Semester-long project

on

"<u>POINT AND SHOOT DEVICE FOR STOPPING</u>

<u>CROSS BORDER INFILTRATION"</u>

(Sept 2019-Dec 2019)

Submitted by

<u>Nipun Verma 2017ME10596</u>

<u>Nikhil Mishra 2017ME10595</u>

Supervised by

DR SP SINGH

Department of Mechanical Engineering

Indian Institute of Technology Delhi

Abstract

As of now, our borders are protected by Spike fences, and watch-towers having soldiers continuously watching over for any infiltrations and intrusion over the border area day and night even in harsh weather conditions. The project won't completely remove the responsibility of the soldiers but manages to assist the present methodology and reduce mistakes on the border. The basic purpose of the project is to improve border security in various conditions electronically with automation and to reduce the hectic workload and ensure the safety of our soldiers. Also, the project can be used on a small scale for home or building security by simply adjusting the range according to the environment. The model is capable of working in outdoor conditions and taking on realistic situations because of a stable platform for mounting the components and an improved image recognition algorithm for fool-proof detections. A combination of PIR sensors and camera are used for detection along with some algorithms, servo motors have been used for changing altitude and azimuth angles of the gun, and Arduino is used for controlling these automotive actions.

Introduction

The point and shoot device which is a type of autonomous shooting system is a camera-based weapon system that uses software to detect and track a target and hardware to point to the target and fire at it. It consists of a combination of hardware and software to point a mounted gun's aim to the target detected and tracked in the camera's view. The gun will have the ability to scan its field of view using a camera and fire automatically at moving targets. The field will be scanned and each frame will be processed by using OpenCV libraries for object detection, motion tracking and colour detection of the detected object. The user interface will show a continuous video stream for the camera. The system will recognize and detect and track a moving object, find its contours and centroid and fire the object until the object is removed. The hardware will contain two servo motors for aiming the gun along the horizontal and vertical axis and another servo motor for trigger control. High definition cameras will be used in conjecture with OpenCV to detect and track targets. When an object enters the field of view of the camera, the system detects it and begins tracking the object. The computer continuously updates the object coordinates and passes this information to the microcontroller which translates the coordinates to pulse widths for the servo motors to move the Automatic target detection and shooting gun to detect and target any living object or any movement within the sensor range. Automation depends on the PIR sensor, camera sensor and Arduino controller and processor. Arduino satisfies all requirements of automation. Arduino gives commands and instructions to control sensors and guns. PIR sensor detects the motion of the objects. If the object tries to cross the boundary, then PIR generates a signal and gives it to Arduino and it produces an alert message. If the object continues to go further then Arduino gives the command to shoot that object. Automatic shooting is done by mounting a gun on a stable platform and giving it the required degrees of freedom by rotating the device and adjusting the height. The mounting device is calibrated initially so that the bullet is fired at the target accurately. The recoil effect and consequential instability caused by it is also taken into account during the calibration. This device can play an important role in the nation's security and defence by detecting various infiltrators or attacks and will act as the first line of defence more efficiently than having humans and while also saving lives of the soldiers and the citizens of the country by preventing cross border terrorism.

Literature Survey

Autonomous shooting system projects have been previously created and implemented in the past. The system as a whole is not a new technology. Various aspects of the system have been done previously for a variety of reasons, such as motion tracking and a gun based system that executes incoming targets. These were the original and unique technologies that when integrated will help in building our autonomous shooting system. The influence of these existing technologies has widened the range for the growth of different prototypes for future solutions. The previous projects failures and successes are determined by researching them for the design of this project which will help to improve our system. By researching all the similar existing technologies implemented, successful image-Based systems can also be further optimized depending on their use but initially, they can be used for local security by businesses or homeowners. It can even be used in battlefields in order to protect a
military base from incoming enemies and can also have the ability to successfully detect and intercept incoming planes, helicopters, and missiles. The system's concept does have several uses and one can use it according to his needs.

1) **Object detection** : For detecting objects the Histogram of Oriented Gradients (HOG) descriptor can be incorporated. This method suggested by Dalal and Triggs in their seminal 2005 paper, Histogram of Oriented Gradients for Human Detection demonstrated that the HOG image descriptor and a Linear Support Vector Machine (SVM) could be used to train highly accurate object classifiers — or in their particular study, human detectors. Thus this descriptor can be used easily to detect human objects in the video stream.

2) **Motion tracking** : Once the objects within the image have been detected, the next logical step is to track those objects. This algorithm uses the most popular method of tracking by optical flow. The algorithm is known as the Lucas-Kanade algorithm, which is a differential method. In this method, we give some points to track and we receive the optical flow vectors of those points. So applying Lucas-Kanade there, we can get optical flow of the detected object.

3) **Color detection** : by detecting various colors, the system can differentiate open allied targets and enemy targets. For detecting colors we use HSV instead of RGB. The primary benefit of a pixel represented by HSV is that it would be relatively easy to detect the various shades of a single color. This method gives you a nice binary image that displays the specified colour as white with all other objects and colors on the screen as black. By specifying a certain range of colour to represent an allied target, the system will be able to calculate the average color of the target and determine if it is a threat or not.

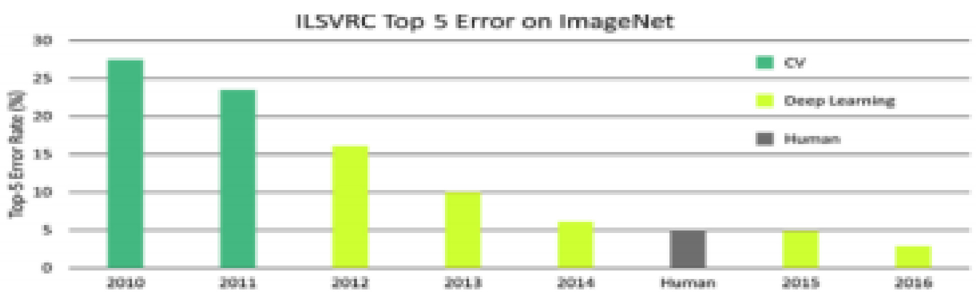
4) **Turret control** : The turret control will be the main class of the software. It will control the firing, target selection and determine the angles needed for the servo motors. The fire function will take the given target and attack that target.

Weapon Systems

The housing design will allow for the sentry turret to utilize a multitude of platforms for target tracking. Anything weighing less can be mounted without sacrificing performance. Therefore, we prefer a non-projectile platform, a laser, flashlight, or something that can be modified to fit. For this project, it was clear that either a paintball gun or airsoft gun would be implemented into the design because of their availability, ease of use and target-hit-verification. However, these two weapon options have some differentiating factors that led to making a decision (further discussed in the work progress section).

NEW DETECTION ALGORITHM:

Anyone familiar with Deep Learning would know that image classifiers have surpassed human level accuracy.



Error rate on the ImageNet dataset over time, for Humans, Traditional Computer Vision (CV) and Deep Learning. So yes, a machine can keep a lookout for objects at the same standard (or better) when compared to a human. With that being said, using technology to perform surveillance is much more efficient. So looking for such algorithms we came across the RCNN (convolutional neural network) which is capable of detecting humans and is fast enough to be incorporated in live feed. Later, we will discuss how we got the basic code structure and the other additions done by us.

CALIBRATION ALGORITHM

While searching for a possible solution to relate the person's pixel and actual location, we came across a research paper. It claims that there exists a function that can do the job. The form of the function is a quadratic polynomial. So we planned to create a quadratic polynomial one for our project as well. In the paper they exploited this relationship to train a system that finds a mapping between an object's pixel height and physical distance. This mapping is then used to find the physical distance of test objects from the pixel height in the image. Experimental results in their study demonstrated the capability of their proposed technique by estimating physical distance with accuracy as high as 98.76%.

Project Objectives and Work Plan

Problem definition/Motivation our country is the one with one of the most volatile borders and many crucial areas which are vulnerable to attack, stealth penetration and points of illegal entry which are some of the leading factors of terrorism in India. It requires a lot of effort, resources and people to maintain a tight surveillance system around all these areas. The concertina fencing technique and other main ultramodern defensive pieces of equipment used as of now are very costly and it needs constant surveillance of the soldiers and effective maintenance and these are being replaced by the UAV at some areas, though they are effective, their cost is very high and needs frequent maintenance and fueling. So these factors lead to the cause for developing an ultimate machine that has a minimum cost and a maximum efficiency which would be able to work on any kind of rough terrains and adverse climatic conditions and defend the borders without any human intervention.

Since this is just a surveillance system, an automatic detection system using various types of sensors can be implemented to detect any sort of abnormality or change in that area and such incidents can be reported to take suitable actions like locating/following the cause of the incident, sending a surveillance team or in cases where it is sure that there is a presence of an enemy, an automatic shooting device must locate the enemy and fire on command. Its a very crucial and sensitive activity, Both the parts - detection and shooting, of the device must work efficiently and in complete coordination with each other. Various challenges in both the activities like accurate detection, coordinate detection, accurate shooting etc. must be overcome to guarantee the smooth working of the device.

b. Objectives of the work

As stated in the mid-term evaluation, we had three major objective heads to complete:

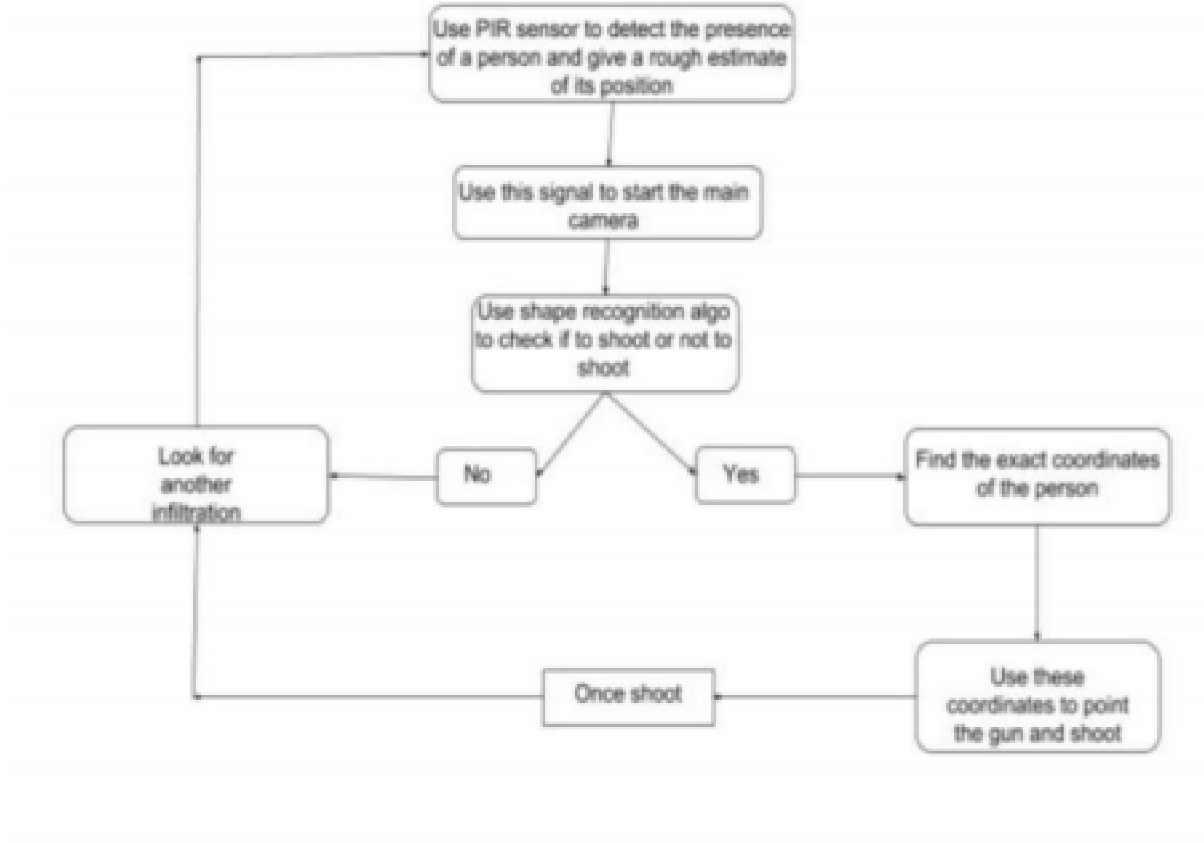
a. Building a stable gun turret mechanism

b. Calibration of the detection algorithm

c. Refining of shooting algorithm

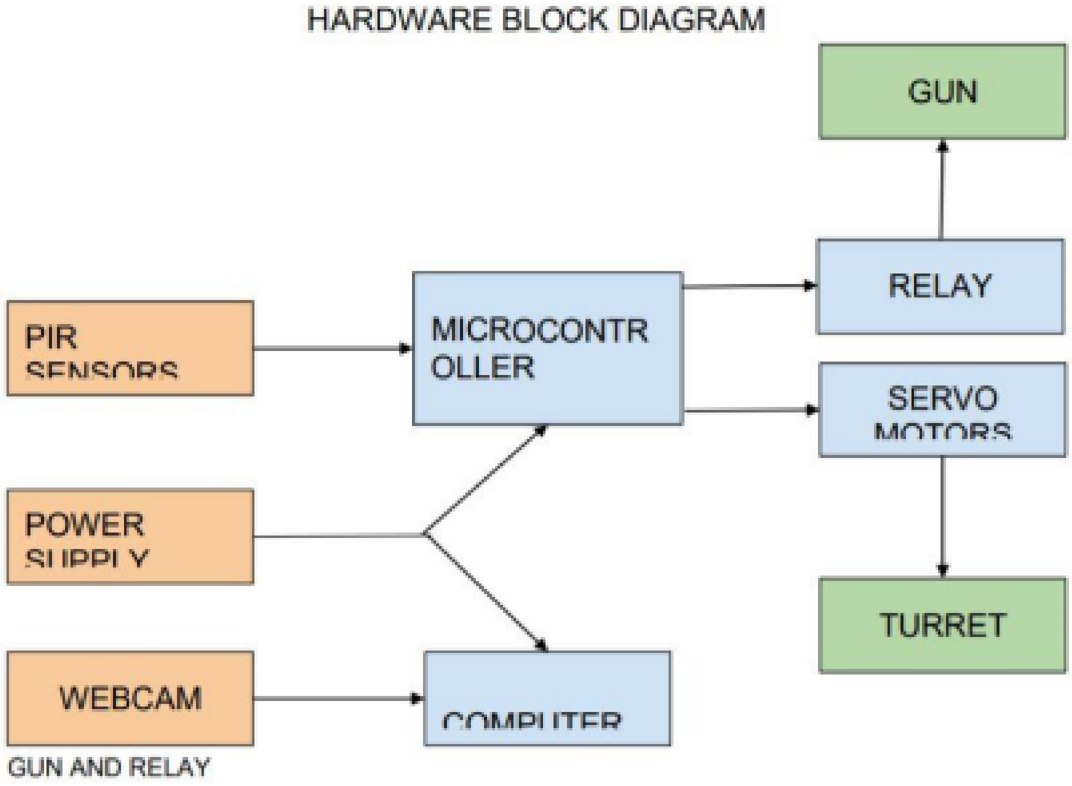
To construct a prototype to detect and eliminate any infiltration attempts that could be handle outdoor conditions and further extended to be deployed at the borders and remotely controlled from a safe distance. The technology had to be functional irrespective of the terrains, weather conditions and lighting conditions. It was also supposed to have foolproof detections and support our existing defence capabilities. The aim was to develop an automatic target identification mechanism using Infrared sensors and high-definition cameras and eliminate it using signal-controlled firing equipment.

c. Methodology:-



Work progress

Starting with all the steps done for building a turret mechanism, we will show how the detection algorithm was formed and calibrated. The journey for the development of gun turrets and algorithm went hand in hand, as they were not related to each other to a great extent.



HARDWARE COMPONENTS

ARDUINO The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analogue inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains components needed to support the microcontroller. It can be connected to a computer with a USB cable or power with an AC to-DC adapter or battery to get started. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language.

Reasons for choosing Arduino over other platforms:

i. Low Cost: The plus point of the Arduino board as compared to other microcontrollers is that it has relatively less cost.
ii. Cross-platform: Arduino can work on Linux, Windows, and Macintosh, while others are only for Windows.
iii. Simple and clear programming environment: Arduino programming is more flexible and easy, even beginners can benefit from it.
iv. Extensible and open source software: Arduino software was published as an open-source tool and is available for extension. All types of source codes and libraries of the Arduino environment are available openly, with which anyone can get used to Arduino.
v. Extensible hardware: The Arduino board is based on Atmel's ATMEGA8, ATMEGA328P PU and ATMEGA168 microcontrollers. It can be connected with any type of sensor, and circuits

CAMERA

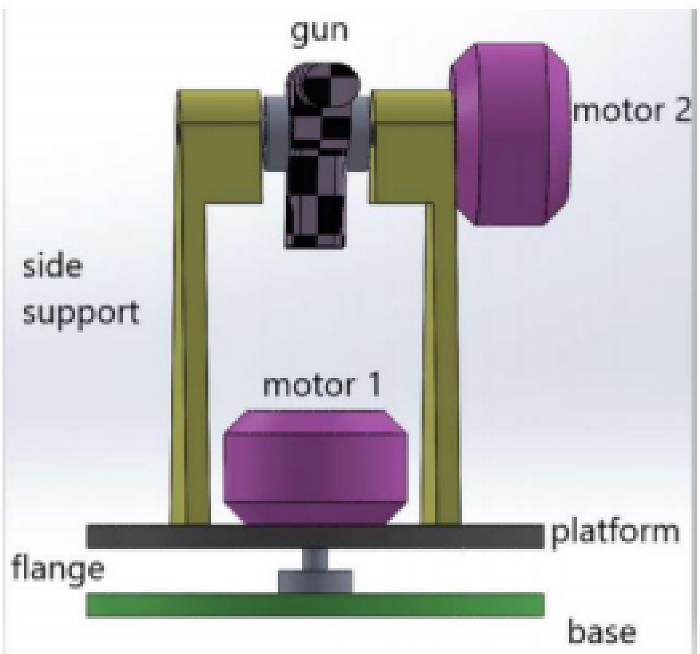
The camera will be a vital component in the system design. The sole purpose of the web camera is to provide optics for the system, from where the software will be implemented so the system has object detection. For these implementations to occur, a high definition camera will need to be used. The camera will be attached to a servo motor to point in the direction of motion detected by the PIR sensor. The camera will also be connected to the laptop via USB and have a real-time video capturing that will display on the laptop.

PIR SENSORS

The PIR sensor is the core part of the system. The system functions based on infrared radiation, which is emitted from the human body. PIR sensor is widely used in a security system to detect the motion of humans. Infrared (IR) light is electromagnetic radiation with a wavelength between 0.7 and 300 micrometres. Human beings are the source of infrared radiation. It was found that the normal human body temperature radiates IR at wavelengths of 10 micrometres to 12 micrometres. Fig. 2 Infrared Radiation of human body PIR sensors are passive electronic devices that detect motion by sensing infrared fluctuations. It has three pins (gate, drain and source). After it has detected IR radiation difference, a high is sent to the signal pin.PIR sensor is made up of crystalline material that generates a surface electric charge when exposed to heat in the form of IR. This change in radiation striking the crystalline surface gives a change in charge. The sensor elements are sensitive to radiation of a wide range but due to the use of a filter window that limits the sensitivity to the range 8 to 14 micrometre which is most suitable to human body radiation.

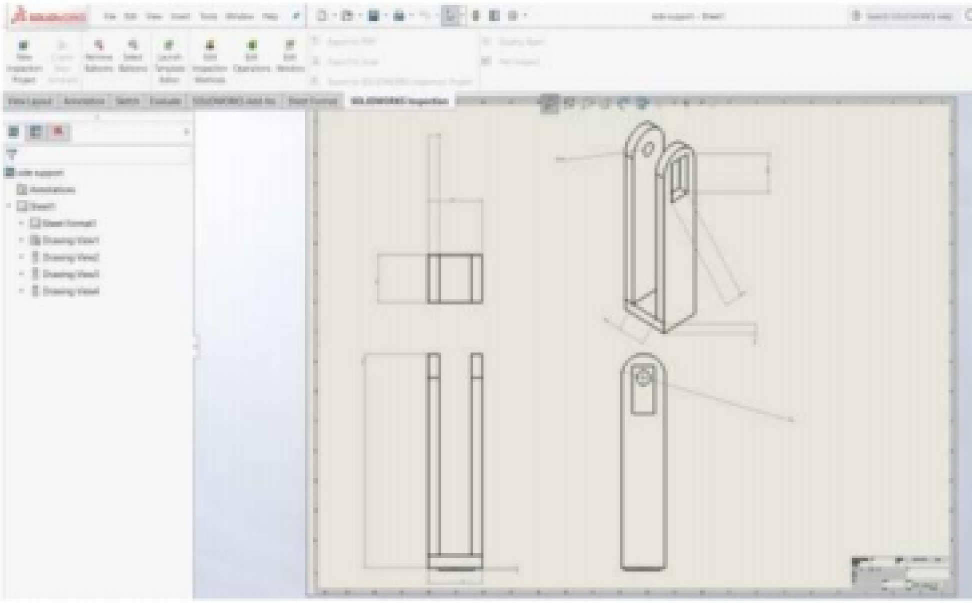
GUN TURRET

1) VISUALISATION OF MODEL ON SOLIDWORKS



2) FABRICATIONS OF TURRET

The material used: Plywood 0.5 cm thick for the platform and side 0.5 INCHES thick for the base. A base was formed using a box and required holes were created for mounting the motor and fitting the wires. A new thought was given to construct the platform along with the side supports and directly mount them on the motor. The solid-works diagram of the platform part is shown in the fig below. The measurements are given below. Height of motor mount 20cm Total height of the side support 25 cm Dimensions of the integrated base 5cm*7.5cm



SERVO MOTORS

SERVO MOTOR VS STEPPER MOTOR:

Servomotor consumes power only when it rotates to its commanded position, while the rest of the time it doesn't consume any power to maintain its commanded position. They are generally replacing Stepper motors due to their high performance. And they contain an opener and controller which is the reason for their high cost, but they improve the performance of the overall system (for all of accuracy, power and speed) relative to the capacity of the basic motor. Stepper motors continuously consume power to hold the commanded position or to move to the commanded position and due to continuous consumption of power, it becomes warm. They have built-in output steps. It can only work properly with a load that is well within its capacity, otherwise, overload may lead to missed steps and as a result we would get positioning errors.

SERVO CONTROL METHOD:

Servo control method Most standard servos have three leads, positive power, negative, and signal. The power lead not only acts as the power source for the servo but can also be utilized to turn the servo either on or off. The typical input voltage for power is between 4.8 volts and 6.0 volts. The negative power lead should be common ground. The signal lead will control the direction of the servo. The primary method of controlling the servo is to send a pulse-width modulation along with the signal lead. This pulse-width modulation is a square width. The length of each pulse of the square wave controls how far the servo will rotate.

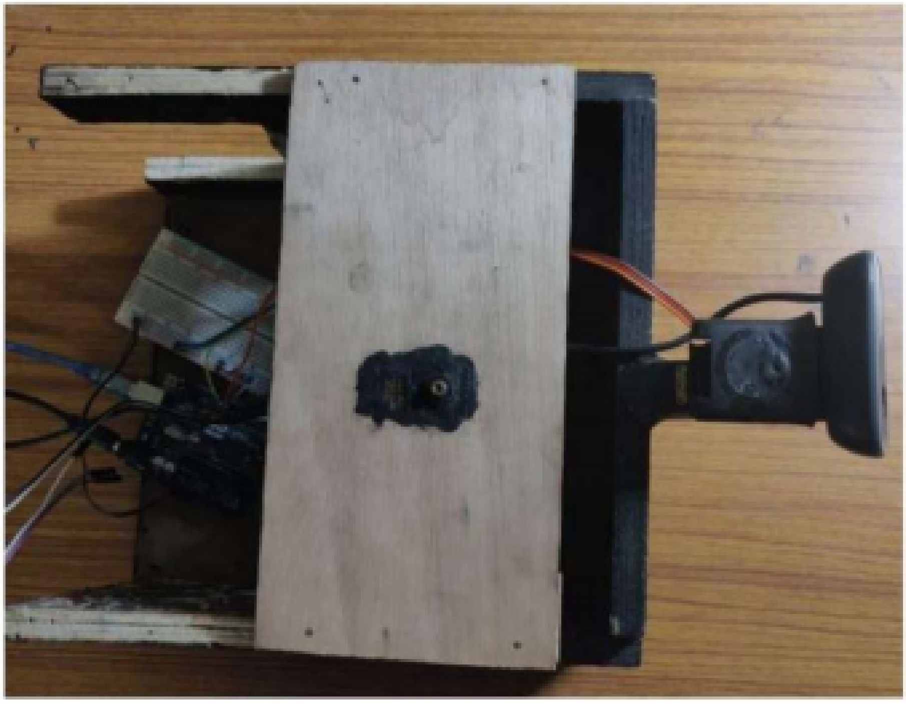
For example, a pulse of 600 microseconds will rotate the servo arm -90 degrees and a 2400 microsecond pulse will rotate the arm positive 90 degrees.

3) FINAL ASSEMBLY OF THE TURRET MECHANISM:

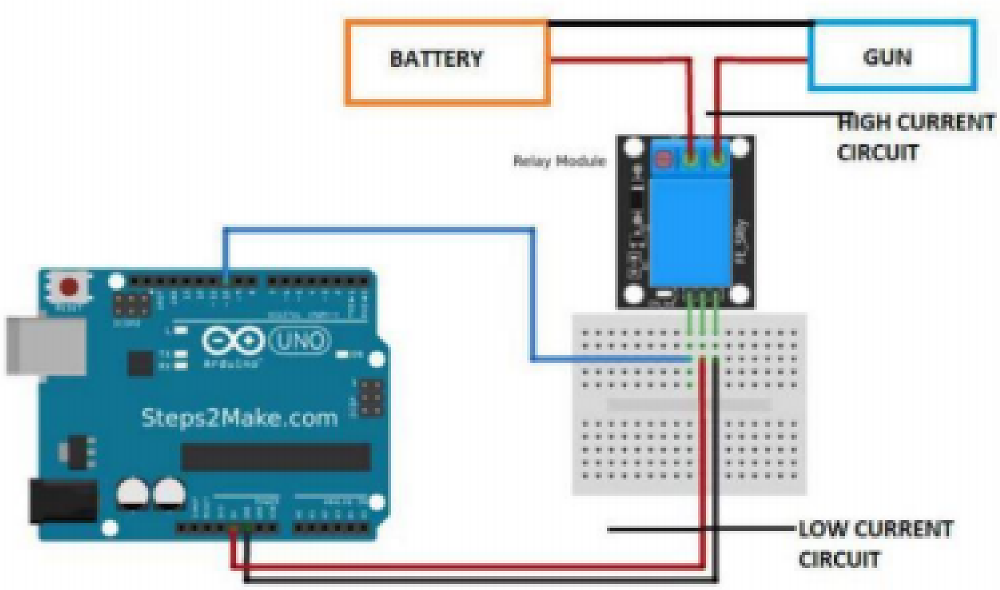
 

There were several improvements made along the way. Some crucial changes in the prototype are mentioned below: The motor of the base had a play and was not stable during the operations. It used to shake a lot while starting and stopping and while shooting the targets. Also, there were no proper passages for connecting wires and mount for the camera and its motor. So a bearing was fitted with the platform and was connected with the shaft of the motor. The special function of this bearing is that it supported the platform away from the base and provided stability to it while rotating and shooting and also it reduced the amount of load on the shaft of the motor thus increasing the accuracy of the turret. Also, a proper mount for the camera was created and wire passages were made for easy functioning.

TOP VIEW OF THE BASE:



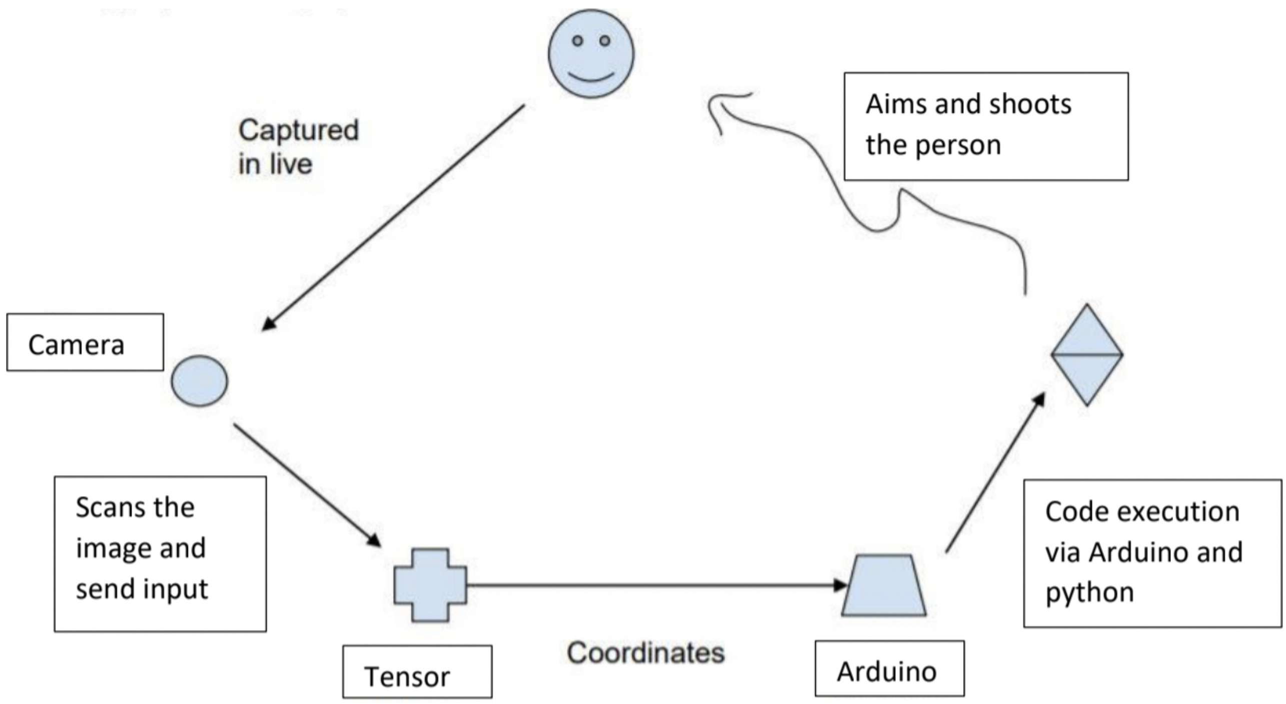
RELAY CIRCUIT FOR GUN



ARDUINO CONNECTIONS WITH MOTORS AND PIRS

SOFTWARE

Detection Algorithm: In a bigger picture the project looks something like this.



The above cycle shows how our overall algorithm works. After a positive signal by PIR's this loop comes into action. This detects the exact coordinates of any person within the reach of the sight of the camera. After successful detection of a person, Arduino gives a final call to aim and keep shooting at the person.

Let's talk about each component:

Arduino: This is responsible for PIR's integration, operation of hardware involved for shooting. Below is a snapshot of the code uploaded to the Arduino board.

Final_BTP_2

```
#include <Servo.h>              //Servo library

Servo servo_1;         //initialize a servo object for the connected servo
Servo servo_2;

int angle = 0;        // for motor to rotate
int b = 1;            //for gun motor
int a1 = 60;          // for camera motor
int a2 = 120;
int a0 = 90;
int ledpin = 9;       // laser pin
int fire = 11;        // pin for shooting
int pir_1 = LOW;      // defining pir states initially to zero
int pir_2 = LOW;
int pir_1in = 12;     // setting pin to as PIR's input
int pir_2in = 13;
int val1 = 0;         // variable to store PIR's state
int val2 = 0;

void setup() {


  servo_2.attach(8);       // attach the signal pin of gun servo to pin9 of arduino
  servo_1.attach(7);       // for camera motor
  Serial.begin(9600);        // defining the band rarte for serial port
  Serial.write(6);           // just to start the communication

  pinMode(ledpin, OUTPUT);    // defining pins as output
  pinMode(fire,OUTPUT);

  pinMode(pir_1in,INPUT);     // defining the pins as input sensors
```

Python:

The python code used in the project was taken from a GitHub account with proper copyrights. It was clearly stated that we can use this for educational purposes. But getting the code in basic structure was not at all enough to incorporate it in our project. We built upon the code so that we could extract the exact coordinate of the object if and only if it is a human with at least 70% probability. So finding the coordinates of the detected box, creating a function to convert 2D pixel coordinate to 3D spatial coordinate were our add on to the python program. But we are completely aware of how that code works and how one is built. This code is a tensor-flow based ML model. Below is a simple flow chart showing how such models are trained.

This process requires a good amount of training data both in terms of quality and quantity. So it's quite a labour-some work. Talking about the capabilities of the kind of distinction it has, it can detect 90 classes of objects ranging from humans to dogs to cars to airplanes to many more. Once the person is detected its pixel coordinates of the centroid are converted to the original coordinates of the person via a proper calibrated function. This function is a relation between pixel coordinates and the actual coordinates of the person. Snapshot of the .py file:

```python
import os
import cv2
import time
import argparse
import numpy as np
import subprocess as sp
import json
import tensorflow as tf
import serial

from queue import Queue
from threading import Thread
from utils.app_utils import FPS, HLSVideoStream, WebcamVideoStream, draw_boxes_and_labels
from object_detection.utils import label_map_util

(width , height, ll) = (0,0,0)
zz = 0
zz1 = 11
nex = str('a')
banex = bytes(nex, encoding="ascii")


ser = serial.Serial("/dev/tty.usbmodem14101",9600)
print("Trying to connect")
connected = False
while not connected:
    serin = ser.read()
    connected = True
# making sure that it is connected and changing the value of connected to true
print(serin)
print("Connected with Arduino")
#l = ser.read()
#print(l)

CWD_PATH = os.getcwd()
```

In this part of the code, we imported all the libraries needed. After that other functions were called to examine the live feed coming from the camera.

```python
#print(rec_points)
#time.sleep(2)  # delay for 5 seconds
# now we will try to print the centre of the rectangle
#print(rec_points[0]["xmin"])

if len(rec_points) != 0:

    X = (0.5*(rec_points[0]["xmax"] + rec_points[0]["xmin"]))
    Y = (0.5*(rec_points[0]["ymax"] + rec_points[0]["ymin"]))
    # angle to be sent to arduino
    X1 = 1 - X
    #zz = int(X1*10)
    zz = int(X1*60)
    #zz = i
#k2 = int(ser.read())
#print("hello")
#print(zz)
#print(tt)
z = str(zz)
ba = bytes(z, encoding="ascii")
#print((X,Y))
#print((X*width,Y*height))
#k2 = 4
"""
if (time.time()-tt) > 2:
    ser.write(ba)
    print(ba)
    tt = time.time()
"""
if zz != zz1:
    ser.write(ba)
    print(ba)
    ser.write(banex)
    #time.sleep(0.1)

zz1 = zz
"""
```

This section of the code extracts the coordinates of the rectangle and after converting it to appropriate dimension and scale it is sent as a signal to Arduino via a serial port.

```python
            i = 30
        """
            #time.sleep(5)
        #if (class_names == [['person: 97%']]):
            #print('LOLOLOL')
        #print(k2)
        #k2 = int(ser.read())
        #print(ba)
        #print(k2)
        class_colors = data['class_colors']
        for point, name, color in zip(rec_points, class_names, class_colors):
            cv2.rectangle(frame, (int(point['xmin'] * args.width), int(point['ymin'] * args.height)),
                          (int(point['xmax'] * args.width), int(point['ymax'] * args.height)), color, 3)
            cv2.rectangle(frame, (int(point['xmin'] * args.width), int(point['ymin'] * args.height)),
                          (int(point['xmin'] * args.width) + len(name[0]) * 6,
                           int(point['ymin'] * args.height) - 10), color, -1, cv2.LINE_AA)
            cv2.putText(frame, name[0], (int(point['xmin'] * args.width), int(point['ymin'] * args.height)), font,
                        0.3, (0, 0, 0), 1)
        if args.stream_out:
            print('Streaming elsewhere!')
        else:
            cv2.imshow('Video', frame)

    fps.update()

    print('[INFO] elapsed time: {:.2f}'.format(time.time() - t))
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

fps.stop()
print('[INFO] elapsed time (total): {:.2f}'.format(fps.elapsed()))
print('[INFO] approx. FPS: {:.2f}'.format(fps.fps()))

video_capture.stop()
cv2.destroyAllWindows()
```

This is the very end of the code that is just the output to be shown on screen. It makes Arduino a rectangular box around the person, shows the type & % of probability score and updates the frame on our screen.

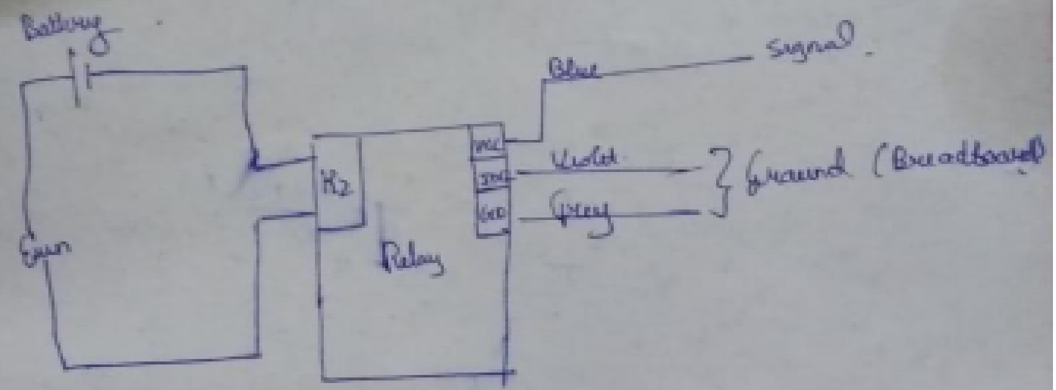
Serial Port:

This is the final link between the python processing power and Arduino control. Using a Serial port we were able to make a successful communication bridge between the two. It is capable of parsing an integer value of how many degrees to rotate is parsed using Serial. present command. So correct connection helped to use the power of both the worlds. Just show the future scope of addition.

Connections:

The following images show all the connections of the device, colour-coded with respect to the wire colour -

Fan + Relay



Arduino
- Pin-2 Orange PIR-1
- Pin-3 Blue PIR-2
- Pin-8 Orange Motor-1
- Pin-9 — Motor-2
- GND Green GND-(Breadboard)
- 5V black (+ve) (Breadboard)

Motor-2
Vertical

Brown	Red	Orange
↓	↓	↓
Violet	Grey	White
↓	↓	↓
Ground	5V (Breadboard)	Pin-9 (Arduino)

IR-Sensor-1	VCC	Out	Ground
	Red.	Orange.	Green.
	Red.	Orange	Black
	Red	Orange	Yellow
	5V		Ground.
	Breaboard (+)	Arduino (2,[illegible])	Breadboard (-ve)

Laser-Pointer	White	Black	{White Violet Grey}
	5V · Breadboard	Ground (Breadboard)	free

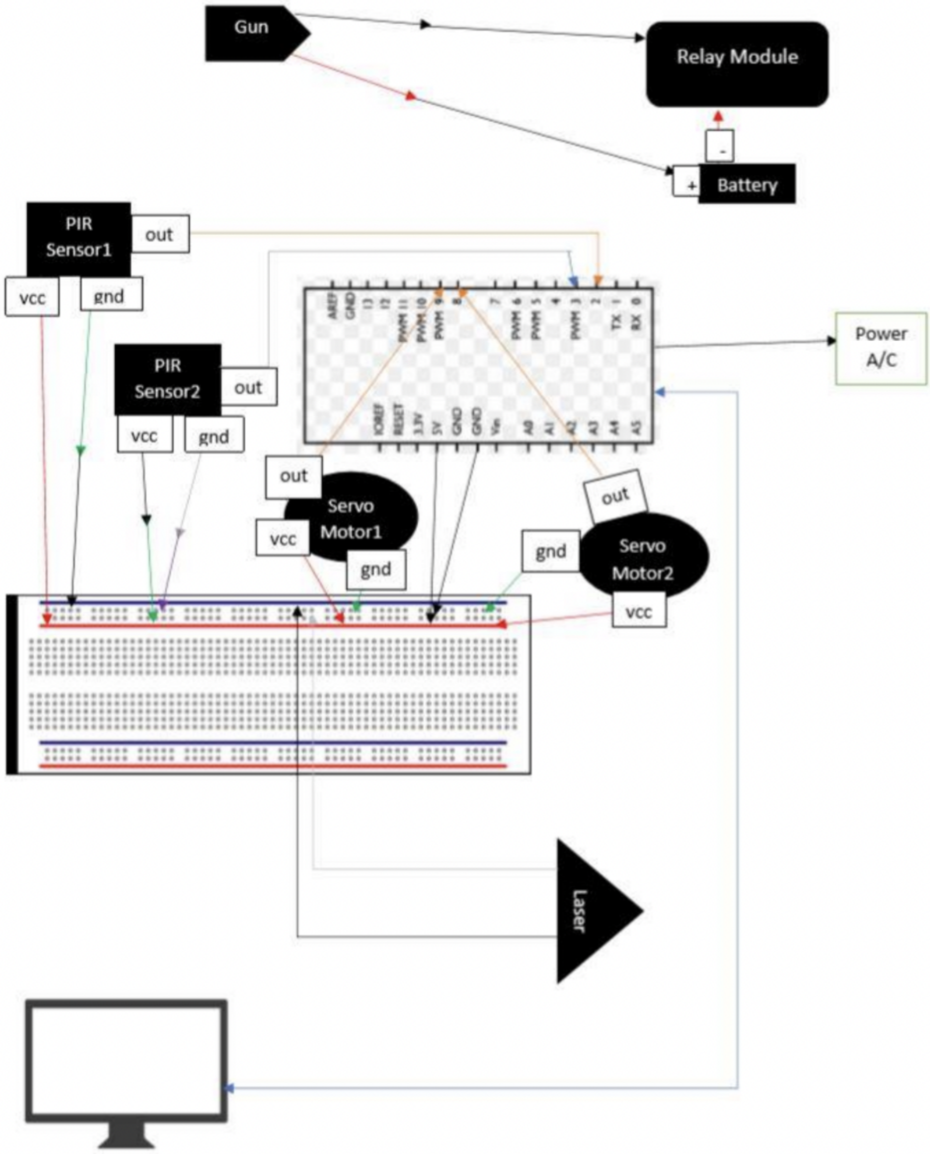
Motors	Orange	Red	Green
	Arduino (8,9)	5V	Ground

IR-2	Black	White	Grey
	Green	Blue	Violet
	Green.	Blue	Violet
	5V · Breadboard.	Arduino (3).	Ground (Breadboard)

NOTE: These colours represent the colours of the corresponding wires in the device. Most of the wires are multiple connections i.e., have multiple wires connecting 2 ports in the device, and these are represented accordingly.

Project Components and their connections

The following image shows the project components (all of them) and their respective connections, alongside mentioning the ports to where the wires are supposed to be connected



NOTE- All the arrows shown are The following colour-coded the connections, representing the corresponding actual wires of the device

Conclusion and Future work

This paper has discussed the design of an autonomous shooting system. The system will focus on finding a target in its vision and then firing at it. This system will make use of the OpenCV libraries for finding and tracking a target and a mounted gun controlled by servos for firing. We have been able to improve the previous prototype and addressed a lot of drawbacks from it. These are the highlights and objectives that we could achieve:

1) Were able to construct a turret and worked out for its stability during the operation in outdoor fields. Also mounted it with an electric-powered gun and relay for instant fires and no delay
2) Incorporated a new detection algorithm that detects a person and is not sensitive to other movements in the frame. Along with this real-time transfer of angle to the servo was done to instantly point at the target.
3) The shooting was calibrated to a great extent and for most of the spaces in front of the camera but there were still some misfires and this area has more scope for improvements. The performance of the proposed system may be further improved by automatic guns, night vision cameras and faster servos. As for the future, this project can be improved by conducting experiments with other object detection and motion tracking approaches available for faster image processing. This would further improve the accuracy and speed of the turret. Also, the construction of the model could be varied easily for different purposes.
4) There's a small issue in-camera detection that the camera detects non-living things as human bodies too, e.g., a plastic body cooler, a stool were recognised as human bodies.
5) The code is only supported in Apple MacBook, not in windows. This problem is further left to be solved. It requires great knowledge of software development and programming. So this problem still remains.

To check out the sample working of the device, follow this link:

Point and Shoot Video.mp4

References

[1] Navneet Dalal and Bill Triggs, “Histograms of Oriented Gradients for Human Detection.”

https://lear.inrialpes.fr/people/triggs/pubs/Dalalcvpr05.pdf

[2] Bruce D. Lucas and Takeo Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision.”

http://wwwcse.ucsd.edu/classes/sp02/cse252/lucaskanade81.pdf

[3] The Sentry Project. Web. 03 Dec. 2011. http://www.paintballsentry.com

[4] Autonomous Targeting System http://www.eecs.ucf.edu/seniordesign/fa2011sp2012/ g12

[5] Self-Targeting Automated Turret System

http://www.eecs.ucf.edu/seniordesign/sp2014su2014/ g08/home.html

[6] An Image-Based the colour-coded following Approach to Compute Object Distance Ashfaqur Rahman* Department of Computer Science, American International University Bangladesh

https://pdfs.semanticscholar.org/7f49/68b9e7188638bf5f4524db3a3a9d5422899f.pdf