YOLOv8 Architecture

Liu Yichao

Outline

Mathematical Notations

- X: Input feature maps
- X_{first}, X_{second}: Split portions of input
- F_i: i-th convolutional block
- $S \times S$: Detection grid size
- p_c: Confidence score
- (b_x, b_y): Center coordinates
- (b_w, b_h): Width and height
- W, b: Learnable parameters
- σ: Sigmoid activation
- L_{cls}, L_{box}, L_{dfl}: Loss components
- IoU: Intersection over Union
- ρ: Euclidean distance between centers
- T: IoU threshold for NMS
- d: Local detection density

C2f Backbone Module

- Replaces traditional CSP blocks with more efficient C2f
- Uses dense connections for better gradient flow
- Reduces computational complexity by 15%
- Mathematical form:

$$C2f(X) = Concat[X_{first}, F_n(F_{n-1}(...F_1(X_{second})))] \tag{1}$$

- X_{first}, X_{second}: Split portions of input
- F_i: i-th convolutional block

Anchor-Free Detection Head

- Fully anchor-free approach
- Each grid cell directly predicts:

$$\hat{y} = \{p_c, b_x, b_y, b_w, b_h, p_1, p_2, ..., p_C\} \tag{2}$$

- No predefined anchor boxes
- More flexible object localization
- p_c: Confidence score
- (b_x, b_y): Center coordinates
- (b_w, b_h): Width and height

Decoupled Head Architecture

- Separate branches for:

$$\text{Classification: } C(F) = \sigma(W_c \cdot F + b_c) \tag{3}$$

$$\text{Regression: } R(F) = W_r \cdot F + b_r \tag{4}$$

$$\text{Confidence: } O(F) = \sigma(W_o \cdot F + b_o) \tag{5}$$

- Independent optimization for each task
- Reduces training conflicts
- W, b: Learnable parameters
- σ: Sigmoid activation

Loss Function Formulation

Composite loss:

$$L_{total} = \lambda_{cls} L_{cls} + \lambda_{box} L_{box} + \lambda_{dfl} L_{dfl} \tag{6}$$

- L_{cls}: Binary Cross-Entropy for classification
- L_{box}: CIoU loss for bounding box regression
- L_{dfl}: Distribution Focal Loss for coordinate modeling

Dynamic NMS Implementation

- Adaptively adjusts IoU threshold based on detection density

$$T = T_{base} \cdot \exp(-\beta \cdot d) \tag{7}$$

- Better performance in crowded scenes
- Allows more detections in dense regions
- Maintains strict filtering in sparse areas
- β: Hyperparameter for adjustment
- T: IoU threshold for NMS
- d: Local detection density

Results

- 53.9 AP on COCO dataset
- Inference time: 0.39 ms on modern GPUs
- Significant advancement in speed-accuracy trade-off
- State-of-the-art for real-time object detection