International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 6 – MAY 2018.

Design and Extraction of Traffic Sign from Dynamic Image

Navarajan.J[1,2], Pooja Satish[2], Irene Sylvia.I[2], Lakshmi Priya.V[2]

[1]Associate Professor, [2] U.G. Scholar
Department of Electronics and Communication Engineering, Panimalar Institute of Technology,Chennai.
nava.cool@gmail.com, poojasatish17@gmail.com, sylviaimmanuel005@gmail.com,
anujayakumarjk@gmail.com

***Abstract*— Traffic sign detection has attracted great interest, as it plays an essential role over a broad range of applications, such as intelligent driver assistance and roadway inventory management. A traffic sign detection system could require real time processing for front view video from a moving vehicle or off-line processing for road images from a database. In either case, it is critical to develop a fast traffic sign detection system with high throughput. Recently, great efforts have been made to design hardware accelerators to facilitate fast traffic sign detection. These accelerators have been implemented with various hardware platforms, including FPGA, digital signal processor, system-on-chip (SOC), CPU, graphics processing unit and so on. In this proposed system, traffic sign is detected from the current frame of the video. This image undergoes feature extraction and further procedure to capture the sign through this technique, accuracy is slightly improved.**

***Index Terms*— Accelerator, cascade classifier, energy efficiency, FPGA, traffic sign detection.**

I. INTRODUCTION

Traffic sign detection has attracted great interest, as it plays an essential role over a broad range of applications, such as intelligent driver assistance and roadway inventory management. A traffic sign detection system could require real-time processing for front-view video from a moving vehicle or off-line processing for road images from a database. In either case, it is critical to develop detection system with high throughput. Numerous algorithms have been proposed for traffic sign detection in the literature, including support vector machine, cascade classifiers, neural networks, and so on. Recently, remarkable advances have been made using deep neural networks (DNNs) and convolutional neural networks (CNNs). They are now the state-of-the-art techniques, offering the highest detection accuracy for many practical applications. However, both DNN and CNN are highly complex and, hence, come with high computational cost. To lower power consumption and reduce chip area, analog circuits with emerging devices have been designed to implement CNN. In a mixed-signal chip with both analog and digital circuits is designed for CNN and it achieves 2G operations per second with power consumption of 20 mW only. However, the design cost may increase due to the limited support offered by analog CAD tools.

Alternatively, surf algorithms are often adopted to build a practical high-throughput traffic sign detection system because of two reasons. First, portable platforms for real-time applications are usually constrained by cost and energy and have limited computational power. Second, for large-scale offline databases, the huge amount of data prevents us from applying complex algorithms to get the results of interest in a timely manner. Therefore, simple algorithms such as cascade classifiers are often the preferred choice in order to achieve high detection accuracy with affordable computational cost. Recently, great efforts have been made to design hardware accelerators to facilitate fast traffic sign detection. These accelerators have been implemented with various hardware platforms, including FPGA], digital signal processor, system-on-chip (SoC) , CPU, graphics processing unit (GPU), and so on. Among them, FPGA generally offers a flexible solution due to its reconfigurability in both high-level system architecture and low-level circuit implementation and, hence, has been adopted by many industrial applications. In this paper, we propose an FPGA-based accelerator for traffic sign detection using cascade classifiers. To efficiently utilize the hardware resources and maximize the detection speed, we propose several novel ideas and heuristics.

Programmable Logic Devices (PLDs) offer wide range of logic capacity, features, speed and voltage characteristics and these devices
can be changed at any time to perform any number of functions. The concept is to have a few PLD blocks or macro cells on a single device with general purpose interconnect in between. Simple logic paths can be implemented within a single block. More sophisticated logic will require multiple blocks and use the general purpose interconnect in between to make these connections. CPLDs are great at handling wide and complex gating at blistering speeds. e.g., 5ns which is equivalent to 200MHz. the timing model for CPLD is easy to calculate, so even before design it can calculate the in to output speeds. CPLDs enable ease of design, lower development costs, more product revenue for money, and the opportunity to speed your products to market, etc.

II. BACKGROUND DETAILS

Historically, FPGAs have been slower, less energy efficient and generally achieved less functionality than their fixed ASIC counterparts. An older study had shown that designs implemented on FPGAs need on average 40 times as much area, draw 12 times as much dynamic power, and run at one third the speed of corresponding ASIC implementations. More recently, FPGAs such as the Xilinx Virtex-7 or the Altera Stratix 5 have come to rival corresponding ASIC and ASSP solutions by providing significantly reduced power, increased speed, lower materials cost, minimal implementation real-estate, and increased possibilities for re-configuration 'on-the-fly'. Where previously a design may have included 6 to 10 ASICs, the same design can now be achieved using only one FPGA.

The primary differences between CPLDs (complex programmable logic devices) and FPGAs are architectural. A CPLD has a somewhat restrictive structure consisting of one or more programmable sum-of-products logic arrays feeding a relatively small number of clocked registers. The result of this is less flexibility, with the advantage of more predictable timing delays and a higher logic-to-interconnect ratio. The FPGA architectures, on the other hand, are dominated by interconnect. This makes them far more flexible (in terms of the range of designs that are practical for implementation within them) but also far more complex to design for.

In practice, the distinction between FPGAs and CPLDs is often one of size as FPGAs are usually much larger in terms of resources than CPLDs. Typically only FPGA's contain more complex embedded functions such as adders, multipliers, memory, and serdes. Another common distinction is that CPLDs contain embedded flash to store their configuration while FPGAs usually, but not always, require an external nonvolatile memory.

Security considerations

With respect to security, FPGAs have both advantages and disadvantages as compared to ASICs or secure microprocessors. FPGAs' flexibility makes malicious modifications during fabrication a lower risk.[25] Previously, for many FPGAs, the design bitstream is exposed while the FPGA loads it from external memory (typically on every power-on). All major FPGA vendors now offer a spectrum of security solutions to designers such as bitstream encryption and authentication. For example, Altera and Xilinx offer AES (up to 256 bit) encryption for bitstreams stored in an external flash memory.

FPGAs that store their configuration internally in nonvolatile flash memory, such as Microsemi's ProAsic 3 or Lattice's XP2 programmable devices, do not expose the bitstream and do not need encryption. In addition, flash memory for LUT provides SEU protection for space applications.

III. RELATED WORKS

The MATLAB® high-performance language for technical computing integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include Math and computation, Algorithm development, Data acquisition, Modeling, simulation, and prototyping, Data analysis, exploration, and visualization Scientific and engineering graphics, Application development, including graphical user interface building. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. It allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or Fortran. The name MATLAB stands for matrix laboratory.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems.

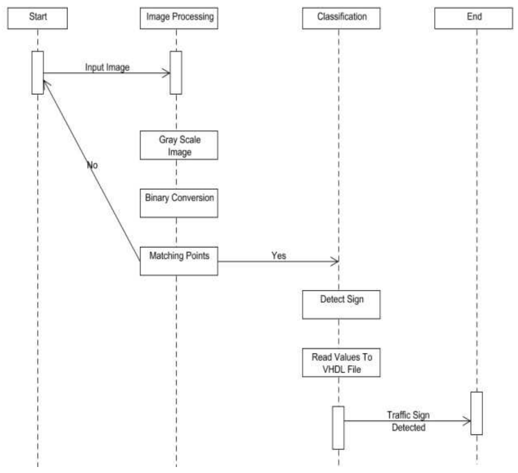
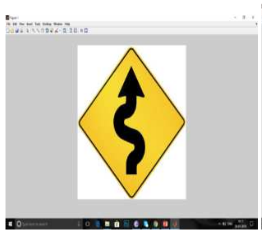
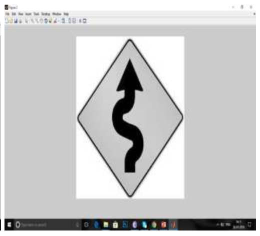


IMAGE EXTRACTION

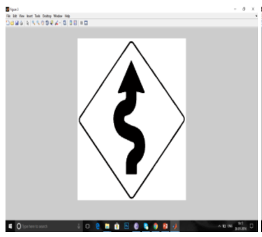
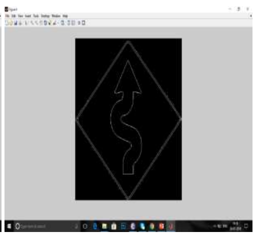
The first step to this process is that of the image extraction. This is obtained through the use of an EBScam which captures the image from a running video. The next step is done by segmenting foreground image from the dynamic background using MATLAB. The traffic sign is then segmented to RGB, binary images etc., as shown in figure. From the foreground image, traffic sign is detected, for that hexadecimal value is synthesized. Then the text file is created for equivalent hexadecimal values.

As we can see in Fig. the traffic sign has been restored in RGB and Fig in binary and Figure. The purpose of restoration is to identify what traffic sign it is as it would be easier for comparison. Hence the foreground image from the dynamic background is being reduced to RGB, binary and edge.

In this way the traffic sign is detected by background, color and edges and shape.

a) The input image b) Binary image

c) RGB d) Edges

TEXTFILE CREATION USING VHDL

For the corresponding text file, VHDL code is written for same process. The VHDL code is written for the purpose of converting the text file which are in the form of pixels to an of 8*8 matrix where the pixels are arranged in the matrix as elements. This is for the comparison purpose carried out in the simulation using MODELSIM.

HEX FILE AND BLOB DETECTION

The third step deals with the detection of the blob from the converted hex file. This process is done through MODELSIM where the input image and the reference image in the database are compared based on their binary values.

The Figure shows the code written for the convertion of pixels to the matrix form.A matrix of 8 rows and 8 columns is created representing 64 elements as the pixel size of the image is 64.Then the XOR implementation is carried out which means if the inputs are same then the output is zero suggesting that the traffic sign is same when compared with the database else they are not. As shown in Fig the XOR implementation is carried out for the input image and the reference in the database. Here the clock and reset are adjusted accordingly for the comparison process.

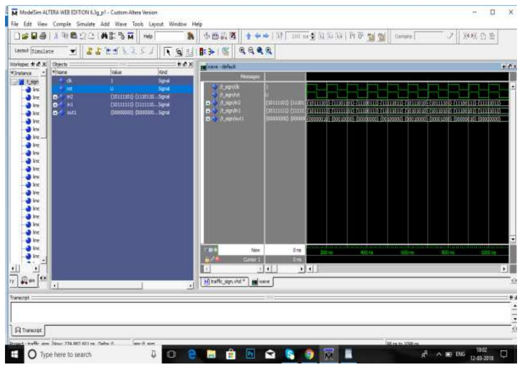
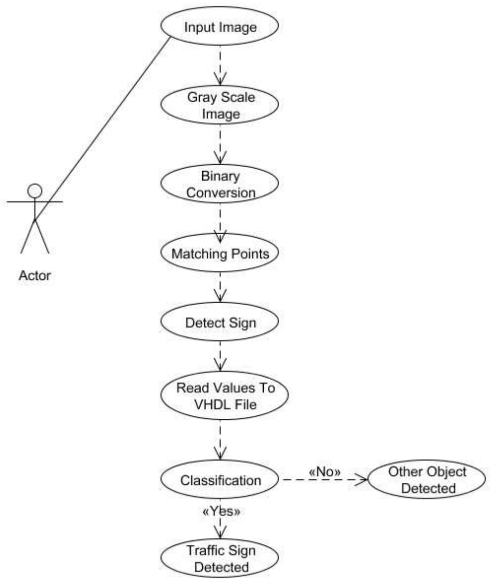


Figure: Simulation output

INTEGRATION

The final step in this process is that of the integration which is dumping the code into the CPLD device which is our case is a Spartan-3E kit and adjusting the pins to obtain the output of what traffic sign is detected through the LED or LCD. Here as we can see the elements of the input image and the reference image are subtracted in order to know the difference through the image is detected. There is also a provision added through which the image detected can differentiate between the traffic sign and any other larger objects through the size mentioned. If the size is less than 4 then the detected sign is a traffic sign else it is a an object of larger magnitude. This helps in the detection of animals which is helpful while travelling in forest areas. Thus the performance is analyzed for power, area and delay measured. Hardware setup is done by Spartan-3E kit and LCD display. An LCD device is interfaced to the Spartan-3E kit for the purpose of displaying the output.



Spartan 3 FPGA kit:

The Spartan-3 family of Field-Programmable Gate Arrays (FPGAs) is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The five-member family offers densities ranging from 100,000 to 1.6 million system gates.

The Spartan-3 family builds on the success of the earlier Spartan-3 family by increasing the amount of logic per I/O, significantly reducing the cost per logic cell. New features improve system performance and reduce the cost of configuration. These Spartan-3 FPGA enhancements, combined with advanced 90 nm process technology, deliver more functionality and bandwidth per dollar than was previously possible, setting new standards in the programmable logic industry. Because of their exceptionally low cost, Spartan-3E FPGAs are ideally suited to a wide range of consumer electronics applications, including broadband access, home networking, display/projection, and digital television equipment. The Spartan-3 family is a superior alternative to mask programmed ASICs. FPGAs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs. The Spartan 3 Starter Kit is a low-priced, compact prototyping module that can be used for rapid proof of concept or for educational environments.

The module is based on the Spartan 3 FPGA from Xilinx along with supporting circuitry to ease prototyping efforts. Designers can use the Spartan-3 kit for general FPGA prototyping, experimenting with multiple FPGA configuration techniques, and proving out low cost design methods.

a) Spartan 3

IV. CONCLUSION

In this paper, we propose design and extraction of traffic sign from dynamic image. In order to achieve high throughput and low energy consumption, we incorporate four novel ideas into our proposed design, including: 1) rearranged numerical operation; 2) shared image storage; 3) adaptive workload distribution; and 4) fast image block integration. The proposed accelerator is implemented and evaluated with a Xilinx ZC706 board. Our accelerator achieves the throughput of 126 frames/s and the energy efficiency of 0.041 J/frame, when processing high-definition video with the resolution of 1920 × 1080.

The proposed design is compared against the CPU-based and GPU-based software implementations and other FPGA-based hardware accelerators to demonstrate its superior performance in terms of throughput and energy consumption. In our future research, we will further integrate the proposed FPGA-based accelerator with other components (e.g., front-view camera, video decoding system, and so on) in order to evaluate the performance for the entire driver assistance system.

V. REFERENCE

[1] Automatic detection and classification of traffic signs by Carlos Philippe Paulo, Paulo Lobato Corriea,2013.

[2] Robust Traffic Sign Detection by means of vision and V2I communications. A. Garc ́ıa-Garrido, M. Ocana, D. F. Llorca, M. A. Sotelo, E. Arroyo and A. Llamazares,2011.

[3] Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives Morten B. Jensen, Mark P. Philipsen, Andreas Møgelmose, Thomas B. Moeslund, and Mohan M. Trivedi,2016.

[4] Multi-column Deep Neural Networks for Image Classification, Dan Cires an, Ueli Meier and Jurgen Schmidhuber,2015.

[5] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[6] . Balali, A. A. Rad, and M. Golparvar-Fard, "Detection, classification, and mapping of U.S. traffic signs using Google street view images for roadway inventory management," *Vis. Eng.*, vol. 3, no. 1, p. 15, Dec. 2015.

[7] S. Lafuente-Arroyo, P. Gil-Jimenez, R. Maldonado-Bascon, F. Lopez-Ferreras, and S. Maldonado-Bascon, "Traffic sign shape classification evaluation I: SVM using distance to borders," in *Proc.*

IEEE Intell. Vehicles Symp., Jun. 2005, pp. 557–562.

[8] X. Baro, S. Escalera, J. Vitria, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 113–126, Mar. 2009.

[9] A. Shafieee*et al.*, "ISAAC: A convolutional neural network accelerator with *in-situ*analog arithmetic in crossbars," in *Proc. ACM/IEEE*

Annu.Int. Symp. Comput. Archit., Jun. 2016, pp. 14–26.

[10] L. Xia *et al.*, "Switched by input: Power efficient structure for RRAM-based convolutional neural network," in *Proc. ACM/EDAC/IEEE DesignAutom. Conf.*, Jun. 2016, pp. 1–6.