# Solution Document: Traffic Light Detection Using Faster R-CNN and YOLOv8

## Introduction

This document outlines a solution for detecting traffic lights in images using two advanced object detection models: Faster R-CNN and YOLOv8. These models are designed to handle the complexities of detecting traffic lights in varying sizes, shapes, and orientations amidst complex backgrounds and lighting conditions.

## Objectives

- **Accurate Detection:** Identify and localize traffic lights in images.
- **Robustness:** Handle different traffic light sizes, shapes, orientations, and lighting conditions.

## Approach

### Part 1: Traffic Light Detection Using Faster R-CNN

**Faster R-CNN** is a two-stage object detection model renowned for its accuracy. It consists of a Region Proposal Network (RPN) that generates potential object locations, followed by a second stage that classifies these proposals and refines their bounding boxes.

#### Key Steps

1. **Environment Setup:**
    - Install necessary libraries like PyTorch, OpenCV, and torchvision.
2. **Model Preparation:**
    - Load a pre-trained Faster R-CNN model,, which includes a class for traffic lights.
3. **Image Preprocessing:**
    - Convert images to a format suitable for model input using transformations that ensure consistency in data handling.
4. **Object Detection:**
    - Perform inference on images to obtain bounding box predictions, labels, and confidence scores.
    - Filter predictions to isolate traffic light detections using class labels and confidence thresholds.
5. **Visualization:**

    - Display detected traffic lights with bounding boxes and confidence scores annotated on the image.

### Expected Outcomes

- **High Detection Accuracy:** Ability to accurately detect traffic lights in diverse conditions.
- **Precision in Bounding Boxes:** Accurate localization of traffic lights, minimizing false positives and negatives.
- **Visualization Clarity:** Bounding boxes clearly indicate detected traffic lights with associated confidence levels.

## Part 2: Traffic Light Detection Using YOLOv8

**YOLOv8** is an efficient real-time object detection model. It uses a single-stage approach to predict bounding boxes and class probabilities simultaneously, making it suitable for applications requiring fast inference.

### Key Steps

1. **Environment Setup:**
    - Install necessary packages, including the ultralytics library for YOLOv8.
2. **Model Loading:**
    - Load a pre-trained YOLOv8 model capable of detecting traffic lights among other classes.
3. **Image Inference:**
    - Use the model to perform inference on input images, generating predictions for various objects including traffic lights.
4. **Post-Processing:**
    - Extract relevant bounding boxes for traffic lights by filtering predictions based on class.
5. **Result Visualization:**
    - Annotate detected traffic lights on the image with bounding boxes and class labels.

### Expected Outcomes

- **Real-time Detection:** Fast inference speeds suitable for real-time applications.
- **High Detection Accuracy:** Reliable detection performance across diverse traffic light conditions.
- **Ease of Deployment:** Suitable for applications requiring quick setup and deployment with efficient model performance.