PROJECT PROGRESS REPORT: WEEK 1

Project Title: Smart Bin Waste Classification System
Module: Deep Learning Model Development & Validation
Date: December 15, 2025

1. SUMMARY

This report documents the technical milestones achieved during the first week of the Smart Bin project. The primary objective was to engineer a computer vision backend capable of distinguishing between **Biodegradable** and **Non-Biodegradable** waste in real-time.

To achieve this, two distinct Deep Learning architectures were trained and evaluated:

1. **YOLOv11n (You Only Look Once - Nano):** An object detection model designed for locating and identifying multiple items simultaneously.
2. **MobileNetV4 (Small):** A lightweight image classification model designed for speed on embedded devices.

Key Outcome:
The YOLOv11n model demonstrated superior performance for the project's requirements, achieving a 80.6% accuracy in identifying Non-Biodegradable recyclables. This model has been selected for hardware deployment in Week 2.

2. METHODOLOGY

2.1 Dataset Engineering

A custom dataset was curated to simulate real-world waste scenarios.

- **Data Sources:** The dataset aggregates images from the TrashNet repository (recyclables) and the Waste Classification Data (organic waste).
- **Class Mapping:** To match the physical sorting mechanism of the Smart Bin, diverse waste labels were consolidated into binary categories:
  - **Biodegradable:** Paper, Cardboard, Food Waste, Organic Matter.
  - **Non-Biodegradable:** Plastic, Glass, Metal, Wrappers.
- **Preprocessing:** All images were resized to 640x640 (for YOLO) and 224x224 (for MobileNet) and augmented with rotation and flips to ensure robustness against different camera angles.

2.2 Model Selection Rationale

- **YOLOv11n:** Chosen for its ability to perform *Object Detection*. Unlike simple classification, this model can identify multiple pieces of trash in a single frame and draw bounding boxes, which is crucial if the user throws multiple items at once.

- **MobileNetV4:** Selected as a benchmark for *Image Classification*. It is highly efficient but treats the entire image as a single object, serving as a fallback option if YOLO proved too computationally heavy for the Raspberry Pi.

3. TRAINING ANALYSIS

3.1 YOLOv11n Analysis (Object Detection)

The YOLO model required a rigorous two-stage training strategy to overcome initial accuracy plateaus.

- **Training Cycle:**
  - **Phase 1 (Epochs 1–50):** The model reached an initial convergence with an mAP@50 (Mean Average Precision) of approximately 68%.
  - **Phase 2 (Fine-Tuning, Epochs 51–100):** A "Fine-Tuning" session was initiated using the weights from Phase 1. Despite an initial dip in accuracy due to learning rate adjustment, the model recovered and stabilized with significantly lower classification error.
- **Performance Metrics:**

Metric	Score	Interpretation
Overall mAP@50	**72.4%**	Strong general detection capability.
Non-Biodegradable	**80.6%**	Excellent reliability for detecting plastics/metals.
Biodegradable	**64.2%**	Moderate reliability; struggles with visual ambiguity of crumpled paper/organic piles.

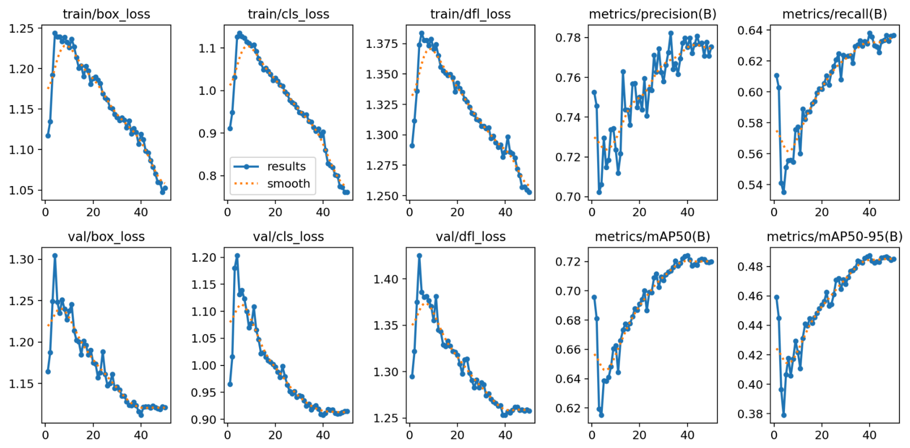


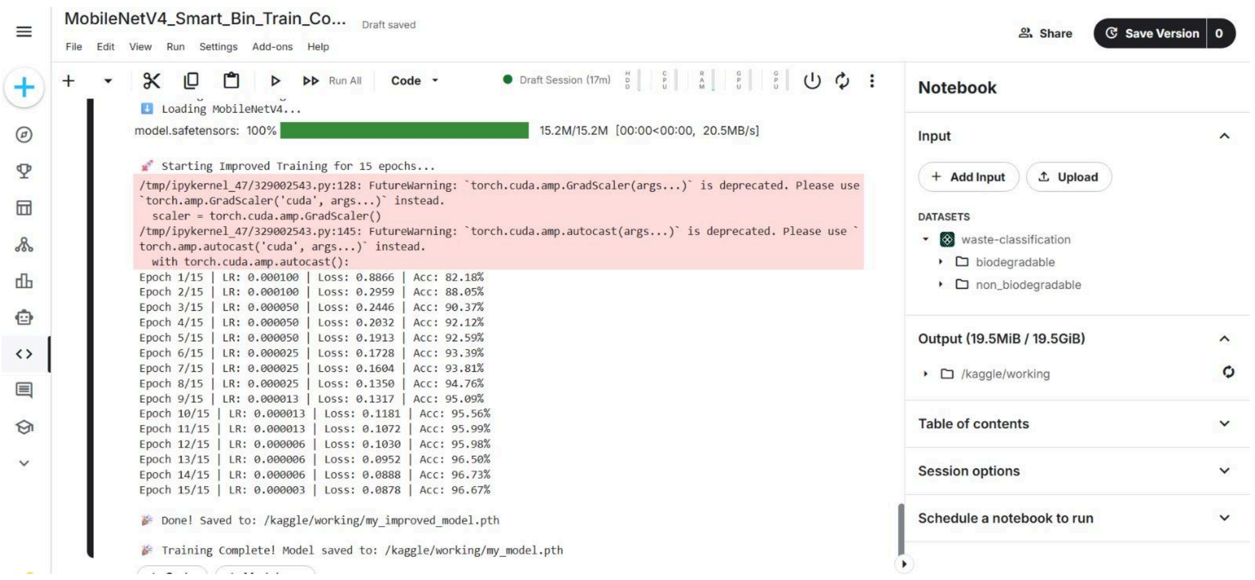
Figure 1: YOLOv11n training curves showing the reduction in Box Loss over 100 epochs.

3.2 MobileNetV4 Analysis (Image Classification)

The MobileNetV4 model was trained to provide a lightweight, high-speed alternative for single-item classification.

- **Training Cycle:**
  - **Configuration:** The mobilenetv4_conv_small architecture was used with Transfer Learning (pretrained on ImageNet).
  - **Progression:** The model was trained for **15 Epochs** using a specialized Learning Rate Scheduler.
  - **Convergence:** The model showed rapid improvement, jumping from ~82% accuracy in Epoch 1 to peak performance by Epoch 9.
- **Performance Metrics:**

Metric	**Score**	**Interpretation**
Training Accuracy	**95.09%**	Extremely high confidence on the training set.
Training Loss	**0.1317**	Very low error rate, indicating the model learned the features well.
Inference Behavior	**High Confidence**	During testing, the model consistently output confidence scores >95% for clear images.



```
Loading MobileNetV4...
model.safetensors: 100%                                  15.2M/15.2M [00:00<00:00, 20.5MB/s]

Starting Improved Training for 15 epochs...
/tmp/ipykernel_47/329002543.py:128: FutureWarning: `torch.cuda.amp.GradScaler(args...)` is deprecated. Please use
`torch.amp.GradScaler('cuda', args...)` instead.
  scaler = torch.cuda.amp.GradScaler()
/tmp/ipykernel_47/329002543.py:145: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `
torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast():
Epoch 1/15 | LR: 0.000100 | Loss: 0.8866 | Acc: 82.18%
Epoch 2/15 | LR: 0.000100 | Loss: 0.2959 | Acc: 88.05%
Epoch 3/15 | LR: 0.000050 | Loss: 0.2446 | Acc: 90.37%
Epoch 4/15 | LR: 0.000050 | Loss: 0.2032 | Acc: 92.12%
Epoch 5/15 | LR: 0.000050 | Loss: 0.1913 | Acc: 92.59%
Epoch 6/15 | LR: 0.000025 | Loss: 0.1728 | Acc: 93.39%
Epoch 7/15 | LR: 0.000025 | Loss: 0.1604 | Acc: 93.81%
Epoch 8/15 | LR: 0.000025 | Loss: 0.1350 | Acc: 94.76%
Epoch 9/15 | LR: 0.000013 | Loss: 0.1317 | Acc: 95.09%
Epoch 10/15 | LR: 0.000013 | Loss: 0.1181 | Acc: 95.56%
Epoch 11/15 | LR: 0.000013 | Loss: 0.1072 | Acc: 95.99%
Epoch 12/15 | LR: 0.000006 | Loss: 0.1030 | Acc: 95.98%
Epoch 13/15 | LR: 0.000006 | Loss: 0.0952 | Acc: 96.50%
Epoch 14/15 | LR: 0.000006 | Loss: 0.0888 | Acc: 96.73%
Epoch 15/15 | LR: 0.000003 | Loss: 0.0878 | Acc: 96.67%

Done! Saved to: /kaggle/working/my_improved_model.pth

Training Complete! Model saved to: /kaggle/working/my_model.pth
```

Figure 2: MobileNetV4 accuracy progression showing rapid convergence.

4. TESTING & VALIDATION

To verify the models before hardware integration, a bulk testing pipeline was developed using Python to process a fresh batch of **500+ unlabelled test images**.

4.1 Automated Inference Script

A custom script was deployed on a local machine to:

1. Load both trained models (.pt and .pth files).
2. Run inference on the test directory.
3. Automatically sort output images into yolo_output and mobilenet_output folders for side-by-side comparison.

4.2 Visual Results

YOLOv11n Results:
The detection model successfully localized waste items, drawing bounding boxes even in cluttered images. It showed high confidence (>90%) for plastic bottles and metal cans.

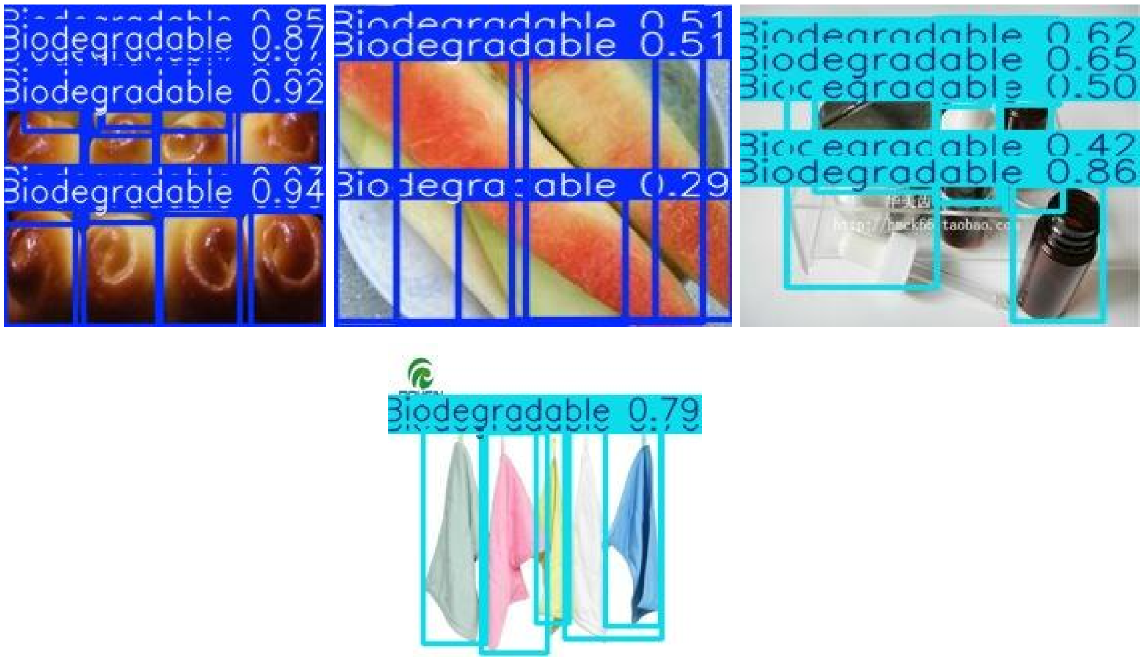


Figure 3: YOLOv11n successfully detecting Biodegradable (Dark Blue) & Non-Biodegradable (Light Blue) with a bounding box.

MobileNetV4 Results:
The classification model performed well on clean, single-item images but lacked the ability to localize items in complex scenes.

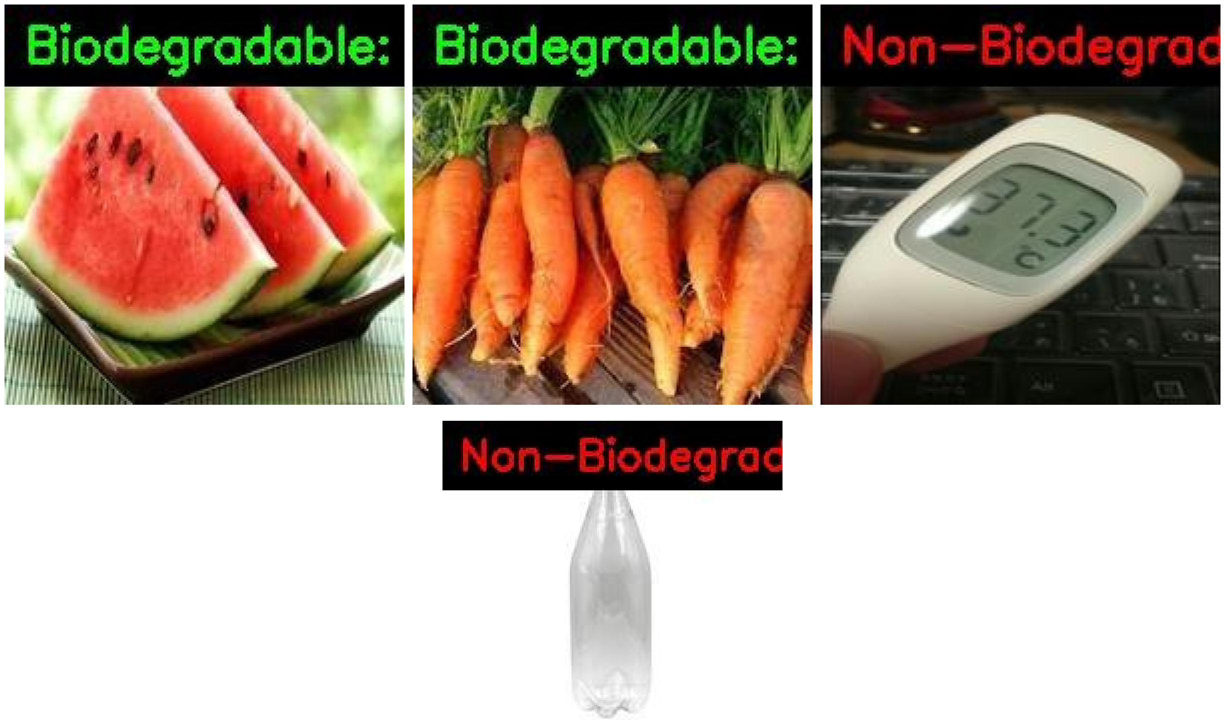


Figure 4: MobileNetV4 classifying Biodegradable and Non - Biodegradable

5. CONCLUSION & NEXT STEPS

5.1 Conclusion

The objectives for Week 1 have been met. The comparative analysis confirms that **YOLOv11n** is the optimal architecture for the Smart Bin. While MobileNetV4 achieved higher raw accuracy (95%) on single images, YOLO's ability to localize multiple objects and its **80.6% accuracy on Non-Biodegradable** items makes it more robust for real-world bin usage.