Traffic signs recognition: a comparison on computer vision approach and deep neural network approach

Han Cao (hcao29), Xi Chen (xchen776), Lichengxi Huang (lhuang236)

1. Background and problem introduction

Traffic signs play a significant role in the modern traffic system, and it becomes even more important in recent years due to the emergence of autonomous vehicles.

The problem we are aiming to solve is to use the computer vision technologies to do the traffic signs recognition (TSR) under different driving scenarios. We may further develop algorithms to recognize traffic signs under extreme cases.

2. Motivation

In the development of autonomous vehicles, recognizing the road signs plays a very important role. Although in the future, when connected autonomous vehicles are developed, this may be solved by using roadside units sending instructions to the vehicles, nowadays, we still need to rely on the vehicles themselves to recognize road lanes and signs.

Furthermore, vehicles that can recognize traffic signs will improve the driving experiences and safety for the drivers. Most drivers may have experienced situations when they were driving too fast on the highways and did not see clearly what the road guidance sign tells, hence driving to the wrong direction. In other cases, drivers may not see speed limit signs when the environment is dark, hence getting a ticket on over speeding.

These scenarios can all be avoided as long as we can develop an application that can automatically recognize the traffic signs. The vehicles that are equipped with this application can extract the key information, and either speak it out or display it on the head-up displays (HUD).

3. Current state-of-the-art

In general speaking of traffic signs detection technology, it’s already well developed per se. It is a significant part of Advanced driver-assistance systems(ADAS). Such technologies are already used broadly by vehicle manufacturers on their products, such as BMW, Ford, Volvo etc., to improve the performance and safety. Most cars with an ADAS are able to show the speed limit on the dashboard. They can also detect other signs like: “Do Not Enter,” and Stop Sign.

The cornerstones of the TSR are based on two facts:

1. The features of traffic signs in the same area are identical. There is a standard of the design and size of all traffic sign published by MUTCD in the U.S.
2. The cutting-edge algorithms enable cars to analysis and understand the content of traffic sign.

The concept was brought up in 1968, but the recent blossom is triggered by the breaking through in hardware and Machine learning algorithms.

Following are some technical details of the technology:

- Blurring
- Color-based detection: e.g. stop signs are red etc.
- Shape-based detection: e.g. stop signs are hexagon in most places
- Cropped and extract features

- Separation

The approaches that researchers commonly use include:

- Histogram of oriented gradients (HOG)
- Machine learning approaches:
    - ◊ RCNN(Mask-RCNN, Fast-RCNN …...)
    - ◊ CNN
    - ◊ Classification
    - ◊ Others

4. Method and innovation

Our major purpose of doing this project is to dive into the principles and implementation details of TSR technologies. For this part, we pursue a thorough understanding of the technology per se. But we also endeavor to bring some innovative factors into the project by using hyperparameter tuning or by integrating traditional computer vision and deep learning approaches together. We plan to take both of these approaches and aim to increase the accuracy and efficiency of detecting by doing some reasonable integrations.

For the traditional computer vision part, our group plan to redesign and refine current existing algorithms and to implement it by ourselves completely. For the deep learning part, we will train a deep neural network based on existing models, such as YOLO/Mask-RCNN etc. We intend to increase the detection accuracy by hyperparameter tuning.

5. Why this method?

So far, although we see there are already a lot of traffic sign recognition systems developed, most of them are based on the deep neural network, and a few of them follow the traditional computer vision approaches. As we also notice, for those which do not utilize deep learning approaches, the applications are usually limited to some specific scenarios, for example, recognizing speed limit signs or stop signs only. However, later applications based on deep learning are usually more generic, being able to recognize multiple signs on the road. The previous works also seldomly mention their performance on extreme settings, for example on rainy or snowy weather, or when the light is dim.

Hence, our project that aims at implementing the application based on both of the traditional computer vision and deep learning approaches will give us a better insight in the advantages and disadvantages of both of methods.

6. Performance evaluation and the expected results

In order to evaluate the outcome of our work, we will compare the performances of our solution with the current existing solutions. The perspectives we will compare on include the accuracies, true positive rates, false positive rates, etc. We will also take the resolution of the picture/video in count. Other aspects like running time and performance on extreme settings will also be taken into consideration.

For the deep learning part, we are expecting an accuracy of around the average of existing solutions. For the computer vision part, we should hold the same record with the current approach. By integrating two methods together, we optimistically look for a result which is better than a deep learning model trained with the same set of data in an observable way.

Finally, we are going to show several driving videos, where our methods recognize the traffic signs and mark the signs along the roadsides.

7. Potential plan

Our tentative plan includes:

- Build website
- Data collection – take photos and videos as raw data, find online data for training
- Model selection – select methods for computer vision and deep neural network
- Build computer vision model – write python codes to apply HOG and etc.
- Build deep neural network – use deep learning packages to build our network
- Train deep neural network
- Test models – use the data we collected to test the two models
- Comparison and evaluation – compute accuracies and other metrics
- Write reports – midterm report and final report
- Final presentation

The detailed timeline is shown in the following table.

Activity	15 Feb	1 Mar	15 Mar	1 Apr	15 Apr	30 Apr
Build website	------	------	------	------	------	------
Data collection	------	------	------	---		
Model selection	----	---				
Build computer vision model		----	------	---		
Build deep neural network		----	------			
Train deep neural network			---	------		
Test models				---	---	
Comparison and evaluation					------	---
Write reports			---	--	--	------
Final presentation						-----