Designing an Automatic Gate: A Machine Learning approach

Utkarsh Ojha
Department of Computer Science and Engineering
Motilal Nehru National Institute of Technology Allahabad
Allahabad, 211004

Abstract

This paper outlines the development of an automatic gate with the help of a microprocessor using a machine learning approach. The first part of the paper describes the traditional methods which have been used to trigger the automatic opening and closing of gates. The second part gives an idea on how a machine learning approach can be used to overcome the shortcomings of traditional triggering methods, and improve the efficiency in detecting when to open or close gates automatically.

1 Introduction

The inconveniences caused by manually operating the closing and opening of gates 24x7 has called for an immense search for solutions. The automatic gate described here can automate the entrances to parking lots of residential areas, organizations, public parks etc.The technology used eliminates gate monitoring and manning by human beings. Specifically, the system described in this paper monitors two gates, the entrance and exit. The automatic gate senses any vehicle approaching it. It automatically opens, waits for a specified time, and closes after the time has elapsed. As soon as the gate closes, the system counts, registers, and displays the number of vehicles. The system also serves as an automobile parking control unity by periodically checking the number of vehicles that have entered the area and computing the available space limit in the parking area. Once the available space limit is reached, the system triggers an alarm for a specified time and the entrance gate remains inaccessible until another vehicle comes out through the exit gate.

2 Brief on traditional techniques for gate automation

The implementation of the system involves segmenting the overall system into modules or sub units, which are individually designed and tested prior to integration. The system design is divided into:

- Sensor unit
- Trigger circuit
- CPU module
- Memory module
- Gate control unit

The block diagram of the system is shown in Figure 1. The system senses, opens and closes the gate, counts, registers, and displays the number of vehicles crossing the gate (both entrance and exit) and triggers an alarm once the space limit is reached. Once triggered, the gate remains inaccessible until another vehicle leaves through the gate.
The only module that differs in this traditional automation system and the one using machine learning

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

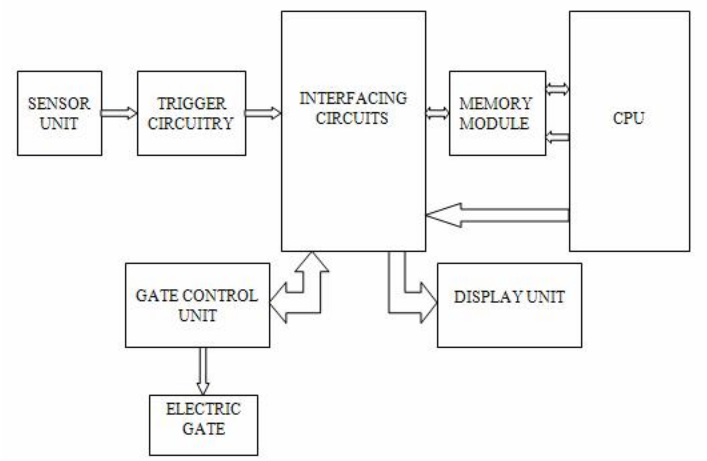

Figure 1: Block diagram of traditional automatic gate system

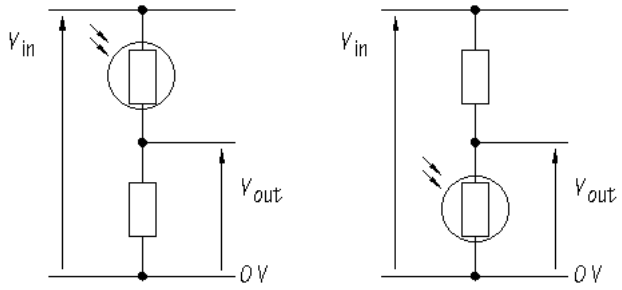

Figure 2: Light dependent resistor based sensors

approach is the *Trigger Circuitry*. So in this section, we'll be discussing the trigger circuitry in this traditional system and its shortcomings, and then will later see how the idea of machine learning can attempt to solve it.

2.1 Sensor Circuitry

The sensor which provides the input signal to the trigger circuitry consists of a *Light dependent resistor* (LDR). When light rays get focused on the LDR, its resistance value goes down, and hence the input to the trigger circuitry is held LOW. But when an object interrupts the beam, the LDR's resistance becomes high and the LDR is said to have dark resistance and the input to the trigger circuitry goes HIGH. The trigger circuitry serves as an Analog–to–Digital converter (ADC), which produces a HIGH signal when the beam is interrupted, and LOW signal when it is not. This HIGH or LOW signal is used by the subsequent modules for further action. From Figure 2, we can calculate:

$$V_O = \left(\frac{R_{Bottom}}{R_{Bottom} + R_{Top}} \right) \times V_{in}$$

Typically, the value of dark resistnce (R_{Bottom}) is approximately $10M\Omega$ (quite large than R_{Top}) . So whenever light beams focus on the LDR, high value of R_{Bottom} causes the value of V_o to be nearly equal to V_{in} i.e., close to 5V. And the case in which the light rays don't get focused sufficiently on the LDR results in V_O to be nearly equal to 0V. Two pairs of sensors (4 total) can be used for the entire system; each pair for the entrance and exit gates.

2.2 Trigger Circuitry

This is made up of a Schmitt-trigger (Figure 3), Quad two-input NAND gate (74LS132), to form a de-bounced switch circuit. This circuit is designed in such way that when there is a positive output

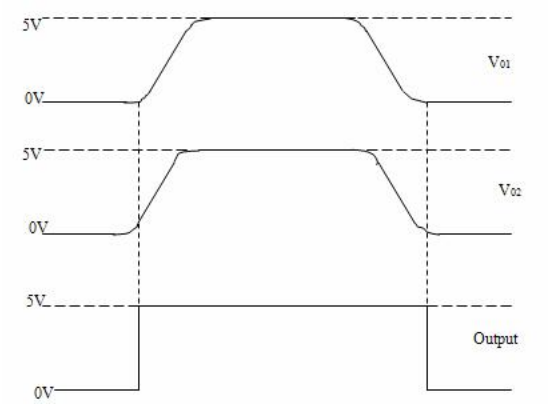

Figure 3: Noisy and denoised images

from both the sensors (nearly 5V), only then a *trigger* signal will be sent to the subsequent modules. Logically, this implies AND'ing the outputs of the two sensors, which is usually performed by using two NAND gates (74LS132). The logic behind AND'ing the inputs from two sensors rather than OR'ing them is that we want only large objects like cars (which can trigger both sensors) to result in automatic opening of the gate, rather than a small object like any stray animal for instance(which will usually trigger one of the sensor).

2.3 Flaws in this trigger circuitry module

The system described above, if deployed in reality, is sure to perform poorly.

- It may fail most of the time when a motorbike or a scooter approaches the gate. Since they are not as large as cars, the two sensors won't be able to signal simultaneously, resulting in the door remaining closed.
- Consider the case in which the weather is sunny, and now gradually dense clouds begin to accumulate, resulting in drastic change in the intensity of light falling on the LDR's. Now the sensor may consider this case as the case of *vehicle approaching*, and open the gates for no reason. Moreover, the gates will most probably remain open for a long period of time, again for no reason.
- Similar kind of problem may arise during the night time. The intensity of street lights **may** be not quite similar to the natural sun rays, and so they may not cross that threshold required to produce the triggering signal.

3 The Machine Learning approach

Our idea is to build a binary classifier which takes 3 input values as its feature vector (described below), and outputs 0 (gate remains closed) or 1 (gate opens). For building such a classifier, we need to train a mathematical model on a labeled data set, each data point of which will have four entries (three for input and the fourth one being 0 or 1). The three input features are explained as follows:

- The output voltage V_{OA} from the first sensor unit.
- The output voltage V_{OB} from the second sensor unit.
- The average of

Reasons for selecting these input features:

The **third input feature** can be understood with the help of an example. Consider a case when there was natural sunlight focused on the LDR. Assume initial value of V_O was 0.8 V. Now a car approaches the gate, increasing the resistance of LDR, resulting in increase in output voltage, say V_O = 4.7 V. This time (say T_1) between which V_O changes from 0.8 V to 4.7 V will be quite small,

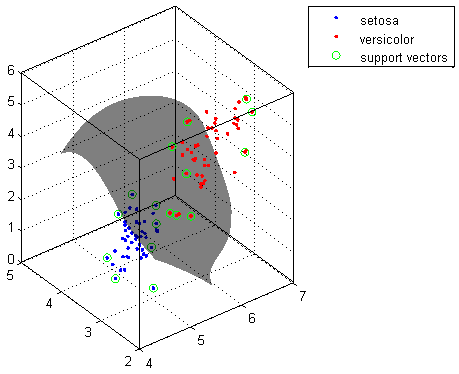

Figure 4: Illustration of binary classification in 3-D

since the car had some speed. Now consider the same initial situation, but in this case instead of the car, heavy clouds get accumulated *slowly* over the sensors, resulting in the output voltage V_O to become 4.3 V. Note that this time (say T_2 required to change the output voltage from 0.8 V to 4.3 V will be very small, which is obvious from the fact that clouds won't be moving with the same speed as the cars. This difference between T_1 and T_2 doesn't play any role in the traditional triggering system. But as we've seen, it gives a fairly intuitive idea about the probability of presence of car or bike, which makes it an important feature to be considered.

The **first and second input features** can be really helpful in overcoming the first shortcoming, i.e. the case in which the gates remained closed even when a motorbike had approached, because only one of the sensor had prominent V_O. Now lets say that our data set contains several examples in which only one of the output voltage values (either V_{OA} or V_{OB}) is prominent, and the output value is 1 (fourth entry in a data point). So if our model has been trained on this data set, it has *learned* intuitively that even if one output voltage is not prominent, chances are that there is a vehicle waiting, and requesting opening of the gate.

Obtaining the data set:

For obtaining the data set, a gate with specifications as described above should be considered, i.e., two sensors for entrance and two for exit. The difference will be that this phase will involve gate opening and closing operations done by a human being, so that we get correct outputs for every situation. A computer program will read in values of sensors, value of output (gate opened or closed), and compute T_{avg} in two cases:

- **Every 30 minutes:** This set of data points will help the model learn about situations when not to open the gate.

- **Whenever the value of outputs of one or both the sensors changes by a pre-defined threshold value:** This set of points will not only help the model learn when to open gates, but also a part of it will tell the model about the cases when not to open the gates even if there has been a change in the output voltages of the sensors (clouds example).

The idea of deploying this machine learning approach, atleast in theory, will be able to tackle the problems mentioned above. When it comes to choosing a classification algorithm, the best choices are neural networks, support vector machines and bayesian classifier. Figure 4 illustrates how a hyper plane can divide a set of points (having 3 input features) in a 3-D space, resulting in binary classification. Both of our sensor units feed their output voltage values, along with T_{avg}, to the classifier. The trained mathematical model then makes a prediction whether to send a trigger signal (output 1) or not send a signal (output 0) to the subsequent modules.

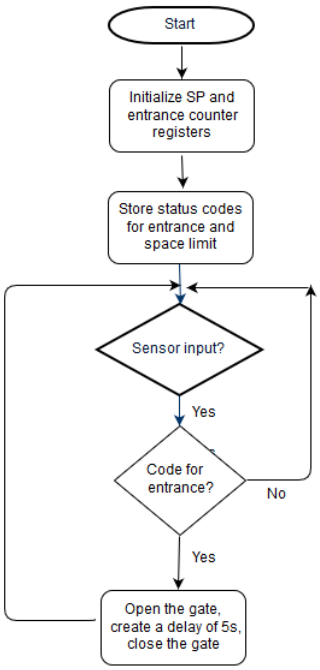

Figure 5: Simplified flowchart of the system

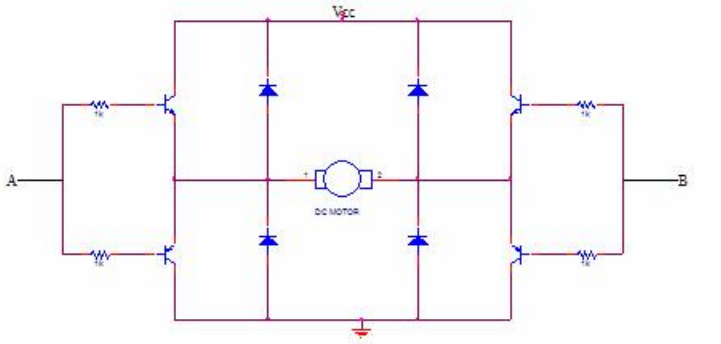

Figure 6: Gate control unit

4 Implementation

4.1 Gate control unit

The gate control unit is made up of NPN and PNP transistors, diodes and an electric motor. The PNP and NPN transistors are arranged in such a way that a pair (PNP and NPN) controls the opening of the gate through the motor and the other pair is responsible for closing of the gate. There is a time difference 5 seconds between opening and closing and vice versa. This delay can be varied by the software implementation described below. The arrangement of diodes serves to protect the transistors from reverse bias polarity. The electric (DC) motor used is the one that has the ability of rotate in both directions simply by reversing the polarity. Figure 6 illustrates a simple circuit that can implement a gate control system.

4.2 Algorithm

The algorithm used to implement the program for the system is described as follows:

1. Start: Initialize the microprocessor.
2. Fetch the status bits from the sensors.
3. Compare the status bits with the entry code.
4. if status == entry code, go to step 5, else go to step 2.
5. Gate open, wait, and close, go to step 2.

Figure 5 summarizes the algorithm.

4.3 Software implementation

The program for implementing this system can be written in assembly language for Z80 microprocessor. When entering an assembly program into the microprocessor, the assembly code must first be converted into the machine code. For longer programs, a separate program called assembler program is used to convert the assembly code into the machine code, which is placed directly into the microprocessor memory. The program is segmented into following modules:

- Main program
- Sensor subroutine
- Delay subroutine
- Gate control subroutine

```
Start:
LD B, 80H
LD C, 00H
LD D, 14H
LD A, 00H
OUT 01H
OUT 04H
OUT O5H

Gate entrance:
LD A, 20H
OUT 01H
CALL Delay
LD A, 40H
OUT 01H  NOP
NOP
NOP
INC H
LD A,H
DAA
OUT 05H
JP Space

Delay:
LD C, 0AH
LD DE, FFFFH
DEC DE
LD A,D
OR E
CP 00H
JPNZ TUNDE2
DEC E
LD A,E
CP 00H
JPNZ TUNDE1
LD C, 01H
LD D, 14H
RET
```

```
Main:
IN A, 00H
CP 00H
JP Z, Main
LD E,A
LD A,B
CP E
JP, Gate

Gate exit:
LD A, 02H
OUT 01H
CALL Delay
LD A, 80H
OUT 01H
NOP
NOP
NOP
INC L
LD A,L
DAA
OUT 04H
JP Space

Input:
NOP
NOP
NOP
NOP
JP Main
```

```
Entrance:
LD A,C
CP E
JP Z, Gate exit:
NOP
NOP
JP Start

Space:
LD A,H
SUB L
CP D
JP NZ, Start
```

5 Conclusions

The design of a microprocessor based gate automation system has been achieved in this project. We had a look over one of the traditional methods which used Schmitt-trigger to send a signal about vehicle detection. We discussed some of the possible cases in which this triggering technique may fail, and introduced a machine learning approach which can help tackle some of the problems faced by the original triggering technique. We discussed a possible strategy to obtain a data set, which should be big enough to help our classifier train properly.

6 Future work

For an improved, effective and security gate system to be implemented and achieved, the following suggestions should be considered for further work.

- A better sensor is recommended to achieve new functionality. For instance, a suitable sensor such as radar sensor that could detect contraband goods in any vehicle.
- To achieve full automation, a real time system should be employed and a Closed Circuit Television (CCTV) system provided for proper monitoring and security purposes. This can helpful in detecting the presence of vehicles before the system is activated.
- **Active learning:** In many cases, gathering data becomes a tedious work, and sometimes due to the unavailability of large amount of data, our machine learning model may not generalize properly. Therefore in cases where we have limited accessibility to data we can make use of *Active learning*, which is a branch of machine learning that deals with this limited data availability problem. So incorporating more advanced level machine learning algorithms may help in improving the efficiency of automatic gate system.

7 References

Shoewu, O. and O.T. Baruwa. 2006. “Design of a Microprocessor Based Automatic Gate”. Pacific Journal of Science and Technology. 7(1):31-44.