TJHSST Computer Systems Lab 2007 - 2008
Automobile Recognition Through the Use of Image Processing Techniques

by Drew Stebbins

Abstract

Many law enforcement agencies have recently shown interest in automated automobile recognition and tracking technologies such as license plate reading or GPS tracking. However, some criminals may drive vehicles that have false license plates or are not equipped with GPS tracking devices, making the pursuit of such vehicles difficult. This project aims to create a computer vision system capable of taking real-time input from a static camera and identifying passing cars by make and model in order to assist law enforcement agencies in the tracking of suspect or stolen vehicles.

Background

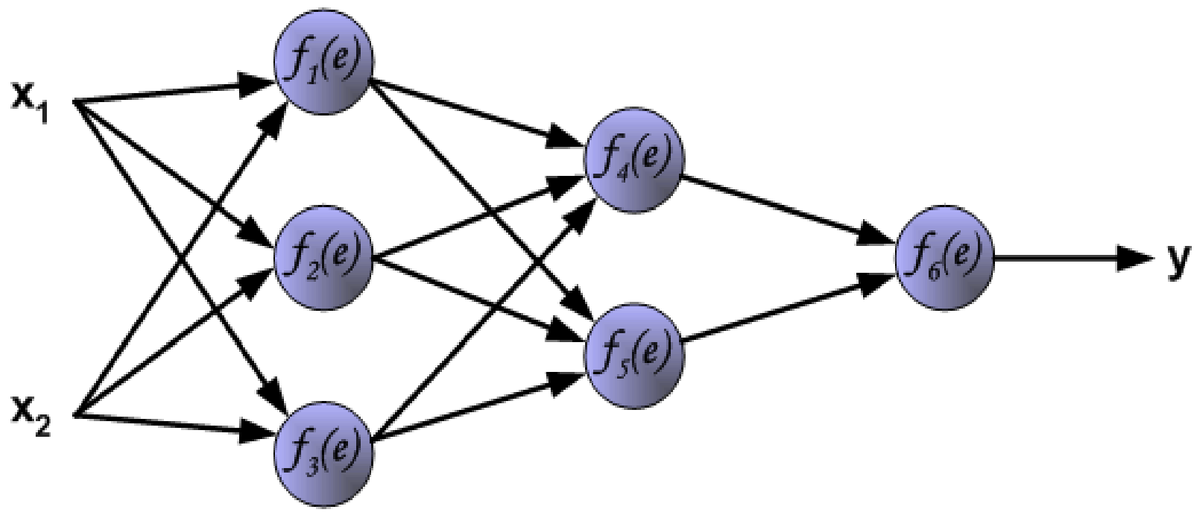
Several computer systems currently exist for the tracking of military and civilian automobiles via License Plate Recognition (LPR) or GPS technology. Such systems are in use by law enforcement entities such as US Customs and Border Protection and the UK police, and have proved very effective in catching criminals. However, these systems fail when an automobile has fake or no plates, and no GPS tracking device, and is able to avoid recognition. My system, on the other hand, will be able to alert law enforcement officers of the presence of any specific shape, color, or size of vehicle regardless of whether or not it is equipped with a GPS receiver or the proper license plates. Some systems already exist that can automatically recognize military vehicles such as tanks, planes, and armored personnel carriers by their shape, size, and color. However, in the course of my preliminary research I found no existing systems capable of automatically recognizing civilian vehicles such as cars or trucks. My system is similar to preexisting systems for the automatic detection of military vehicles in that it defines a certain set of characteristics for comparison, extracts those characteristics from the image of a single vehicle, and searches amongst a list of known vehicles and characteristics for a possible match. The primary difference between previous types of these systems and my own is that mine is much more precise in terms of characteristics such as size and shape, and, when completed, will select possible matches from a much more diverse database.

Procedure and Methods

At the moment, my final program exists in the form of many individually compiled component parts. My image processing programs make use of the Canny edge detector, Hough transform line detector, and my own vertex detection algorithm. My streaming video input GUI uses the video4linux API and GTK+ graphical toolkit, and will eventually be integrated with the code in my image processing programs. My single-hidden-layer, feed-forward neural net uses the backpropagation learning algorithm to do simple optical character recognition (I will most likely not be using a neural net in the final version of my program due to the advantages of other methods of object recognition). The language I am doing all of my programming in is C++, which I have selected based on its fast speed and hardware access capabilities.

Results

The various components of my program currently perform as expected, accurately detecting lines, recognizing handwritten characters, and displaying video input. The frame rate of my streaming input viewer is somewhat low, a problem which will have to be addressed before accurate motion detection can be attempted. I hope to be able to test an assembled version of my final program on short video segments of cars in motion by the end of the third quarter.



A simple feed-forward, double-hidden-layer neural net

Webcam used, and streaming input as viewed with GUI

Output of Canny edge detector