SOLUTION TESTING

Test Plan

The testing on the program will have five steps, each step we will analyze how the system dealt with each section.

1. Set of signs that are fully visible
2. Set of signs on an angle.
3. Set of signs that are poorly visible.
4. Set of signs of different distances away.

Test Cases

1. Fully visible images


These images all contain clearly visible road signs/traffic lights. The aim of this is to get an over recognition figure of the program with no interferences.

2. Signs on a angle

These images are all of signs that are at an angle to see how the program would deal with detecting signs when driving round a corner for example.

3. Poorly viable

All sorts of things can affect how the sign might be recognised. The above images were used to test the program's limitations in terms of image quality.

4. Distance Test

For the distance test we showed the system a range of frames of a car approaching a set of traffic lights, these were the images shown.

Test Report

Results

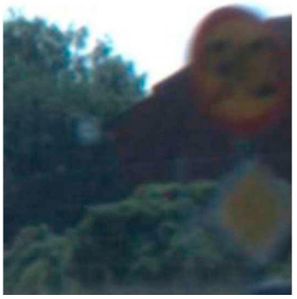
1. **Fully visible images**

As shown in the image test results the system accurately dictated and classified all the signs given.

2. Signs on a angle

As shown in the image results haven the sign on an angle did not affect the accuracy of the program as it still accurately detected and classified every image

3. Poorly Visible Signs


Clearly the quality of image the system needs is an issue as shown in the image results only three of the five given images correctly detected the sign. The program seemed to struggle with the overexposure due to sunlight and the blurry images.

4. Distance Test


Distance seemed to be another issue for the system. As shown in the image results the system only detected the traffic light in this test instance till it was approximately two car lengths away.

Overall analysis

The program we set out to build worked as expected. However, there were issues in a number of areas of the program.

From the testing, it showed that the program struggled with low image quality. If the image was slightly overexposed, or blurry it failed to detect the sign at all. If we turned the accuracy threshold down recognition improved, but this was to the detriment of identification accuracy. Accuracy was extremely important in this use case, so allowing identification accuracy to slide was not an option.

The second issue found with the program was the distance at which the sign/traffic light was detected. From our tests it was discovered that the program did not detect anything until it was within two car lengths away from the road sign. This would require significant improvement in later editions of the program. It would be essential to improve recognition from distance if the program was going to be able to scale to the level of aiding driverless cars. Currently, in many cases the program would detect the sign far too late in order to accurately acknowledge the sign and make a timely decision based on the information.

Recommendations

Thay main issue I can see from this is the distance issue. From looking at the and the results it doesn't seem to be the program. It more seems like a camera quality issue which also ties into it failing to detect the poor quality images. The distance especially could be solved with a different sort of lens for example the sort can see better quality in the distance or even zoomed in. If the system was implemented into an autonomous car you could have 2 cameras, one for distance and one for close up, maybe even a wide angle to make sure that all aspects are visible to the system and can always accurately detect and classify the signs.