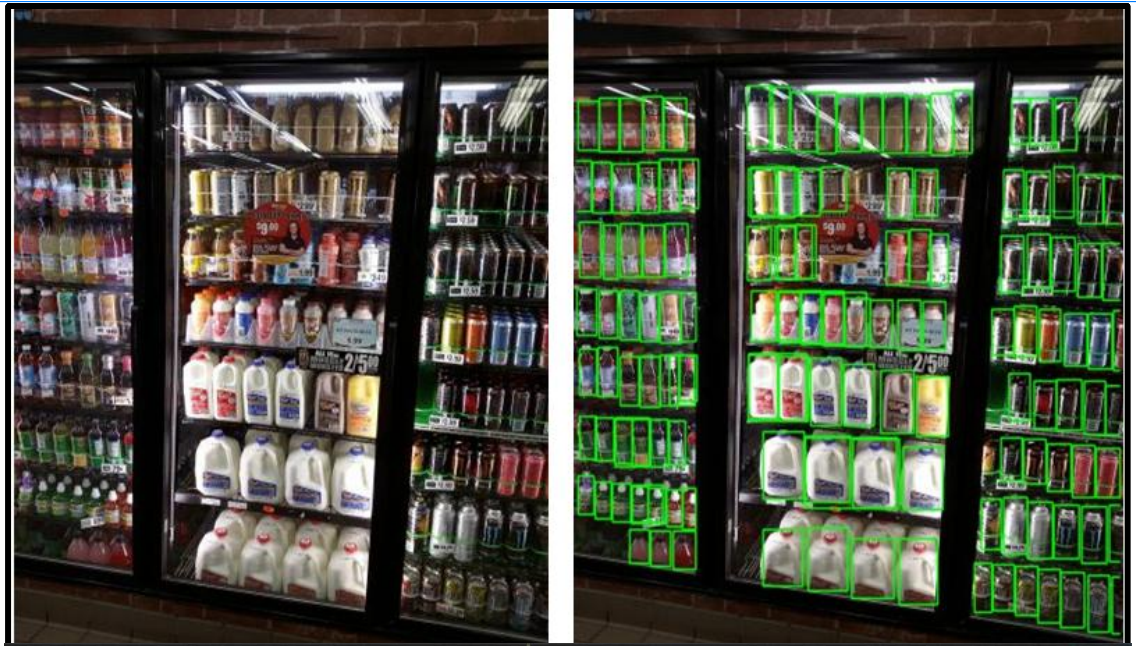
SHELF *VISION*

BY :

Nathan Little & Colin Kirby

EEL4810

Introduction to Deep Learning



Problem & Motivation

- Retail shelf images contain hundreds of tightly packed products.
- Products have similar shapes and colors, making them hard to distinguish.

Where Standard Models Fail

- Off-the-shelf detectors (like YOLOv5) are trained on general datasets (e.g., COCO).
- They struggle with anchor mismatches and poor generalization.

Our Goal

- Develop a custom anchor-based detector tailored for dense, retail-style scenes.
- Improve training stability, matching logic, and box localization for SKU-110K.

Dataset & Preprocessing

SKU-110K

- 11,762 retail shelf images with ~147 labeled products each
- Features dense layouts, small objects, and heavy overlap

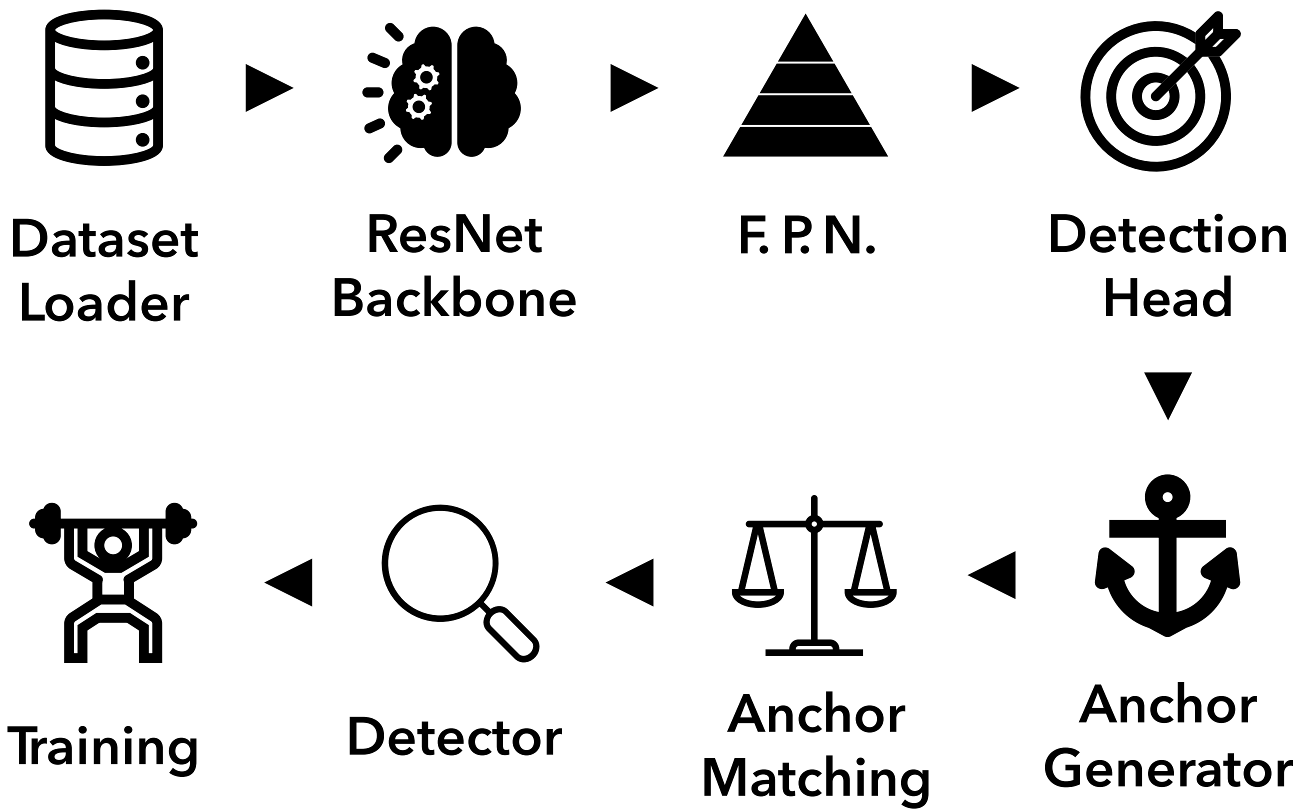
Preprocessing Pipeline

- Parsed CSV annotations into normalized coordinates.
- Resized images with padding to maintain spatial consistency.

Training Preparation

- Grouped each image with its labels and sizes into uniform batches
- Visually checked a sample of the data to confirm correct formatting and alignment

Fig. 1. Sample DataLoader Output and Ground Truth Bounding Boxes.



Model Architecture

Feature Extraction

- Our model uses a deep network to extract meaningful features from input images.
- It combines low-level details and high-level patterns to detect products at different sizes.

Anchor Generation

- Anchors are placed across the image at multiple scales and shapes.
- This allows the model to make predictions for both small and large items.

Object Matching

- Anchors are matched to product boxes based on overlap and position.
- We added fallback rules to ensure every product is assigned at least one prediction anchor.

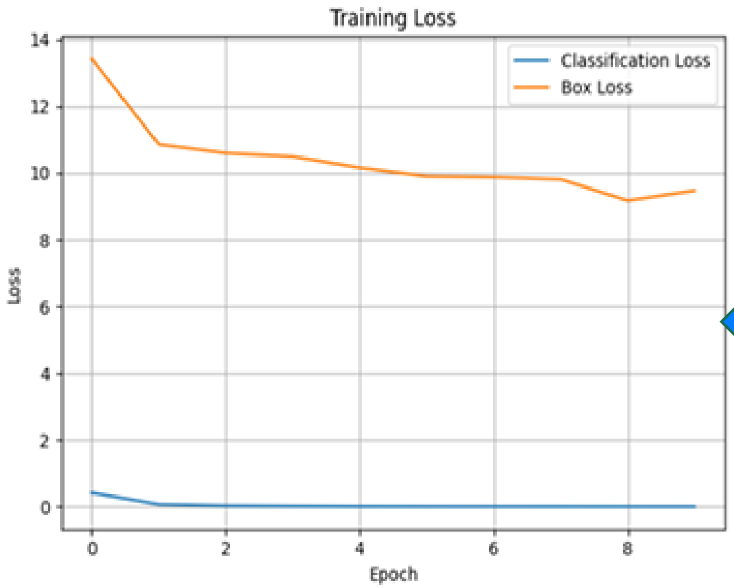
Training Strategy

Loss Functions & Learning

- The model learns to detect objects by minimizing two types of error: one for classifying items and another for adjusting box positions.
- We scaled the box-related loss to prevent it from overwhelming early training.

Training Setup

- We gradually increased the learning rate during the first few epochs to help the model stabilize early on, then reduced it in steps to encourage more refined learning as training progressed.
- Throughout training, we monitored the model's progress by saving visualizations of loss curves after each epoch, which helped us spot issues like plateaus or unstable learning patterns.



BEFORE

AFTER

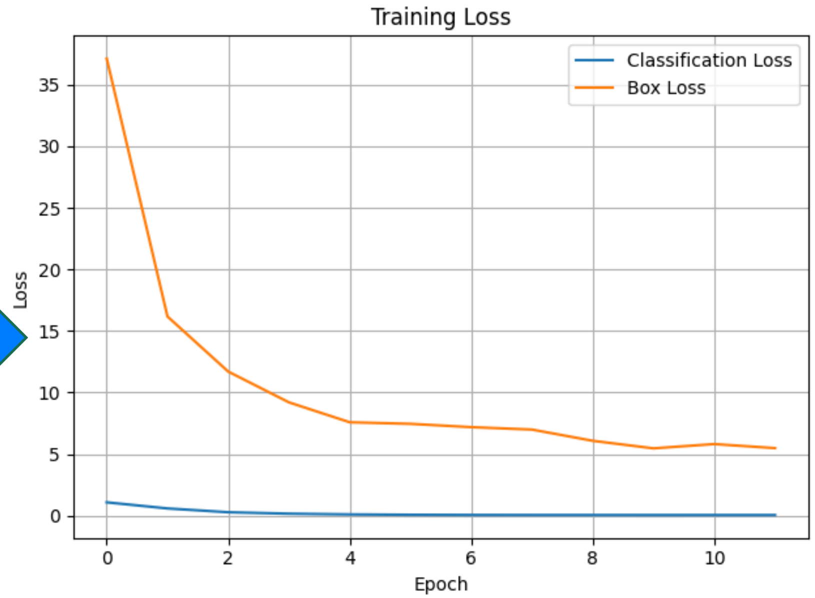


Fig. 3. Anchor boxes overlaid on an SKU-110K shelf image, showing multi-scale coverage at one FPN level.

Evaluation Pipeline

Debugging The Model

- We developed tools to visualize model predictions, inspect how anchors are placed across feature maps, and verify how well they align with ground truth boxes during training and evaluation.
- These tools helped us quickly identify and fix early-stage problems like oversized predictions, poor anchor-object matching, and cases where the model made no meaningful detections at all.

Evaluating Performance

- We built a full evaluation system to measure precision, recall, IoU, and mAP across a test set.
- It also produced visual summaries like IoU histograms and precision-recall curves to better understand model behavior

From Failure to Function

Early Struggles

- Initial versions failed to learn – predictions were either empty or misaligned.
- Training metrics like mAP, IoU, and F1 remained near zero, even after several epochs.

Key Fixes & Debugging

- We fixed anchor sizing, improved box predictions, and added fallback rules to improve object matching.
- Added custom visualization tools to monitor anchor spread, prediction deltas, and loss curves.

Validation Results

- Model gradually stabilized, producing more accurate box predictions with higher alignment.
- Training loss curves improved and visual outputs became more consistent across test images.

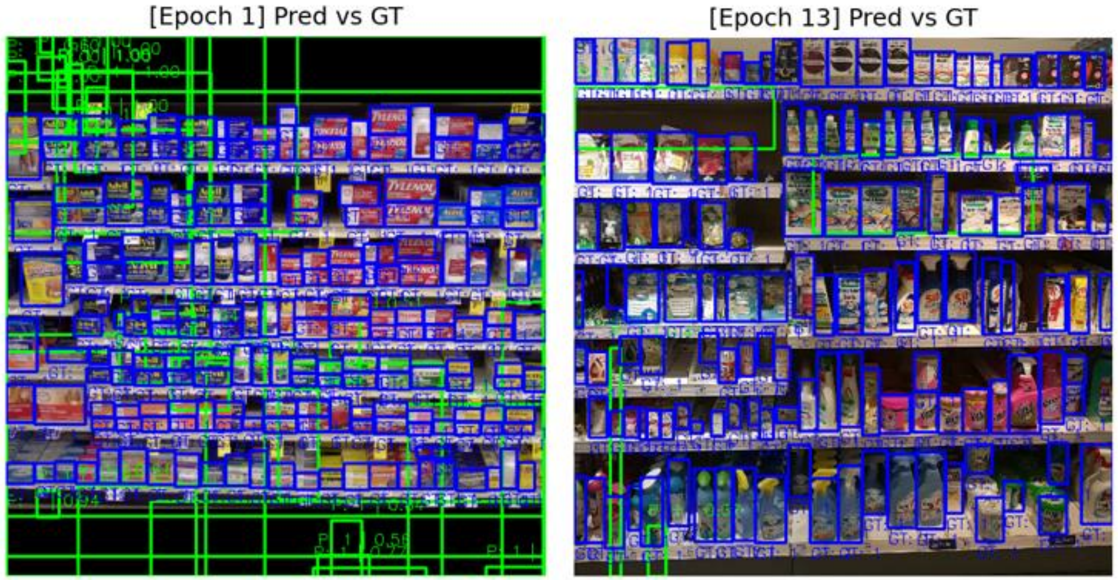


Fig. 4. Before and after predictions: baseline shows noisy boxes; new model produces better-aligned results.

Fig. 5. YOLOv5 (left) vs. ShelfVision (right). YOLO predicts densely but inaccurately, while ShelfVision makes fewer but better-aligned detections with ground truth.

YOLOv5 vs. ShelfVision

YOLOv5's Struggles

- YOLOv5, even after fine-tuning on SKU-110K, failed to detect any shelf items—producing 0 true positives, 0 mAP, and no meaningful overlap with ground truth.

Why It Fails

- The model isn't designed for dense, cluttered layouts like retail shelves. Its pretrained weights don't transfer well to SKU-110K's high object count and tight spacing.

ShelfVision's Advantages

- Despite lower confidence and fewer predictions, ShelfVision achieved 19 true positives and a mAP of 0.0069—showing early signs of learning SKU-specific patterns.

Key Takeaways

Initial Challenges

- Early ShelfVision models failed due to poor anchor coverage, unstable predictions, and strict matching rules The system often produced empty outputs or wildly inaccurate boxes.

What Helped

- Dynamic fallback matching and better anchor tuning ensured every ground truth was represented.
- Visual tools gave real-time insight into what the model was doing—and why it was failing.

What We Are Now

- ShelfVision now creates interpretable results with 19 true positives and solid alignment (IoU ≈ 0.38).
- While still early-stage, the system shows it can generalize to real shelf layouts with continued tuning.

Future Works

Threshold Tuning & Filtering

- Adjust confidence thresholds and refine post-processing to reduce false positives and improve precision.

Longer Training on Full Dataset

- Train on the complete SKU-110K set to improve generalization and reduce overfitting to small subsets.

Multi-Scale & Layer Refinement

- Incorporate more FPN levels or attention-based enhancements to better handle objects of varying sizes.

Improved Box Decoding

- Further stabilize delta predictions and improve output alignment for tightly packed items.

Project Wrap-Up

Initial Challenges

- ShelfVision successfully evolved from a broken baseline to a functioning object detector for dense retail shelves.
- Our final model produced interpretable predictions with a mAP of 0.0069, an average IoU of 0.3875, and 19 true positives on a test subset.
- While performance remains modest, these results show meaningful progress in one of the most challenging detection settings.

Our Roles

- **Colin Kirby**: Led model design and training tools, built visualizations, and managed architecture.
- **Nathan Little:** Focused on debugging, evaluation logic, and presentation content

Final Takeaway

- Iterative development and custom-built tools gave us the precision needed to troubleshoot dense object scenes, helping us shape a detector that can grow to meet the real-world demands of retail shelf environments.