Traffic Light Detection and Recognition

Introduction

The code implements a computer vision system designed to detect and recognize traffic lights in images taken from a vehicle driving along a road. The system identifies traffic lights by their color state (red, green, and amber) and locates the traffic light housing, displaying these detections with colored boxes. Additionally, the system visualizes ground truth traffic light locations with pink boxes and performs automatic performance evaluation based on the overlap between the detected traffic light boxes (white) and the ground truth boxes (pink).

Code Logic and Theory

The code employs several key computer vision techniques and concepts, as outlined below:

1. Color-Based Detection: The system uses HSV (Hue, Saturation, Value) color space to define color ranges for green, red, and yellow, which are typical colors of traffic lights. HSV separates color information (hue) from intensity (value), making it less sensitive to lighting changes.

2. Back Projection: Histogram back projection is used to map the detected colors back to the original image, highlighting areas that match the predefined color histograms for traffic light colors.

3. Morphological Operations: The code uses a smoothing filter to enhance the detection results, improving the clarity of the detected areas.

4. Contour Detection: By applying contour detection to the thresholded back projection results, the system identifies contiguous areas that likely represent traffic lights.

5. Bounding Boxes and Filtering: The detected contours are encapsulated in bounding boxes, which are then filtered based on size to eliminate detections that are too small or too large to be traffic lights.

6. Overlap with Ground Truth: The system calculates the overlap between detected traffic lights and ground truth data to evaluate the performance of the detection algorithm.

7. State Determination: The system attempts to determine the state of the traffic light (red, green, amber, or red+amber) based on the number of detected lights in each color.

8. Edge Detection and Expansion: Canny edge detection is used in combination with a light color mask to identify the housing of the traffic light. The system then expands these detections until they

meet a proper edge, aiming to more accurately define the traffic light housing.

9. Performance Metrics: The code calculates Intersection over Union (IoU) and area similarity between detected boxes and ground truth to quantitatively assess performance.

10. Traffic Light State Determination:

An integral part of the code system is its ability to determine the current state of the traffic lights based on the detection results. The system counts the number of detected boxes corresponding to each traffic light color (red, green, and amber) and applies specific logic to deduce the traffic light state, which is then displayed on the top left corner of the output image.

Green: The system identifies the state as "GREEN" if there are at least two green detections and fewer than two detections for both red and yellow colors.

Amber: The state is determined to be "AMBER" if there are at least two amber detections and fewer than two green and two red detections.

Red: The state is identified as "RED" if there are at least two red detections and fewer than two detections for both green and yellow colors.

Red-Amber: A special "RED-AMBER" state is recognized if there are at least two detections for both red and amber colors and fewer than two green detections.

11. Initial White Box Dimensions and Adjustments

The initial dimensions of the white bounding boxes, used to encapsulate detected traffic light colors, play a significant role in the accuracy of the detection system. The initial dimensions are calculated based on the area of the detected color contour, using a specific formula that aims to approximate the aspect ratio of a typical traffic light. The width is derived from the square root of the area divided by 15, and the height is then set to 15 times the width. This calculation is designed to create an initial box that is likely to encompass the entire traffic light when expanded.

After the initial bounding boxes are expanded to capture the traffic light housing accurately, further adjustments are made based on the determined state of the traffic light:

Green State Adjustment: For traffic lights in the green state, the bounding boxes are moved up by 25 pixels. This adjustment accounts for the typical positioning of the green light at the bottom of the traffic light housing.

Red State Adjustment: Conversely, for traffic lights in the red state, the bounding boxes are moved down by 15 pixels. This adjustment acknowledges that the red light is usually at the top of the traffic light housing.

Image Processing Loop

1. Image Reading: Each image is read and converted to HSV color space.

2. Color Masking and Histogram Calculation: Masks for green, red, yellow, and black are created, and histograms for each color are calculated and normalized.

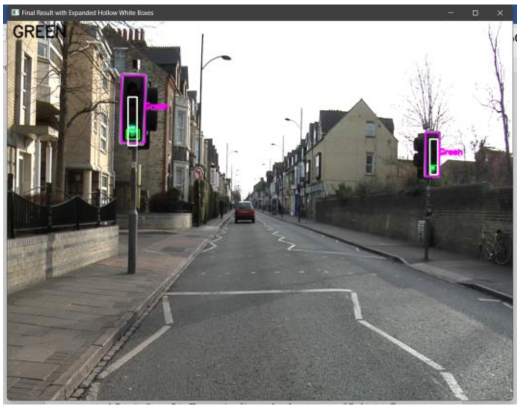
3. Back Projection and Smoothing: Back projection is applied for each color, followed by smoothing.

4. Thresholding and Contour Detection: The back-projected images are thresholded, and contours are detected.

5. Bounding Box Processing: Detected contours are encapsulated in bounding boxes, filtered, and drawn on the image.

6. Edge Detection and Expansion: Canny edge detection is applied to the original image, and the results are used to expand the initial bounding boxes to more accurately encapsulate the traffic light housing.

7. Overlap and Performance Evaluation: The overlap with ground truth data is calculated, and performance metrics are displayed.




Final Result with Expanded Hollow White Boxes
RED-AMBER


Final Result with Expanded Hollow White Boxes
GREEN


Final Result with Expanded Hollow White Boxes
AMBER


Final Result with Expanded Hollow White Boxes
RED


Final Result with Expanded Hollow White Boxes
GREEN


Final Result with Expanded Hollow White Boxes
GREEN


Final Result with Expanded Hollow White Boxes
GREEN


Final Result with Expanded Hollow White Boxes
GREEN

(Note that Image No.13 and No.14 failed to detect any of the traffic light for unknown reason)

Summary Statistics

1. Detection IoU: The average IoU across every single image serves as a key indicator of the system's accuracy. It measures the overlap between detected traffic light boxes and the ground truth, with higher values indicating better detection precision.

2. Area Similarity: This metric assesses the size accuracy of the detected boxes compared to the ground truth. An average area similarity close to 1 indicates that the detected boxes are very close in size to the actual traffic lights.

3. Detection Success Rate: The proportion of images where the system successfully detects traffic lights and accurately estimates their sizes, as evidenced by high IoU and area similarity scores.

4. Failure Analysis: A breakdown of failure cases, including missed detections (false negatives), incorrect detections (false positives), and instances of poor size estimation (IoU or Area Similarity lower than 50%), offering a nuanced view of the system's limitations.

By focusing on these revised summary statistics, the report will offer a detailed and nuanced assessment of the system's performance, highlighting its strengths in detecting and sizing traffic light boxes.

```
Pair 1: IoU: 0.2569, Area Similarity: 0.2799 | Pair 2: IoU: 0.4714, Area Similarity: 0.4714 | Low accuracy both on Pair 1 and Pair 2
Pair 1: IoU: 0.4765, Area Similarity: 0.4765 | Pair 2: IoU: 0.5745, Area Similarity: 0.5745 | Low accuracy on Pair 1
Pair 1: IoU: 0.4077, Area Similarity: 0.4832 | Pair 2: IoU: 0.6508, Area Similarity: 0.6508 | Low accuracy on Pair 1
Pair 1: IoU: 0.6027, Area Similarity: 0.8770 | Pair 2: IoU: 0.6425, Area Similarity: 0.6425
Pair 1: IoU: 0.1919, Area Similarity: 0.1919 | Pair 2: IoU: 0.7417, Area Similarity: 0.9728 | Low accuracy on Pair 1
Pair 1: IoU: 0.6024, Area Similarity: 0.7709 | Pair 2: IoU: 0.4858, Area Similarity: 0.9626 | Low accuracy on Pair 2
Pair 1: IoU: 0.4862, Area Similarity: 0.7975 | Pair 2: IoU: 0.5276, Area Similarity: 0.9259 | Low accuracy on Pair 1
Pair 1: IoU: 0.6182, Area Similarity: 0.9944 | Pair 2: IoU: 0.6838, Area Similarity: 0.7642
Pair 1: IoU: 0.2268, Area Similarity: 0.2268 | Pair 2: IoU: 0.3313, Area Similarity: 0.5347 | Low accuracy both on Pair 1 and Pair 2
Pair 1: IoU: 0.8667, Area Similarity: 0.8667 | Pair 2: IoU: 0.6167, Area Similarity: 0.9796
Pair 1: IoU: 0.6253, Area Similarity: 0.9143 | Pair 2: IoU: 0.7961, Area Similarity: 0.9907
Pair 1: IoU: 0.0000, Area Similarity: 0.0000 | Pair 2: IoU: 0.0000, Area Similarity: 0.0000 | Low accuracy both on Pair 1 and Pair 2
Pair 1: IoU: 0.0000, Area Similarity: 0.0000 | Pair 2: IoU: 0.0000, Area Similarity: 0.0000 | Low accuracy both on Pair 1 and Pair 2
PS C:\Users\M2-Winterfell\Downloads\CamVidLights>
```

Weakness Analysis (Robustness of the Approach)

While the provided code demonstrates a comprehensive approach to traffic light detection, several potential weaknesses may impact its robustness and general applicability:

The robustness of the traffic light detection system is evaluated through a detailed analysis of its weaknesses, considering various operational scenarios and environmental conditions.

1. Dynamic Lighting Conditions: The system's reliance on color-based detection using fixed HSV ranges might not be robust to drastic changes in lighting, such as those encountered during different times of the day or under varying weather conditions. Adaptive color thresholding or additional lighting compensation techniques could be considered to enhance robustness.

2. Distance and Scale Variability: The fixed size thresholds for filtering bounding boxes may not account for the varying sizes of traffic lights due to different distances from the camera. Implementing dynamic size filtering based on perspective or depth information could improve detection accuracy across distances.

3. Generalization to Diverse Environments: The manual setting of parameters like color ranges and threshold values might limit the system's applicability to different geographic locations with varying traffic light designs and backgrounds, such as an Amber state traffic light might be mis-detected as a red one. A more adaptive or learning-based approach could help the system generalize better to new environments.

4. Complex Backgrounds and Occlusions: The presence of complex backgrounds, such as trees, buildings, and other vehicles, can interfere with edge detection and expansion, leading to incorrect housing identification. In this traffic light detection system, the program fails to detect any traffic light color when there exist other vehicles in the image (13 and 14) with an unknown reason. Advanced segmentation techniques or deep learning-based methods could provide more robust differentiation between traffic lights and background objects.

5. Robustness to Atypical Traffic Light Configurations: The system may struggle with atypical traffic light configurations, such as horizontal arrangements or unconventional color patterns. For example, not every single traffic light housing a rectangle shape, but also be a polygon. Incorporating a more flexible model that can learn from a wider range of traffic light configurations could mitigate this issue.

6. Error Analysis and Feedback Loop: Incorporating a feedback mechanism where detection errors can be analyzed and used to continuously improve the system could significantly enhance its robustness and adaptability over time.