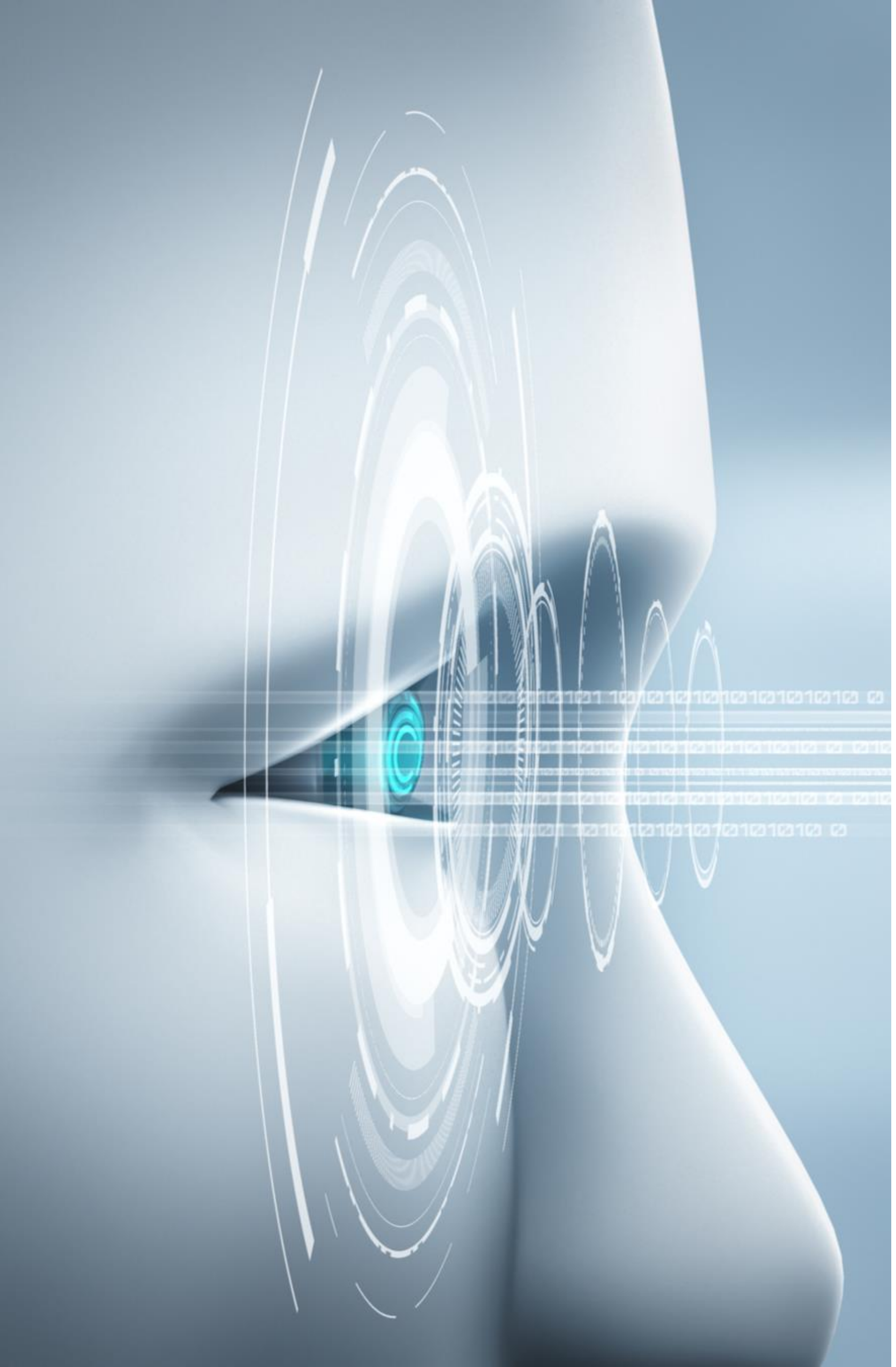


# Computer Vision Applications

BY QUADEER SHAIKH

# About me

**Work Experience**

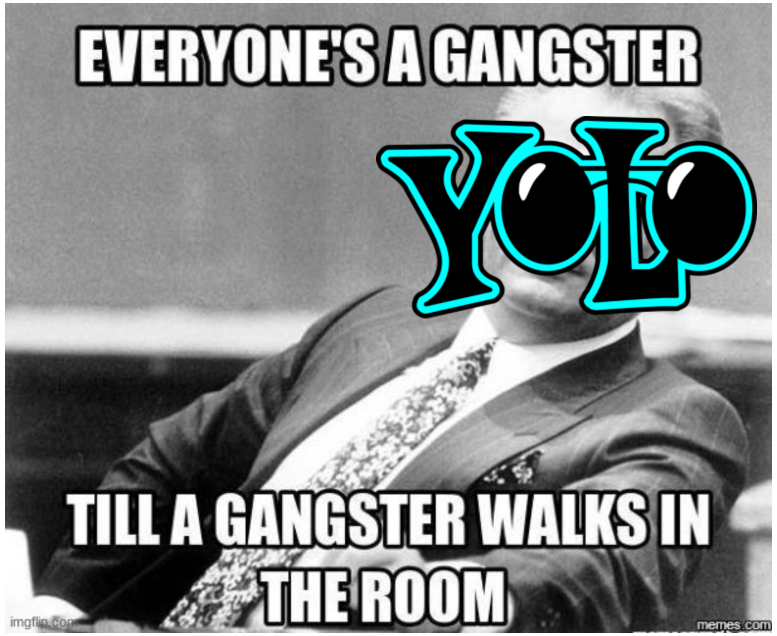
- Risk Analyst
  - Morgan Stanley (Jan 2023 – Present)
- Data Science Intern
  - AkzoNobel Coatings International B.V. Netherlands (Feb 2022 – Dec 2022)
- Data Science Intern
  - EzeRx Health Tech Pvt. Ltd. (Jan 2022 – July 2022)
- Associate Engineer
  - Tata Communications Ltd. (July 2019 – Aug 2020)
- Network Automation and Analysis Engineer Intern
  - Cisco (June 2018 – July 2018)

**Education**

- M.Tech – Artificial Intelligence
  - NMIMS (2021 - 2023, currently pursuing)
- B.E. – Computer Engineering
  - Mumbai University (2015 - 2019)

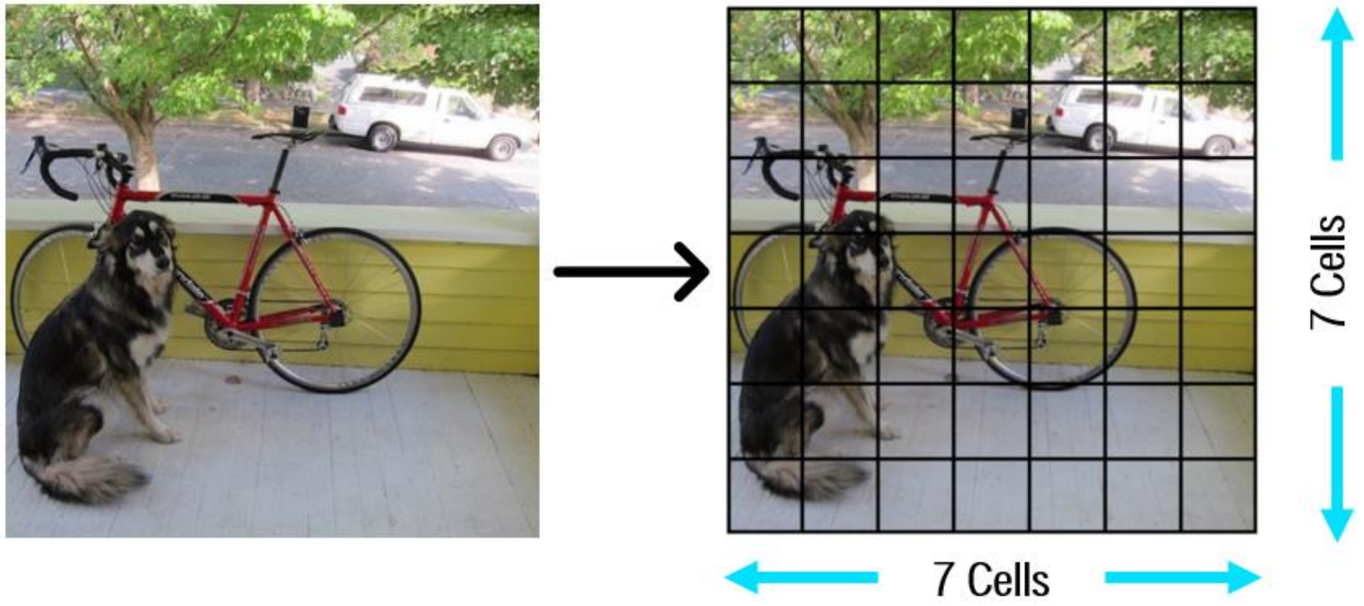
# YOLOv1: You Only Look Once

- The network only looks at the image once to detect multiple objects. Thus, it is called YOLO, You Only Look Once.
- Unlike the two stage detectors like the RCNN family YOLO is a on stage detector and hence the name YOLO

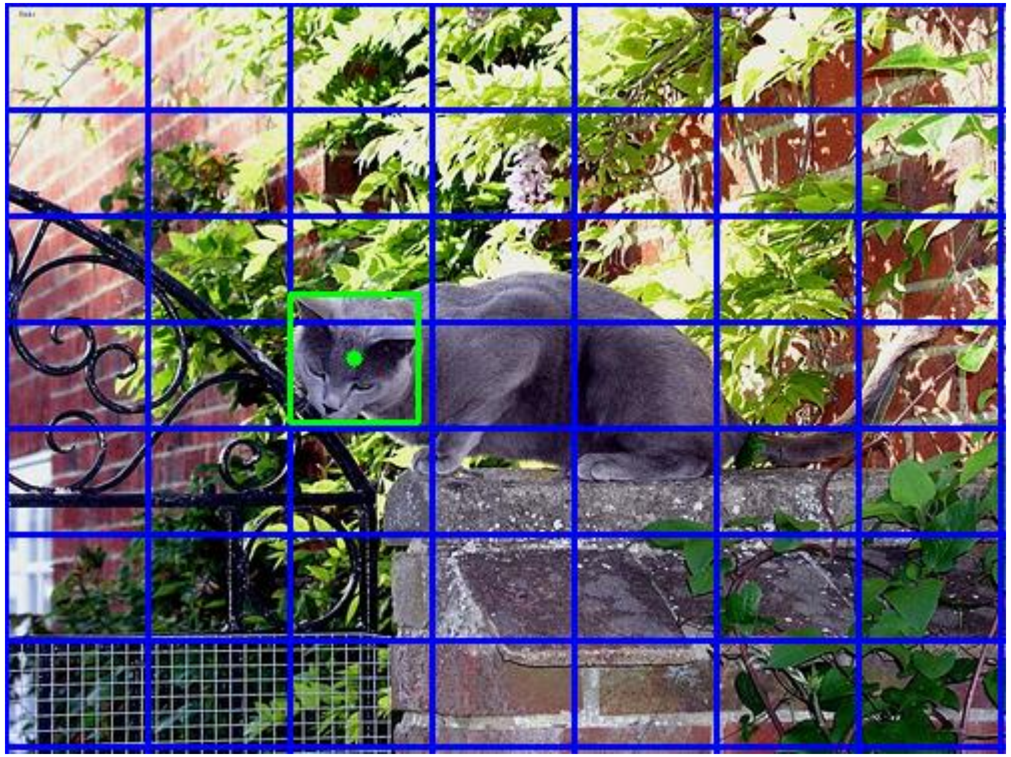


# YOLOv1: Unified Detection/Detection at one go

- **The input image is divided into an S×S grid (S=7)**. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.



# YOLOv1: Unified Detection/Detection at one go

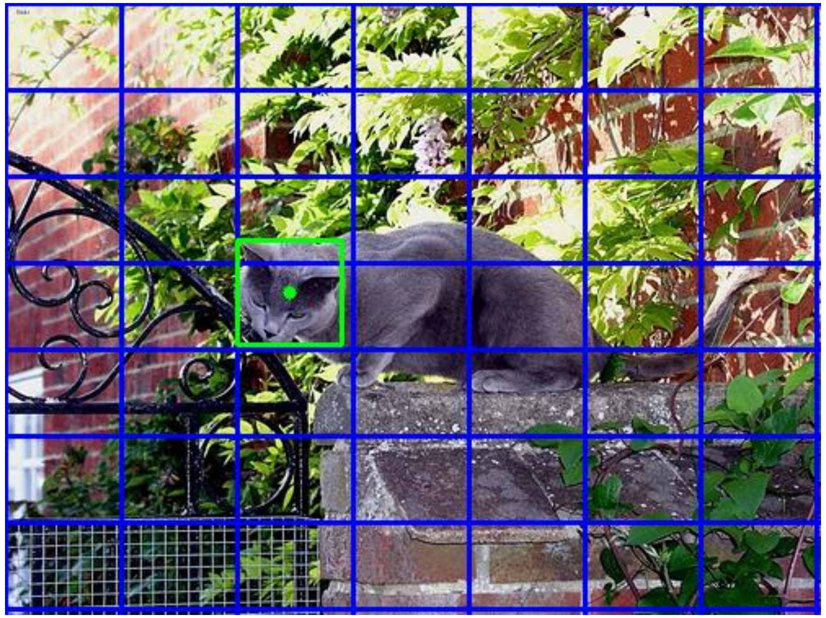


# YOLOv1: Unified Detection/Detection at one go

1. Each grid cell predicts B bounding boxes (B=2 in the paper) and confidence scores for those boxes.

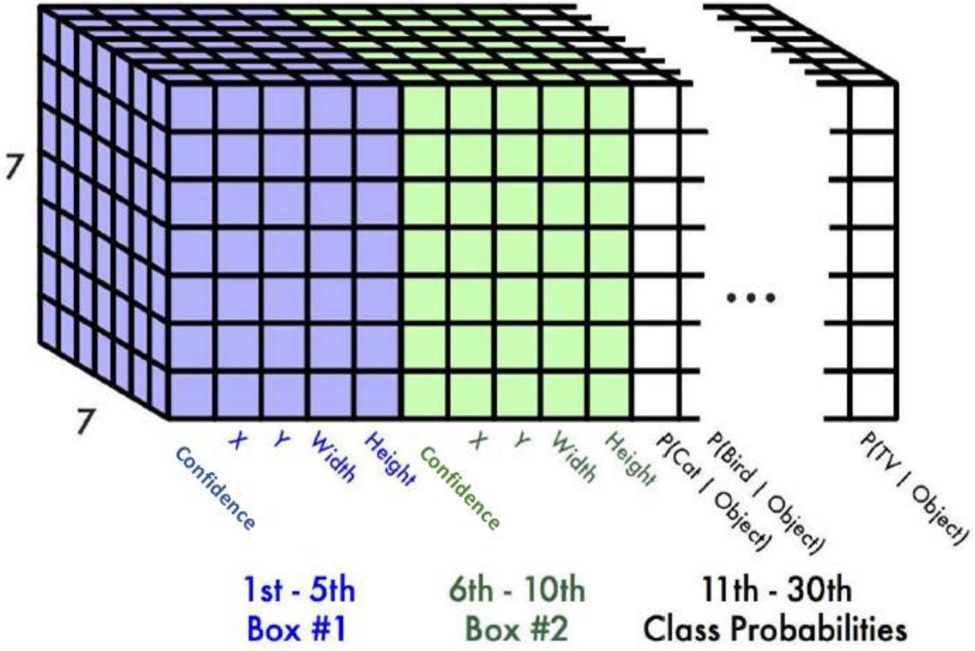
2. These confidence scores reflect how confident the model is that the box contains an object. i.e. the probability of prediction of the box

3. Each bounding box consists of 5 predictions: x, y, w, h, and confidence.
   - The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell.
   - The width w and height h are predicted relative to the whole image.
   - The confidence represents the Intersection Over Union (IOU) between the predicted box and any ground truth box.



# YOLOv1: Unified Detection/Detection at one go

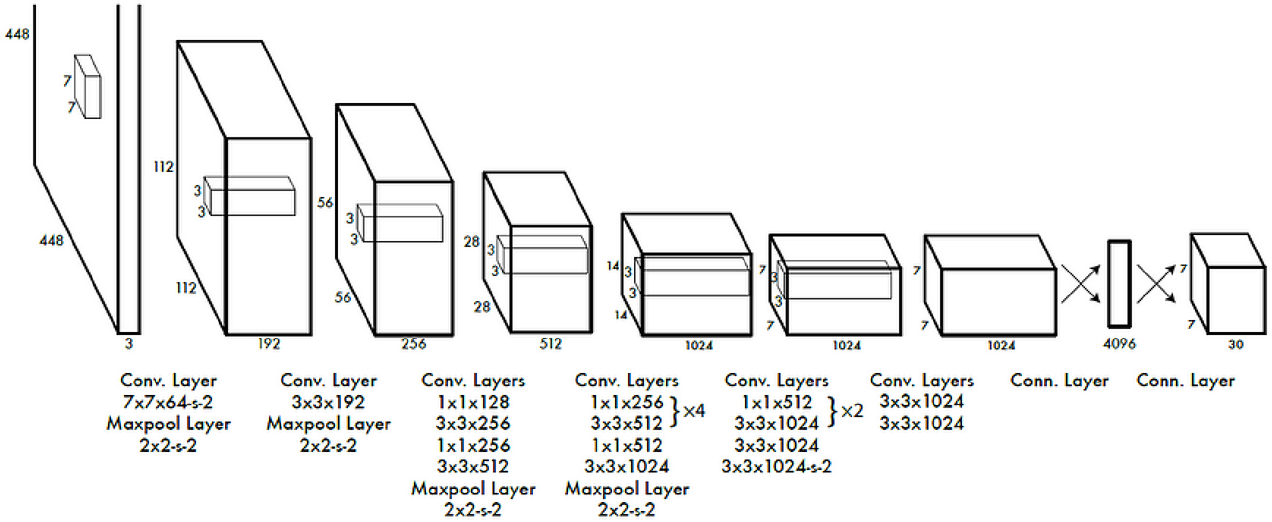
1. Out of the B bounding boxes for each grid cell only one is selected based on the highest confidence.
2. Each grid cell also predicts C conditional class probabilities, where C is the number of classes. YOLOv1 was originally trained on PascalVOC2007 dataset which had 20 classes so in that case C=20
3. S = 7, B = 2, C = 20
4. S x S x (B x 5 + C) = 7x7x(2x5+20) = 7x7x30 is the output shape of the network
5. The ground truth data for object detection consisting of bounding box coordinates and class predictions should be structured in the shape of the above output



# YOLOv1 Architecture

**The model consists of 24 convolutional layers followed by 2 fully connected layers.** Alternating 1×1 convolutional layers reduce the features space from preceding layers.

**Input Size:** 448x448x3, **Output size:** 7x7x30 (PascalVOC2007)



# YOLOv1 Loss Function

Surprisingly YOLO’s loss function is the most odd one to have ever been defined as it comprises of squared errors for both box coordinates and class predictions.

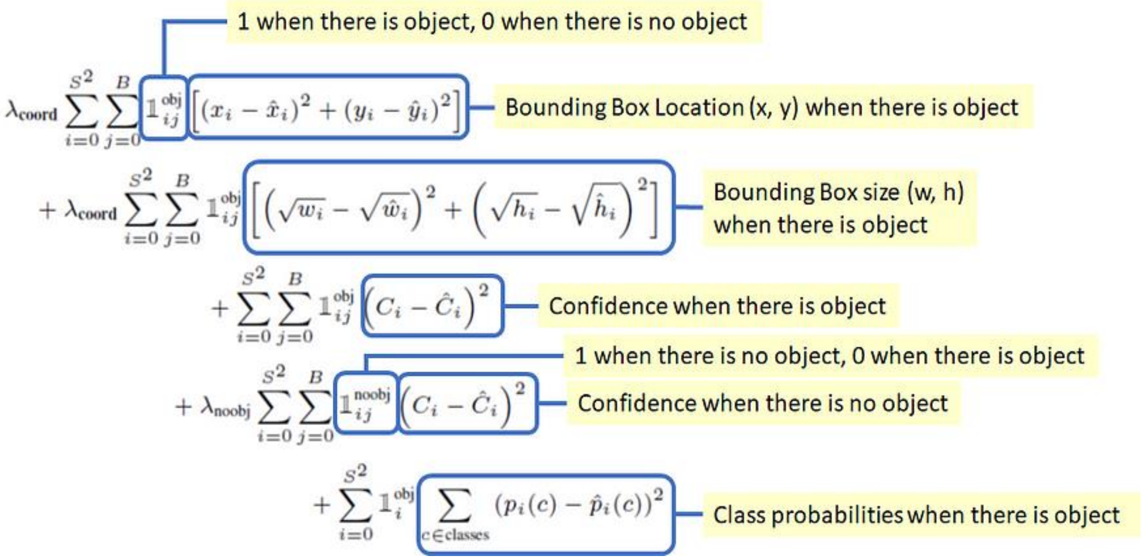
1. The bounding box x and y coordinates is parametrized to be offsets of a particular grid cell location so they are also bounded between 0 and 1. And the sum of square error (SSE) is estimated only when there is object.

2. The bounding box width and height are normalized by the image width and height so that they fall between 0 and 1. SSE is estimated only when there is object. Since small deviations in large boxes matter less than in small boxes. square root of the bounding box width w and height h instead of the width and height directly to partially address this problem.

3. In every image many grid cells do not contain any object. This pushes the “confidence” scores of those cells towards zero, often overpowering the gradient from cells that do contain objects, and makes the model unstable. Thus, the loss from confidence predictions for boxes that don’t contain objects, is decreased, i.e. **λnoobj**=0.5.

4. SSE of class probabilities when there is objects.

5. Due to the same reason mentioned in 3 and 4, λcoord = 5 to increase the loss from bounding box coordinate predictions.



1 when there is object, 0 when there is no object

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

Bounding Box Location (x, y) when there is object

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

Bounding Box size (w, h) when there is object

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

Confidence when there is object

1 when there is no object, 0 when there is object

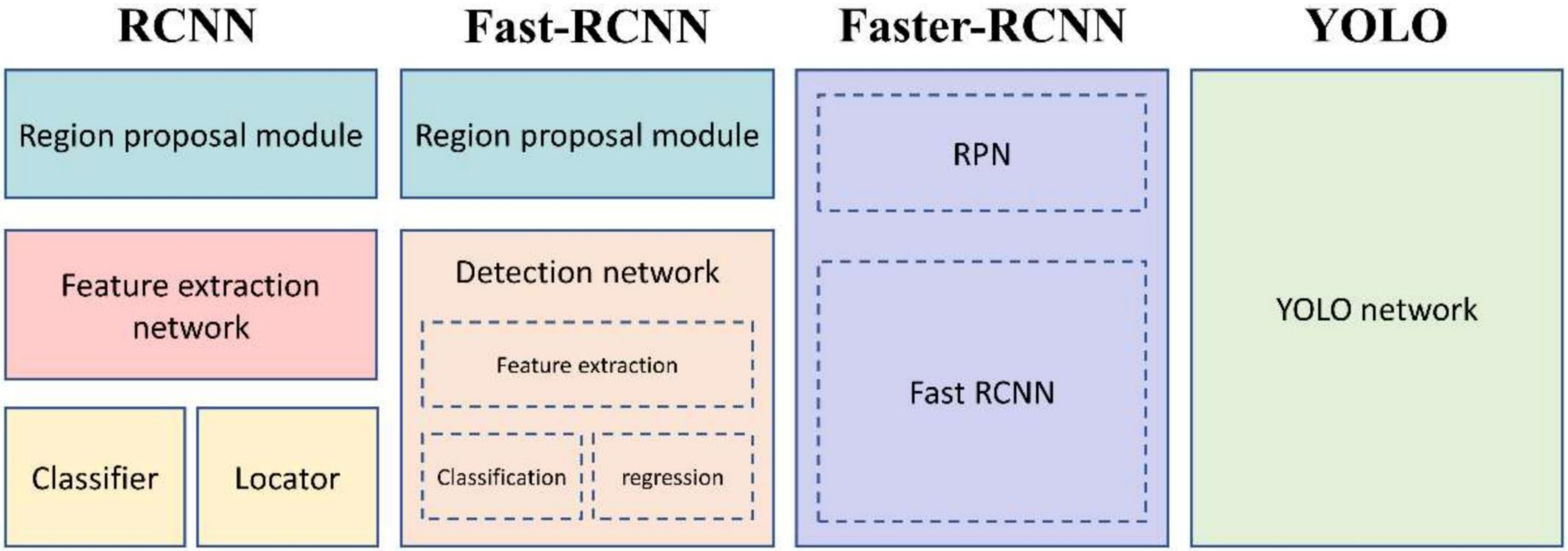
$$+\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \boxed{\mathbb{1}_{ij}^{\text{noobj}}} \boxed{\left(C_i-\hat{C}_i\right)^2}$$

Confidence when there is no object

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

Class probabilities when there is object

# RCNN Family vs YOLOv1



# *Thank you*

For any queries drop an email at: quadeershaikh15.8@gmail.com