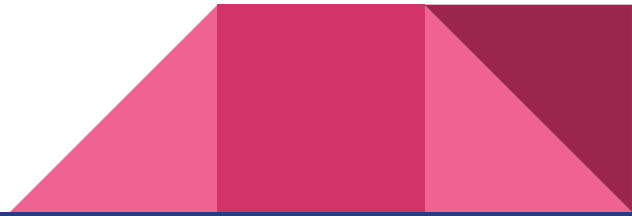
Detection: Yolo 7, Yolo NAS

Object Detection

- Want to identify and localize objects within an image
- Combining classification (which objects are present) and object localization (where the objects are)
- Algorithms draw bounding boxes around objects and perform classification
  - Multiple objects + multiple classes in single image
- Challenges:
  - Variable # of objects per image
  - Variable object sizes
  - Multiple objects at same location



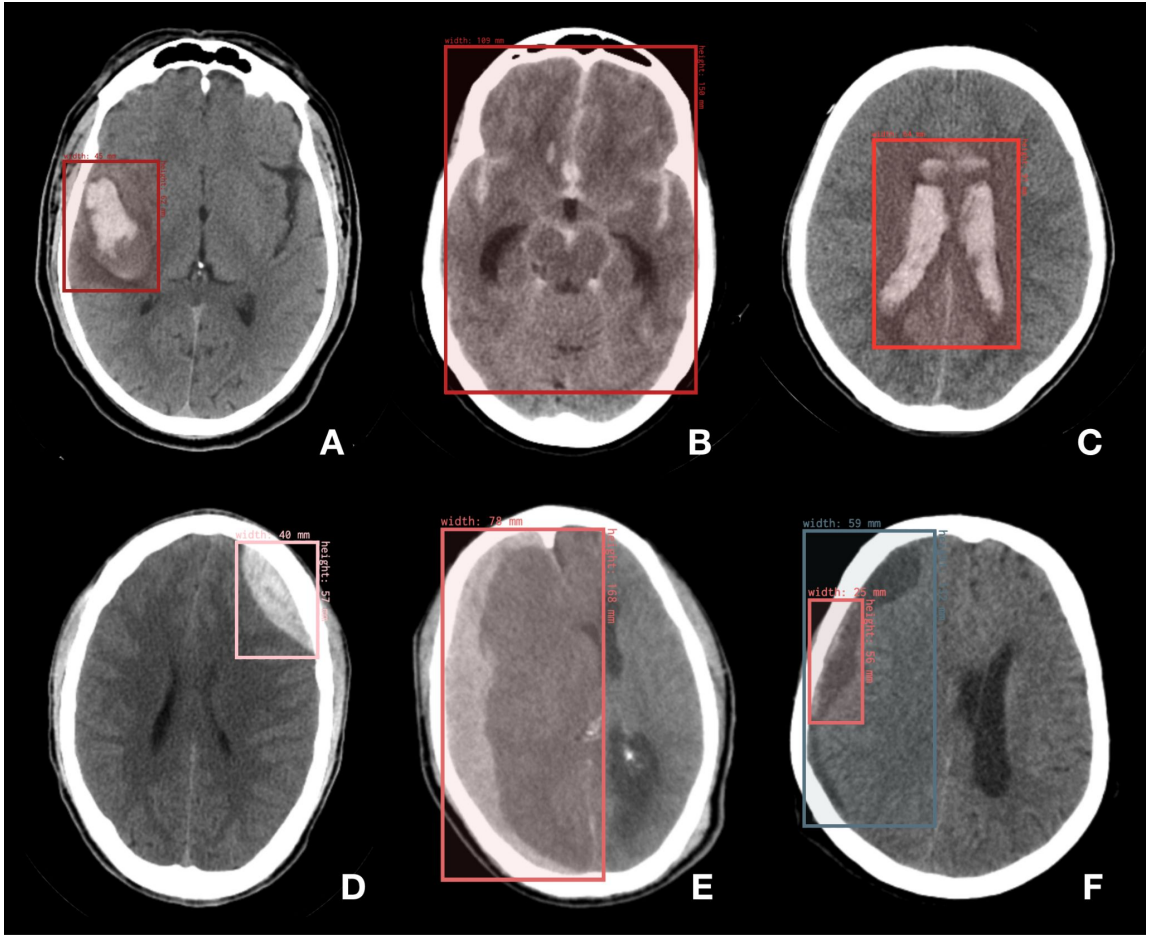


Image source: https://physionet.org/

Object Detection

Two-Stage (Region-Based) Detectors

2 major steps:

1) Region Proposal
    a) Look for regions that are most likely to contain objects, and only look at those
    b) Generate potential regions of interest (RoIs)
2) Classification
    a) RoIs are classified into object categories
    b) Localize bounding box

Single-Stage Detectors

- Directly predict object bounding boxes and class labels in a single pass through the network
- No explicit region proposal step!

YOLO!

YOLO (You Only Look Once)

What?

Object detection algorithm that divides an input image into a grid and predicts bounding boxes and class probabilities for each grid cell

How?

Makes predictions of bounding boxes and class probabilities all at once (completes within a single iteration)

Compared to Faster RCNN

Network?

CONV layers are pre-trained using ImageNet (average pooling, FC layer)

Convert this to perform detection

Different versions of YOLO introduced changes to architecture

Iterative Steps

Grid Based Detection

Divide input image into SxS grid and B subdivisions

Iterate over each B subdivision and predict B bounding boxes + associated class probability

Prediction Outputs

Bounding Box

(x,y), width, height

Class Probability

Corresponds to the total number of classes in dataset

Non Maximum Suppression

NMS

Filter out redundant and overlapping detections by selecting the bounding box with the highest confidence score for each object that doesn't overlap

YOLO 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

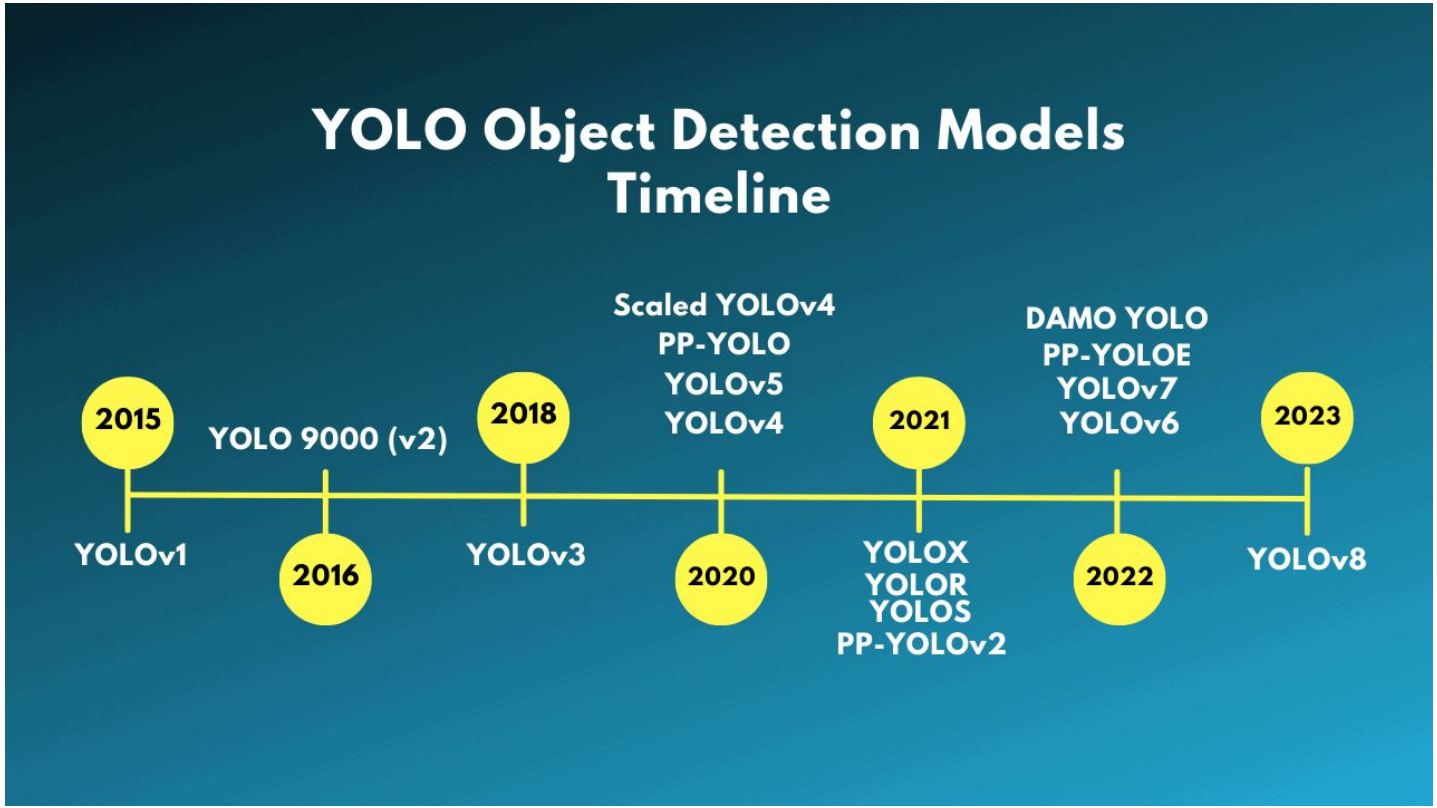
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

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

Classification loss

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

YOLO Family


YOLO Object Detection Models
Timeline
2015
YOLOv1
YOLO 9000 (v2)
2016
2018
YOLOv3
Scaled YOLOv4
PP-YOLO
YOLOv5
YOLOv4
2020
2021
YOLOX
YOLOR
YOLOS
PP-YOLOv2
DAMO YOLO
PP-YOLOE
YOLOv7
YOLOv6
2022
2023
YOLOv8

YOLO 7

Released in 2022

https://arxiv.org/pdf/2207.02696.pdf

Limitations

- Struggles to detect small objects
  - In crowd scenes or far away
- Sensitive to changes in lighting or other environment conditions
- Computationally intensive

Main Improvements

- Uses 9 anchor boxes: detects a wider range of object shapes / sizes
  - Reduces the number of false positives
- New loss function: focal loss
  - Weighs down the loss for well classified examples to focus on harder to detect objects
- Higher resolution
  - Processes images at a resolution of 608 x 608 pixels
- Speed
  - Processes images at a rate of 155 frames per second

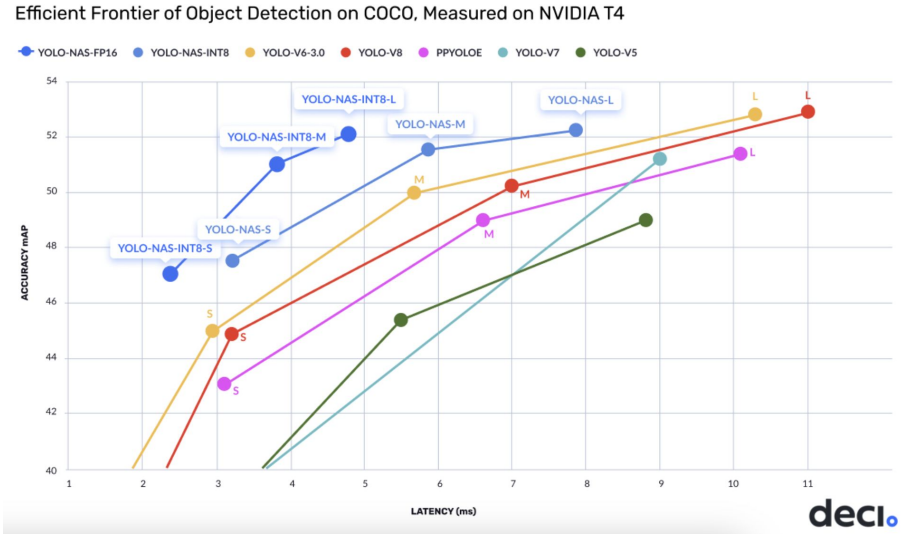
YOLO 7


airplane 0.73
airplane 0.58
airplane 0.68
airplane 0.34
airplane 0.77
airplane 0.80
airplane 0.55
truck 0.75
truck 0.37
truck 0.37
truck 0.71
erson 0.43
37
car 0.82
bus 0.744
car 0.51
truck 0.72
truck 0.75
bus 0.708
bus 0.78

YOLO-NAS

Released in 2023

https://github.com/Deci-AI/super-gradients/blob/master/YOLONAS.md



Main Improvements

- Generated by AutoNAC Engine
  - Neural Architecture Search Technology
- Pretrained on well known datasets
  - COCO, Objects 365, Roboflow 100
- Hybrid Quantization method
- Attention mechanisms
- Inference time reparametrization
- Deci

Current state of the art.

Neural Architecture Search (NAS)

- Technique used to automatically search for optimal neural network architectures for a given task
- Looks for best trade-off between accuracy, computational complexity, and model size
  - Employs optimization algorithms to discover the most suitable architecture for a given task
- **AutoNAC**
  - Architectures of YOLO-NAS models were “found” using Deci’s proprietary NAS technology: AutoNAC
  - Were 10^14 possible architecture configurations in the NAS search space

YOLO-NAS: SOTA Real Time Object Detection Models

LearnOpenCV.com