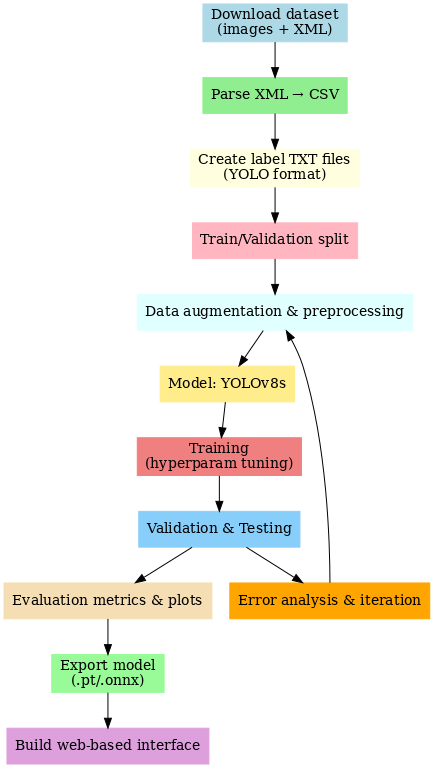
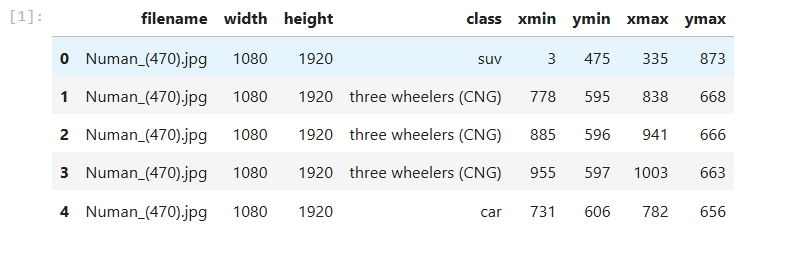
**Task 3 – Vehicle Detection:**

* **Flowchart**

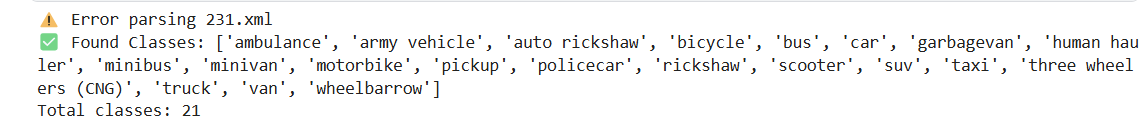


**YOLOv8s Vehicle Detection Flowchart**

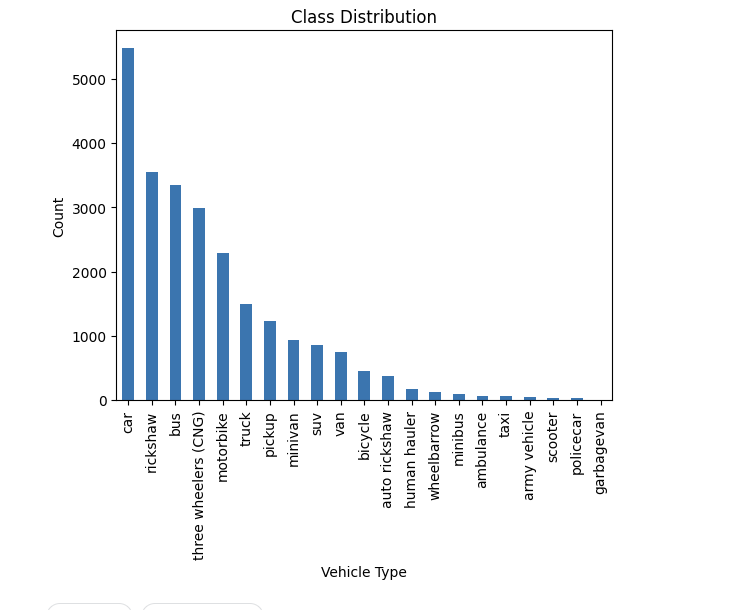
* **Dataset Download**
  + Source: Kaggle dataset vehicle-detection-dataset
  + Contents: Images + XML annotations
* **XML → CSV (Data Understanding & Visualization)**
  + Parse each XML to extract: filename, class, xmin, ymin, xmax, ymax, width, height
  + Save as annotations.csv
  + Use CSV to compute per-class counts, bbox areas, widths/heights for visualization



**Sample of Parsed Annotation Data (CSV Head)**



**List of Vehicle Classes in the Dataset**

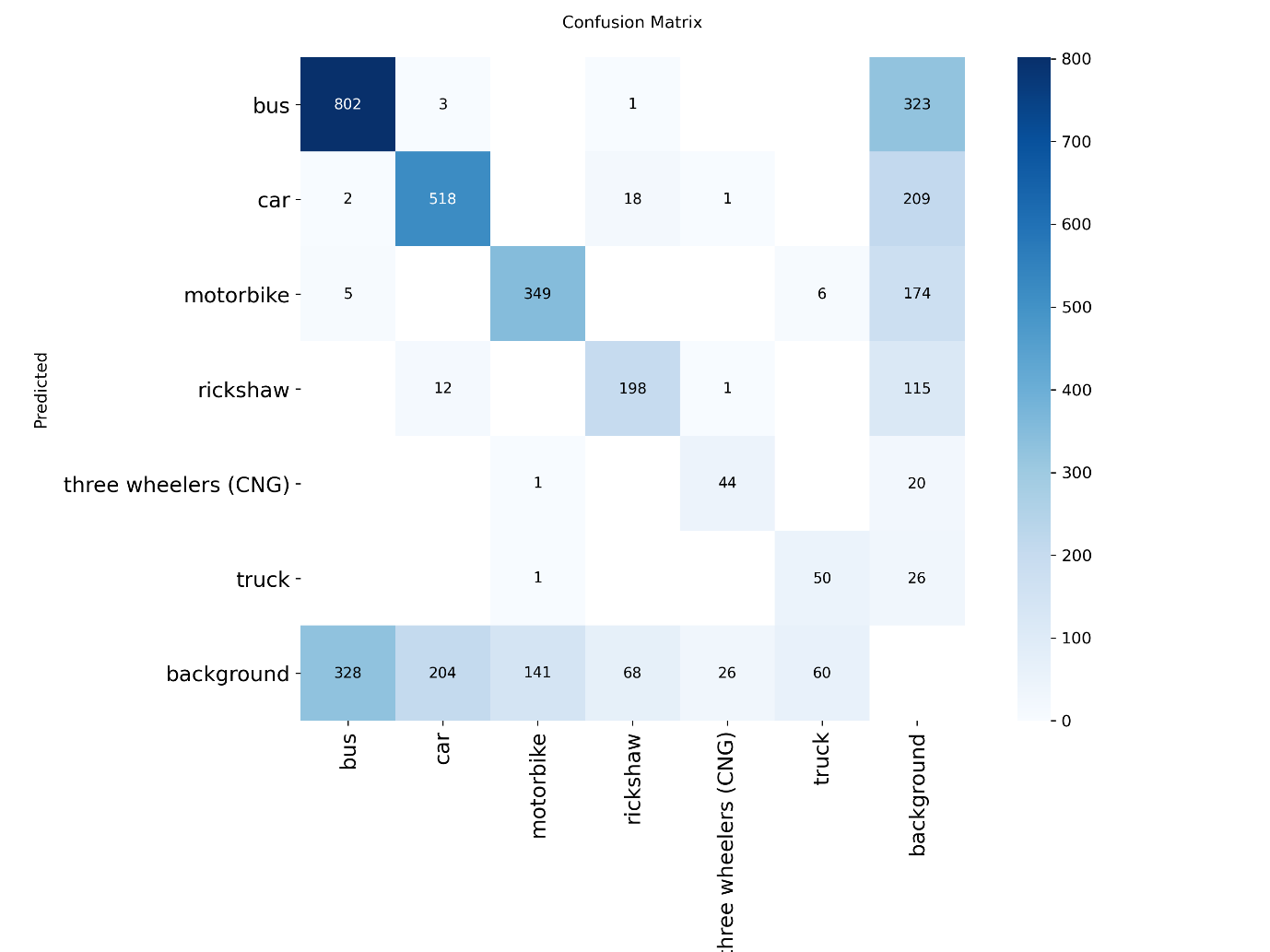


**Distribution of Vehicle Classes in the Dataset**

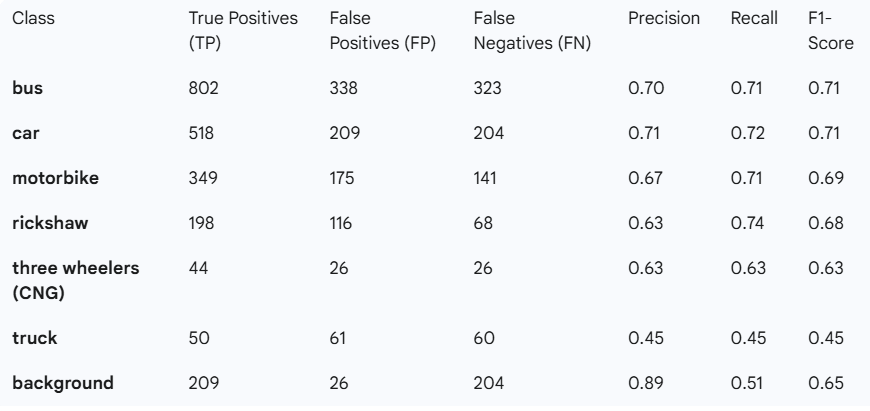
* **Convert XML → YOLO TXT labels**
  + Convert bounding boxes from absolute coords to normalized YOLO format (class x\_center y\_center w h)
  + Save one .txt label file per image
* **Train/Val Split**
  + Split images and labels into train/ and val/ (e.g., 80/20)
  + Ensure class balance or use stratified sampling if needed
* **Data Augmentation & Preprocessing**
  + Resize, flip, color jitter, mosaic (YOLO), random crop
  + Normalize images and optionally compute class weights
* **Model: YOLOv8s**
  + Prepare data.yaml (classes, train/val paths)
  + Choose yolov8s config and hyperparameters (batch size, epochs, lr)
* **Training & Hyperparameter Tuning**
  + Train model, save best checkpoints
  + Monitor loss, mAP@0.5, precision, recall
* **Validation & Testing**
  + Run inference on validation set
  + Produce confusion matrix, PR curves, mAP per class
  + Visualize detections on sample images
* **Evaluation Metrics & Plots**

Here are the metrics for each class, based on the True Positives (TP), False Positives (FP), and False Negatives (FN) derived from the confusion matrix.

* + **Precision**: The accuracy of the positive predictions. It's calculated by dividing the true positives for a class by the sum of true positives and false positives for that class.
  + **Recall**: The fraction of all actual instances of a class that were correctly identified. It's calculated by dividing the true positives for a class by the sum of true positives and false negatives for that class.
  + **F1-Score**: The harmonic mean of precision and recall. It provides a single score that balances both metrics.



**Confusion Matrix for Object Detection Model**



**Performance Metrics**

* + **Overall Accuracy**

The overall accuracy of the model is calculated by summing the diagonal values (true positives for all classes) and dividing by the total number of predictions (sum of all values in the matrix).

Total Correct Predictions (Diagonal Sum): 802+518+349+198+44+50=1961

Total Predictions (Sum of all values): 2608

**Overall Accuracy:** 26081961​≈0.752 or **75.2%**

* **Web Based Result**
  + Build a responsive **web-based interface** using **HTML, CSS, and Bootstrap to** allow users to upload images and display detection result

