CS190I: Generative AI, Spring 2025
Programming Assignment 1

Joe Lee

April 27, 2025

1 Algorithm and Dataset Selection

We implemented YOLOv1 following the architecture in "You Only Look Once: Unified, Real-Time Object Detection" [1]. Our implementation uses the Pascal VOC dataset with a 7×7 grid, 2 bounding boxes per cell, and 20 classes.

2 Implementation Details

2.1 Enhanced Loss Function

The total loss function combines four components:

$$L_{\text{total}} = \lambda_{\text{coord}} L_{\text{coord}} + L_{\text{obj}} + \lambda_{\text{noobj}} L_{\text{noobj}} + L_{\text{class}} \tag{1}$$

where $\lambda_{\text{coord}} = 5$ and $\lambda_{\text{noobj}} = 0.5$. The coordinate loss uses square root scaling for width and height:

$$L_{\text{coord}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \tag{2}$$

The objectness and no-object losses are calculated as:

$$\begin{aligned} L_{\text{obj}} &= \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ L_{\text{noobj}} &= \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ L_{\text{class}} &= \sum_{i=0}^{S^2} \mathbf{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \tag{3}$$

2.2 Training Pipeline

Key optimizations include:

- Adam optimizer ($lr = 2e-5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$)
- Validation every 9 epochs with mAP:

$$\text{mAP} = \frac{1}{|C|} \sum_{c \in C} \text{AP}(c) \tag{4}$$

2.3 Data Processing

Grid cell assignment and normalization:

$$\begin{aligned} i &= \lfloor y_{\text{center}} \times S \rfloor \\ j &= \lfloor x_{\text{center}} \times S \rfloor \\ x_{\text{norm}} &= \frac{x_{\text{center}} - j}{S} \\ y_{\text{norm}} &= \frac{y_{\text{center}} - i}{S} \end{aligned} \tag{5}$$

2.4 Post-processing

Non-maximum suppression uses IoU:

$$\text{IoU}(box_1, box_2) = \frac{\text{area}(box_1 \cap box_2)}{\text{area}(box_1 \cup box_2)} \tag{6}$$

3 Training Statistics

3.1 Progress

Epoch	Loss	mAP	Time/Epoch
1	168.08	0.1543	29.13
15	60.23	0.2255	28.54
30	30.72	0.2222	29.34
45	21.04	0.2207	30.04
60	16.35	0.2259	29.80
75	12.95	0.2293	32.82
90	10.86	0.2213	30.40
105	9.33	0.2138	30.39
120	8.08	0.2131	32.84
135	7.22	0.2203	29.81

3.2 GPU Requirements

- Batch size: 80
- Memory: 24 GB
- Training time: 3.08 hours

4 Running Instructions

4.1 Setup and Training

Required packages:

```
torch>=1.7.0
torchvision>=0.8.0
numpy>=1.19.0
opencv-python>=4.4.0
matplotlib>=3.3.0
tqdm>=4.50.0
```

Training configuration:

```
LEARNING_RATE = 2e-5
BATCH_SIZE = 80
EPOCHS = 135
NUM_WORKERS = 4
PIN_MEMORY = True
```

4.2 Testing on Custom Images

```
python predict.py --image path/to/image.
    jpg --model checkpoints/best_model.
    pth.tar
```

5 Lessons Learned

Loss function weight tuning proved critical for model performance, while square root scaling helped handle varying object sizes. Grid-based detection required special attention to boundary objects.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv:1506.02640*, 2016.