Robot vision application on embedded vision implementation with digital signal processor

Xiaochun Guan1,2, Jianlin Huang2 and Tinglong Tang3

Abstract
The great development of robot vision represented by deep learning places urgent demands on embedded vision implementation. This article introduces a hardware framework for implementation of embedded vision based on digital signal processor, which can be widely used in robot vision applications. Firstly, the article discusses implementation of a pretrained typical convolutional neural network on the digital signal processor embedded system for real-time handwritten digit recognition. Then, the article introduces the migration of OpenCV software packages to digital signal processor embedded system and the implementation flow of face detection algorithms with OpenCV on digital signal processor. The experimental results are remarkable with convolutional neural networks for handwritten digit recognition. This article provides a convenient and feasible design scheme of digital signal processor system for implementation of embedded vision.

Keywords
Embedded vision, TMS320C6748, LeNet-5, handwritten digit recognition, machine learning

Introduction
Today, artificial intelligence has been widely applied in the field of computer science.1,2 As we all know, machine learning is an important part of artificial intelligence, which is regarded as a new technology that would be integrated into the embedded system. Embedded system based on digital signal processor (DSP) or advanced RISC machine (ARM) is a technology development direction, which has been widely recognized in computer, communication, and information industries with its powerful and flexible applicability.3,4 Embedded system has been widely used in industrial control, traffic management, information appliances, family intelligent management system, network and electronic commerce, environmental monitoring, and robot control. At present, embedded/robot vision applications are mainly implemented on hardware platforms such as DSP, field-programmable gate array (FPGA), and ARM.

In recent years, with the increase of task complexity, the hardware scheme of FPGA + ARM, DSP + FPGA, and FPGA + DSP + ARM is proposed in succession. Thus the task assignment can be further divided for each hardware.
so as to improve the performance of the system. LeCun et al. implemented a back-propagation (BP) network for handwritten digit recognition on a commercial DSP. The final network was trained by BP algorithm. It showed that DSP is an important carrier for implementation of machine learning algorithm. In recent years, researchers have been exploring and studying the implementation of advanced algorithms based on DSP. The DSPs introduced by Texas Instruments (TI, Dallas, Texas) is the typical representatives of DSPs, which have been widely used for their powerful digital signal processing abilities based on optimized architecture. In recent years, there are many successful implementations of advanced algorithms on DSP. Sheng et al. introduced a real-time infrared signal processing system based on TMS320C6748. Feng et al. managed to implement detection and analysis of electromyography based on a DSP system. Vaishnavi et al. implemented brain MR image segmentation algorithm on DSP. Zoubir and Wejdan proposed an intelligent control system architecture based on TMS320C6748. The system can accomplish signal acquisition and control for power conditioning using a novel recursive stochastic optimization. Phalguni et al. designed a system for automatic recognizing the traffic signs based on TI OMAP-L138. Li et al. describe a detection and tracking system for aerial target in the dynamic background on TI AM5728.

In recent decade, with the development of machine learning, the study of robot vision has made a series of achievements. It has solved many problems that are difficult to overcome by traditional methods. For example, the authors proposed a new method for cooperative autonomous localization among air-ground robots in a wide-ranging outdoor industrial environment based on the intelligent collaborative between the aerial robot and the ground robot. And also a novel formulation of the object association problem based on a hierarchical Dirichlet process is proposed, which shows a very impressive improvement with respect to the traditional SLAM.

There would be more and more application requirements of robot vision algorithm in various fields. How to implement these algorithms on embedded systems is of great significance for robot vision application. Furthermore, the continuing growth of robot vision applications places great demands on embedded vision. Industrial vision, video surveillance, and automotive vision are representative of the main directions of the field of embedded vision. For industrial vision application, where the main task is to detect, classify, and sort objects on the assembly line with computer workstations, migrating vision algorithms from costly computer workstations to an embedded DSP is an effective way to save price and power consumptions. As vision algorithms improve and become more stable and capable, video surveillance will incorporate more automatic monitoring and analysis of the recorded data and become more intelligent and sophisticated. Smart camera is in great demand. This presents a particular requirement to embedded video surveillance systems, such as people counting, license plate reading, and so on. For automatic vision, transferring the latest vision algorithms from high-level PC software to DSP is a critical step in the development process of automatic vision applications. So, DSP plays a critical role in the embedded vision applications.

In this article, we introduce a hardware framework for implementation of embedded vision based on TMS320C6748. TMS320C6748 is a fixed-point and floating-point DSP based on a C674x DSP core by TI. This article adopts this framework for implementation convolutional neural network (CNN) on DSP embedded system. The hardware architecture of the system mainly incorporates the peripherals of TMS320C6748, a liquid crystal display module (LCDM) and an image sensor module. It utilizes the powerful digital signal processing capability of TMS320C6748 to implement a pretrained CNN, which is trained on PC based on TensorFlow beforehand.

The article also discusses the port of open-source computer vision library OpenCV to TMS320C6748 to facilitate algorithm development. TI manages to optimize vision systems by running OpenCV on the C6000™ DSP architecture. A face detection application is introduced based on the algorithm function from OpenCV. This article provides a convenient and feasible design scheme of the DSP application system for implementation of embedded vision algorithm.

The rest of this article is organized as follows. The second section literates on the hardware architecture and software development environment of DSP embedded system on TMS320C6748. The third section discussed the porting of OpenCV to TI’s C6000 DSP architecture and its application for face detection. The fourth section describes the experiments of handwritten digit recognition based on CNN LeNet-5 algorithm. The fifth section draws the conclusion remarks and discusses our future work.

System architecture
This section will literates on hardware architecture and software development environment for implementation embedded vision on TMS320C6748.

Hardware architecture
For the hardware, RK6748 is adopted for implementation. RK6748 is a development board with TMS320C6748 as the control core. Its main frequency is as high as 450 MHz. Rich peripherals are integrated into TMS320C6748, such as liquid crystal display controller (LCDC), video port interface (VPIF), external memory interface A (EMIFA), general-purpose input output (GPIO), enhanced direct memory access (DMA) 3, universal asynchronous receiver/transmitter (UART), and so on. TMS320C6748 provides 144 general-purpose pins that can be configured as either inputs or outputs. All these pins are multiplexed by
several peripherals. Pin multiplexing is controlled by the pin multiplexing registers. The peripherals use their corresponding multiplexing pins to interface with the device out of TMS320C6748. For example, VPIF, LCDC, and EMIFA need multiplexing pins to realize their specific peripheral functions. EMIFA memory controller can provide a means for the CPU to connect to SDRAM and asynchronous devices. It can also enhance the ease and flexibility of connecting CPU to external SDRAM and asynchronous devices. TMS320C6748 can efficiently handle image and audio processing tasks. The main hardware block diagram of the embedded system based on TMS320C6748 for implementing CNN is shown in Figure 1. The hardware framework of the system is mainly based on the peripherals VPIF, LCDC, GPIO, and EMIFA of TMS320C6748 to interface with the camera module ATK-OV5640, LCDM, keys, and flash memory, respectively. ATK-OV5640 is a 500 W pixel high-performance camera module. ATK-OV5640 module uses the image sensor OV5640 as the core component, which integrates active crystal oscillator, low dropout regulator, auto focus function, and two high brightness light-emitting diode flashes. The raw image captured by ATK-OV5640 is transferred to TMS320C6748 by VPIF. VPIF supports ITU-BT.656 format, ITU-BT.1120 format, and SMTPE 296 format video receiving and transmitting; VPIF also supports raw data capture, video blanking interval data storage, and clipping of output data. VPIF has two video input channels and two video output channels. Channels 0 and 1 share the same receive architecture, and channels 2 and 3 share the same transmit architecture. In this article, VPIF is configured to support raw data capture mode. With VPIF’s raw data capture mode, the output signal of the camera module ATK-OV5640 can be transferred to the memory of the DSP.

The peripheral LCDC is capable of supporting an asynchronous (memory mapped) LCD interface and a synchronous (raster type) LCD interface. LCDC consists of two independent controllers, the Raster controller and the LCD interface display driver (LIDD) controller. Each controller operates independently from the other and only one is active at any given time. The Raster controller provides the synchronous LCD interface. It supports timing and data for constant image refresh to a passive display. It can support a variety of monochrome and full-color display types and sizes by use of programmable timing controls, a built-in palette, and a gray-scale/serial-izer. Image data are processed and stored in LCDC’s frame buffers. The frame buffer is a contiguous memory block in the system. A built-in DMA engine can transfer the image data from a corresponding frame buffer to the raster engine efficiently, which, in turn, outputs to the data to LCDM by GPIO pins. The LIDD controller supports the asynchronous LCD interface. It can provide full-timing programmability of control signals and output data. In our design, we use the Raster controller of LCDC peripheral to control the interface with LCDM. Code 1 shows the initialization code of LCDC peripheral. The LCDM with a resolution of $640 \times 480$ is used to display the image and the recognition result. It can display pictures, Chinese characters, numbers, and English. The system uses ATK-OV5640 to obtain the raw image in real time and then transmits the captured raw image data to the memory by DMA engine via VPIF. The CPU reads the image data from the memory and recognizes the handwritten digit based on the pretrained CNN algorithm.
**Software environment**

The development environment for DSP system adopts Code Composer Studio v6.0 integrated development environment. RK6748 development board from Rock Embed™ is used as the hardware with xds100v2 as its emulator. Code Composer Studio comprises a suite of tools used to develop and debug embedded applications. It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other features. Code Composer Studio combines the advantages of the Eclipse software framework with advanced embedded debug capabilities from TI resulting in a compiling feature-rich development environment for embedded developers. And also, in order to facilitate the application of TMS320C6748, StarterWare is adopted to support the configuration of the peripherals. StarterWare is a free software development package that provides no-operation-system platform support for TMS320C6748. StarterWare includes Device Abstraction Layer libraries and example applications that demonstrate the capabilities of the peripherals. With the API functions based on the StarterWare, we can easily and friendly configure numerous registers of the peripherals LCDC, VPIF, GPIO and EMIFA for the system. Before using the driver source code to assist the initialization programming of the hardware, you need to download StarterWare software installation package from TI’s official website and then install it. After its installation, there will be multiple folders under the installation directory, among which the folder named “drivers” stores the driving source code of related peripherals. Then we need to add the path of the StarterWare’s installation directory and the path of “drivers” folder together with its subfolders to C6000 Compiler’s include options; we also need to add the library file of StarterWare “drivers.lib” to C6000 Linker’s include options. Now we can use the functions from the driving source code of StarterWare to assist the initialization of the peripherals.

**Porting OpenCV to TMS320C6748**

OpenCV is a free and open-source software package that provides a variety of functions for computer vision. OpenCV is under the permission of Berkeley Software Distribution license, and it is one of the most popular libraries in the field of computer vision. It is written in C++ language and it is usable through C or Python language applications. OpenCV

```c
Code 1. LCDC peripheral initialization.

LCDCPinMaxSetup(); //GPIO pins multiplexing configuration
RasterDisable(SOC_LCDC_O_REGS); // disable raster controller
RasterClkConfig(SOC_LCDC_O_REGS, 38000000, LCD_CLK); // LCDC clock configuration
RasterDMAConfig(SOC_LCDC_O_REGS, RASTER_DOUBLE_FRAME_BUFFER,
  RASTER_BURST_SIZE_16, RASTER_FIFO_THRESHOLD_8,
  RASTER_BIG_ENDIAN_DISABLE); // LCD DMA controller configuration
RasterModeConfig(SOC_LCDC_O_REGS, RASTER_DISPLAY_MODE_TFT,
  RASTER_PALETTE_DATA, RASTER_COLOR, RASTER_RIGHT_ALIGNED); // mode configuration
RasterLSBDataOrderSelect(SOC_LCDC_O_REGS); // frame buffer data ordered by LSB style
RasterNibbleModeDisable(SOC_LCDC_O_REGS); // disable Nibble mode
RasterTimingConfigure(SOC_LCDC_O_REGS, RASTER_FRAME_CLOCK_LOW
  | RASTER_LINE_CLOCK_LOW | RASTER_PIXEL_CLOCK_HIGH | RASTER_SYNC_EDGE_RISING | RASTER_SYNC_CTRL_ACTIVE | RASTER
  _AC_BIAS_HIGH, 0, 255); // raster controller polarity configuration
RasterHparamConfig(SOC_LCDC_O_REGS, 800, 30, 210, 45); // horizontal parameter configuration
RasterVparamConfig(SOC_LCDC_O_REGS, 480, 10, 21, 22); // vertical parameter configuration
RasterFIFODMADelayConfig(SOC_LCDC_O_REGS, 2); // FIFO DMA delay configuration
RasterEnable(SOC_LCDC_O_REGS); // enable raster controller
RasterDMABConfig(SOC_LCDC_O_REGS, (unsigned int)(g_pucBuffer+PALETTE_OFFSET),
  (unsigned int)(g_pucBuffer+PALETTE_OFFSET) + sizeof(g_pucBuffer) - 2 - PALETTE_OFFSET, 0);
RasterDMABConfig(SOC_LCDC_O_REGS, (unsigned int)(g_pucBuffer+PALETTE_OFFSET),
  (unsigned int)(g_pucBuffer+PALETTE_OFFSET) + sizeof(g_pucBuffer) - 2 - PALETTE_OFFSET, 1);
// frame buffer 0,1 configuration
```
is still under active development, with continuous updating to eliminate bugs and add new algorithm function. OpenCV’s greatest characteristic is the rich of algorithms included in its standard distribution. Its algorithms range from low-level image filtering and transformation to sophisticated feature analysis and machine learning algorithms. OpenCV can serve as a comprehensive toolbox of useful and well-tested algorithms, which can be used as building blocks for many specialized applications. So it will be a great appeal to use OpenCV in DSP embedded vision for robot vision applications. Since OpenCV’s original development focuses for use with PC workstations, the task of migrating OpenCV to DSP embedded platforms poses some challenges in C++ implementation, memory constraints, and floating-point support. A great effort has made by TI to optimize vision systems by running OpenCV on DSP architecture. Meanwhile, TI also provides optimized vision and imaging libraries, such as VLIB and IMGLIB.16,17 The IMGLIB supports a wide range of image applications that include compression, video processing, machine vision, and medical imaging. The VLIB provides a collection of C-callable high-performance routines that can serve as key enablers for a wide range of image/video processing applications. The VLIB even provides application programming interface (API) functions of CNN for C66x DSP architecture, such as 2D Image convolution for N Input and M Output channels and ReLU and MaxPooling operations. The functions in these algorithm libraries exploit the high-performance capabilities of TI’s DSPs. These libraries make implementation of vision- and image-related algorithms on TMS320C6748 more efficient.18,19 IMGLIB and VLIB libraries can replace some function of OpenCV. Coupled with these libraries, the OpenCV library can accelerate high-level OpenCV APIs on DSP embedded vision application.

Here, we introduce a flow of face detection based on the ported OpenCV on TMS320C6748. The adopted hardware framework is the same as shown in Figure 1. The main flow diagram is as shown in Figure 2. Firstly, ATK-OV5640 module is used to capture images with VP1F interface of TMS320C6748, and preprocessing is done for the captured images. Then, function cvHaarDetectObjects() in OpenCV is used for face detection in the images. The function cvHaarDetectObjects() uses cascade classifier trained for a target object for target detection; a sequence of rectangular boxes containing the target objects will be returned by this function. Combined with this function from OpenCV, it can realize multi-face tracking and detection on TMS320C6748.

Experiment
The experiment is based on the MNIST database. The MNIST database of handwritten digits has a training set of 60,000 examples and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

With the development of artificial intelligence, deep learning, which is typically represented by deep CNN, has been widely applied to machine vision and image processing system.20,21 It greatly promoted the efficiency and accuracy of machine learning for object detection and machine vision recognition. LeNet-5 is a convolution neural network designed by LeCun in 1998 for handwritten numeral recognition. Most American banks used it to recognize handwritten numerals on checks. It is one of the most representative experimental systems in the early convolution neural network. LeNet-5 has seven layers (excluding the input), each layer contains a different number of training parameters.3 In experiment, we train a LeNet-5 on TensorFlow framework to get a model based on MNIST database, then implement the pretrained model on TMS320C6748. The pretrained CNN LeNet-5 is trained with TensorFlow framework on a PC. By using the function save() in TensorFlow framework, the trained parameters are extracted and then adopted on the DSP. With the trained weights and biases, the forward propagation algorithm of LeNet-5 can be implemented for handwritten digits recognition on DSP. The recognition results can be displayed on TFT LCD.

The software flow diagram of implementation of CNN on TMS320C6748 for handwritten digit recognition is as
shown in Figure 3. Firstly, the image is captured by image sensor Ov5640 in ATK-OV5640 module, setting the output window size of the sensor to \(640 \times 480\), 16 bits per pixel, and RGB565 data format. The received image data are stored in memory by DMA through VPIF interface. When initializing LCDC, the initial address of its frame buffer is set in accordance with the storage address of the raw image data captured by image sensor, which makes the image information captured by image sensor can be seen on LCD in real time. Then, the input of keys is used as the start signal for the processing of captured image data, such as gray processing, down-sampling, and binarization. Finally, the pre-processed image is recognized by the pretrained CNN algorithms and the recognition result can be outputted on LCD.

The parameters of each convolution kernel of the pretrained CNN can be saved and exported to the files. It can be observed that the derived convolution kernel parameters are four-dimensional (4-D), which are columns, rows, output channels, and input channels. Using C language, the 4-D pointers can be established to store the 4-D convolution kernel parameters, so as to facilitate the implementation of the algorithm. In order to reduce the error caused by data underflow, all the parameter data can be multiplied by 10,000 and then exported to the files. Then, in the process of algorithm implementation, the calculation results of the parameters can be divided accordingly, so as to solve the problem of data overflow. Due to the relatively large number of parameters, attention should be paid to the reasonable allocation of memory and the release of memory during algorithm implementation. Thus, it further improves the efficiency of the algorithm.

Table 1 presents the classification accuracy of MNIST on DSP embedded system and PC, respectively.

Table 1. Classification accuracy of MNIST on DSP embedded system.

| Platform                  | MNIST |
|---------------------------|-------|
| PC based on Tensorlow     | 99%   |
| DSP embedded system       | 99%   |

DSP: digital signal processor.

Figure 4 shows an example of capturing the raw handwritten digit “2” by image sensor Ov5640 on ATK-OV5640 and display image on LCD with its recognition result on the left upper corner. LCD: liquid crystal display.
This article takes the application of LeNet-5 convolution neural network on TMS320C6748 DSP as an example to study the implementation of embedded vision on DSP. CNN LeNet-5 is trained with gradient-based BP algorithm. LeNet-5 has been proved to have an accuracy of about 99.2% on MNIST database of handwritten digits.

Conclusion

In this article, we propose a convenient and feasible design scheme for implementation of embedded vision based on single core DSP TMS320C6748. The experimental results are remarkable with a CNN for handwritten digit recognition based on TMS320C6748. Machine learning technology plays an important role in the embedded system, especially in human–computer interaction, self-learning, and so on. With the support of machine learning technology, the self-improvement and learning implementation will become a reality for the embedded vision system, and with the further development of the machine learning technology and robot vision, the embedded vision will have a wider development space. In future, we will want to study on the implementation of more complicated algorithms on TI AM5728 for more complicated embedded vision application.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Youth Science Foundation Project of Zhejiang Natural Science Foundation: Study on Grouping Characteristics of High Dimensional Data in Spectral Data Analysis (LQ19F020006).

ORCID iD

Xiaochun Guan https://orcid.org/0000-0003-3021-9102

References

1. Zhang J, Zhang J, and Chen S. Discover novel visual categories from dynamic hierarchies using multimodal attributes. IEEE T Ind Inform 2013; 9(3): 1688–1696.
2. Zhang J, Chen J, Wang Q, et al. Spatiotemporal saliency detection based on maximum consistency superpixels merging for video analysis. IEEE T Ind Inform. Epub ahead of print 12 October 2019. DOI: 10.1109/TII.2019.2933999.
3. Diao L, Tang J, Loh PC, et al. An efficient DSP–FPGA-based implementation of hybrid PWM for electric rail traction induction motor control. IEEE Trans Power Electron 2018; 33: 3276–3288.
4. Wu Q, Song T, and Li X. Research on the high-speed image acquisition and storage technology based on TMS120C6748. In: National conference on electrical, electronics and computer engineering, Xi’an, China, 12–13 December, 2015, pp. 788–791.
5. LeCun Y, Boser B, Denker JS, et al. Handwritten digit recognition with a back-propagation network. Adv New Inform Process Syst 1997; 2(2): 396–404.
6. Sheng C, Tang X, and Liu S. Design of real-time infrared signal processing system based on TMS320C6748 and SYS/BOIS. Semiconduct Optoelectron 2012; 33(2): 265–269.
7. Yanmei F, Xuefei Z, Junjun W, et al. Electromyography collection and processing based on DSP. Chin J Electron Dev 2014; (5): 830–834.
8. Vaishnavi A, Milind M, and Yogita D. Implementation of brain MR image segmentation algorithm on DSP. In: IEEE international conference on communication and signal processing, Chennai, India, 6 April–8 April, 2017, pp. 2066–2070.
9. Zoubir H and Wejdan AE. Power conditioning with intelligent control using a novel recursive stochastic optimization. IEEE Trans Ind Electron 2019; 66(5): 3721–3730.
10. Phalguni, Kishan G, Venugopala M, et al. Design and implementation of an automatic traffic sign recognition system on TI OMAP-L138. In: IEEE international conference on industrial technology, Western Cape, South Africa, 25–28 February, 2013, pp. 1104–1109.
11. Li L, Zhao G, and Xu S. Design of video aerial target detection and tracking system based on AM5728. In: 30th China conference on control and decision-making, Shenyang, China, 9–11 June, 2018, pp. 6525–6531.
12. Zhang J, Liu R, Yin K, et al. Intelligent collaborative localization among air-ground robots for industrial environment perception. IEEE Trans Ind Inform 2019; 66(12): 9673–9681.
13. Zhang J, Gui M, Wang Q, et al. Hierarchical topic model based object association for semantic SLAM. IEEE Trans Visual Comput Graphics. Epub ahead of print 12 October 2019. DOI: 10.1109/TVCG.2019.2932216.
14. Coombs J, Prabhu R, and Peake G. Overcoming the challenges of porting OpenCV to TI’s embedded ARM + DSP platforms. Int J Elec Eng Educ 2012; 49: 260–274.
15. Texas Instruments. TMS320C6748 DSP technical reference manual (SPRUI79A). 2011.
16. Texas Instruments. Vision library (VLIB) application programming interface reference guide (SPRUG00). 2009.
17. Texas Instruments. TMS320C64x image/video processing library programmer’s guide (SPRUF30A). 2007.
18. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Machine Intell 1986; 8: 679–714.
19. Senthil Kumar Y. Canny edge detection implementation on TMS320C64x/64x+ using VLIB, Application Report. Report no. SPRAB78, Texas Instruments, 2009.
20. Krizhevsky A, Sutskever I, and Hinton GE. ImageNet classification with deep convolutional neural networks. In: International conference on neural information processing systems, Nevada, USA, 3–6 December, 2012, pp. 1097–1105.
21. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, San Diego, USA, 7–9 May, 2015, pp. 1–14.