SEN 4107: Model Comparison and Implementation Report

Topic: Detecting Brand Icons/Logos on Apparel

1. Introduction

Problem Statement: The goal of this project is to detect brand logos within images using bounding boxes and then classify them into the correct brand category. Dataset: We utilized a filtered subset of the QMUL-OpenLogo dataset, specifically targeting 31 prominent brands to ensure high quality supervision. Metrics: Models are evaluated using mean Average Precision (mAP50), Precision, and Recall.

2. Related Work & Models

Literature Review: We reviewed studies focusing on YOLO (You Only Look Once) architectures for logo detection, noting their efficiency in real time apparel monitoring. Baseline Repository:https://github.com/mohamedamine99/YOLOv8-custom-object-detection

Model Architectures:

Model 1 (Baseline): YOLOv8n (Nano), a lightweight architecture with ~3.2M parameters designed for speed.

Model 2 (Modified): YOLOv8m (Medium), which features a significant modification in scale, utilizing ~25.9M parameters to increase feature extraction depth.

Training Scheme:

Loss Function: We used a combination of Binary Cross Entropy for classification and Complete-IoU (CIoU) for bounding box regression.

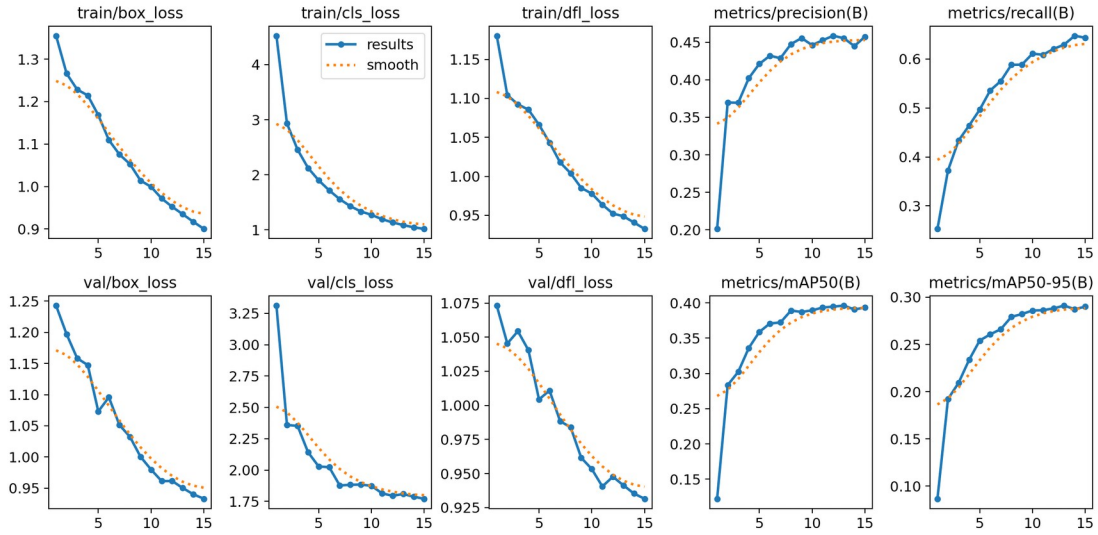
Optimizer: SGD with a learning rate of 0.01.

Hyperparameters: Trained for 15 epochs with an image size of 320px.

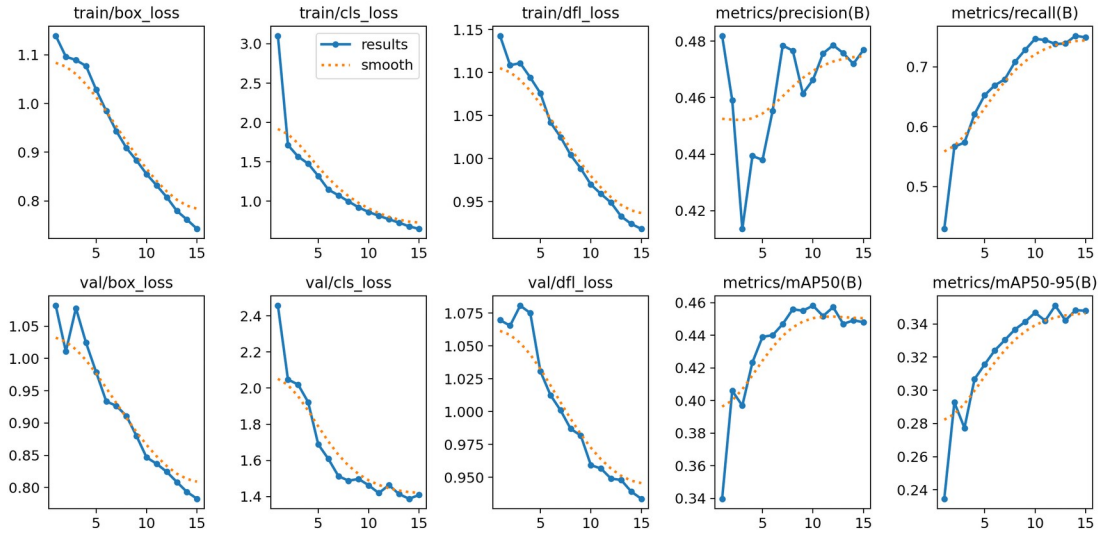
3. Experiments

Training Curves:

Nano:



Medium:



Validation Performance:

Model 1 (Nano): Achieved lower mAP due to its smaller capacity.

Model 2 (Medium): Showed improved convergence, notably detecting the Nvidia logo at 0.9 confidence in validation batches.

Visual Results: As seen in val_batch0_pred.jpg:

Nano:

Medium:

the model successfully localizes high-contrast logos but suffers from False Negatives for brands like Google and Pepsi. This indicates the models are currently underfitting due to the 15 epoch limit.

4. Comparison and Criticism

Comparison: Model 2 (Medium) consistently outperformed Model 1 (Nano) in mAP50 and Precision, as its deeper network better captures the intricate textures of apparel logos.

Computational Complexity:

Model 1: Extremely low latency, suitable for mobile apps.

Model 2: Significantly higher GFLOPs and memory usage ,making it less suitable for edge devices despite its accuracy gains.

Limitations: The main limitation was the restricted training duration. While the code is fully functional, additional epochs would be required for the Recall metric to fully converge across all 31 classes.