Embedded Linux Facial and Text Recognition Project

Rob Christenson

Michael Tingey

Project Description

Using the Raspberry Pi device, software is to be developed that allows the Pi to interface with a Logitech C920 webcam to discern the following:

- Is a face on screen?
- Is text on screen?
- What does the text say?

Project Goals (Overall)

The overall goal for the project is to be able to present a document or a face to the webcam connected to the Pi and, on the press of a button, have the Pi determine if a face is present or not, and if text is present, display the text.

Implementation Plans

The project will be implemented using the C++ programming language and the OpenCV API (Open Source Computer Vision). OpenCV provides a relatively straight forward API which contains the capabilities needed for our project. The specifics aspects of OpenCV that will be used include the facial recognition, optical character recognition (OCR) and webcam image capture modules.

Project Components

The project has been broken up into 3 major software components. These components are the image capture, optical character recognition, and facial recognition aspects of the project.

Image Capture

The image capture aspect of the software is responsible for interfacing with the physical webcam and upon button press, capturing image data from the camera. As of now the image is captured and saved for later use, however the button press aspect remains to be implemented.

Optical Character Recognition

The OCR component of the project is almost completely implemented. As of now the OCR uses the given image, sets the color spectrum to grey scale and using a pixel brightness threshold algorithm forces each pixel to either completely white or black in order to aid the openCV OCR API in the detection of text. The text is then detected and sent to the standard output of the pi.

Facial Recognition

The facial recognition component of the project begins by opening the image, converting it from the RGB color space to grayscale, and then equalizes the image's histogram (which helps to improve facial detection). Then, the program utilizes a local binary pattern (LBP) cascade designed for faces to detect possible faces within the image. Following that, a haar cascade designed for eyes is applied to the possible faces and, if two eyes are detected in at least face, the program outputs the result.

So far, the program does work for a subset of sample images, however it has yet to work for an image taken with the C920.

Project Challenges

The design of the project thus far has been a relatively smooth experience. The easy and challenging aspects can be summarized as follows:

Easy

The image capture itself turned out to be a simple matter that just required the use of simple OpenCV commands to capture, format, and save the requisite image file. The actual use of the OCR was also a simple matter, however preprocessing of the image to guarantee greater accuracy turned out to be a challenge.

Difficult

The most challenging aspect completed as of now would be the preprocessing of the image before the OCR algorithm can be run on it. The image needed to be set to greyscale and a brightness threshold algorithm applied in order ensure the text appeared in stark contrast to the background. The greyscale and threshold algorithm had to be determined on our own through the use of photo editing software in order to find the right combination of processes.

Goals (Coming weeks)

The goals for the coming weeks will be to implement the button press aspect in order to capture the image, as well as increase the accuracy of the OCR and facial recognition algorithms. The raspberry pi device itself and the camera will be fashioned together in order to make the project resemble a compact device.

Group Members

Rob Christenson

OpenCV image capture, optical character recognition

Michael Tingey

Facial recognition software, Raspberry Pi hardware