Surveillance Camera

Kuldeep Soni
Indian Institute of Technology Kanpur
ksoni@iitk.ac.in

Devendra Sunariwal
Indian Institute of Technology Kanpur
devend@iitk.ac.in

June 22, 2014

Abstract

We have made an advanced surveillance camera capable of face detection and at the same time recognizing the face detected. The face detection and face recognition has been done using `OpenCV` library. The code was written in C++. And this all processing has been done on `Raspbian OS` on `Raspberry Pi`. For capturing the images we are using `Pi Camera Board`. With face recognition capability it can help to improve the security systems.

Introduction

The main aim of our project is to make an advanced surveillance camera with face recognition features enabled on it. For achieving this task we have chosen Raspberry Pi as our Processing hardware with Raspbian OS. The visual input is taken by Pi Camera Module attached to Raspberry Pi via cable into the **CSI port** of Pi.

The whole code for face detection and face recognition is written in C++ using OpenCV libraries. Once the power is switched on, the system is programmed to function on its own. For this we have used bootable scripts to directly run the program for face recognition.

Once a face is detected in a frame, the face is cropped out of the frame and it is then matched against the trained database on Pi. If the face is recognized as a familiar one then it is identified as a visitor. If an unknown face is detected then the cropped image is stored. A particular face needs to be recognized only once. The number of recognizable faces depends on the data with which the program is trained.

Further, the updated data regarding the recognized and non-recognized faces can be accessed via a common network shared by Raspberry Pi and the host server.

Motivation Behind the Project

The facial recognition system is a system that is capable of identifying individuals based on their facial characteristics. Most importantly it does not need to make any direct contact with the individual in order to identify him. Whereas this feature is lacking in other biometric identification systems.

With real time face recognition enabled on security cameras, it can help to identify a particular suspect in a huge mob. With such features in mind, we started working on this project as this can be advantageous for surveillance and tracking in clean environments, and in automation systems.

We have decided to make this project to develop a system that will keep a record of the visitors visiting a place. With the help of face recognition it will identify the visitors and will be updating the database at the server further. That database can be accessed by the admin. The whole procedure will be autonomous.

Moreover building such a project using Raspberry Pi and Pi Cam is economical too. As though there are such security cams available in market but they are too expensive.

Background Theory

For developing a fully automated face recognition device we needed a smart processing device and a camera for the visual input. Raspberry Pi seemed to be fit for the job as it has a 700 MHz processor and 512 MB of RAM. For taking visual input we considered Pi camera board with 5 MP picture quality.

The images taken by camera are processed on Raspberry Pi using a C++ code with OpenCV library. The faces identified are cropped from the frame and these are stored for future reference. The data stored can be accessed offline using `Putty` and `Xming` software. It can also as be accessed online as the data will be updated there using LAN connection.

Implementation Details

We have implemented following hardwares and softwares to get our project working:

Hardware

Raspberry Pi: For processing the whole code, we have used Raspberry Pi as it has 700 MHz processor and 512 MB RAM. It even has two USB ports, HDMI port and one 100mb Ethernet port. It works on `Raspbian OS` based on `Debian` which is Linux OS. It is a small sized single-board computer.
Pi Camera: For efficient face recognition we required good camera compatible with our Pi. For this, we used Pi Camera Board which is capable of taking HD videos and stills. It has a 5 MP fixed-focus camera that supports 1080p30, 720p60 and VGA90 videos, and still captures. It can be accessed through MMAL and V4L APIs. It connects to Pi through a ribbon cable at the CSI port.

Software

OpenCV: For image capturing and image processing we have used OpenCV on Raspberry Pi. We have used `Haar feature-based Cascade Classifiers` for face detection. OpenCV i.e. Open Source Computer Vision is a library of programming functions that helps in real-time image processing. OpenCV is written in C++ and its primary interface is in C++.
PuTTY: For accessing Pi through command line on windows PC we needed PuTTY. PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including

SCP, SSH, Telnet, rlogin, and raw socket connection.
Xming: For using remote desktop on Pi Xming was used. It is an `X11 Display Server`. Xming may be used with implementations of Secure Shell (SSH) to securely forward X11 sessions from other computers.
Userland-Masters: Since we are not using Webcam and the Pi Cam is not a **USB** device so we need to install Userland-Masters on Pi in order to make OpenCV compatible with Pi Cam.

First of all we downloaded **NOOBS** zip on our laptop and extracted it. The extracted files were then transferred to the SD Card and when the Pi was booted for the first time we installed Raspbian on it. After that Pi camera was enabled and tested. We could have used Webcam but the frame rate achieved with Pi cam in videos is much higher than in webcam and the resolution is also better.
Then OpenCV library was downloaded and installed on Pi which took nearly 10 hours. Though it was installed but to make it compatible to work with Pi cam we installed **Userland-Masters** library. After doing all that we were then able to access Pi cam using OpenCV.
With face recognition working on Pi, our next step was to make the whole program run on every single boot itself. For this we made the program bootable usin bootable scripts. The processed data can then be accessed using **WinSCP** on Windows.

Limitations

As our Pi was not detected by Router, which was supposed to be used for data updation on network, so we are now giving network connection to Pi via `USB Tethering`.
The accuracy that can be attained for face recognition using OpenCV on Raspberry Pi depends on experimental conditions. So for that the program needs to be trained with many test images and that too in different test conditions.

Future Scope

In future it can replace biometric system for identification such as fingerprint scanner, retina scanner etc. It can be used for `Law Enforcement` as a suspect can be kept under surveillance, and it can even help to identify a criminal in a huge mob.

References

- Installing Raspbian
  - http://www.raspberrypi.org/help/noobs-setup/
- Setting up Pi Cam
  - http://www.raspberrypi.org/help/camera-module-setup/
  - http://thepihut.com/pages/how-to-install-the-raspberry-pi-camera
- Setting Static IP
  - http://robertcastle.com/2014/02/installing-opencv-on-a-raspberry-pi/
  - https://www.youtube.com/watch?v=8Dw2ow9uAiQ

  - http://stackoverflow.com/questions/22383080/raspberry-pi-unable-to-connect-to-internet-with-wifi-shared-over-ethernet

- Remote Access Pi
  - $http : //elinux.org/RPi_{R}emote_{A}ccess$
  - $http : //sourceforge.net/projects/xming/$

- SD Card Backup
  - http://www.electroons.com/blog/2013/02/how-to-recover-the-unallocated-space-on-your-sd-card-due-to-raspberry-pi-os/
  - http://sourceforge.net/projects/win32diskimager/

- Face Recognition
  - 0 $http : //docs.opencv.org/trunk/modules/contrib/doc/facerec/tutorial/facerec_{v}ideo_{r}ecognition.htm$
  - $http : //www.raspberrypi.org/facial - recognition - opencv - on - the - camera - board/$
  - http://thinkrpi.wordpress.com/2013/05/22/opencv-and-camera-board-csi/

- Bootable Scripts
  - http://www.raspberry-projects.com/pi/pi-operating-systems/raspbian/auto-running-programs
  - http://www.stuffaboutcode.com/2013/05/time-lapse-video-with-raspberry-pi.html

- Miscellaneous
  - http://www.raspberrypi.org/forums
  - http://raspberrypi.stackexchange.com/