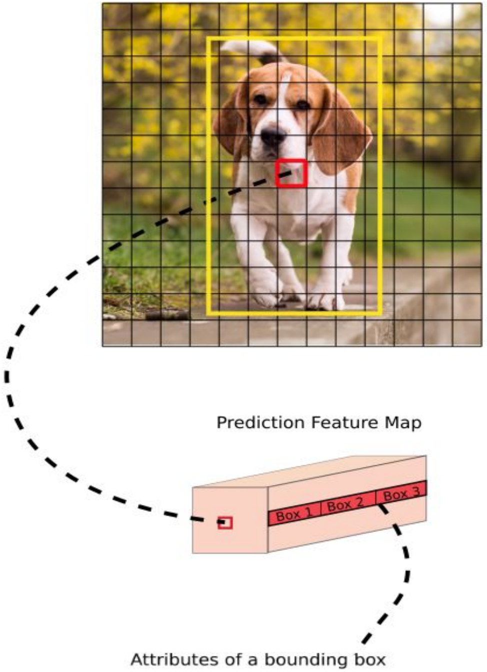
# Live Coding Session

# Data Pre Processing




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| P(obj) | x | y | w | h | P(c1) | P(c2) | P(c3) |

Now for each grid cell:

P(obj) = 1 if the center of the object bounding box lies in the grid cell.

P(c1) = 1, P(c2) = 0, P(c3) = 0
if the object is of class c1

Similarly for the others.

Y_label = np.zeros((8,8,8))

1. **Computing which grid cell the center of the object lies in.**

`center_x = (x1+x2)/2` `center_y = (y1+y2)/2` where (x1,y1) and (x2,y2) are given as bbox coords.

```
grid_x = center_x/grid_size_x  ; grid_y = center_y/grid_size_y.

grid_size_x = img_width/8 = 16; grid_size_y = img_height/8 = 16.

Now we have Y_label[0,grid_x,grid_y] = 1
```

2. **Computing x and y.**

x and y represent the center of the box relative to the bounds of the grid, therefore they lie in (0,1).

Within the box the coordinates of x will be = center_x % grid_x, and since x is relative the bounds we normalize with grid_x.

```
x = (center_x%grid_size)/grid_size  , y = (center_y%grid_size)/grid_size
Now we have Y_label[1,grid_x,grid_y] = x, Y_label[2,grid_x,grid_y] = y
```

## 3. Computing w and h where w and h are relative to the whole i

```
w = (x2-x1)/image_height ;  h = (y2-y1)/image_width
```
where (x1,y1) and (x2,y2) are given as bbox coords.

```
Now we have Y_label[3,grid_x,grid_y] = w, Y_label[4,grid_x,grid_y]
```

## 4. Class Probabilities.

If class from the raw_labels = ‘traffic light’ = 1

Then,

```
Y_label[5,grid_x,grid_y] = 0 , Y_label[6,grid_x,grid_y] = 1 and Y_label[7,grid_x,grid_y] = 0
```

# Yolo Loss.

Each grid cell predicts box confidence score, bounding boxes and conditional class probabilities.

**Classification loss**

If *an object is detected*, the classification loss at each cell is the squared error of the class conditional probabilities for each class:

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

where

$\mathbb{1}_i^{obj} = 1$ if an object appears in cell $i$, otherwise 0.

$\hat{p}_i(c)$ denotes the conditional class probability for class $c$ in cell $i$.

### Localization loss

The localization loss measures the errors in the predicted boundary box locations and sizes. We only count the box responsible for detecting the object.

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

where

$\mathbb{1}_{ij}^{obj} = 1$ if the $j$ th boundary box in cell $i$ is responsible for detecting the object, otherwise 0.

$\lambda_{coord}$ increase the weight for the loss in the boundary box coordinates.

## Confidence Loss:

If *an object is detected in the box*, the confidence loss (measuring the objectness of the box) is:

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

These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts.

We want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth box

C_target  = P(objectness) * IOU(predicted_box, ground_truth_box) = IOU(predicted_box, ground_truth_box)

C_predicted = output of the n/w.

**If no object exists in that cell, the confidence scores should be zero, C_target = 0.**

Workbook links:

https://colab.research.google.com/drive/1I2I1hCa6XuE7oOeiAnYhdnRQfs9-2sG1?usp=sharing

https://colab.research.google.com/drive/1iTD-Jez1jP2YAX12ID9PDWXqZTdE9xd8?usp=sharing

https://colab.research.google.com/drive/1Tjh6W0C-aptaRZem6ep-qnLxJI7rU3h8?usp=sharing

TA- assignment link -

https://docs.google.com/spreadsheets/d/1Ls9eOFbvaxIYnOS_TmROW1qIOuIj9k2tO-zHFAoXryo/edit?usp=sharing

Go to the assigned TA’s debug channel on discord.