RESEARCH ARTICLE OPEN ACCESS

Real-Time Detection of Multiple Objects from High Resolution Video Feeds Using Multiple Raspberry Pi's And OpenCV

Preeja Priji [1], Pranav JI [2], Rashmi S Nair [3]
MTech Student [1], Assistant Professor [3]
Department of Computer Science
Mohandas College of Engineering, and Technology, Trivandrum
Department of Computer Science [2]
Kerala University
Kerala - India

ABSTRACT

Raspberry Pi is a credit card sized computer that runs on ARM processor. Mobility, lower power consumption, scalability and lesser cost are the highlights of Raspberry Pi. Image processing is a task that requires a lot of CPU resource. Image processing in Raspberry Pi using OpenCV is comparatively slower than in a desktop machine. A video may contain many objects. If we use Haar Cascading for finding a single object from an HD image, using Raspberry Pi, it takes approximately 1 second. For multiple objects, the process requires more than 2 seconds making it impossible to process in real-time. As a solution, we propose a method to detect multiple objects by using multiple Raspberry Pi's working together by sharing the source feed and using Haar cascading for detection.

Keywords :— Multiple object detection, Cluster computing, Raspberry Pi, Haar Cascading, Image Processing.

I. INTRODUCTION

Raspberry Pi is a low cost, low energy, credit card sized computer that works on ARM processor. The latest version of Raspberry Pi is version 3 which has a 1.2GHZ quad core processor ,1 GB RAM, Wi-Fi, Bluetooth,40 GPIO pins, 4 USB ports, 1 Ethernet Port, 1 HDMI port, 1 3.5 mm audio jack and ports for camera and display. Raspberry Pi runs variety of operating systems based on Linux kernel and runs Windows based IOT OS. The operating system we are going to concentrate is the official Raspbian, a Debian based operating system for Raspberry Pi.

We use Python 2.7 and OpenCV 3.0 for image processing. The basic idea is to implement a cluster computer that can process high resolution images from video and detect multiple objects using Haar Cascading [5]. For this we require at least 2 Pi's, A master and slave configuration is used. One of the Pi act as the master. The master is responsible for synchronization, interaction, and scheduling. Only the master requires a display and input devices. Slaves and master will be connected to a router. Video source can either be a camera attached to Raspberry Pi or a video file or another camera stream in the network. This should be selectable by the user. For all sources, same Procedure is followed. Since Raspberry Pi has 4 cores we will divide each image into two slices and process with multi-threading to increase speed and the slave pi only require a lite version of Raspbian since it doesn't require GUI. Storage requirement is also low so just 4 GB class 10 memory card is enough for the slaves.

This paper is divide into five chapters. Chapter II deals with the setup and preparation of Raspberry Pi's and required software. Chapter III deals with the Structure and working of the system. Chapter IV will be the conclusion and Chapter V will be the future works.

II. SETUP AND PREPARATION OF SYSTEM

There are two sections in this chapter, section A explains the setup and hardware connection of Raspberry Pi, Section B explains about the basic software required for the operation.

A. Hardware Setup.

We require at least two raspberry pi for the system. Choose one of the Pi as the master to which we connect the display and input devices for interface. We should name this device as master. The rest of the Pi's can be appropriately name. We use one Pi for one object cascading. So accordingly, we can choose Pi for our need. The rest of the Pi along with the Master Pi should be connected using a highspeed router. In the router, we should port forward the master Pi for global access Total Pi's can be calculated as

$$No\ of\ required\ Pi's = No\ of\ Objects\ to\ be\ recognized + 1$$

The Pi can be arranged in racks with appropriate cooling if necessary. The Master Pi will need a 5V 2.5A power supply and for slaves a maximum of 1.5 ampere is more than enough. It is better to use small length cables instead of Wi-Fi, if there are enough number of ports in the router.

In brief, we need to connect the components together as shown in Fig.1.

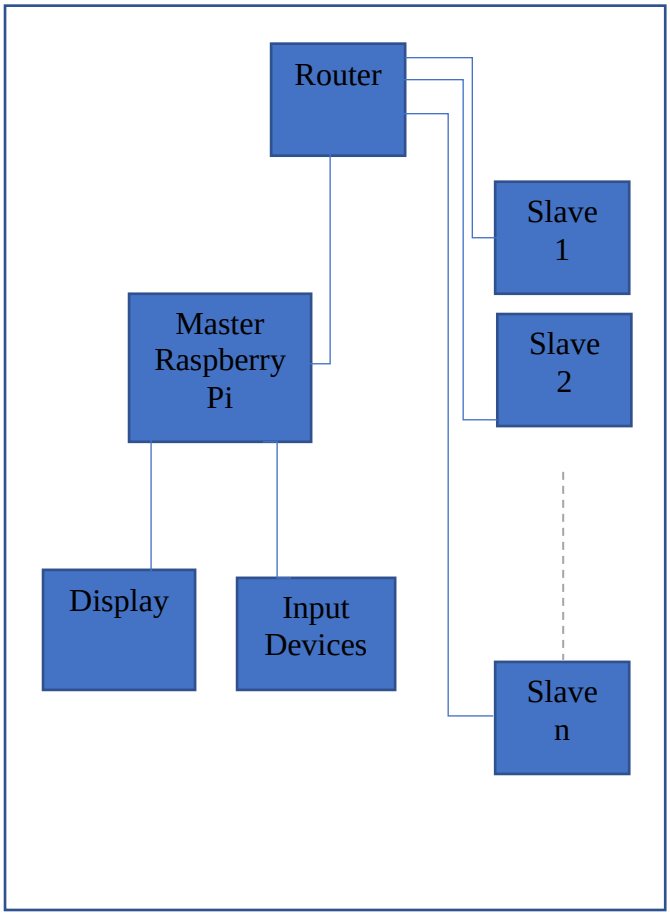

Fig.1 System Model

B. Software Setup

The operating system we are going to use is Raspbian and Raspbian lite, which is a raspberry pi, Debian based OS. We setup the heavy HUI focussed system on master Pi, Master Pi must have at least 16 GB class 10 memory. After naming the Pi as master, connect and port forward the Pi. Now install latest version of Python from repository. Download OpenCV 3.0 sources and compile for raspberry pi and install with Python bindings and all dependencies. Finally install OpenCV Python binding from repository. For slaves, Haar cascade XML [1], [2] for the object has to be saved in the local system or can be distributed during initialization by the master.

For slave Pi's, use command line to install latest version of Python. Same as done in master, compile, and install OpenCV without GUI kit.

For Streaming from camera connected to Master Pi, install motion, a software for video streaming and configure it to stream outside local host.

Once the software setup is complete, check the reachability of slaves by pinging from master. It is a good practice to reserve local IP for Pi's MAC addresses to prevent DHCP from changing IP.

The SSH can be turned on for remote access or VNC server can be activated in the Master Pi for Visual Control of Pi remotely via Internet.

III. STRUCTURE AND WORKING OF THE SYSTEM.

The System is now setup for working. The processing strategy can be applied. This chapter is divided into 2 sections. Section A explains the working of the Master Pi and Section B explains the working inside a slave node.

A. Working of Master Pi

Master Pi controls and coordinates all the tasks. During the The master goes through four phases

1) Initialization
During initialization, the master calculates the number of slaves needed for the job. Then the IP list of the Pi are stored along with the objects they are going to recognize or object id. Then the cascading XML are transferred to the slaves using FTP along with a unique object id.

2) Preparation
During this phase, the capture device is prepared. TCP Connection is established with slaves and master checks if all slaves are ready.

3) Process frames
A frame from the video source is extracted and stored locally, the location will be the source of an FTP server. The current time stamp is updated and if any previous frame is being processed is stopped.
The FTP link along with time stamp is sent as request to all slaves. The slaves download the image simultaneously.

4) Process Results

As soon as a result is received from any of the slave, the time stamp and object id is checked and the region is marked. If replies with more accuracy again, the list regions are remarked. If a slave completes and master is waiting for anther reply, the master assigns the free slave to process the 2nd slice of the frame with an object which any one of the slave is still processing and hasn't replied not even for the first time and waits for another slave to reply. The waiting process continues till a live time for a frame expires. As soon as live time expires another frame is grabbed. The ideal live time will be less than 1 second.

B. Working of Slave Nodes

Slave Pi listens to the master Pi and is synchronized by the master Pi. The slaves are connected to masters using TCP/IP, where master is the server and slaves are the clients. The ports that are going to be used are predefined. As soon as slave receive the request with the URL to a frame image hosted by master and synchronization time stamp. The slave drops current job irrespective of its status and makes a copy of the image. The image is then read into memory and divided into two slightly overlapping horizontal slices. The image is then sub sampled to a lower quality image. The three images are then subjected to Haar cascading as in [3], [4] under multiple thread with subsampled image with higher priority. If objects are detected by any of the thread, the object regions, time-stamp, and object id are immediately passed to the master. If the next frame is not received immediately, the slave continues to process the sliced image and try to detect the regions and transfer it to master after refining the first result. If time is still left, the slicing area is altered and haar cascading is done.

Thus, the process is done so that time is the most prioritized factor even if all objects in a frame are not processed, the next frame is processed.

IV. CONCLUSION

The proposed system is very fast and cost effective compared to super computers. The system can perform well for large number of objects and images. The hit rate can be increased by performing multiple subsampling. The power consumption is also significantly low. The setup and working is simple and can be done with minimum time and expertise.

V. FUTURE WORKS

As future work, haar cascading can be replaced with Artificial neural networks and use the same methodology. For more efficiency, more than one Pi can be used for detection of same objects. The system is easy and faster to implement than most super computers.

REFERENCES

[1] K.S.Shilpashree , Lokesha.H , Hadimani Shivkumar, Implementation of Image Processing on Raspberry Pi, M.Tech, VLSI Design and embedded system, E & C Department, Kalpataru Institute of Technology, Tiptur, India1 Senior Scientist, ALD Division, DSP Lab, CSIR-National Aerospace Laboratories, Bangalore, India2 Associate Professor, E & C Department, Kalpataru Institute of Technology, Karnataka, India, ISSN (Online) 2278-1021 ISSN (Print) 2319-5940 International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015

[2] Vijayalaxmi1 , K.Anjali2 ,B.Srujana3 , P.Rohith Kumar4, OBJECT DETECTION AND TRACKING USING IMAGE PROCESSING. Department of ECE, Vignan Institute of Technology and Science, Hyderabad. Global Journal of Advanced Engineering Technologies, Special Issue (CTCNSF-2014) ISSN (Online): 2277-6370 & ISSN

[3] Preeja Priji, Rashmi S Nair," A Survey on multiple face detection and tracking in crowd", IJIET, Vol 7, Issue 4 Dec 2016

[4] Preeja Priji, Rashmi S Nair," Real time multiple face detection from multiple angles using multiple haar cascade classifiers and convolutional neural network", IJARTET, Vol 4, Issue 6, April 2017

[5] Afshin Dehghan, HaroonIdrees, Amir Roshan Zamir, and Mubarak Shah, "Automatic Detection and Tracking of Pedestrians in Videos with Various Crowd Densities", Computer Vision Lab, University of Central Florida, Orlando, USA e-mail: adehghan@cs.ucf.edu U. Weidmann et al. (eds.), Pedestrian and Evacuation Dynamics 2012, DOI 10.1007/978-3-319-02447-9__1