You Only Look Once: Unified, Real-Time Object Detection

- [[📚 1506.02640.pdf]]
- **Basic Idea**:
  - Can we have a single-stage detector which is faster than two-stage detectors (region-proposal + detection/classification) such as RCNN family?
  - Allow the network to look at the entire image to predict the bounding boxes rather than just local RoI (as in two stage detector). This is indicated by high false positives by RCNN than YOLO.
- **Unified Detection**
  - 🟡 P2

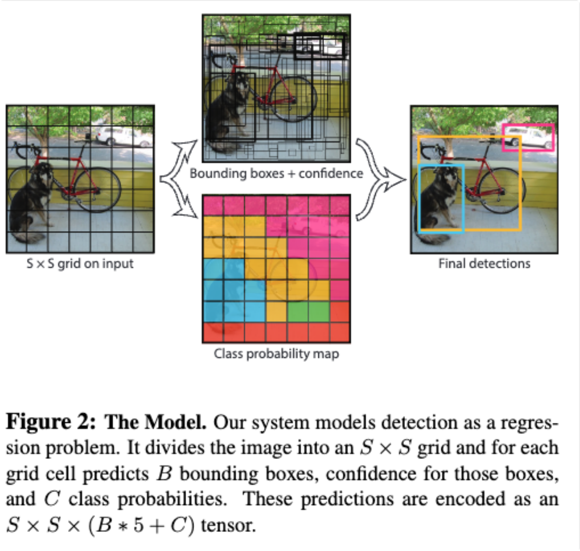
    

    Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.
  - 🟡 P2 Our system divides the input image into an S × S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.
  - 🟡 P2 Each grid cell predicts B bounding boxes and confidence scores for those boxes.
    - 🟡 P2 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.
    - Confidence (C):

      $$C = \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

      - $C = 0$ if no object belongs to the cell.
      - Else, $C = \text{Intersection over Union (IoU)}$ between predicted box and ground truth.
    - 🟡 P2 Each bounding box consists of 5 predictions: x, y, w, h, and confidence.
      - 🟡 P2 The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell.
        The center coordinate is wrt the grid cell because this would enable $[x, y] \in [0, 1] \times [0, 1]$. Moreover, since convolutions have local information, knowing the center wrt to the whole image would be difficult.
      - 🟡 P2 The width and height are predicted relative to the whole image
    - 🟡 P3 YOLO predicts multiple bounding boxes per grid cell. At training time we only want one bounding box predictor to be responsible for each object. We assign one predictor to be "responsible" for predicting an object based on which prediction has the highest current IOU with the ground truth. This leads to specialization between the bounding box predictors. Each predictor gets better at predicting certain sizes, aspect ratios, or classes of object, improving overall recall.
  - 🟡 P2 Each grid cell also predicts C conditional class probabilities, Pr(Classi|Object). These probabilities are conditioned on the grid cell containing an object.
    - Only predict one set of class probabilities per grid cell.

- Implies that an object occurs only once in a grid cell (Doubtful).
- This kind of formulation for sharing conditional probability across bounding boxes helps save number of parameters.

- **Test time**: Multiply conditional probability and individual box confidence predictions which gives class-specific scores for each box:

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

🟡 **P2** These scores encode both the probability of that class appearing in the box and how well the predicted box fits the object

- **Questions**
  - How do they handle cases where there are more than 2 objects being assigned to the same grid cell?
  - 🟡 **P4** each grid cell only predicts two boxes and can only have one class.
    How is this true? The conditional probability is estimated using a linear activation function, hence there should be no competition among classes. 1

- **General Problems with Object Detection/Tricks of the Trade**
  - *Learning fine-grained features.*
    Image classification networks trained at 224×224 whereas the detection network trained at 448×448. 1
    🟡 **P4** Our model also uses relatively coarse features for predicting bounding boxes since our architecture has multiple downsampling layers from the input image
    - *The classic problem of retaining the dense spatial information throughout the network which often gets lost in the classification networks due to max-pooling operation.*

  - *Detection of smaller objects*
    🟡 **P4** YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as flocks of birds 1
    Also don't have any mechanism for multi-scale detection.

  - *Loss function not directly aligned with the goal of maximizing average precision.*
    🟡 **P3** We use sum-squared error because it is easy to optimize, however it does not perfectly align with our goal of maximizing average precision.

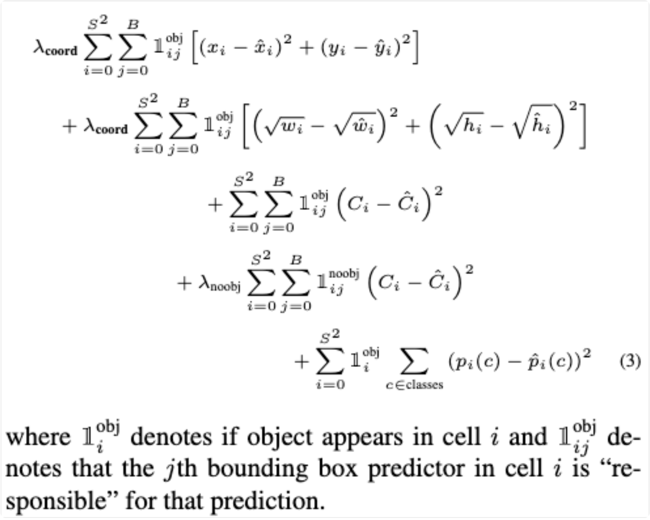
  - *Class imbalance: A lot of bounding boxes don't have any object*
    🟡 **P3** Also, in every image many grid cells do not contain any object. This pushes the "confidence" scores of those cells towards zero, often overpowering the gradient from cells that do contain objects. This can lead to model instability, causing training to diverge early on.
    🟡 **P3** To remedy this, we increase the loss from bounding box coordinate predictions and decrease the loss from confidence predictions for boxes that don't contain objects.
    Parameters $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$ accomplish this.

  - *Small error/deviation in large boxes matter less than in small boxes*
    🟡 **P3** Sum-squared error also equally weights errors in large boxes and small boxes
    🟡 **P3** To partially address this we predict the square root of the bounding box width and height instead of the width and height directly.

  - *Early divergence in training with high learning rate*
    🟡 **P4** Our learning rate schedule is as follows: For the first epochs we slowly raise the learning rate from 10−3 to 10−2. If we start at a high learning rate our model often diverges due to unstable gradients. We continue training with 10−2 for 75 epochs, then 10−3 for 30 epochs, and finally 10−4 for 30 epochs

  - *Generalization to new or unusual bounding boxes*
    🟡 **P4** Since our model learns to predict bounding boxes from data, it struggles to generalize to objects in new or unusual aspect ratios or configurations.

- **Loss Function**
  - 🟡 **P4**



$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i-\hat{x}_i)^2 + (y_i-\hat{y}_i)^2\right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right] \\
&+ \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i-\hat{C}_i\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i-\hat{C}_i\right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c\in\text{classes}} (p_i(c)-\hat{p}_i(c))^2
\end{aligned}
\tag{3}
$$

where $\mathbb{1}_i^{\text{obj}}$ denotes if object appears in cell i and $\mathbb{1}_{ij}^{\text{obj}}$ denotes that the jth bounding box predictor in cell i is “responsible” for that prediction.

  - C_i is the confidence score of object for which bounding box j is "reponsible" in grid cell i
  - $p_i(c)$ conditional probability of class c being in grid i given there is an object in grid i.
  - The last sum happens only if object exists in the grid cell i indicated by 1_i^{obj}. This is required to ensure that the predicted probability represents the conditional probability $\Pr(\text{Class}_i|\text{Object})$.
  - λ_{noobj} can be determined using the fraction of bounding boxes that have no object in them.
    🟡 **P3** In practice α may be set by inverse class frequency or treated as a hyperparameter to set by cross validation.

- **Discussion**
  - *Spatial diversity in bounding box*: 🟡 **P4** The grid design enforces spatial diversity in the bounding box predictions. Often it is clear which grid cell an object falls in to and the network only predicts one box for each object. However, some large objects or objects near the border of multiple cells can be well localized by multiple cells. Non-maximal suppression can be used to fix these multiple detections. While not critical to performance as it is for R-CNN or DPM, non-maximal suppression adds 23% in mAP

  - *Spatial constraints on grid cell proposals*: 🟡 **P5** However, our system puts spatial constraints on the grid cell proposals which helps mitigate multiple detections of the same object
    One can think that by spatial constraints they are able to get rid of the region proposals.

  - *Global context reasoning*: 🟡 **P2** Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method [14], mistakes background patches in an image for objects because it can’t see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN.

▸ Unlinked References