



**YOU ONLY
LOOK ONCE**

**UNIFIED, REAL TIME
OBJECT DETECTION**

WHAT IS YOLO

- A CNN BASED OBJECT DETECTION MECHANISM

SINGLE NEURAL NETWORK PREDICTS

```
graph TD; A[SINGLE NEURAL NETWORK PREDICTS] --> B[BOUNDING BOXES]; A --> C[AND CLASSIFIES PROBABILITIES];
```

BOUNDING BOXES AND CLASSIFIES PROBABILITIES

- Normal Yolo: 45 Frames / Second

- Fast Yolo : 155 Frames / Second

- mAP (Yolo) 2x > Other Real Time Detectors

- Localization Error > State Of Art Detection System

- Less Likely To Predict False Positive On Background

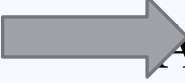
WHY YOLO?

- Most other detection systems repurpose classifiers to perform detection.

Example:

- DPM (deformable parts model): uses **sliding window approach** where **classifier** is run at **evenly spaced locations** over the **entire image**.
- **R- CNN** uses region proposal methods to **generate bounding boxes** then **run a classifier on these proposed boxes**.

- In yolo object detection is reframed as a **single regression problem**
- Image pixel **bounding box coordinate and class probabilities.**
- Helps to answer the **where** and **what**

- YOLO : sees **an entire image** while training and testing time.
-  Able to implicitly encode contextual information about classes and appearance.

R- CNN makes two times more mistakes on background patches.

- Yolo also works well with artwork after being trained with actual image.

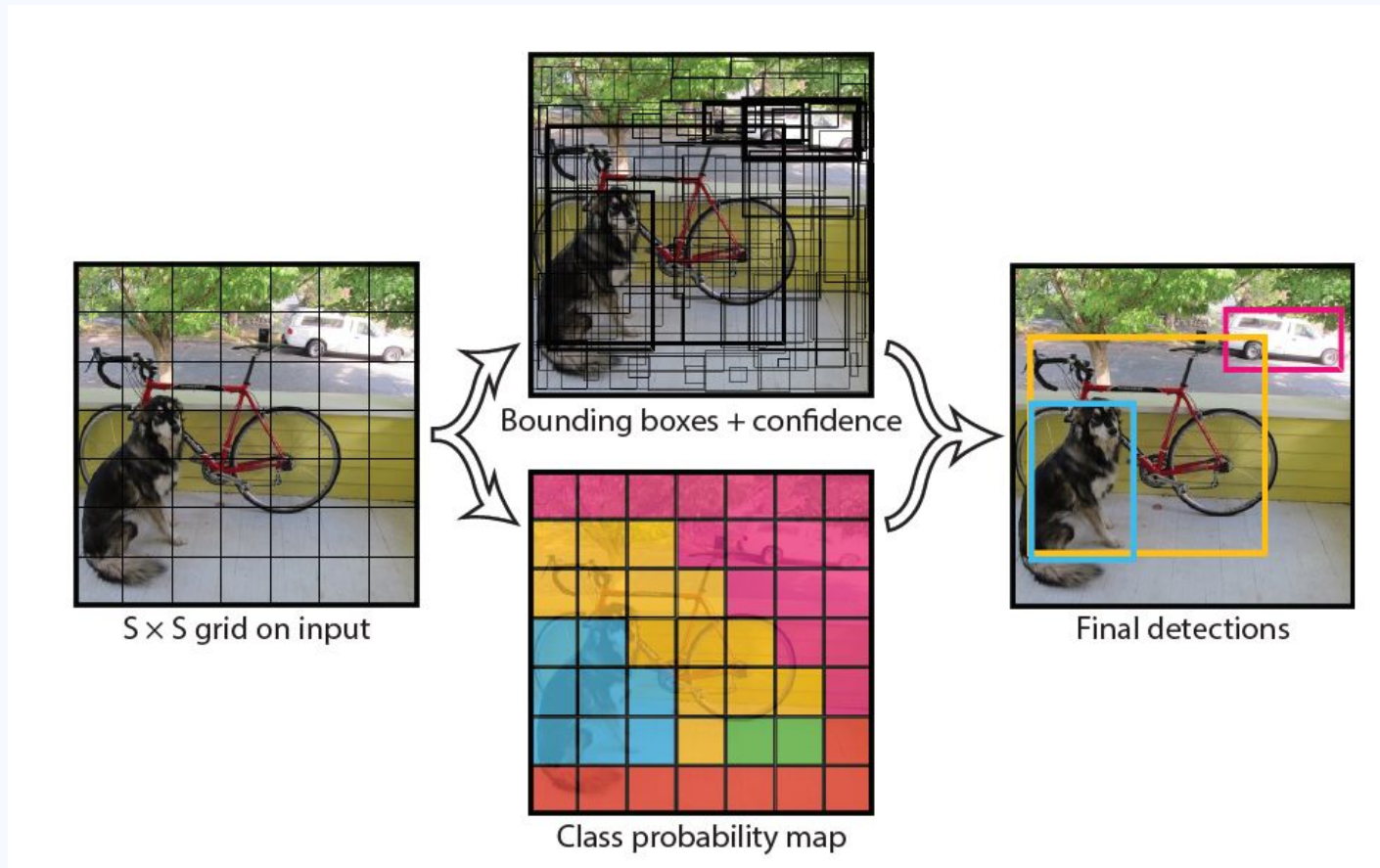
HOW DOES IT WORK

1. Divides image in to an SXS grid
 - 2.If a center of an object falls on a cell then its responsible for detecting the object
 - 3.Each grid predicts bounding boxes and confidence score.
- =Confidence score = $\text{pr}(\text{object}) * \text{IOU}(\text{intersection over union})$
- If the object doesn't exist confidence = 0

- Bounding box consists of 5 predictions : x, y, w, h, C
- Confidence(C) can be predicted as intersection over union between predict box over ground truth
- (x, y) = represent coordinates relative to grid cells

- Each grid also predicts the conditional class probabilities $\text{pr}(\text{class}_i/\text{object})$
- It only predicts for one set of class probabilities per grid cell regardless of the number of boxes.

- Conditional class probabilities x individual box confidence = class specific confidence score for each box
- $\text{Pr}(\text{classi}/\text{object}) * \text{pr}(\text{object}) * \text{IOU} = \text{pr}(\text{classi}) * \text{iou}$



Prediction is encoded as = $S \times S \times (B*5+C)$ tensor

Network Design

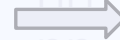
- Uses CNN
 - Evaluated on PASCAL VOC detection dataset.
 - 24 convolutional layers/9 incase of fast yolo +fewer filters
- 2 fully connected layers

- **Convolutional layers:** extract features from image
- **Fully connected layers:** predict out probability and coordinates.
- The model was inspired by GoogleLenet
- 1x1 reduction layer and 3x3 convolutional layer

Training

To pretrain convolutional layers on imagenet 1000-competition dataset

20 convolutional layers average pooling layer fully connected layer.



After being trained for about a week it was able to achieve accuracy of 88% on imagenet 2012 validation set.

- Darknet framework was used for training and inferences
- When implemented we use 24 convolutional layers and 2 fully connected layer
- And input resolution is increased from 224x224 to 448x448

One bounding box predictor is responsible for each object based on the center of the object

This is based on which prediction has the highest iou over ground truth

The final layer predicts location and probability

COMPARISON TO OTHER DETECTION SYSTEMS

Deformable Parts Models (DPM):

- Scan entire window
- Use sliding windows
- Detect individual parts of an object
- Predict class of the object as a whole

COMPARISON TO OTHER DETECTION SYSTEMS

R-CNN

- Extract regions (~2000 regions) from an image
- Uses search algorithms to extract regions
- Region Proposals: vertical bounding boxes
- Huge amount of time to train since all 2000 regions need to be classified

COMPARISON TO OTHER DETECTION SYSTEMS

Fast R-CNN

- Instead of a search algorithm to extract regions, it uses a CNN to generate features
- Then identify the region proposals from features
- Use selective search on the features
- It's faster because the whole image is fed to CNN instead of all 2,000 regions (25X faster than R-CNN)

COMPARISON TO OTHER DETECTION SYSTEMS

Faster R-CNN

- Both R-CNN and Fast R-CNN use selective search to find region proposals
- No selection search
- Instead model learns the region proposals
- 250X faster than R-CNN

COMPARISON TO OTHER DETECTION SYSTEMS

	Mean Average Precision (mAP)	Speed
R-CNN	66.0	.05 FPS
Fast R-CNN	70.0	.5 FPS
Faster R-CNN	73.2	7 FPS
YOLO	63.4	45 FPS

COMPARISON TO OTHER DETECTION SYSTEMS

	Mean Average Precision (mAP)	Speed
R-CNN	66.0	.05 FPS
Fast R-CNN	70.0	.5 FPS
Faster R-CNN	73.2	7 FPS
YOLO	63.4 69.0	45 FPS

Limitations Of Yolo

Loss of Accuracy caused by

- Strong constraints on the bounding boxes
- Down sampling images to lower resolution

Hard to recognize small objects

Hard to recognize groups of object

RECENT TECHNOLOGY

