



IDC410

A course on Image Processing and Machine Learning

(Lecture 19)

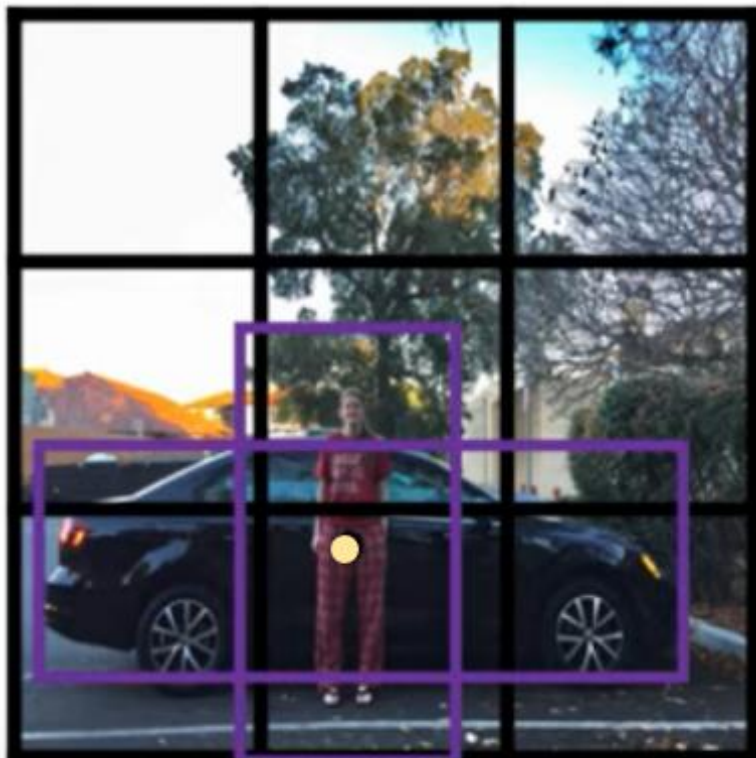
Shashikant Dugad,
IISER Mohali



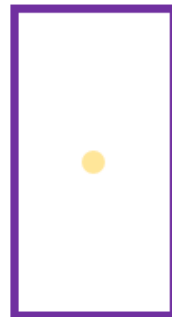
Anchor Box

- **YOLOv3 architecture has 3 Anchor boxes ($N_A=3$) with different aspect ratio and they are used as a starting point to predicts 3 bounding boxes (corresponding to each anchor box) for each grid cell**
- **YOLOv3 uses anchor boxes (predefined bounding box shapes) to predict object locations and sizes. The confidence loss helps the model learn to refine these anchor boxes to accurately capture objects.**
- **Neck in the YOLOv3 model has 3 outputs with each having grid size; 52x52, 26x26, and 13x13**
- **Therefore, # of predicted boxes for a) 1st output (high-resolution) will be $N_A \times 52 \times 52 = 8112$, b) 2nd output will be $N_A \times 26 \times 26 = 2028$ and c) 3rd output (high-semantic) will be $N_A \times 13 \times 13 = 507$**
- **Total Number of predicted boxes are: $8112+2028+507 = 10652$**

Anchor Box



Anchor Box 1



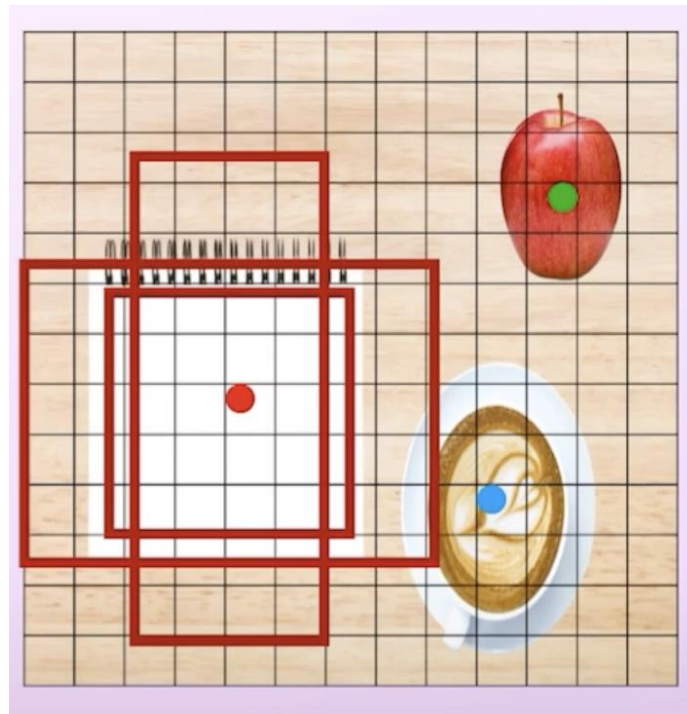
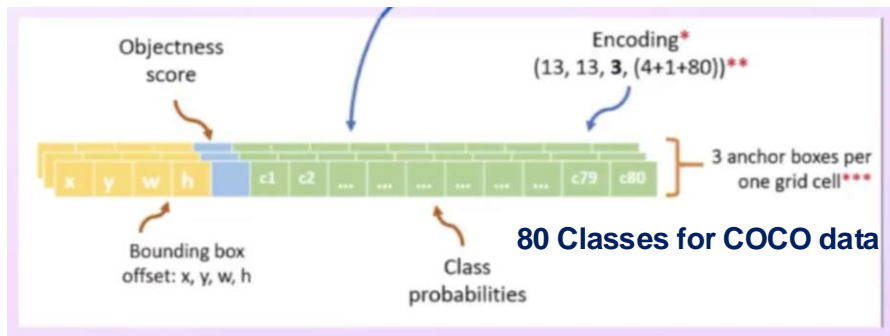
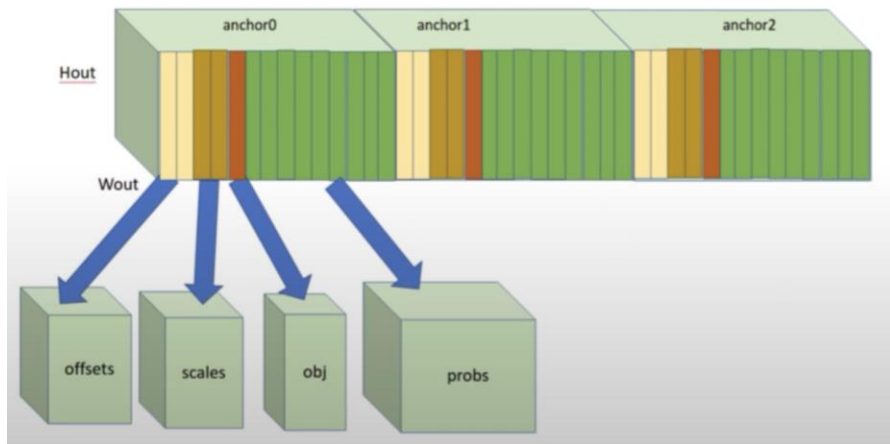
Anchor Box 2



Anchor Box

- Each predicted box has following information: $[t_x, t_y, t_w, t_h, t_0, p(N_c)]$
- t_x and t_y represents measure of coordinates of the center of the predicted box
- t_w and t_h represents measure of width and height of predicted box
- t_0 is indicator of objectness (0 to 1) indicating presence or absence of object in a grid cell
- $p(N_c)$ is an array of class probability of each class (Example: COCO dataset has 80 classes)

Anchor Box



Bounding Box Prediction

- The **YOLOv3** model predicts 3 bounding boxes for each grid cell
- First 5 elements of a given predicted bounding box within that grid cell are: $[t_x, t_y, t_w, t_h, t_o]$
- t_x and t_y represents measure of coordinates of the center of the predicted box that is expected to lie within the same grid cell
- Coordinates of each cell is normalized between 0 to 1
- However, t_x and t_y are kept as free parameter in the loss function thus, predicted value varies between $-\infty$ to $+\infty$
- But, t_x and t_y have to be mapped between 0 to 1 to get physical center coordinates of the box which is done by applying sigmoid function on t_x and t_y and adding the known offset
- Therefore, actual (physical) box center coordinates are:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x & (c_x \text{ is the offset from the left of the image}) \\ b_y &= \sigma(t_y) + c_y & (c_y \text{ is the offset from the top of the image}) \end{aligned}$$

Bounding Box Prediction

- t_w and t_h represents measure of width and height of predicted box in a given grid cell
- However, t_w and t_h are kept as free parameter in the loss function, thus, predicted values of t_w and t_h varies between $-\infty$ to $+\infty$
- But, t_w and t_h have to be mapped between 0 to $+\infty$ to get relative width and height of the box w.r.t. the width (p_w) and height (p_h)
- Exponential function is applied on t_w and t_h to get values between 0 to $+\infty$
- Therefore, actual (physical) dimensions (width and height) of the predicted box in a given grid cell are:

$$b_w = p_w e^{t_w} \quad (p_w \text{ is the width of anchor box})$$

$$b_h = p_h e^{t_h} \quad (p_h \text{ is the height of anchor box})$$

Bounding Box Prediction

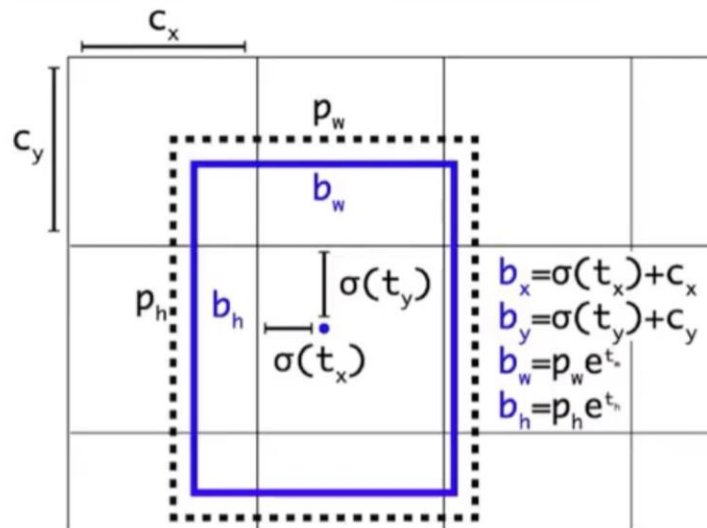
Model outputs - t_x, t_y, t_w, t_h, t_o

$$b_x = \sigma(t_x) + c_x \quad (c_x \text{ is the offset from the left of the image})$$

$$b_y = \sigma(t_y) + c_y \quad (c_y \text{ is the offset from the top of the image})$$

$$b_w = p_w e^{t_w} \quad (p_w \text{ is the width of anchor box})$$


$$b_h = p_h e^{t_h} \quad (p_h \text{ is the height of anchor box})$$



Intersection over Union

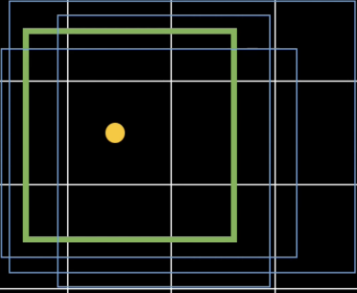
Jaccard Index
 coefficient, is a statistic used for gauging the similarity
 and diversity of sample sets.

Union



$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$$= \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$



Intersection over Union



Loss Function

Objectness Prediction: YOLOv3 predicts a confidence score (C_i) for each bounding box, indicating how likely that box contains any of the trained object

Localization Loss: Penalizes the difference between the predicted and ground truth bounding box coordinates (x, y, width, height)

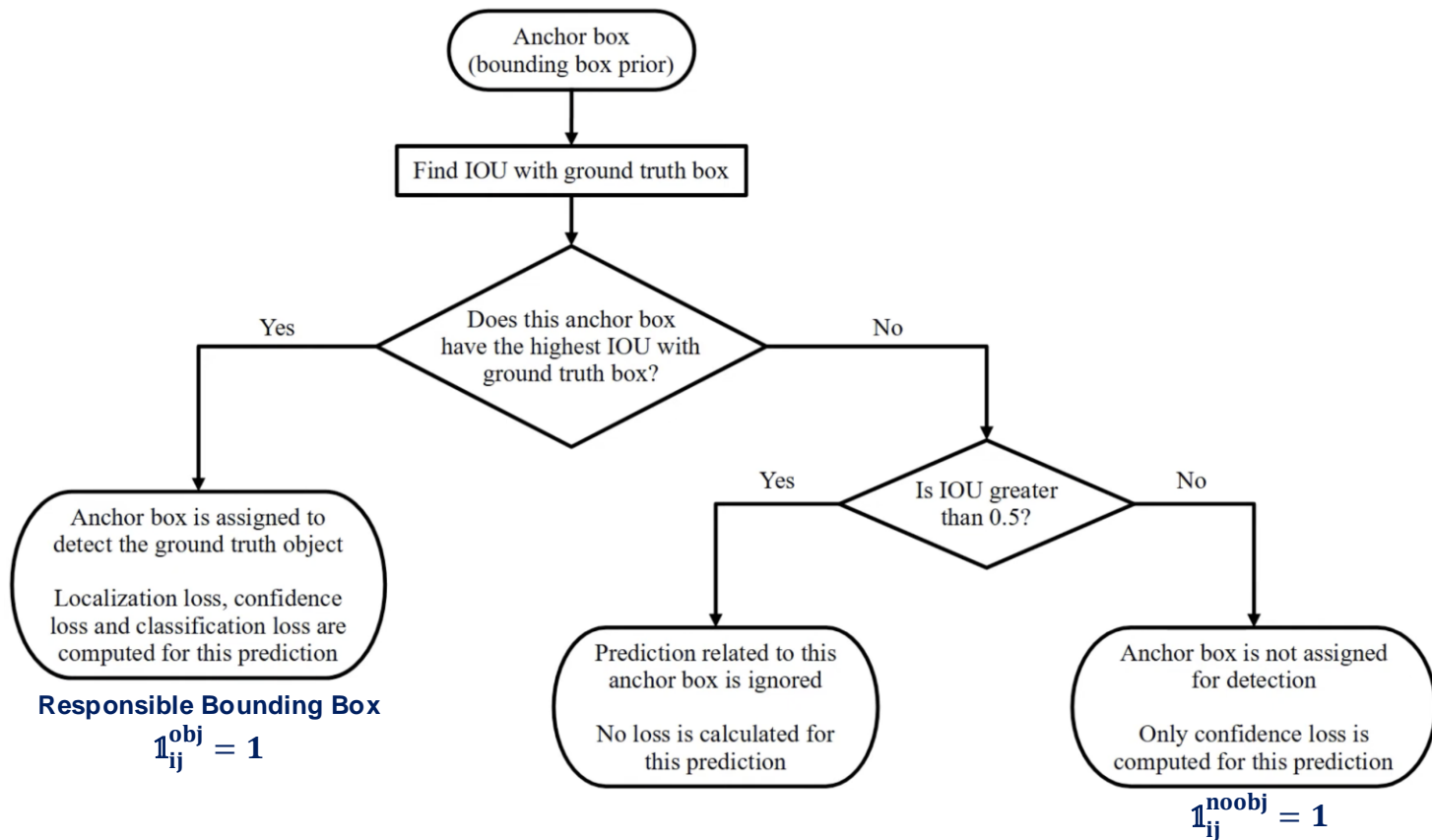
Confidence Loss: Penalizes the model when its predicted confidence score deviates from the ground truth (whether an object is present or not) by penalizing the difference between the predicted objectness score and the ground truth (whether an object is present or not)

Class Imbalance: To address the issue that most bounding boxes don't contain objects (class imbalance), the confidence loss for boxes without objects is weighted down, preventing the model from over-training on background

Bounding Box Accuracy: The confidence loss also contributes to the overall loss by penalizing inaccurate bounding box predictions, especially when an object is present in the box.

Training Process: During training, the model adjusts its parameters to minimize the overall loss, including the confidence loss, leading to improved object detection performance.

Responsible Bounding Box



Loss Function

Regression
loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Confidence
loss

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

Classification
loss

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

PyLessons.com

Loss Function

We not only need to train the network to detect an object if there is an object in a cell, we also need to punish the network, if it predicts an object in a cell, when there wasn't any.

Various terms used in Loss Function:

- The prediction of YOLO is a $S*S*(B*(5+C))$ vector: B is number of boxes predicted for each grid cells (3 in case of YOLOv3), S is the size of the square grid and C is the number of classes
- The λ are just the constants to take into account relative contributions of several terms in the overall loss function. λ_{coord} is the highest in order, indicating more importance. It is used in the first term in the loss function
- The cell is referred with index i and the box number in the grid cell is referred with index j
- The 5 elements of box outputs of the box j of cell i are coordinates of the center of the box x_{ij} , y_{ij} , height h_{ij} , width w_{ij} and a confidence score C_{ij} indicating whether there is an object or not

Loss Function

Various terms used in Loss Function:

- The values with a hat on C_{ij} correspond to labels (\widehat{C}_{ij}) and C_{ij} without hat are predictions. The measure \widehat{C}_{ij} is the confidence score of that bounding box (in cell i) which has the highest Intersection Over Union (IoU) with the ground truth.
- $\mathbb{1}_{ij}^{obj}$ is 1 if the j^{th} bounding box predictor has the highest IoU with the ground truth (It is referred as responsible bounding box), else it is 0
- $\mathbb{1}_i^{obj}$ is 1 when there is an object (precisely, object's center) of a particular class in cell i and 0 elsewhere. This term is used classification loss

Description of each of the 5 terms in the Loss Function:

- 1st and 2nd term in the loss function penalizes bad localization of bounding box center and the inaccuracies in its height and width. The square root is present so that errors in small bounding boxes are more penalizing than errors in big bounding boxes.
- It is to be noted that the loss due to bound box center OR its dimensions (width w and height h) comes from ONLY from THE responsible bounding box of each grid cell. This is considered irrespective of the object is present OR not present in the grid cell as ground truth



Loss Function

Description of each of the 5 terms in the Loss Function:

- 3rd term in the loss function forces the confidence of the most accurate bounding box to be close to the confidence score of the ground truth, if object is in the grid cell and the box is the "responsible" one.
- Loss due to the confidence in each grid cell comes ONLY from THE responsible bounding box, even if the object is present or not present in the grid cell as ground truth
- 4th term, tries to make confidence score close to 0, when there is no object in the cell
- 5th term, aims to align the class of the grid cell with the ground truth, if there is an object.

Loss Function with Cross Entropy

Regression
loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Confidence
loss

$$\sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i \right) \log(1 - C_i) \right] + \\ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{noobj}} \left[\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i \right) \log(1 - C_i) \right] +$$

Classification
loss

$$\sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in \text{classes}} \left[\hat{p}_i(c) \log(p_i(c)) + \left(1 - \hat{p}_i(c) \right) \log(1 - p_i(c)) \right]$$



Non Maximum Suppression (NMS)

- **Step 1:** The bounding box with the greatest score is moved from *list B* to *list D*
- **Step 2:** Compared with all the other boxes left in the *list B* w.r.t. IoU which defines overlap index between two bounding boxes.
- **Step 3:** If the IoU is greater than a certain threshold, that box in *list B* is REMOVED since it is too similar to the one just extracted and has a lower score w.r.t. that in *the list D*.
- **Step 4:** Once the comparison is over, then Step 1 is repeated which is moving the best bounding boxes from remaining B to D
- The loop is repeated until the *list B* is empty.
- **Note:** You may eliminate the bounding boxes in the *list B* by rejecting those with a score lower than a fixed threshold OR eliminate the bounding boxes in the *list D* by rejecting those with a score lower than a fixed threshold

Performance of YOLOv2

Accuracy - COCO

	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
<i>One-stage methods</i>							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2

Performance of YOLOv3

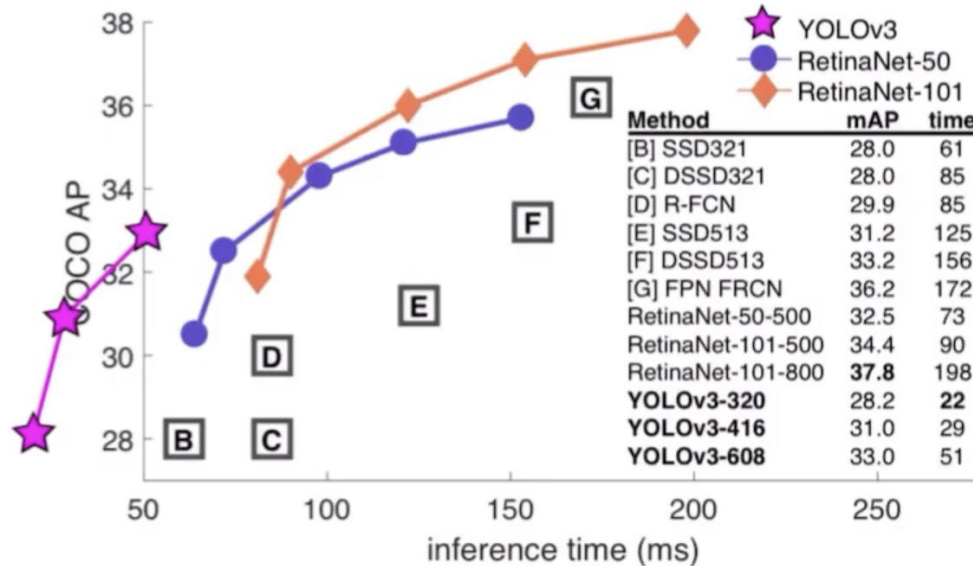
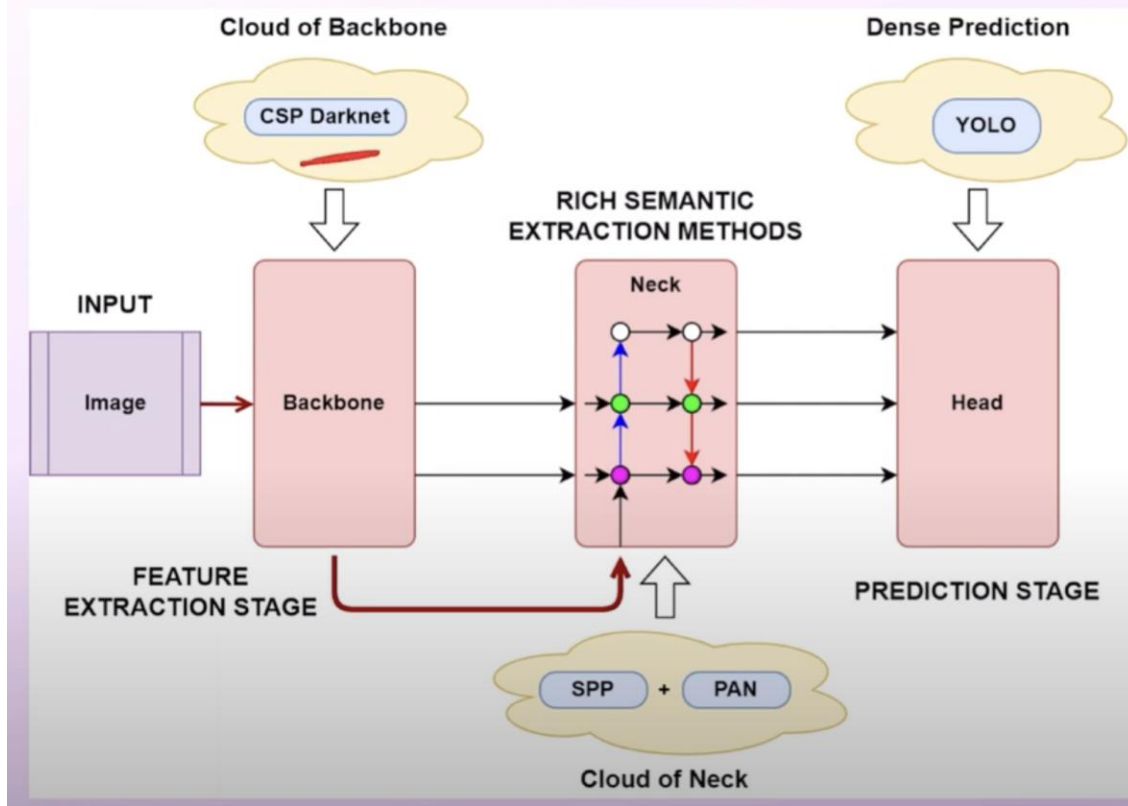


Figure 1. We adapt this figure from the Focal Loss paper [9]. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

YoloV4



Performance of YOLOv3/YOLOv4

