A Real Time Object Detection Security System With Automated Response For Closed Spaces

Team SkyEye

Chenjie(Lavi) Zhao (lavizhao@bu.edu), Yaying Zheng (yyzheng@bu.edu),
Junwei Zhou (junwei23@bu.edu)

Abstract

This project aims to design and build a security system for enclosed spaces using real time image processing and object recognition algorithms. Traditional security systems like CCTV only record and capture surveillance videos. They do not possess the ability to issue warnings if there are suspicious objects within the frame. The proposed project implemented using Gumstix have resolved this issue. The design consists of four components: gun turret, webcam, gumstix microprocessor and computer. The webcam is connected to the laptop through USB. Images are captured and analyzed by the computer using opencv for image processing and Tensorflow SSD+mobilenet network for object detection. Once an object (e.g. human) is detected, instructions are sent back to the Gumstix using bluetooth. At this point, the Gumstix will provide GPIO outputs to the motors to move the gun and shoot the target. The evaluation of the project showed that it takes an average of 1 second to detect that a target is within the frame and shoot.

Introduction

Security within closed spaces, like buildings and homes, has been a recurring concern. The traditional approach for this issue is a closed circuit TV (CCTV). CCTV is a surveillance system that uses video cameras to transmit video feeds to a monitor. However, these particular systems lack the ability to alert the user if a suspicious individual is detected. As such, even if an intruder enters the home, the owner will not be alerted unless he is monitoring the screen. An alternative to CCTV is a security system implemented based on IoT concepts. These security systems have been designed using sensors (e.g. passive infrared (PIR) motion sensors) and equipped with feedback mechanism to detect suspicious activities and send alert signal to user.

Security system using sensors limits the detection range to only moving life forms and therefore is almost exclusively used for home intrusions. As such, in cases where a user needs to detect potentially dangerous or banned objects, these kind of systems will not be able to provide the necessary functionality. For example, cell phones are strictly banned in some music studios and movie sets to prevent premature leak of production. Furthermore, such security systems still require human intervention as the user has to decide what action to take in response to each alert. A security system with a passive response like this can be ineffective in preventing damages if the user is unable to react to the alert. For example, in home burglary situations, if the owners are absent when the alert is triggered, they might not be able response in time to prevent damage to their personal possessions. Even if they immediately notify the local police, it would still take time for responders to arrive and in this time frame, the suspect may have already escaped. No matter how fast the local police reacts, it can hardly be considered a real time response.

To improve these potential problems, a more versatile real time security system with an active response is needed. The system should be able detect suspicious objects including humans in real time and react automatically to minimize potential damages without user intervention. The user should be able to easily modify the system to detect their desired targets. With these specifications in mind, the objective of this project is to design a real time security surveillance system using object detection algorithms. This design will be equipped with its own automated defense system to immediately neutralize suspicious targets.

This project has several components including bluetooth, webcam, Gumstix microprocessor, computer, hardware and hardware control. The motorized nerf gun is mounted on a wooden turret with the bottom plate and middle rotating piece attached to step motors for horizontal and vertical movements respectively. The motors and nerf gun trigger are controlled by the Gumstix using GPIO outputs. The webcam is mounted on the nerf gun and connected to a computer. The computer runs the object detection program and sends instructions for motor movements and trigger actions to the Gumstix through one way bluetooth connection.

Bluetooth connection was successfully established from the computer to the gumstix and data is sent with minimal delay every second. Bluetooth communication from Gumstix to computer is able to send 400x400 jpeg images with an approximate one second delay, but was not integrated into the final project design because Gumstix failed to capture images from the webcam. The object detection algorithm is implemented to detect human or other common objects of choice in real time with minimal to no delay. Using one way bluetooth communication from the computer to the Gumstix, the system is able to use object detection to identify desired target and shoot with an approximately delay of one second.

Design Flow

The proposed design is implemented using Gumstix Verdex Pro PXA270. Other components used in the design include a webcam (Logitech QuickCam Orbit Webcam Model V-UU22), stepper motors, motor driver, relay module, and nerf gun. All the said components are connected and mounted on a wooden frame. An initial projection of the security system is shown in the sketch in Figure 1. In the initial design implementation, the webcam is connected to the Gumstix by USB. Two way bluetooth connection is established between the Gumstix and user computer using socket programming with RFCOMM protocol in order to send images and instructions. Using the Linux USB Class Video driver (UVC driver) and Video4Linux2 (V4L2) API on the Gumstix, video images are captured and sent to the computer running the object detection algorithm in the background. Using opencv in Python3, images from webcam is converted for input to the Tensorflow SSD+mobilenet network algorithm for object detection. Once a target is identified, an instruction string is sent by the computer to the Gumstix, which then converts the string into commands for the motors and gun trigger using GPIO outputs. However, during the design process, the provided drivers was discovered to be incompatible with the specific webcam model chosen for the project. As such, the design was modified to provide only one way bluetooth communication between the Gumstix and user computer. In the final system implementation, the webcam is connected to the computer rather than the Gumstix. Video images from the webcam are be captured and converted using opencv running on the computer. The process following target identification remained unchanged. An image of the initial and final design flow is shown in Figure 2 and 3 below for comparison.

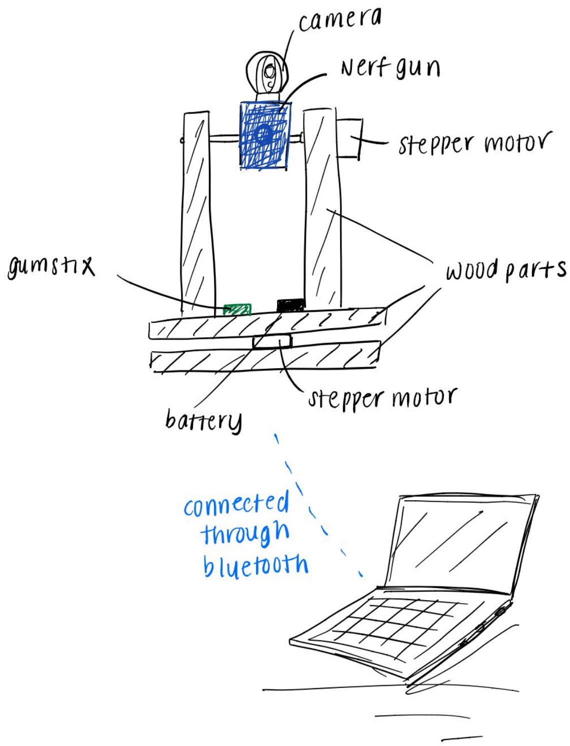

***Figure 1**: Initial Design for Security System*

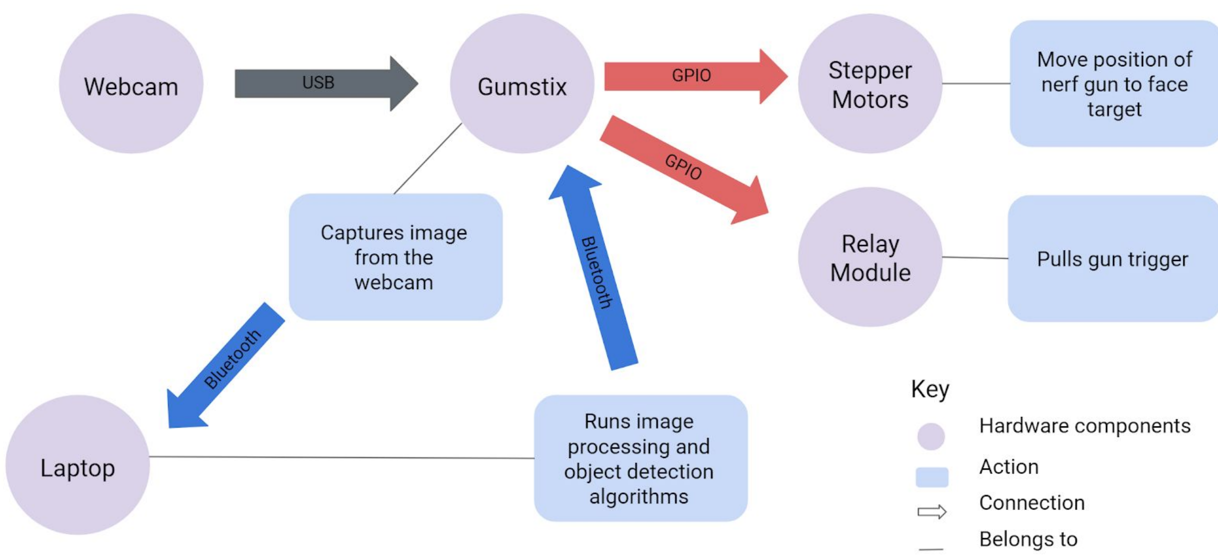

***Figure 2**: Initial Design Flow of Security System*

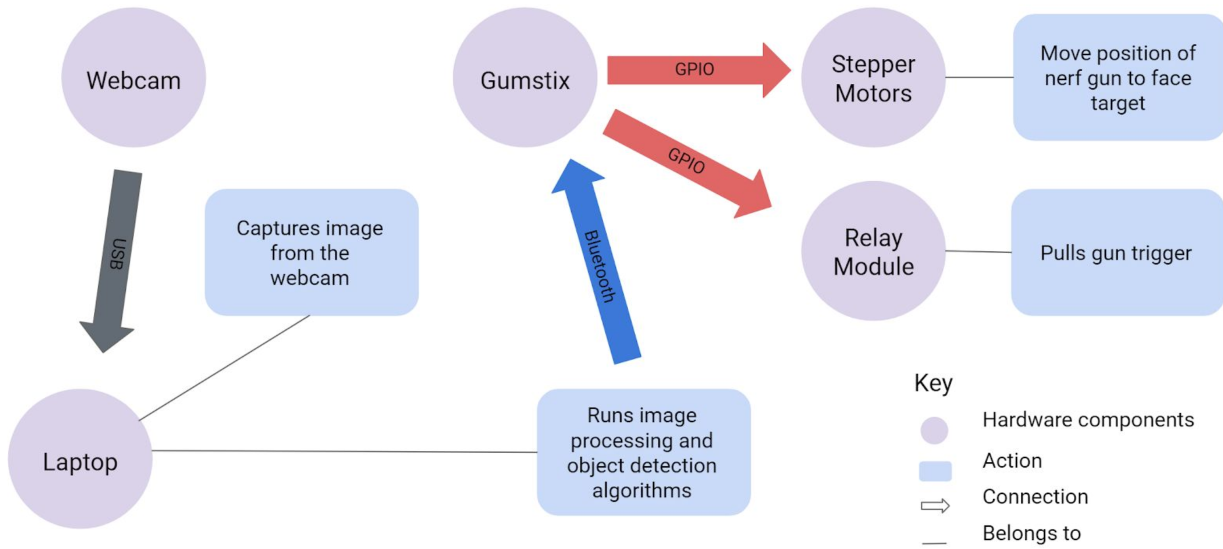

Figure 3*: Final Design Flow of Security System*

The work distribution is shown below:

- Hardware design (i.e. gun turret frame) - Junwei Zhou
- Two way bluetooth connection - Yaying Zheng
- Kernel modules for motor control - Yaying Zheng, Junwei (Antony) Zhou
- Webcam driver configuration + user level application - Chenjie (Lavi) Zhao
- Object detection code - Chenjie(Lavi) Zhao, Junwei (Antony) Zhou

The overall contribution to the project is as follows:

- Chenjie(Lavi) Zhao - 30%
- Yaying Zheng - 35%
- Junwei Zhou - 35%

Project Details

Webcam
Refer to /source/Webcam for the source code

In the initial project design, the webcam was decided to be connected to the Gumstix. As such, the Linux USB Class Video driver (UVC driver) and Video4Linux2 (V4L2) API was needed in order for the webcam to work on a target arm-linux embedded system like the Gumstix. The provided kernel modules v4l1-compat.ko, v4l2-common.ko, videodev.ko, uvcvideo.ko, compat_ioctl32.ko were used to set up the required drivers. The compat_ioctl32.ko, v4l1-compat.ko and v4l2-common.ko set up the V4L2 in a 32-bit system. The videodev.ko and uvcvideo.ko built the USB device as a video device.

In order to interact with the installed device drivers, a user level application need to be cross-compiled using arm-linux-gcc and loaded onto the Gumstix. The uvccapture module and v4l2grab module were the two user level applications chosen for this design. Both of the said modules require the jpeglib library, which was not included in the host kernel used to cross compiled user applications. As such, the library dependencies also needed to be built using arm-linux-gcc from the source. The jpeglib library was built from the jpeg9c source and statically linked to the user level applications during cross compilation.

The uvccapture module contained many functions such as pan-and-tilt to give a more complete control of the webcam movement. On the other hand, the v412grab module is more lightweight and perfect for just continuously capturing and saving video images as jpegs. However, it does require an additional library, the libv4l2 library. The uvccapture and v4l2grab application was tested on a Ubuntu 18.04 system. The uvccapture application was additionally tested on the Gumstix. Both applications compiled and executed successfully. For the purpose of this project design, v4l2grab module would be enough because the camera was mounted on top of the gun. As such, the gun will follow the movement of the gun and no additional movement would be required.

Once the drivers and user application were loaded, the webcam was connected to the Gumstix through USB for testing. The webcam model used is Logitech QuickCam Orbit/Sphere. Vendor ID is 046d and device ID is 08b5. Unfortunately, the installed drivers were unable to register the webcam as a video device. It may be a compatibility issue with the provided drivers and this particular webcam model. Failure to use the Gumstix to capture video images compromised the initial project implementation for two way bluetooth connection between the gumstix and computer. The project had to regress to connecting the webcam to a linux based computer through USB in order to capture the necessary video images.

Object Detection

Refer to /source/ObjectDetection for the source code

This project uses the Tensorflow MobileNet + SSD (Single Shot MultiBox Detector) model pre-trained with COCO dataset to classify and detect common objects, such as human and cars. MobileNet is the base network for classification and is used for its optimal accuracy and versatility compared to the popular YOLO (You Only Look Once) framework. The source code is written in python and uses opencv library for pre-processing and displaying images on the computer. The detection is limited to human for the purpose of this project, but the source can be easily modified to detect other objects instead. Face recognition function to narrow down the detection range to specific target is also evaluated, but is not integrated to the final project because of its low accuracy.

After detecting a person, the algorithm finds the bounding box of that person using SSD and establishes the center of the bounding box as the center of the target. If the center of the frame is aligned or within a small distance from the center of the target, then the program outputs instructions to shoot into a .txt file. Otherwise, the program sends instructions for motor movements in attempts to align the center of the frame with the center of the target in order for the gun to aim at the center of the target once it is found. If there are several people detected, whoever has a center closest to the center of the frame becomes the target. The instruction the program outputs is always three character bytes, in the format of shoot (1/N) + horizontal movement (L/R/N) + vertical movement (U/D/N). In other words, NRD means do not shoot, move the gun right and down. In the end, the program uses opencv to display the video stream annotated with bounding boxes.

Hardware and Circuitry

This project is built from a motorized nerf gun and a self-designed wooden turret. The frame contains four big pieces of wood. To perform horizontal rotation, two pieces of round wood with same radius are applied at the bottom of the turret frame. The motor performing horizontal rotation is fixed to the upper part and its shaft goes through the upper board and fixed to the lower board. When the motor rotates, the upper board will rotate. To perform vertical rotation, two pieces of wood with same length are applied by two side of the gun. The motor performing vertical rotation is fixed to one board and its shaft goes through the board and attaching to the gun. There is a long screw going through the other standing board and attaching to the gun too. So when the motor rotates, the gun will be able to rotate vertically. Four 90 degree bracket are used to fix the standing boards to the round rotating board.

To control the stepper motors, two stepper motor drivers and a 12V DC power supply are used. Each motor driver requires two GPIO inputs from the Gumstix to control rotating step and rotating direction, four outputs to motor, power supply for the motor and power supply from gumstix to the driver itself. To trigger the gun, a relay module is used. The relay module requires power supply for the module and one signal input. Two power wires of the gun are disconnected first and then connected to the relay module. The relay module is powered by gumstix and is controlled by one GPIO input. When the GPIO output is set, two power wires will be connected by the relay module and the gun is triggered.

Bluetooth

Refer to /source/Bluetooth for the source code

In the initial project implementation, two way bluetooth communication is used to transfer data between the user computer and Gumstix. Bluetooth communication is first attempted between the Gumstix and a computer with a Windows environment. Interaction with the Gumstix is established using minicom, which sets up a terminal for communication. Using terminal commands like hcitool and rfcomm in minicom and terminal on user computer with a linux subsystem, bluetooth connection was attempted continuously. After numerous failed attempts, it was discovered that a linux environment is also required on the user computer in order to successfully establish bluetooth connection. As such, Ubuntu 18.04 was installed as a dual boot. After the installation, bluetooth connection between Gumstix and user computer was successfully established using rfcomm commands. Then, the next step is to set up a way for the two devices to communication. After further research, it was decided that socket programming using RFCOMM protocol was the optimal option. With socket programming, one device would act as the client that sends data and the other would act as the server that received the data. Initially, it was decide that the Gumstix and computer would act as the client and server since both devices need to send and receive data. Once a socket is established, the Gumstix and user computer would be able to relay information. After further testing, the Gumstix and user computer was able to successfully send and receive information from each other. However, due to compatibility issues with the webcam and gumstix, the design was modified to use only one way bluetooth communication. Therefore, the user computer would take the role of the client and the Gumstix would take the role server.

Motor and Trigger Control
Refer to /source/Motor for the source code

The motors and trigger are controlled by 5 GPIO output signals in total. The relay module and two stepper motors are powered by Gumstix. The relay module requires one GPIO signal to control the connection of the circuit and each motor driver requires two GPIO signals to control the rotating step and rotation direction of the motor. When there is no object detected, the gun is set to scan horizontally by rotating 180 degrees back and forth automatically. When the object is detected, the gun will move towards the object and aim at it. When the gun is aiming at the object, it will be triggered and start shooting until the object is off the center of the camera. A timer is set every 10 ms in kernel module and checks the command. When there is no object detected, a default command will be passed to kernel module. When the rotating angle reaches to 90 degrees to either left or right, the rotating direction will be changed and the direction signal will be set and passed to the horizontal motor. When there is object in camera but not at the center, kernel module will receive a command indicating rotating direction for both horizontal and vertical motors. Direction signals will be passed to two motors. When the object is at the center of the camera, a shooting command will be passed to the kernel module. The trigger signal will be set and signals to two motors will be cleared to make sure the gun will not move when it is shooting.

Summary

This project focuses on developing a real time security surveillance system with automated response to detected threats. The components implemented including bluetooth, webcam, gumstix microprocessor, a computer that runs the object detection program, hardware and hardware control connect and communicate correctly with each other. The end product can operate real time with minimal delay.

One of the biggest challenges remaining is to configure the webcam as a video device on the gumstix so that the computer is completely detached from all the other parts. This problem should be resolved by getting a new camera compatible with the Linux UVC drivers. The other technical difficulty is that the step motors used do not have enough torque to rotate the base plate and the nerf gun. Other future work that can be done include replacing bluetooth connection with cloud computing so as to freeing the computer. The object detection model can also be optimized to detect weapons and face recognition function can be incorporated to real time detection to narrow down the target to unauthorized people. The software side of the project can be further secured against malicious attacks. A passive response such as alerting the user through text messages can be combined with the current active response approach to make the system even more versatile.

References

[1] Real-Time Object Detection on Raspberry Pi Using OpenCV DNN and MobileNet-SSD, https://heartbeat.fritz.ai/real-time-object-detection-on-raspberry-pi-using-opencv-dnn-98827255fa60

[2] Tensorflow detection model zoo,
https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md
[3] v4l2grab - Grabbing JPEGs from V4L2 devices,
https://www.twam.info/software/v4l2grab-grabbing-jpegs-from-v4l2-devices
[4] Independent JPEG Group, http://www.ijg.org/
[5] Linux UVC driver and tools, http://www.ideasonboard.org/uvc/
[6] CONFIG_USB_PWC: USB Philips Cameras,
https://cateee.net/lkddb/web-lkddb/USB_PWC.html
[7] Video4Linux, https://github.com/qtec/build-qt5022-core/wiki/Video4Linux
[8] RFCOMM sockets, https://people.csail.mit.edu/albert/bluez-intro/x502.html
[9] hcitool man page, https://www.systutorials.com/docs/linux/man/8-hciconfig/
[10] Serial Bluetooth Communication, https://unix.stackexchange.com/questions/92255/how-do-i-connect-and-send-data-to-a-bluetooth-serial-port-on-linux
[11] Bluetooth serial connection between Ubuntu and Android,
https://avilpage.com/2017/10/bluetooth-communication-between-ubuntu-android.html
[12] How to Make a Raspberry Pi Motion Tracking Airsoft / Nerf Turret,
https://youtu.be/HoRPWUl_sF8
[13] How To Wire It! Stepper Motors, https://youtu.be/GgfgWU0bpHk
[14] How to use 2 channel relay to control AC and DC loads in Arduino,
https://youtu.be/EclIomn_3dI