An FPGA-Based Hardware Accelerator for Traffic Sign Detection

Abstract:

Traffic sign detection plays an important role in a number of practical applications, such as intelligent driver assistance and roadway inventory management. In order to process the large amount of data from either real-time videos or large off-line databases, a high-throughput traffic sign detection system is required. In this paper, we propose an FPGA-based hardware accelerator for traffic sign detection based on cascade classifiers. To maximize the throughput and power efficiency, we propose several novel ideas, including: 1) rearranged numerical operations; 2) shared image storage; 3) adaptive workload distribution; and 4) fast image block integration. The proposed design is evaluated on a Xilinx ZC706 board. When processing high-definition (1080p) video, it achieves the throughput of 126 frames/s and the energy efficiency of 0.041 J/frame. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Existing System:

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. For instance, a multicolumn DNN is trained for traffic sign classification, requiring 6G floating-point operations to process a single image with 48×48 pixels. To lower power consumption and reduce chip area, analog circuits with emerging devices have been designed to implement CNN. A mixed-signal chip with both analog and digital circuitry 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, lite 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...
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) [19], [20], and so on. Among them, FPGA generally offers a flexible solution due to its reconfigurable ability in both high-level system architecture and low-level circuit implementation and, hence, has been adopted by many industrial applications.

Disadvantages:

- Speed is low
- Resource utilization is low

Proposed System:

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.

1) Rearranged Numerical Operations: For high-throughput image processing, the need to access a huge amount of data poses enormous challenges on memory bandwidth. To address this issue, we propose a feature extraction scheme with rearranged numerical operations in order to reduce the amount of memory access. In addition, the proposed approach greatly reduces the overall computational cost by combining similar numerical operationstogther.

2) Shared Image Storage: With limited on-chip static random access memory (SRAM), it is difficult, if not impossible, to transfer and store a large amount of image data within on-chip SRAM. To address this challenge, we propose a novel buffer architecture which takes advantage of overlapped pixels between successive image blocks and stores them in a compact form. As a result, it substantially reduces the amount of data for both on-chip communication and storage.
3) Adaptive Workload Distribution: When the cascade classifiers are applied, the number of image blocks that pass each classifier is highly random and varies substantially both spatially and temporally, resulting in significant workload variations. To appropriately handle such large variability, we develop a novel parallel architecture with adaptive workload distribution. It dynamically balances the workload among different parallel processing units to achieve high resource utilization.

4) Fast Image Block Integration: For a high-definition image, the detecting results from millions of small image blocks must be combined, resulting in expensive computational cost. To reduce the computational complexity, we propose an efficient bit array to store the detection results of image blocks and a novel iterative filter with highly parallelized architecture in this paper.

Overall System Architecture:

As shown in Fig. 1, the proposed system is composed of four major components: a detection accelerator, a CPU, a direct memory access (DMA), and an external dynamic random-access memory (DRAM). The detection accelerator, CPU, and DMA are all on a Xilinx SoC chip and they are connected to each other. A C-based embedded program is implemented on the CPU to configure and control other hardware components. Because the accelerator is designed with easy-to-use control instructions, the controlling code running on CPU is relatively simple, including less than 300 lines. Under the command of the CPU, DMA transfers video frames from DRAM to the detection accelerator, where cascade classifiers are applied for traffic sign detection. As will be demonstrated by the evaluation results in Section V, three stages of cascade classifiers can already offer sufficiently high detection accuracy in our experiment.

Figure 1: Overall architecture of the proposed traffic sign detection system.
Do Your Projects With Technology Experts…

In order to minimize external memory access and simplify controlling logic, a streaming architecture is designed to implement the proposed traffic sign detection accelerator. Our accelerator contains four major function modules: 1) imagescaling and integration; 2) image cropping; 3) classification; and 4) image block integration, as shown in Fig. 1.

Hardware implementation:

In order to facilitate an efficient accelerator design with high throughput, we further implement four novel ideas at the hardware level: 1) feature extraction with rearranged numerical operations; 2) shared image storage for data buffer; 3) adaptive workload distribution; and 4) fast image block integration. In this section, we describe their implementation in detail.

Feature Extraction With Rearranged Operations:

For each stage of cascade classifiers, a set of Haar-like features are extracted from the integral image. We need to read out several pixels of the integral image to calculate one feature. When calculating multiple features in parallel, a large number of pixels must be accessed simultaneously. It, in turn, constrains the overall system throughput due to limited memory bandwidth. To address this issue, we propose a novel feature extraction module that largely reduces the amount of data transfer by rearranging the required numerical operations.

Data Buffer With Shared Image Storage:

First, when an image block passes the second stage classifier, its pixels must be stored in the data buffer so that the third-stage classifier can process the image block at a later time. If different image blocks are stored independently, it consumes a large amount of on-chip SRAM resources. Second, transferring the image blocks into data buffers is another challenging task. In the worst case, if several adjacent image blocks are continuously transferred to a data buffer, the required communication bandwidth is prohibitively high. For instance, if one image block has 32×32=1k pixels and each pixel is represented by 1 byte, we need to transfer 1 kB data for each image block.

Several techniques have been proposed in the literature to store and transfer image blocks in a compact form so that the amount of data for both on-chip communication and storage can be significantly reduced. For instance, Tikekaret al. propose to store image blocks with overlapped pixels in cache memory. In our application, however, the data entering our buffer are steamed...
from the second-stage classifier, instead of being fetched from the external DRAM. Furthermore, the data should be passed to the third-stage classifier and then be erased from the buffer. For these reasons, a dedicated controlling block must be appropriately designed to manage data access.

Adaptive Workload Distribution:

At the hardware level, it is implemented with a control module that dynamically assigns the image blocks to the appropriate processing unit. Our proposed control module is composed of four major components: 1) a priority comparator; 2) a multiplexer; and 3) two buffer trackers, as shown in Fig. 2.
Figure 2: Circuit architecture for our proposed adaptive workload distribution

Advantages:

- Speed is high
- Resource utilization is high

Software implementation:

- Modelsim
- Xilinx ISE