Bishop's University
CS 590 – Master's Project
Challenge 3: Advanced Deep Learning Architectures for Object Detection and Image Segmentation

Part I – Object Detection with YOLO Variants (Total: 100 points)

Part I of this project is dedicated to advanced object detection using the YOLO framework. Students will begin by studying the foundational YOLO architecture and its loss function, then progress to modifying the architecture by replacing the backbone feature extractor with a variety of modern networks including VGG, ResNet, DenseNet, Inception, and Transformer-based models. Each model will be trained and evaluated on the PASCAL VOC dataset, with a focus on analyzing how different backbone choices impact detection performance in terms of mAP, loss convergence, and computational efficiency. This part provides hands-on experience in deep model integration, performance tuning, and critical analysis of results.

Q1. Understanding the Existing YOLO Framework (15 pts)

Explain the YOLO loss function, bounding box predictions, and the modifications introduced in the provided baseline architecture.

Expected: Written explanation (1–2 pages) of objective function, supporting diagrams or sketches illustrating model flow and bounding box logic.

Q2. Backbone Replacement and Implementation (30 pts total)

Q2.1. Replace DetNet in the starting code with VGG-16 and train on PASCAL VOC. (7 pts)

Q2.2. Repeat with ResNet-50 and analyze improvements. (7 pts)

Q2.3. Implement and evaluate DenseNet-121. (7 pts)

Q2.4. Integrate Inception v3 and discuss complexity trade-offs. (5 pts)

Q2.5. Integrate a Transformer-based backbone (ViT or Swin). (4 pts)

Expected: Clear architectural modifications in code, description of the process in the report, successful model training runs, and concise evaluation logs with commentary.

Q3. Training & Evaluation (30 pts)

Train each model, monitor loss and mAP. Compare performance metrics and include visualization of detections.

Expected: Metric tables comparing mAP, training loss over epochs, and qualitative results with at least four annotated images per model.

Q4. Analysis & Discussion (25 pts)

Discuss which backbone yielded the best trade-offs and include architectural, performance, and resource analysis.

Expected: A well-structured PDF report (2–3 pages) covering performance vs. efficiency, GPU usage, and rigorous interpretations of the obtained results.

Part II – Image Segmentation (Total: 100 points)

Part II of the project focuses on image segmentation techniques, transitioning from object detection to pixel-level understanding of visual scenes. Students will explore semantic and instance segmentation by implementing and experimenting with key models such as U-Net, DeepLabv3+, Mask R-CNN, and transformer-based architectures like SegFormer. These models will be evaluated on their ability to segment objects accurately under varying conditions, using PASCAL VOC dataset. Emphasis will be placed on understanding segmentation architectures, tuning hyperparameters, and interpreting both qualitative outputs and quantitative metrics such as mIoU and pixel accuracy. This part encourages deeper reflection on the challenges of dense prediction in vision.

Q1. Baseline U-Net Implementation (15 pts)

Implement U-Net and train on PASCAL VOC segmentation. Evaluate with mIoU and pixel accuracy.

Expected: Training log, sample predictions on test images, mIoU and pixel accuracy, and discussion of results.

Q2. DeepLabv3+ with ResNet Backbone (20 pts)

Train or fine-tune DeepLabv3+. Compare with U-Net and report segmentation metrics.

Expected: Training curves, metric comparison table, and segmentation map visualizations.

Q3. Instance Segmentation with Mask R-CNN (20 pts)

Train a Mask R-CNN model on PASCAL VOC. Evaluate segmentation masks using AP metrics.

Expected: Evaluation on instance masks (AP), mask overlay visualizations on at least four images, and architecture notes.

Q4. Transformer-based Model: SegFormer or SETR (25 pts)

Use or train a Transformer-based model. Compare its performance against CNN-based methods.

Expected: Results from pretrained/fine-tuned transformer model, qualitative comparisons, and metric tables.

Q5. Comparative Analysis and Discussion (20 pts)

Create comparative tables across all models. Discuss practical deployment and performance trade-offs.

Expected: A structured document summarizing key comparisons (accuracy, complexity, runtime) with critical analysis.

Resources

- Lecture on object detection,
- Original YOLO paper,
- Great post about YOLO on Medium,

Submission

You will need to submit all programmed solutions. Please provide a PDF report that details the team members, each person's contribution, and the results obtained, named ***final_report_CS590_challeng3.pdf***. Additionally, consolidate all your code (Python files and Jupyter notebook) into a single ZIP file named ***final_report_CS590_challeng3_code.zip***. Convert your Jupyter notebook with output cells into a separate PDF format, named ***final_report_CS590_challeng3_output.pdf***. In both the ***final_report_CS590_challeng3.pdf*** and the Python notebook file, answer all questions from Part I and Part II, ensuring to highlight these answers for clarity. The

distinction is important: ***final_report_CS590_challenge3.pdf*** focuses on contributions, textual answers and screenshots of the obtained results, while ***final_report_CS590_challenge3_output.pdf*** contains the actual code outputs.

Good luck :)