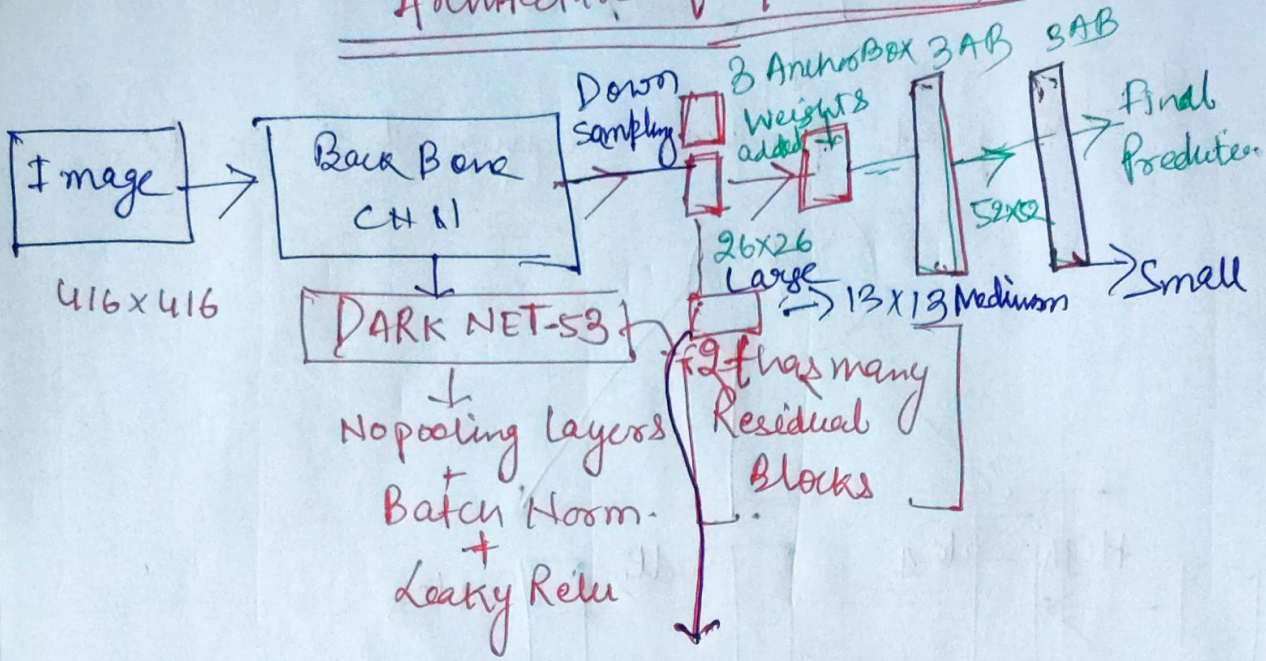
In this part prediction came of 3 parts.



YOLO V_2 → Back Bone CNN → uses Darknet-19

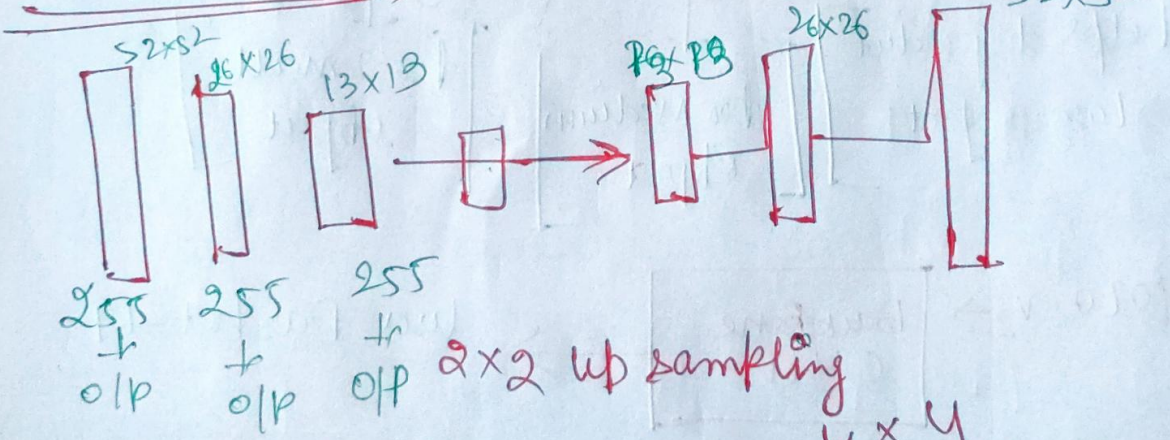
YOLO V_3 → Back Bone CNN → uses Darknet-53 + like 53 layers network

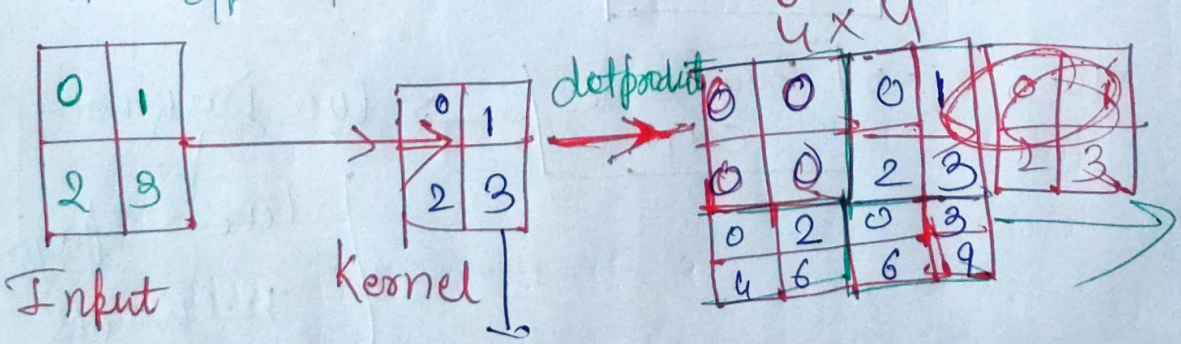
Architecture of YOLO V3 ⇒



$$\text{Feature map} = \frac{416}{32} = 13 \times 13$$

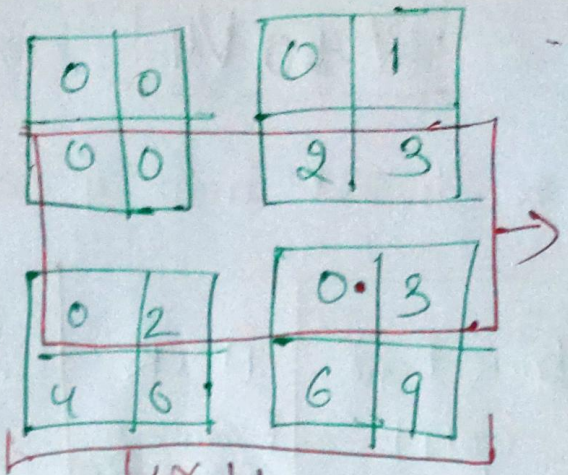
Up sampling ⇒

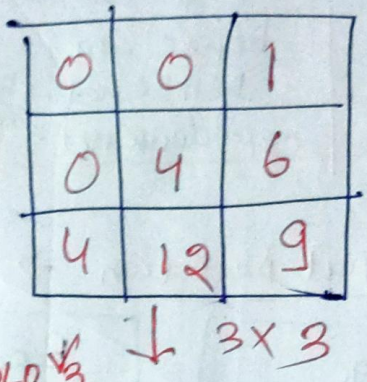




Transpose conv → Reverse conv
↓
upsampling

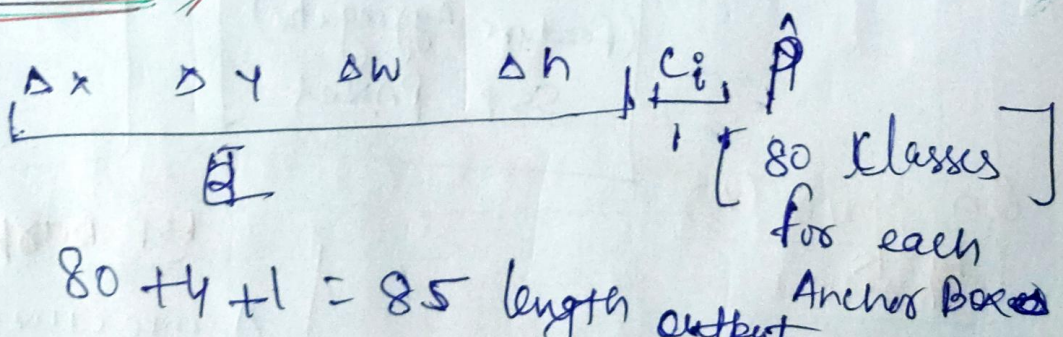
Add →





* In YOLO v_3, All Anchor Box sizes were predetermined based on the analysis of the dataset used for training YOLO v_3. They cover a range of object commonly found in the dataset, and the model learns to adjust its predictions based on these anchor Boxes during the training period.

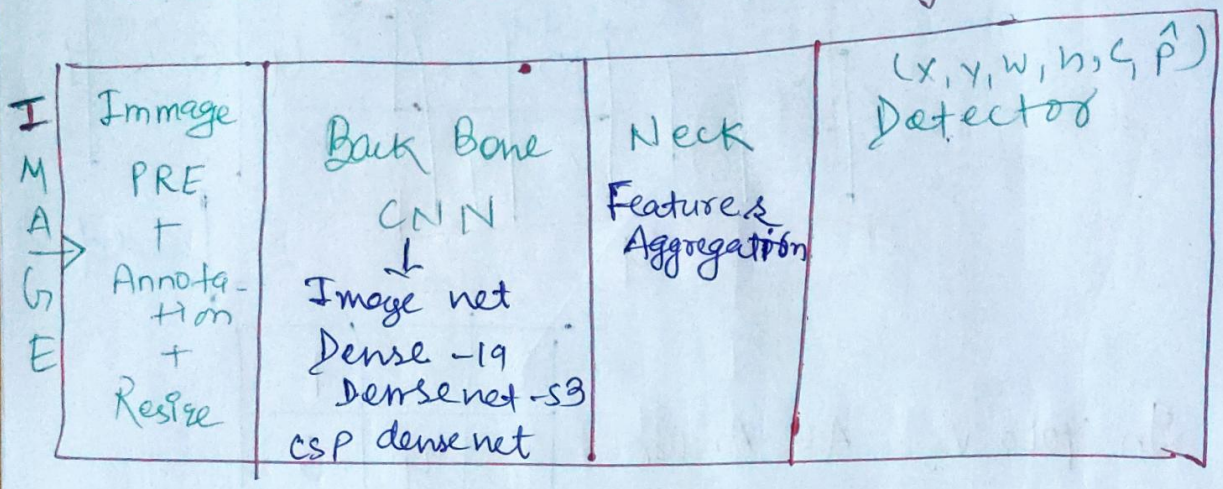
* Predictions ⇒



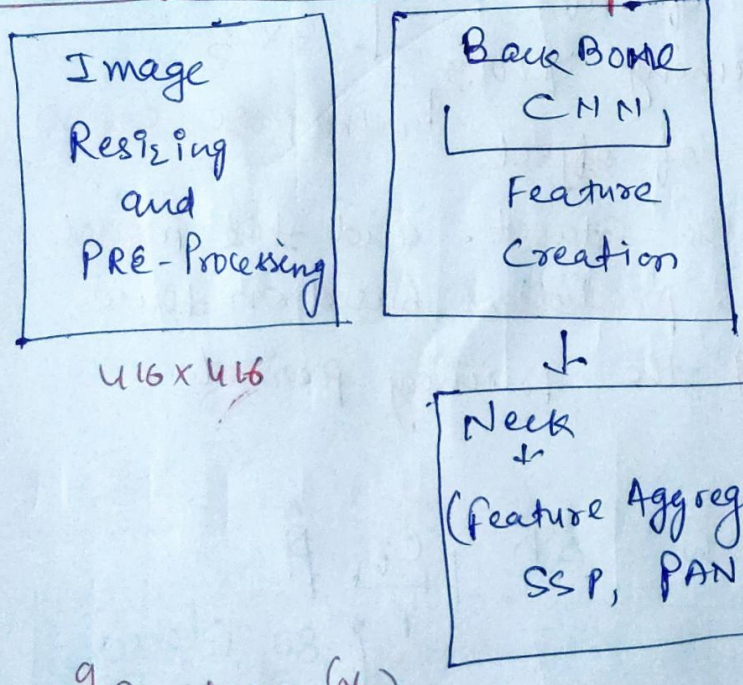
80 + 4 + 1 = 85 length output

YOLO V4

It will be divided into 4 categories.



In object Detection →



On YOLO (V_3)

Dark ~~Dense~~ Net

A convo NN that acts as a backbone for the YOLO V3 object detection.

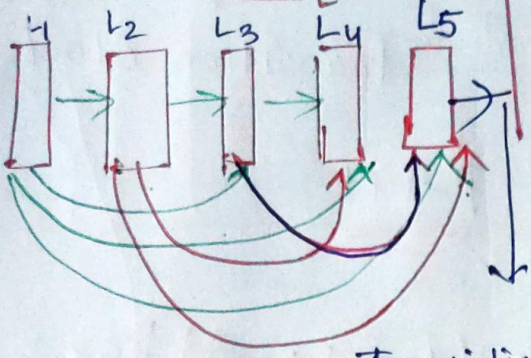
CSP ~~Dark~~ net

A convo CNN and backbone for object Detection.

→ Partition the feature map of the base layer into two parts then merge.

• In Darknet Leaky Relu used → Mish used as activation function

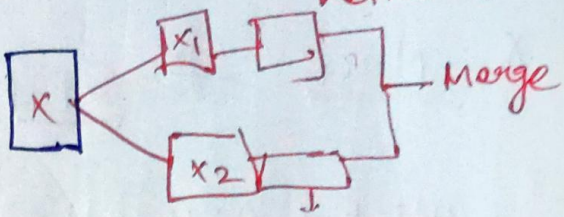
Dense Block [YOLOv3]



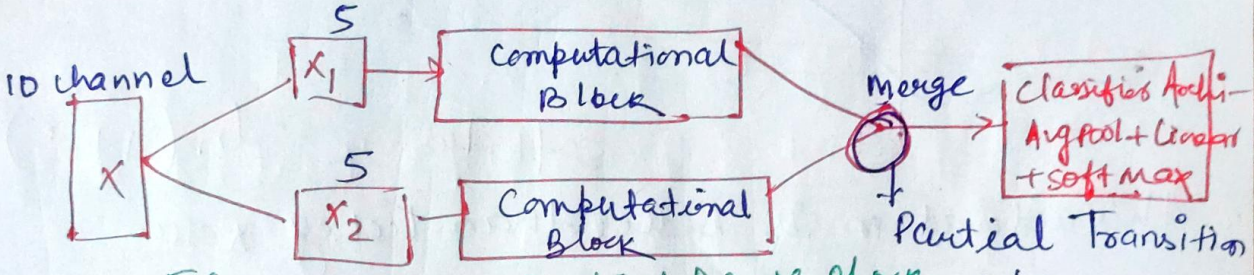
Transition Block
+
Down sampling

CSP Dense Block

(CSP - Cross Stage Partial network)



CSP Dense Block



[X = x1 + x2]

Dense Block →

We connect all layers (with matching feature-map size) directly with each other.

Partial Dense Block

10 Channels

Base Layer

↓

Dense Block

↓

Transition

↓

Down sampling

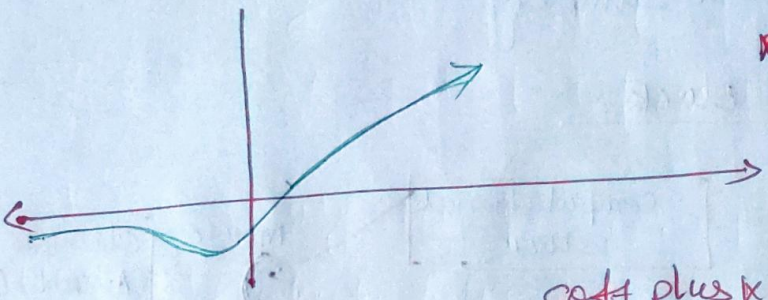
In YOLO V_4

* Back Bone CNN → CSP + Darknet + Dense Block + Transition Block

* Neck → SPP + PAN

* Mish (Activation)

→ Derived from soft plus → $\tanh(\ln(1+e^x)) \times x$



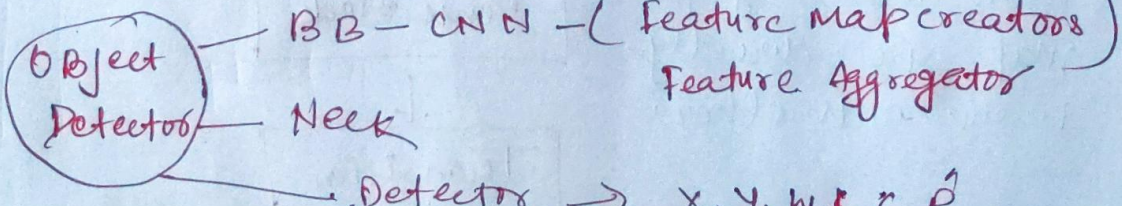
$x * \tanh(\text{softplus}(x))$

$\text{softplus}(x) = \ln(1+e^x)$

this activation is better than Leakey Relu.

→ In YOLO V_4, CSP Darknet used.

Neck [



We use typically two feet aggregator in your YOLO – V_4

(i) PAN (Partial Aggregation Networks)

↳ It is modified Version of FPN.

↓

(Feature Pyramid Network)

(ii) SPP (Spatial pyramid pooling)

* FPN (Feature Pyramid Network) ⇒

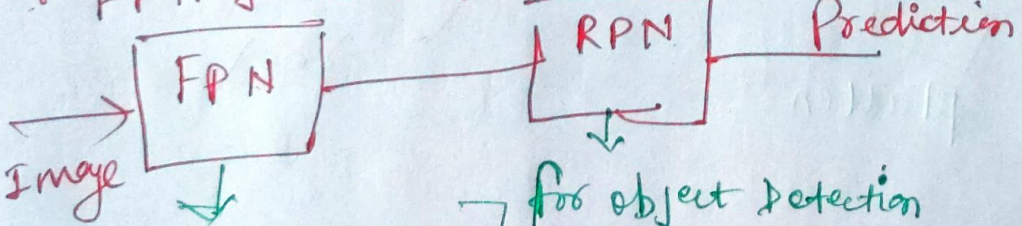
⇒ It is used for smaller object

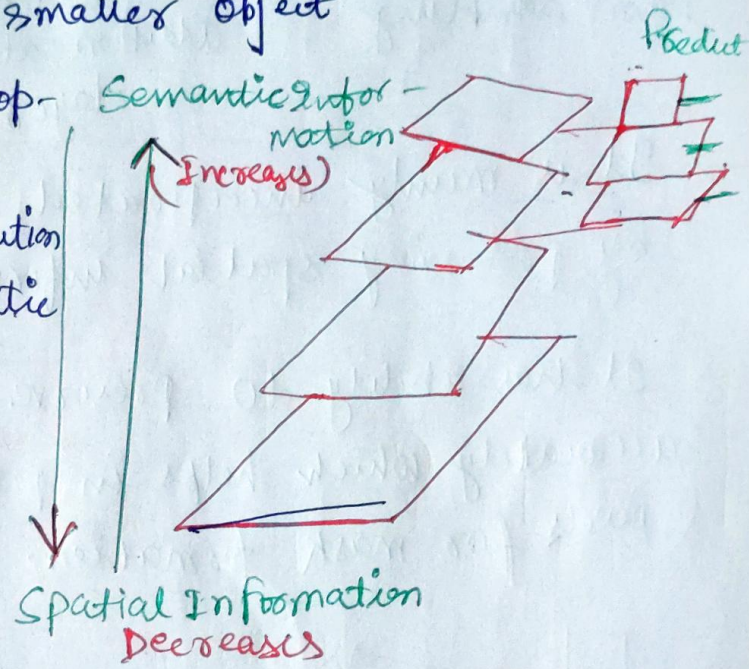
→ FPN provides a top-down pathway to construct higher resolution layers from a semantic rich layer.

→ FPN is not an object detector by itself.

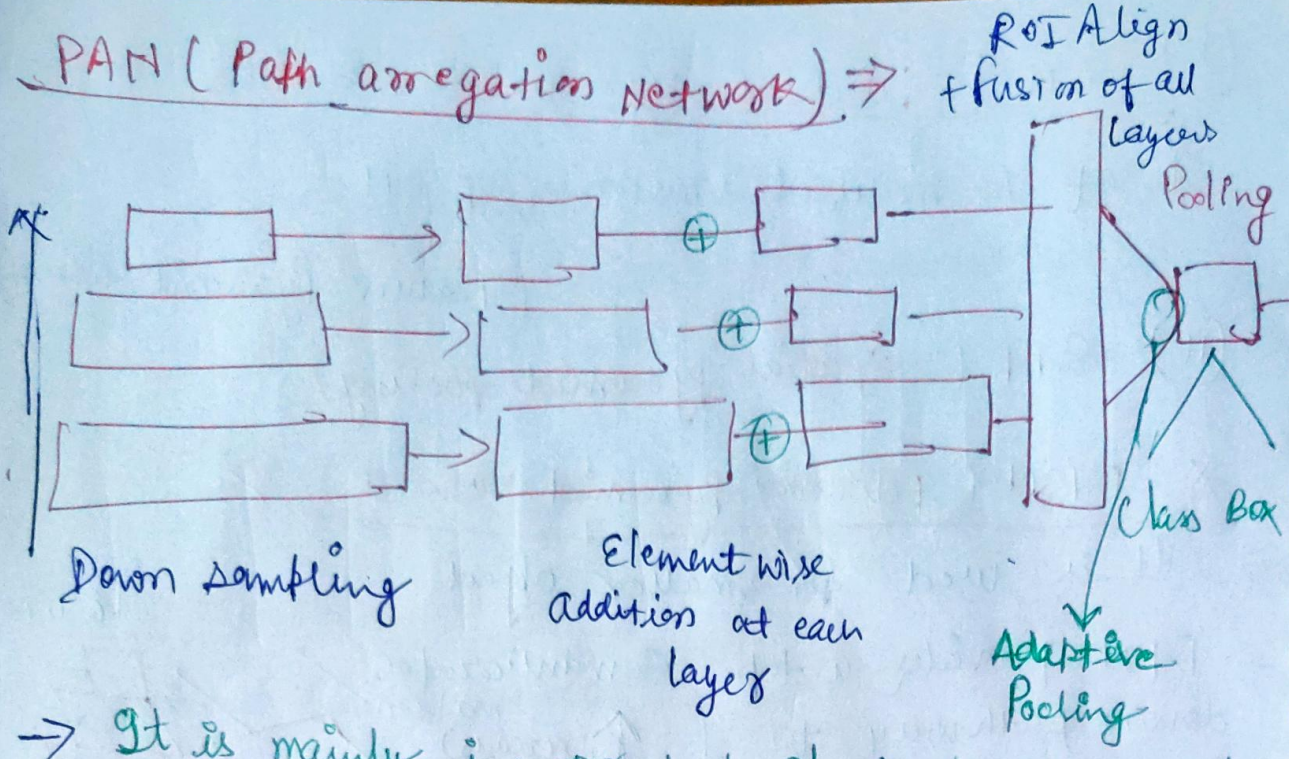
-> It is a feature extractor that works with Object detectors.

- FPN gives better Result or MAP.



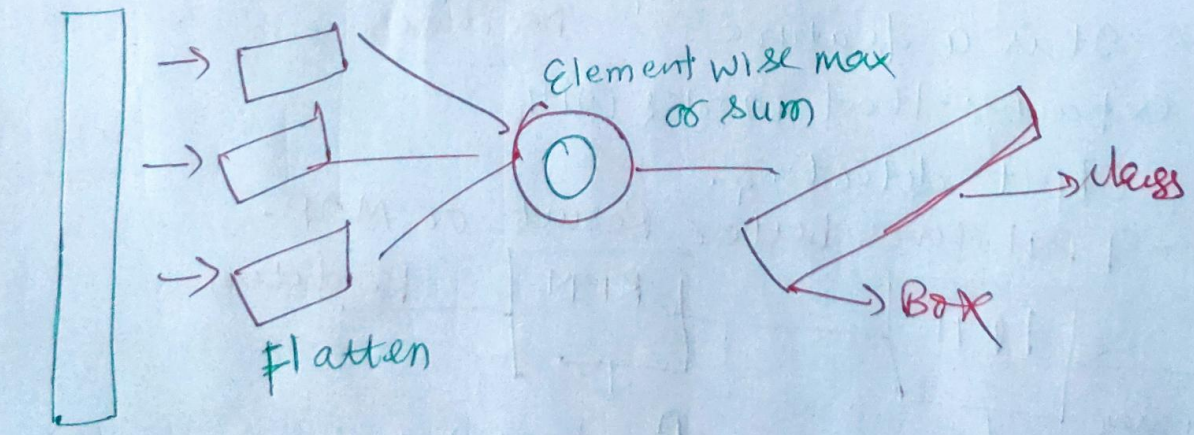


PAN (Path aggregation Network) ⇒



→ It is mainly incorporated of instance segmentation by preserving spatial information.

→ It has ability to preserve spatial information accurately which helps in proper localization of pixels for mask formation.



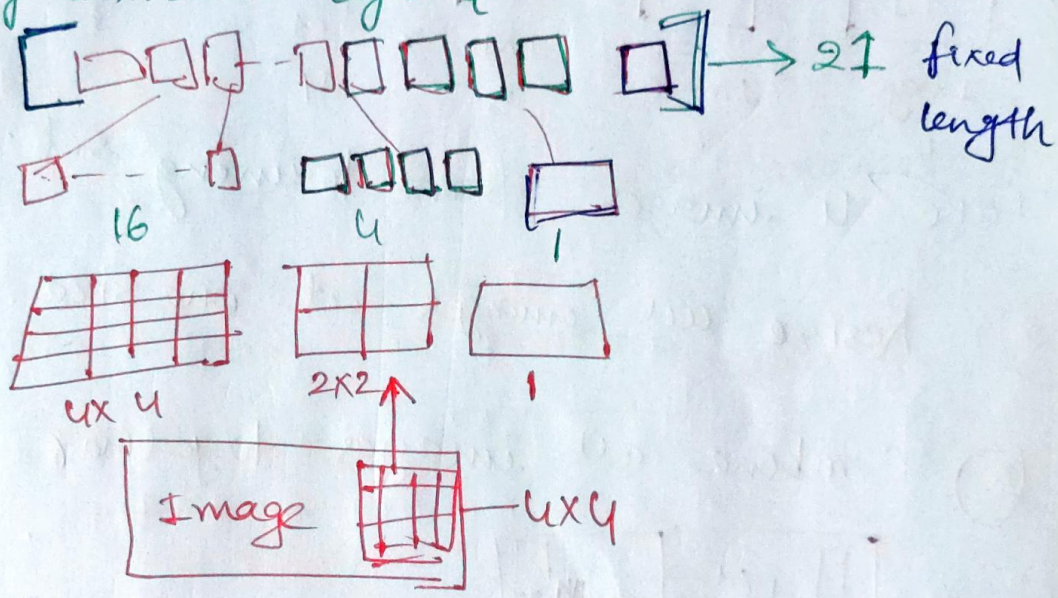
→ In YOLO v4 Modified PAN used as we add conv. layer for concategnation.

→ In YOLO V4, Modified PAN uses concat in replace of adding.

SPP (spatial pyramid pooling)

→ It is used in replace of max pooling.

→ It pools the features and generates fixed-length outputs. which are then fed to the fully connected layers.



→ It gives better contextual information through scaling

* YOLO V5

there are mainly two changes

(i) Mosaic Data Augmentation Technique

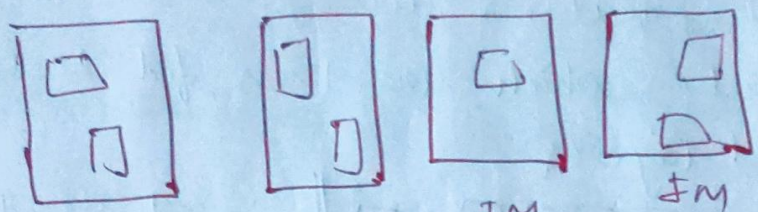
(ii) Focus layer

* MOSAIC Data Augmentation Technique

→ It helps to design image that out of Context object also detected.

→ It is used to increase data.

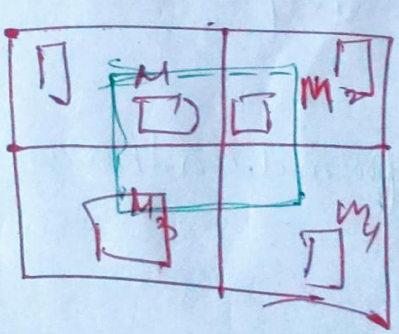
①



Take 4 images from training set.

② Resize all images at one size.

③ Combine all images together



→ Mosaic will help to recognise the objects that the model is not used to looking at together.

→ All the cases where objects appear in different Contexts.

→ Mosaic used during training time.

* Bage of FreeBies →

~~During~~ training time increases and Model will not change.

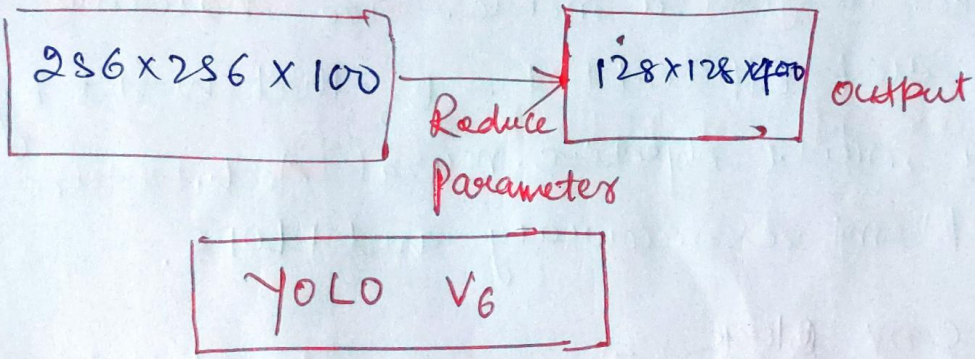
So, Mosiac data aug. is this type.

* Bage of Specials →

→ Mager changes in architecture which increases the complexity & accuracy.

② Focus layer →

→ It is used to reduce no of parameters.



YOLO V6

* Whenever we are building the mode we will using the latest object detection which is based on accuracy & speed.

* Efficient Net used here as BackBone CNN.

*

* Efficient Net →

3 things

B_0
B_1
.
.
.
B_7

→ Width scaling
→ depth scaling
→ Resolution Scaling

⇒ It expands the network's width, depth and resolution.

Hence, it is critical to have a good baseline network.

the authors designed a mobile-size baseline network called Efficient Net – B_0, that works by using a multi-objective neural Architecture that optimizes accuracy and FLOPS.

* MB conv Block →

(1) Squeeze and excitation phase
(2) Expansion Phase.

YOLO V_7

- New backBone CNN used.
- this version is very accurate and fast.
- ELAN (efficient light weight Attention)

→ It is a backbone CNN architecture that was designed to achieve high efficiency and effectiveness in computer vision tasks.
It was introduced in the research paper titled "ELAN" for real time 'Object Detection' by Yuanho Cai, et al., which was published in 2020.

YOLO V_7 [Conv, pooling, CBS, Concat, ELAN, SPP, CSVC]