Acceleration of computer vision applications on the Raspberry Pi using FPGA

Yisong Qiao

29,September,2014

fml – Lehrstuhl für Fördertechnik Materialfluss Logistik
Prof. Dr.-Ing. Dipl.-Wi.-Ing. W. A. Günthner
Technische Universität München

1. Project background and Objectives

2. Project approach

3. Experiment and Result

4. Conclusion

Background

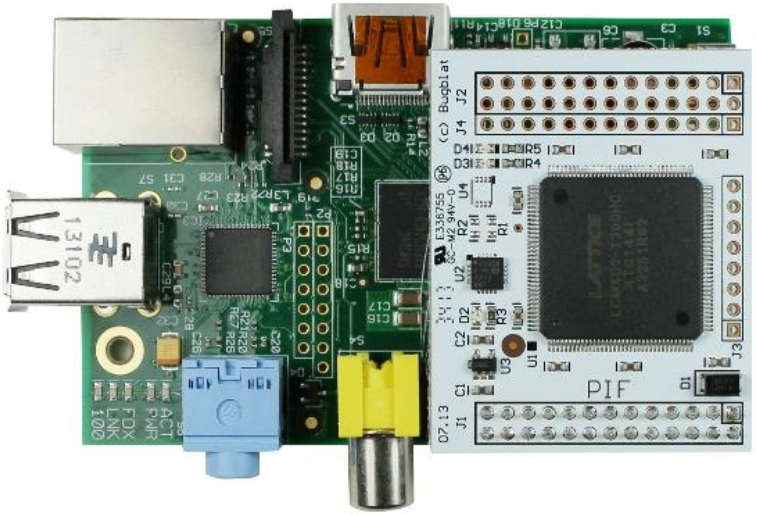
As we know that the forklift trucks are indispensable tools for manual pallet handing. In order to aid the forklift driver especially under limited view condition, we usually equip the forklift with image processing ability by two ways.

1. Using powerful PC to meet highly computation.
2. Doing computation by designing special-purpose logic.

The goal of this project is to develop the image processing system under the FPGA and Raspberry Pi, then integrated into the forklift, in stead of the powerful PC with cheaper, more potable and faster speed.

System Platform

- Raspberry Pi
- PIF (FPGA for the Raspberry Pi).



Raspberry Pi

Differences between Model A and Model B

Features	Model A	Model B
RCA Video Port	✓	✓
HDMI Port	✓	✓
Ethernet Port	✗	✓
USB Port	1x USB 2.0	2x USB 2.0
3.5mm Audio Jack	✓	✓
SD Card Slot	✓	✓
MicroUSB Power Port	✓	✓
Memory	128MB	256MB
CPU	Broadcom 700MHz	Broadcom 700MHz
Cost	$25	$35

PIF - FPGA for the Raspberry Pi

- a complete FPGA development target
- plenty of on-chip 4-input LUTs
- plenty of on-chip 9-Kbit SRAM blocks
- up to 256Kbits user flash memory
- hard coded I2C, SPI, PLL, and timer/counter blocks

Linear filter, nonlinear filter and neural network are the most common and useful tasks when we do the image processing.

System Tasks:

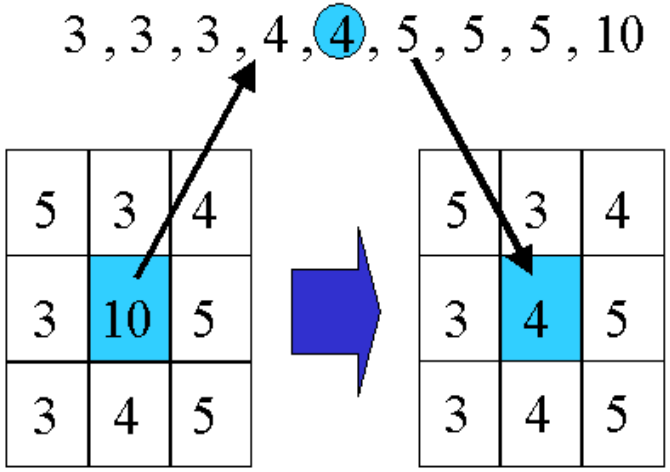
1. Develop a concept for porting different kinds of image filter
- Convolution algorithm (linear filter).
- Sorting algorithm (nonlinear filter).
- Artificial neural network.

2. Implement one filter of each kind in Hardware
- Gaussian filter (blurring).
- Median filter (noise reduction).
- Multilayer perceptron (MLP) neural network (auto detection).

3. Compare the Performance with pure software version
- FPGA vs OpenCV

Sorting algorithm

Set a window of 3 by 3 then sort all the value.
Make the middle value of the window as the Nth value.
Median Filter : N equals 5.

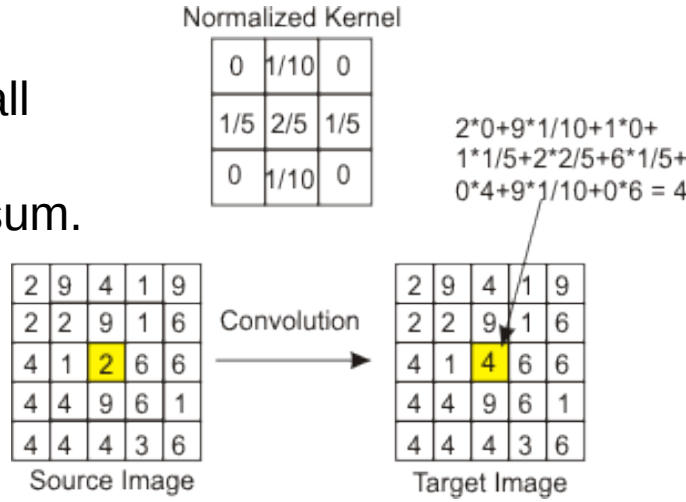


Convolution Algorithm

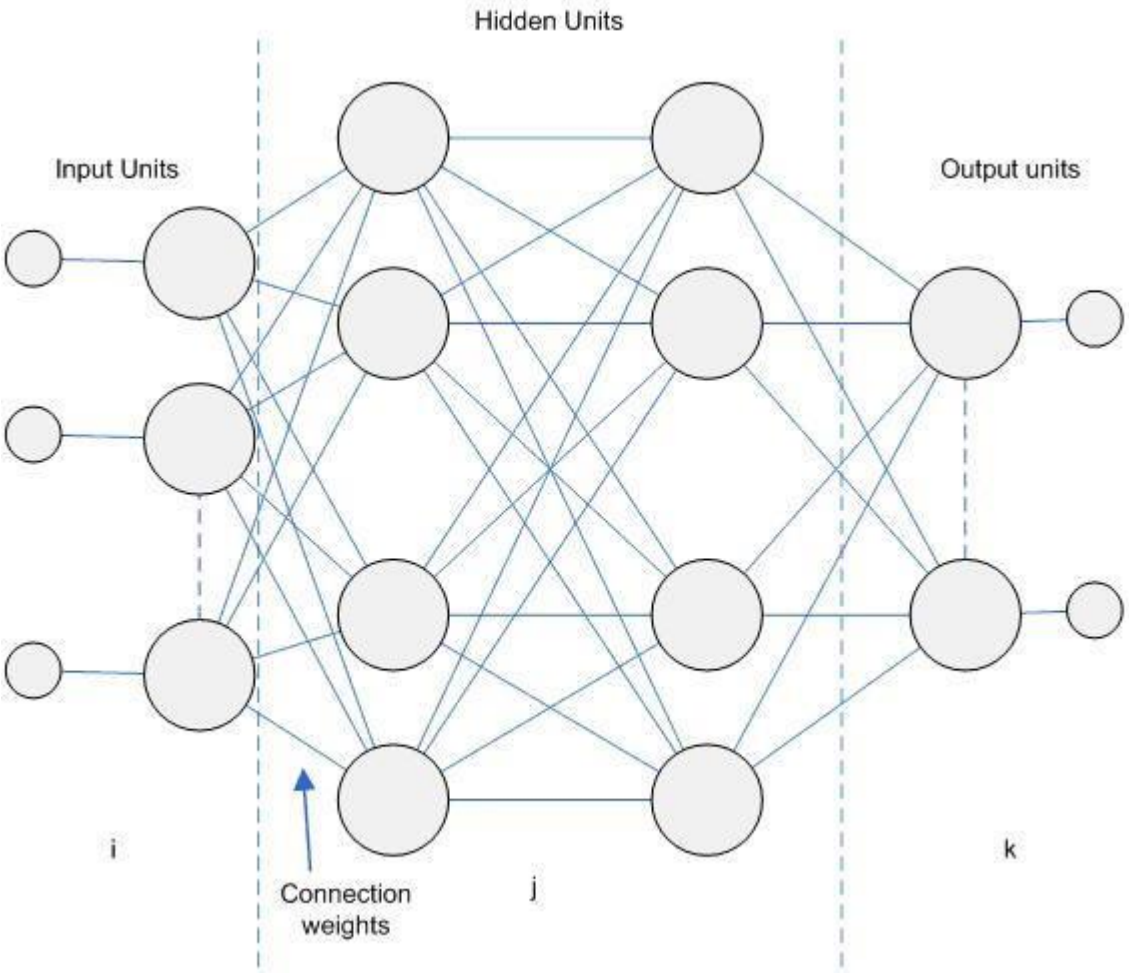
Set a kernel window of 3 by 3 , multiple the correspondent value between kernel and data, sum all the value.
Make the middle value of the kernel window as the sum.
Gaussian Filter : Gaussian Kernel.

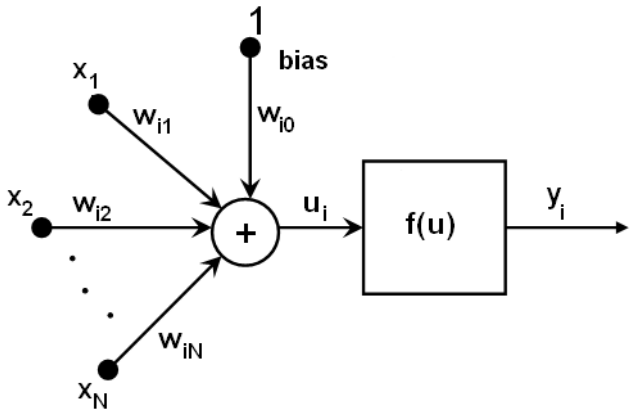
1	2	1
2	4	2
1	2	1

* 1 / 16



Multilayer perceptron neural network





weighted sum :

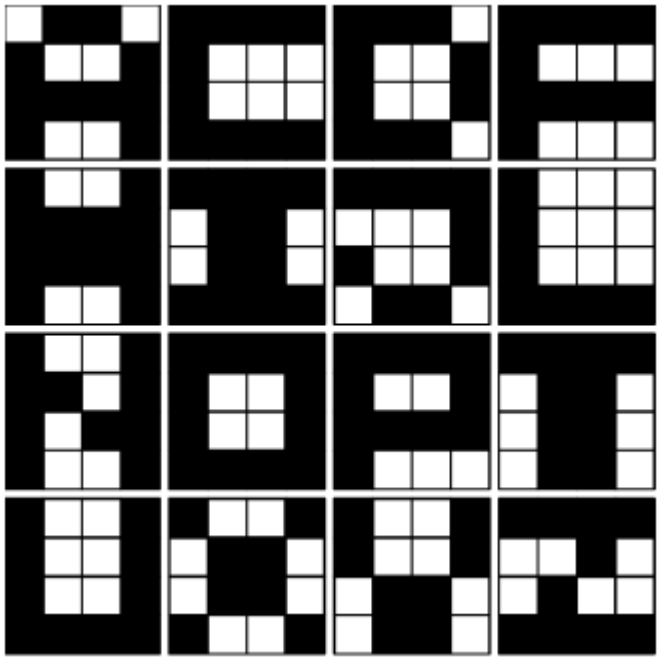
$$s_j = \sum_{i=0}^{n} w_{i,j} o_i$$

passed through an activation function, usually sigmoid:

$$o_j = \frac{1}{1 + e^{-s_j}}$$

Parameters and hardware interface

Image input :
640x360, 8 bit, grayscale.
“Salt-and-pepper noise” for median filter.
Normal image for Gaussian filter.
Size : 230400 bytes



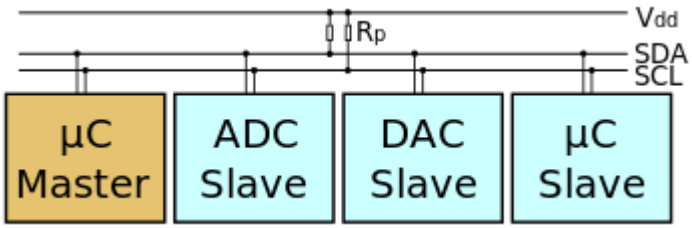
Pattern input :
4x4, 32 bit, grayscale.
Neural network structure: 16,32,16.
16 letters are currently trained to be recognized: A, C, D, F, H, I, J, L, N, O, P, T, U, X, Y, Z.
Example:
A : {0,1,1,0, 1,0,0,1, 1,1,1,1, 1,0,0,1}.
B : {1,1,1,1, 1,0,0,0, 1,0,0,0, 1,1,1,1}.
Size : (32+17*32+33*16)*4 = 4416 bytes .

Hardware interface:

The pif board only has two kinds of hardware interface.

1. I2C , Inter-Integrated Circuit, 500kHz.
2. SPI , Serial Peripheral Interface, 7.8125MHz.

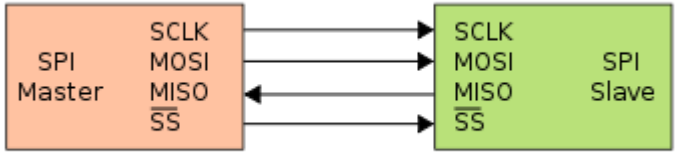
I2C interface



I2C is half-duplex, synchronous, multi-master bus

SPI bus is a synchronous serial data link and operates in full duplex mode.

SPI interface



Duo to the huge size of the input, the speed of transfer is considered as the key component. Using the SPI interface could have better result than I2C.

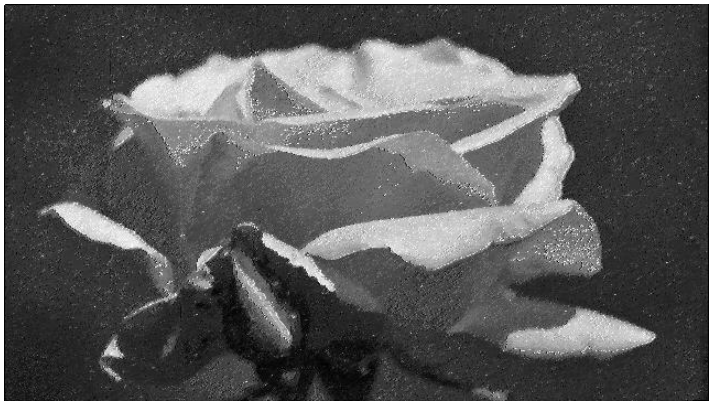
Result (only transfer)

640x360, grayscale, original

SPI interface

input size	time / speed
230.4KB	0.120s / 1.831MB/s

Result (Median Filter)



PIF (FPGA)

OpenCV

	Time (hardware with SPI)	Time (total - transfer)
FPGA	0.180s	0.060s
OpenCV	0.060s	0.060s

Result (Gaussian Filter)

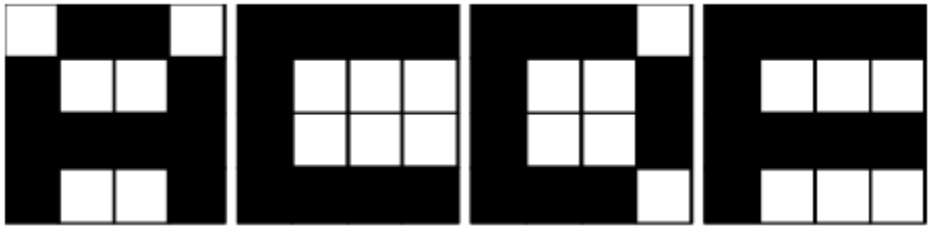
PIF (FPGA)

OpenCV

	Time (hardware with SPI)	Time (total - transfer)
FPGA	0.120s	~ 0.000s
OpenCV	0.020s	0.020s

Multilayer perceptron neural network

Output : Detect for “Z”
0.97503 0.998393 0.968699
1.00423 -0.00983099 0.0638887
1.00992 -0.0138897 0.049473
0.975621 0.0662289 -0.020735
0.957592 0.986483 0.990517
0.970726



Pattern Recognition

100 samples	Transfer	Time (hardware with SPI)	Time (total - transfer)
FPGA	0.006s	0.014s	0.008s
OpenCV	0.000s	0.010s	0.010s

Conclusion

1. From the experiment result, the most time consuming part of the hardware system is the transferring part. The computation of the FPGA is faster than the software but need more time to load the image or weights of neural network.
2. As the image size, weights and bias of neural network are large number of data, larger of internal storage is needed. FIFO (first-in-first-out) buffer is used to store one line of image data at each time, but weights and bias should load into the storage one time.
3. The windows size of the convolution may need to be developed more generic.

Thank You
Q&A