Assignment5 Report

1. includes metrics (IoU, mAP), plots, visualizations, and a brief discussion.

Metric	Part A (10K images)	Part B (55K images)
mAP@0.50	0.9211	0.9769
mAP@0.50-0.95	0.6275	0.8010
Average IoU	0.8439	-
Precision	0.88	0.9715
Recall	0.811	0.9297

2. Short discussion on YOLO performance vs two-stage detectors;

YOLO Advantages:

1. **Speed:** Single-stage architecture processes images in one forward pass (~1ms per image), making it significantly faster than two-stage detectors like R-CNN/Faster R-CNN
2. **Real-time Performance:** Achieves high throughput (~1000 FPS on GPU) suitable for real-time applications
3. **Simplicity:** End-to-end learning with fewer components to tune
4. **Resource Efficiency:** 3M parameters, 8.1 GFLOPs

YOLO Limitations:

1. **Accuracy:** While achieving excellent performance (0.977 mAP@0.50), two-stage detectors typically achieve slightly higher mAP@0.50-0.95
2. **Small Objects:** Limited resolution per grid cell can struggle with very small objects
3. **Dense Detection:** Less effective when objects appear very close together

In this project YOLO is ideal:

- 64x64 RGB images with 1-2 digits per image are well-suited for YOLO's architecture
- Speed is valuable for practical deployment
- The achieved accuracy (0.977 mAP@0.50) is excellent

3. Clear reasoning on speed vs accuracy;

Speed vs Accuracy Trade-off:

Model	Speed	mAP@0.50	mAP@0.50-0.95
YOLOv8n (subset)	~1ms	0.921	0.628
YOLOv8n (full)	~1ms	0.977	0.801

- **0.977 mAP@0.50 is sufficient** for digit detection in practical applications
- **1ms inference time enables real-time processing** at 30-60 FPS, critical for video applications
- **Speed is more valuable** than the potential 1-2% accuracy gain from two-stage detectors for this use case
- The model achieves optimal balance: **fast enough for real-time deployment while maintaining high accuracy**

For MNISTDD, the speed-accuracy trade-off favors YOLO because the task does not require the absolute highest accuracy that two-stage detectors provide, and real-time processing capability is more important for practical digit detection scenarios.

4. Motivate your choice for learning rate, confidence, and NMS thresholds

Learning Rate (lr0 = 1e-3):

- **Standard for YOLO fine-tuning** on COCO pretrained weights - proven to work well
- **Not lower:** With 15-20 epochs on a relatively simple task, higher LR ensures adequate learning within limited training time
- **Not higher:** Risk of instability and overshooting optimal weights during fine-tuning

Confidence Threshold (0.35):

- **Determined through systematic grid search** on validation set comparing [0.25, 0.3, 0.35, 0.4]
- **Why 0.35 vs default 0.25:** The model on full dataset produces high-confidence predictions due to consistent digit appearance, allowing higher threshold to filter false positives without losing true positives
- **Why 0.35 vs 0.40:** Maintains optimal balance - 0.35 preserved high recall (0.930) while improving precision to 0.972
- **Impact:** Reduced false positives without significant loss in true positive detections

NMS Threshold (0.45):

- **Determined through systematic grid search** comparing [0.4, 0.45, 0.5, 0.55]
- **Why 0.45 vs default 0.5:** Digits in MNISTDD are typically well-separated with little overlap, so slightly lower threshold prevents over-suppression of close-but-distinct detections
- **Why 0.45 vs 0.4:** Still effectively removes duplicate detections while being more conservative than 0.5
- **Impact:** Optimal suppression of duplicates without over-suppressing valid detections