EEE-1007 NEURAL NETWORK AND FUZZY CONTROL

J COMPONENT PROJECT FINAL REPORT

Prof. GEETHANJALI P

PIR BASED AUTOMATED SECURITY SYSTEM

(IEEE FORMAT APPLIED)

NAME- SOHAN DAS

REG. NO. - 20BEE0330

SLOT-D2

Machine Learning Based Automated Security System Using PIR Sensor

Sohan Das, Kevin Gerard A and Abhinav Choudhury

Department of Electrical and Electronics Engineering

VIT University

Vellore, India

e-mail: {sohan.das2020, kevingerard.a2020, abhinav.choudhury2020} @vit.ac.in

***Summary*- An Automated Security System uses a digital Pyroelectric Infra-red (PIR) Sensor which detects any movement in the vicinity of the sensor. To increase the efficiency, we are dealing with the signals produced from PIR sensor rather than using pictures as input. To do this we are taking an analog PIR sensor and an object detection system using machine learning. The analog PIR sensor produces an output based on various voltage scales within a certain area rather than producing binary outputs using a threshold value. From the signals produced by the analog PIR sensor, a Fast Fourier Transform (FFT) processed frequency is produced and used as an input of the Artificial Convolution Neural Network (CNN). The artificial CNN then studies the signal pattern of the human motion and animal motion and then detects whether it is human or animal that intruded.**

Keywords* ——— *machine learning; object detection system; PIR sensor; artificial CNN

I. INTRODUCTION

A Pyroelectric Infra-red (PIR) sensor uses the pyroelectric effect of electromotive forces that occurs when it absorbs infrared rays and polarization changes result in electronic charges abandoned. The PIR sensor detects an object based on the temperature change of the surrounding environment and hence produces signals. Using this signal, PIR detects whether it is human or animal. The PIR sensor collects and detects the signal, hence we prefer PIR over IR sensor to make the instrument as small as possible.

The PIR sensor which is based on the differential change in temperature between object and surrounding environment, is highly sensitive when the object is closer to it and henceforth it is less sensitive when the object is far enough to not warm up the environment and creates a problem. Again, during summer when the temperature is closer to the temperature of the human body, the sensor will also face problem in detection. So these are the drawbacks of the PIR sensor.

This report combines these mechanisms and comes up with a new form of PIR sensor and machine learning based algorithm to identify an object and a human. This sensor will collect the signal and FFT processed frequency will be obtained which will be compared with the threshold values of the sensor. If the digital logic values goes above the threshold, then it produces HIGH which means intrusion, else LOW which means no intrusion. The threshold values vary by objects and situations and hence can detect only when there is an object.

II. GATHERING SENSOR AND CNN LAYERS

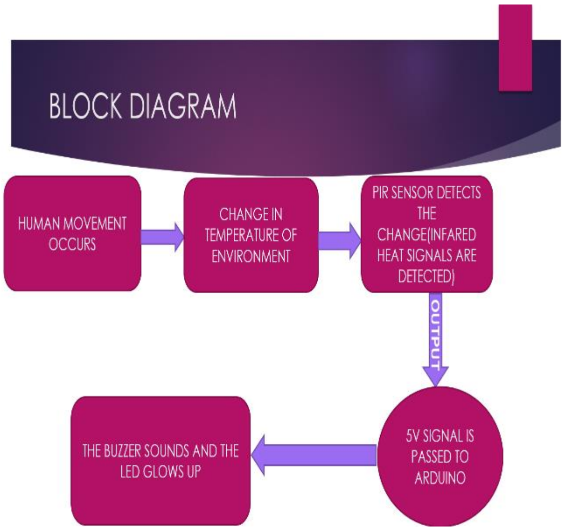
In this section, we will discuss in detail about the sensor and CNN layers.

A. Gathering Sensor Data

To collect the infrared signals in the device, we are using a Fresnel lens. The advantage of this lens is that due to its thin construction, it behaves exactly like a convex lens. The range of the sensor is divided in two parts: the detection area and the non-detection area. The detection area represents the area where humans can be detected while non-detection area represents the distance where human cannot be detected. The detection distance means the distance of the object from the detection area. But since an infrared detection system is installed, the non-detection area should also be considered.

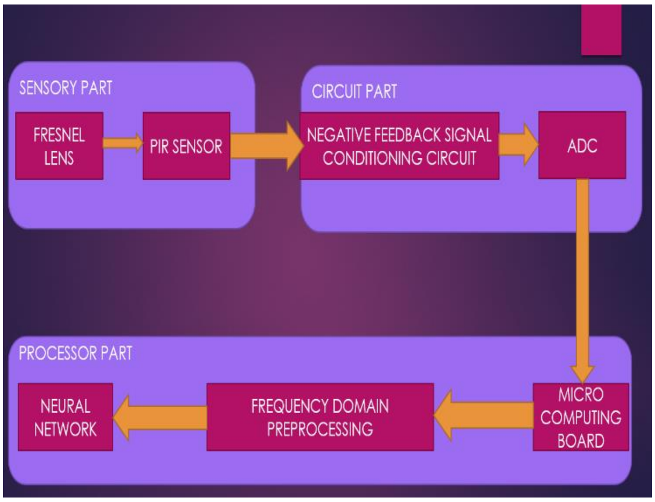
B. CNN Layers

We have used artificial CNN which is complete different from the Multi-Layer Perceptron (MLP). The big difference is the presence of the convolution layer which is present in CNN but bot in MLP. This layer is used to extract features and reduce the complexity of calculation in each neuron. A pooling layer also helps to reduce the amount of calculation and the number of parameters and controls overfitting. Each neuron of each layer is connected and the results go through the activation function and Rectified Linear Units (ReLU) is used as it has much faster calculation speed than hyper tangent or sigmoid function which is needed when training Deep Convolution Neural Network (DCNN).



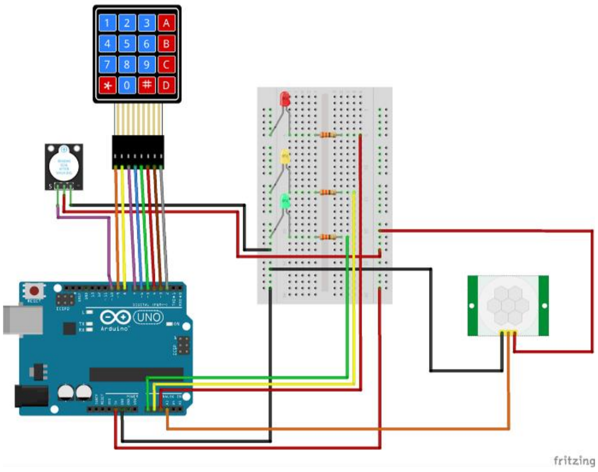
III. SUGGESTED PIR SENSOR-BASED AUTOMATION SECURITY

In this section, we will be building up the automated security system. We have divided the entire mechanism in three parts: Sensor part, Circuit part and the Processing part. Each part is described as each has their own function.



This diagram represents the entire working part of the security system. The sensor part has the role in extracting signals from an external stimulus. The circuit part amplifies this signal and converts all the analog signals to digital signals so that it can transferred to the processing part or to the processor. The processing part transforms the incoming digital signal from time domain to frequency domain. This signal becomes the newly input of the artificial CNN. We have used total five layers that consist of one input layer, one output layer and three convolution layer. Each input layer uses its own feature information and the output layer consist of two nodes needed to distinguish between human and animal.

IV. IMPLEMENTATION



The circuit part of the system amplifies the signal by 69 dB through a non- inverting two stage amplifier. It filters the high and the low frequency noise which could have interfered and led to incorrect detection and that is why the direct current components are removed. Then the input signals in the analog to digital converter (ADC) are set to the range of 0 to 3020, the reference voltage is set to 0.37 times the maximum voltage. An LHI-878 is used as a PIR sensor and a PD23-6020 is used for Fresnel lens. Arduino Uno or Raspberry pi 3 model B is used as the processor part.

The software development tool of machine learning used keras library- based deep learning studio, CPU used is Intel Core(i)i5-6600 and GPU used is Geforce GTX 1050 Ti.

V. CODING

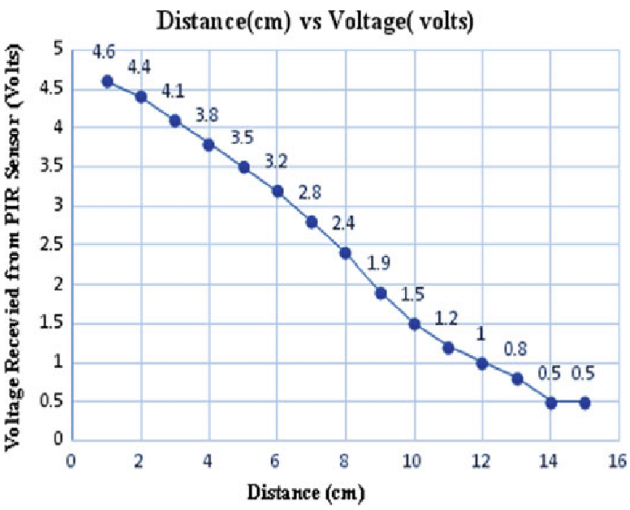
```
#include <Keypad.h>              // library for keyboard

#include <Password.h>            // library for password


Password password = Password( "1234" );  // password


const byte rows = 4;                  // four rows

const byte cols = 4;                  // three columns

char keys[rows][cols] = {                // keys on keypad

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'*','0','#','D'},

};

byte rowPins[rows] = {9,8,7,6};

byte colPins[cols] = {5,4,3,2};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);

#define sensorz A3      // pin for PIR sensor data

//#define contact 0     // pin for on/off alarm

#define alrm 12      // pin for siren, buzzer

#define redLed A2        //  pin for red led

#define greenLed A0        // pin for green led

#define yellowLed A1    // pin for blue led

int contact = 10;       //used to immediately on/off alarm

int val;

int ledBlink;

int sensorzData;

unsigned long ceas, timpmemorat;


int intarziereactivare = 20;    // To delay for standby to armed

int intarzieredezactivare = 10; // To delay for triggered to alarm activated

int timpurlat = 10;            // Time of alarm is on

// This is the variable for states "0"

char caz = 0;

int sistem = 0;      // system is 0 for off and 1 for on

/*

States for   0. - off

  1. - stand-by

  2. - waitting

  3. - countdown

  4. - alarm

  */

void setup()

  {

  keypad.addEventListener(keypadEvent); // an object is created for tracking keystrokes


  Serial.begin(9600);  //Used for troubleshooting

  pinMode(alrm, OUTPUT);

  pinMode(sensorz, INPUT);

  pinMode(contact, INPUT);

  pinMode(redLed, OUTPUT);

  pinMode(yellowLed, OUTPUT);

  pinMode(greenLed, OUTPUT);

  digitalRead(contact);

  Serial.println("System startup"); //Used for troubleshooting
```

```
  Serial.println("Alarm button status:"); //used for troubleshooting
  Serial.println(digitalRead(contact)); //used for troubleshooting
  }
void loop(){
ceas = millis();    // read the internal clock
  val = digitalRead(contact);
  keypad.getKey();
   if (sistem%2 == 0){
    // alarm is off
    digitalWrite(greenLed, LOW);
    digitalWrite(redLed, LOW);
    digitalWrite(yellowLed, HIGH);
    //Serial.println(contact); //Used for troubleshooting
     digitalWrite(alrm, LOW);
    caz = 0;
   // Serial.println("System is OFF !"); // Used for troubleshooting
}
else {
    // alarm is on
    if(caz == 0) {
     caz = 1;
     timpmemorat = ceas;
     digitalWrite(yellowLed, HIGH);
     }
   if(caz == 1)              // system waiting
      {
     if ((ceas%1000)<500) digitalWrite(greenLed, HIGH);
      else digitalWrite(greenLed, LOW);
 keypad.getKey();
      if(ceas >= timpmemorat + intarziereactivare * 1000) {caz = 2;}
      //Serial.println("System is arming !"); // Used for troubleshooting
      }
      if(caz == 2)              // system is armed
      {
      digitalWrite(greenLed, HIGH);
 keypad.getKey();
    sensorzData = digitalRead(sensorz);
      //Serial.print("sensorzdData = "); //Used for troubleshooting
      //Serial.println(sensorzData); //Used for troubleshooting

  //    if(sensorzData > 600) {caz = 3; timpmemorat = ceas;}
     if(sensorzData == HIGH){
       caz = 3;
       timpmemorat = ceas;
       digitalWrite(greenLed, LOW);
       }
      Serial.println("System is armed !"); // Used for Troubleshooting
      }
    if(caz == 3){              // system is triggered and countdown
       if ((ceas%500)<100) digitalWrite(redLed, HIGH);
      else digitalWrite(redLed, LOW);
 keypad.getKey();
      if(ceas >= timpmemorat + intarzieredezactivare * 10) {caz = 4; timpmemorat = ceas;}
      Serial.println("System is triggered and is countdown !"); //Used for troubleshooting
      }
      if(caz == 4) {             // siren (buzzer) is active
      //digitalWrite(alrm, HIGH);
      digitalWrite(redLed, HIGH);
      Serial.println("Siren is active !"); //Used for troubleshooting
// For siren
  //tone( 10, 10000, 100);  // Simple Alarm Tone
    for(double x = 0; x < 0.92; x += 0.01){  // Elegant Alarm Tone
```

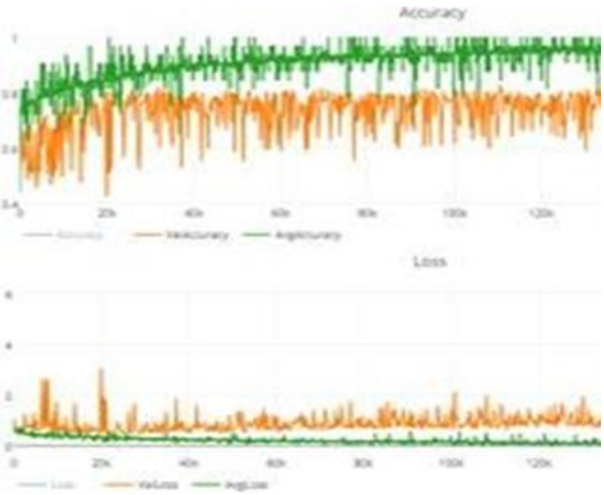
```
        tone(10, sinh(x+8.294), 10);
        delay(1);
        }
    keypad.getKey();
      if(ceas >= timpmemorat + timpurlat * 1000)
{caz = 2; digitalWrite(alrm, LOW);
digitalWrite(redLed, LOW);}
      }
    }
  }
  //take care of some special events
void keypadEvent(KeypadEvent eKey){
  switch (keypad.getState()){
    case PRESSED:
  Serial.print("Pressed: ");
  Serial.println(eKey);
  switch (eKey){
    case '*': checkPassword(); break;
    case '#': password.reset(); break;
    default: password.append(eKey);

 }
  }
}
 void checkPassword(){
  if (password.evaluate()){
    Serial.println("Success"); //Used for
troubleshooting
  sistem++;
  password.reset();
    Serial.println("Disarmed");//Add code to run if it
works
  }
else
{
    Serial.println("Wrong"); //Used for
troubleshooting
    //add code to run if it did not work
    ledBlink = 0;
    while (ledBlink <= 5){
    digitalWrite(redLed, HIGH);
    delay(100);
    digitalWrite(redLed, LOW);
    delay(100);
    ledBlink++;
    }
    password.reset();
  }
}
```

VI. EXPERIMENTATION AND EVALUATION

Many experiments has been conducted based on factors that affect the amount of ambient infrared that PIR Sensor detects. The graph represents the plot between the voltage obtained from the sensor and distance. The y-axis is varied from 0 to 5V in quantization rate of 3200. Data is collected for distance of 2m, 4m, 6m, 8m, etc. These data are then Fast Fourier Transform (FFT) processed to detect whether the object is human or animal using CNN. The accuracy and the loss rate were identified by distinguishing when there is a person and when there is none toward the training, validation and test.



Training accuracy is 97.62%, validation accuracy is 80.5% and test accuracy is 72.8%. The figure below shows the accuracy and loss rate distribution toward the training, validation and test.



VII. CONCLUSION

This report designs a PIR sensor by using CNN based learning algorithm using analog signals to improve a PIR Sensor based Security System. The signal processing algorithm is used in experiments to distinguish between humans and animals so as to improve the efficiency of the sensor.

In future, research on this topic will be conducted so that the efficiency can be increased more by directly comparing the frequency of the signal obtained with the stored data of humans and animals and this data will be used as a parameter of machine learning algorithm.

VIII. ACKNOWLEDGEMENT

This report is a group work and is done under the guidance of our respective Prof. Geethanjali P. She has helped us to deal with the errors and the complete work. We have done zeroth, first and second reviews and we were able to get many informations from her.

IX. REFERENCES

1. https://create.arduino.cc/projecthub/thehack904/motion-sensing-alarm-with-keypad-password-53f05c

2. https://www.elprocus.com/motion-detector-circuit-with-working-description-and-its-applications/#:~:text=Nowadays%2C%20most%20of%20the%20motion,in%20the%20frequencies%20they%20emit.&text=The%20main%20element%20in%20the,or%20any%20other%20detecting%20sensor.

3. https://www.arrow.com/en/research-and-events/articles/how-motion-sensors-work

4. F. Nelli, “PIR motion detector - a sensor for Arduino and Raspberry Pi”, http://www.meccanismocomplesso.org/en/pirmotion-detector/, February 2016.

5. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278-2324, November 1998.

6. V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In Proc. 27th International Conference on Machine Learning, 2010.