Real-Time Traffic Sign Detection and Recognition Using GPU

Zhilu Chen and Xinming Huang
Department of Electrical and Computer Engineering
Worcester Polytechnic Institute, MA 01609, USA

***Abstract*—This paper presents a Graphic Processing Unit (GPU) implementation of real-time traffic sign detection and recognition which can classify 48 classes of traffic signs in the library. The GPU-based system has three stages: pre-processing, feature extraction and classification. The proposed design is to optimize the tradeoff between accuracy and computing time which has not been well studied previously. The GPU-based system can achieve the frame rate of 21.3 frames per second (fps) for the image resolution of 1,238 by 1,628. The experimental results show the detection rate of 91.69% and the recognition rate of 93.77%.**

***Index Terms*—GPU, traffic sign, recognition, image processing**

I. Introduction

Traffic sign detection and recognition are important functions in an Advanced Driver Assistance System (ADAS). A traffic sign detection and recognition system often contains three stages: pre-processing, detection and recognition. Many existing research have been carried out to improve the accuracy of detection and recognition. This work is focused on improving the hardware efficiency that is to minimize the processing time using a GPU-based accelerator.

Typically, feature extraction and pattern classification algorithms are computationally intensive. Many researches have been done to optimize the algorithms themselves to improve the accuracy, but there are only a few works focused on the hardware implementations. In this paper, we propose to utilize the many-core architecture in a GPU to accelerate the traffic sign detection and recognition algorithms through massively parallel processing. The objective is to reduce the computing time considerably such that the traffic signs can be detected and recognized at real-time.

II. Related Work

There are several existing works that detect and recognize multiple traffic signs using the common features such as shapes and colors [1], [2]. However, these works are primarily focused on the algorithms themselves instead of the actual processing time on hardware platforms, which prevents those designs from becoming practically useful. On the other hand, there are some works that consider the tradeoff between accuracy and computing time [3], [4], but the experimental data sets that they use were varied. Without evaluating using the same standard data set, it is difficult to compare their reported results.

Standard traffic sign data sets, such as BelgiumTS Dataset [1] and German Traffic Sign Benchmark [2], are built to address this issue. Researchers can use the same training and testing data set to measure the performance of their designs. In addition, image resolution is another important factor that can affect the processing time and accuracy.

III. Traffic Sign Detection and Recognition System

The proposed system contains three main stages: pre-processing, detection and recognition, as shown in Fig. 1. At first, we perform red and blue color extractions respectively and select regions of interest (ROI) accordingly. Next, Histograms of Oriented Gradients (HOG) [5] features are extracted from the gray scale image. A sliding window searches the image exhaustively to find the candidates using a linear Support Vector Machine (SVM). Color-based HOG detectors are then performed on these candidates to reduce false positives, followed by a rectangle grouping operation to locate the detected traffic signs. In addition, these color-based HOG detectors also classify the detected traffic signs into several super-classes, such as red circle, red triangle, blue circle, etc. Finally, the detected sign candidates are sent to a cascade classifier which contains several linear SVMs. Each of the final detected windows is classify by the SMVs to determine and re-assure its category. Once its category is determined, it is classified by a one-vs-all SVM for the specific category. SVMs are trained using k-fold cross-validation for better accuracy. The proposed system can detect and recognize 48 classes of traffic signs that are listed in the BelgiumTS dataset library. We use the BelgiumTS dataset to train the SVMs to classify different classes of traffic signs in each category.

Although HOG and SVM are commonly used in detecting and recognizing objects, it is still a challenging task to find the balance between accuracy and efficiency. For example, linear kernel SVMs run faster but may results lower accuracy. In order to obtain a desirable accuracy, we use multiple HOG features and SVMs in our design as shown in Fig. 1,

IV. Parallel Processing on GPU

Since pre-processing and HOG algorithms are complex computations, in this section we describe the GPU-based acceleration. For pre-processing, it is a typical point operation

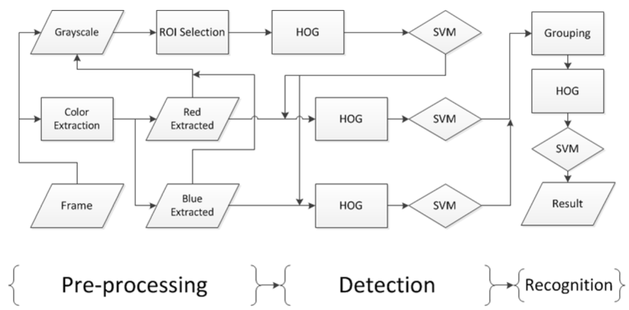

Figure 1. Three stages in our proposed system.

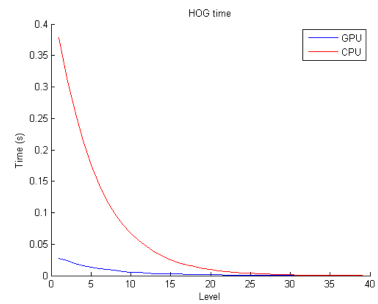

Figure 2. HOG computing time on CPU and GPU.

which is suitable for GPU implementation. The HOG computation is more complicated. We modify an existing GPU version of HOG in the OpenCV library which can accelerate the computation significantly if compared to the CPU version. HOG features need to be computed at multiple levels of scaling for the input image, and gaps between levels can be reduced or eliminated. Once the input data of each level is prepared, there is no data dependency during HOG computation between different levels. In OpenCV implementation, each level stalls until the computation of previous level is done to ensure data synchronization between kernels. Such stalls are unnecessary and can be avoided by using CUDA streams. Kernels can execute in multiple CUDA streams concurrently and can be synchronized in a certain stream without affecting others. By using CUDA streams, we reduce the gaps between levels significantly and thus improve the efficiency of HOG computation. However, we should also consider the overhead of launching multiple CUDA kernels. Therefore, we recommend this approach only for large input images. In our case, the input image size is 1236 by 1628 pixels, which is large enough to hide the overhead of CUDA kernel launching.

V. Experimental Results

The proposed traffic sign detection and recognition algorithms are evaluated on a Tesla K20 GPU platform. The preprocessing stage on GPU takes about 13~17 ms. The detection and recognition stages account for most of the processing time. At first, we compare the HOG computing time on CPU and GPU at each scaling level. As shown in Fig. 2, the speedup of GPU acceleration is significant when the scaling level is small and declines rapidly as the scaling level increases. The original size of testing image is 1236 by 1628 pixels. The OpenCV library is employed for comparing the HOG computing time between CPU and GPU.

Secondly, we test our GPU implementation using 2000 images in the BelgiumTS dataset [1]. Each image is in the size of 1236 by 1628 pixels. The total execution time for all three stages is recorded. The initialization time is ignored such as reading images and SVM parameters. Post-processing time is also ignored such as displaying the results. The average frame rate on GPU is 21.3 fps.

Finally, we evaluate the detection rate and classification rate of our proposed system using the BelgiumTS dataset. We test 1918 images and the detection rate is 91.69%. We also measure the false positive rate by using background images provided by the BelgiumTS dataset. We extract over 20 million windows from those images in different scaling levels. The number of false positives is 684. Thus the False Positives per Window (FPPW) is 3.39×10^{-5}. Similarly, we use the BelgiumTSC dataset to evaluate the classification rate. Each image in BelgiumTSC dataset contains one traffic sign with some background. We resize each image to our window size of 32 by 32 pixels before computing HOG and performing SVM classification. We test 2520 images from BelgiumTSC dataset and the classification rate is 93.77%.

VI. Conclusions

This paper presents a real-time traffic sign detection and recognition system on a GPU platform. It can detect and recognize 48 traffic signs with reliable accuracy. It runs at an average frame rate of 21.3 fps and each frame is of high resolution with 1236 by 1628 pixels. Each frame is processed individually and no information from previous frames is required. However, information from previous frames can be useful when tracking traffic signs and improving the detection accuracy. This will be part of our future work to further improve the overall system performance.

Acknowledge

We acknowledge support from The MathWorks, Inc. for this work.

References

[1] M. Mathias, R. Timofte, R. Benenson, and L. V. Gool, "Traffic sign recognition - how far are we from the solution?" in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2013)*, August 2013.
[2] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *International Joint Conference on Neural Networks*, no. 1288, 2013.
[3] E. Herbschleb and P. H. N. de With, "Real-time traffic sign detection and recognition," pp. 72 570A–72 570A–12, 2009. [Online]. Available: http://dx.doi.org/10.1117/12.806171
[4] K. Par and O. Tosun, "Real-time traffic sign recognition with map fusion on multicore/many-core architectures," *Acta Polytechnica Hungarica*, vol. 9, no. 2, 2012.
[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, June 2005, pp. 886–893 vol. 1.