Annex II

The computational complexity of the YOLO model is a summation of the complexity of individual layers L, i.e.

$$C = \sum_{i=1}^{N} C(L_i)$$

As YOLO is a combination of different types of layers, their types and complexities are discussed below:

. Convolutional Layers

. Routing Layers

. Shortcut Layers

. Upsampling Layers

. YOLO Layers

0.1 Convolutional Layers

In convolutional layers, the convolution operation is followed by an activation operation. Now in the convolution operation, for each element of an input feature map $I \in \mathbb{R}^{H \times W \times C}$, convolution with a filter $K \in \mathbb{R}^{H_k \times W_k \times C}$ is defined as:

$$\phi(m,n) = \sum_{i=0}^{H_k-1} \sum_{j=0}^{W_k-1} \sum_{n=0}^{C} I_n(m+i, n+j) \cdot K_n(i,j) \tag{1}$$

Each convolution operation combines multiplication and summation. Thus, for an input I convolved with N filters, the computational complexity $C(cl)$ is:

$$C(cl) = 2 \times (H \times W \times C \times H_k \times W_k) \times N \tag{2}$$

0.2 Routing, Shortcut, and Upsampling Layers

Routing and Shortcut layers do not contribute directly to computational complexity, as they simply add outputs from previous layers without extra computation. In the Nearest Neighbor Upsampling method, pixel values from the input tensor are duplicated to form a larger spatial grid. This process does not involve floating-point arithmetic and is purely a memory operation.

0.3 YOLO Layers

To estimate the computational complexity of a YOLO layer in FLOPs (Floating Point Operations Per Second), we consider the major operations performed by the layer:

1. Bounding Box Prediction

For each anchor in each grid cell, the model predicts:

- 4 bounding box attributes: center (x, y), width w, and height h.
- 1 objectness score.
- C class probabilities.

Total predictions per anchor:

$$P_{\text{per anchor}} = 5 + C$$

2. Grid and Anchor Coverage

Total predictions across all anchors and grid cells:

$$P_{\text{total}} = H \times W \times A \times P_{\text{per anchor}}$$

Where:

- $H \times W$: Grid size (e.g., 13×13).
- A: Number of anchors (e.g., 3, as per mask configuration).

3. Floating Point Operations

a. Sigmoid Activation

Applied to each prediction (objectness + class probabilities).

FLOPs for sigmoid:

$$\text{FLOPS}_{\text{sigmoid}} = P_{\text{total}} \times \text{Cost}_{\text{sigmoid}}$$

Sigmoid approximately requires 4 FLOPs (1 exponential + 1 division), so:

$$\text{FLOPS}_{\text{sigmoid}} \approx P_{\text{total}} \times 4$$

b. Bounding Box Transformation

Each bounding box requires additional operations to compute:

- Scaled offsets (e.g., $b_x = \sigma(t_x) + c_x$)
- Scaling widths and heights by anchor sizes

FLOPs per bounding box:

$$\text{FLOPS}_{\text{bbox}} = H \times W \times A \times 4$$

Combining all operations across all predictions, the computational complexity of the YOLO layer can be written as:

$$C(yolo) = P_{\text{total}} \times 4 + H \times W \times A \times 4$$

0.4 Combined Computational Complexity

From the discussion above, the combined computational complexity of a YOLO detector with A convolution layers and B YOLO layers can be written as:

$$C = A \times C(cl) + B \times C(yolo)$$