**Technical Document: Pallet Detection and Counting System**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# 1. Data Collection

- Data Sources:

• Manually Annotated Datasets for training (bounding boxes around pallets)

- Tools & Formats:

• File types: .jpg, .png, .json, .jpeg, .avif, etc…

• Annotation tools: Roboflow

• Dataset formats: YOLOv8, YOLOv5 PyTorch, COCO, etc…

# 2. Data Preprocessing

- Image Preprocessing:

• Auto-Orient: Applied

• Resize images (e.g., 640×640 for YOLOv8 and Yolov5 models)

• Auto-Adjust Contrast: Using Adaptive Equalization

• Normalize pixel values

• Data augmentation (rotation, flip, brightness/contrast adjustments)

- Data Augumentation:

• Outputs per training example: 3

• Flip: Horizontal

• Saturation: Between -25% and +25%

• Brightness: Between -15% and +15%

• Exposure: Between -1" and +10% (or -10% and +1")

• Blur: Up to 2.Spx

• Noise: Up to 0.1% of pixels

- Annotation Handling:

• Convert annotations to training format (YOLO format)

• Verify label quality and class balance

# 3. Model Development

- Model Selection:

• YOLOv5 / YOLOv8 (fast and accurate for object detection)

• We used various versions of these models including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv5n, YOLOv5m

- Training:

• Loss: objectness, classification, localization

• Hardware: GPU-enabled environment (In Google Colab T4 GPU version)

- Sample Training Command:

• model.train(data='/content/Pallet-Detection-and-Counting-3/data.yaml',epochs = 50, imgsz=640) #(--weights yolov8n.pt/yolov8s.pt/yolov8m.pt/yolov8l.pt/ yolov5n.pt/ yolov5m.pt)

# 4. Model Evaluation

- Metrics:

• Precision, Recall, mAP@0.5, mAP@0.5:0.95

• Counting Accuracy = (correct count / actual count) × 100%

- Visual Validation:

• Bounding box overlays

• False positives/negatives analysis

- Model Performances:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **Map** | **Epoch** |
| Yolov8n | 96 | 95 | 97 | 50 |
| Yolov8s | 95 | 93 | 96 | 50 |
| Yolov8m | 95 | 92 | 96 | 50 |
| Yolov8l | 97 | 94 | 98 | 50 |
| Yolov5n | 93 | 93 | 95 | 50 |
| Yolov5m | 71 | 59 | 65 | 50 |

# 5. Deployment

- Inference Pipeline:

• Streamlit

• Upload image section for counting the number of pallets in an image

### 1. **Environment Setup**

1. **Install Python** (if not already installed): Download from [python.org](https://www.python.org/downloads/). Verify installation:

bash

CopyEdit

python --version

1. **Create a Virtual Environment** (optional):

bash

CopyEdit

python -m venv myenv

myenv\Scripts\activate # On Windows

source myenv/bin/activate # On Mac/Linux

### 2. **Install Dependencies**

1. **Install Streamlit**:

bash

CopyEdit

pip install streamlit

1. **Install YOLOv8 and other dependencies**:

bash

CopyEdit

pip install ultralytics pillow numpy opencv-python

### 3. **Run the App Locally**

1. In the project folder (where app.py is located), run:

bash

CopyEdit

streamlit run app.py

1. Open http://localhost:8501/ in your browser to view the app.

### 4. **Considerations**

* **Model Size**: Ensure the model isn’t too large for your local machine or environment.
* **Security**: Ensure your app is secure, especially if handling sensitive data.
* **Performance**: Test the model’s speed and consider using a machine with higher performance if needed.