X-Bot
Badminton Shuttlecock Collector

Xi Wang[1], Li Wang[2]

Electrical & Computer Engineering Department,
University of Massachusetts Amherst,
Amherst, Massachusetts 01002

[1]xwang4@umass.edu, [2]liw@umass.edu

www.umass.edu

2018.12

X - BOT
Badminton Shuttlecocks Collector

Xi Wang[1], Li Wang[2]

Electrical & Computer Engineering Department,
University of Massachusetts Amherst,
Amherst, Massachusetts 01002

[1]xwang4@umass.edu, [2]liw@umass.edu

Abstract—More and more people choose professional gymnasiums to practice badminton. The main way to collect and sort badminton shuttlecocks are by hand or mechanical machines. In order to improve the efficiency of collecting and sorting and save the salary paid for staff in gyms, we began the research of a semi-automatic robot, X-Bot. X-Bot can see, collect and sort the shuttlecocks much more quickly and help people relax while playing badmintons. With X-Bot, professional gymnasiums can decrease workload to save money. X-Bot consists of several components and is controlled by software. People can control the Bot by laptop or cell phone via a Wi-Fi connection. There is a chassis with move all wheels, a Raspberry Pi 3 B+ board, a brush and a pulley system in X-Bot. With the designed structure, X-Bot can collect and sort shuttlecocks in order in containers.

I. INTRODUCTION

THERE are so many people playing badminton in professional gymnasiums that collecting and sorting shuttlecocks is exhausting work for both players and staff. The main ways for people to do such jobs are using their hands or with mechanical machines. In order to improve efficiency and cut the salary paid for the staff in gymnasiums, we proposed the idea to design a robot to solve the problem.

In the beginning, we came up an idea to create a robot. We planned to use Arduino as the development board. In consideration of image identification function in the future, we chose Raspberry Pi 3 B+ later as a result of camera settings. The Raspberry Pi 3 B+ boasts a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet and PoE capability via a separate PoE HAT. The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving cost and time to the market. [1]

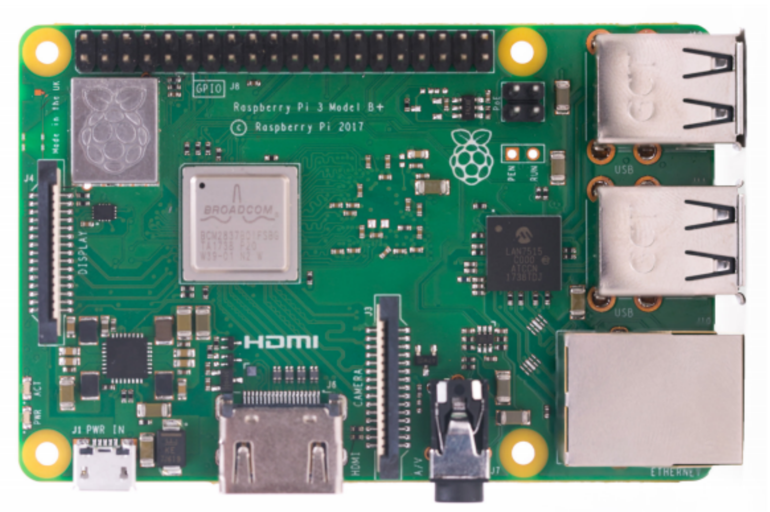

Figure 1-1: Raspberry Pi 3 B+ [1]

The Bot is designed to move as a smart vehicle with a chassis and wheels. At first, we set up the frame of the vehicle. Programming is essential in this part. The vehicle could move forward, backward, forward-left, forward-right, backward-left etc. The Bot could be started and stopped by using the GUI programmed in Python. There are two ways to control X-Bot. The first way is to control the robot by cellphone or laptop to collect shuttlecocks all around. Then we added a 480 * 640 camera to do object detection. And we can use the GUI to start and stop the robot to collect the shuttlecocks that are in one line.

In order to fix all the components into the vehicle, we bought a lot of raw materials to do tests. We chose aluminum alloy as the material of the mechanical frame. Then we ordered several brushes and components used in pulley system. The chassis and wheels for pulley system are printed by a 3D (3-Dimension) printer.

1.1 System Requirements

There are several system requirements in X-Bot.

1) Portable

X-Bot should be easily portable and move flexibly in gymnasiums. The weight should be low and the presence should be as small as it can be. The size and weight of components and installation manner should be taken into consideration.

2) Battery

There are several kinds of batteries on the market. We use a battery unit to supply power for the main board and driver boards. The main board is used to control the whole system. Four motors are installed in the chassis to control four wheels separately in the vehicle. One of the driver boards is used to control the brush and another one is used to control the pulley system.

Lithium batteries have lithium as an anode. Taking portability and rechargeability into consideration, we choose a rechargeable lithium battery to supply efficient power for different modules and to make sure the Bot could be continuously used at least 1 hour after every charging. The voltage of the two modules is 12v and another one is 6V. The battery is used to supply enough power for every module.

3) Remote control

The Bot is designed to be used in professional gymnasiums. In order to use fewer wires and make the Bot move more flexibly, the Bot is connected to the main board via Wi-Fi. A Wi-Fi module is in the Raspberry Pi which is installed in the chassis of smart vehicle. Raspberry Pi 3 B+ is used as a mini laptop.

We can login into the Raspberry Pi via a hand phone by using the IP address, user name and password of the Raspberry Pi. In order to use X-Bot in any network, it is better to set a static IP address to avoid complex settings in different networks with different IP addresses generated by DHCP (Dynamic Host Configuration Protocol) technology.

A hand phone is used to visit the official Raspberry Pi website to access programs uploaded there. A control GUI can be clearly seen. First, we power on the Bot and Raspberry Pi. Then we connect the Bot and the main board via Wi-Fi module in the Raspberry Pi. The vehicle could move all around supported by Mecanum wheels. Then we can start the brush and pulley system to achieve the function of collecting, transmitting and sorting shuttlecocks in order.

4) Move All Around

Four Mecanum wheels are installed in the vehicle to make the Bot move all around. As the image shows below, the Bot can move forward, backward, right, left, forward-right, forward-left, backward-right, backward-left, turning-right, turning-left, curved trajectory and lateral arc. The four wheels are driven by four separate motors.

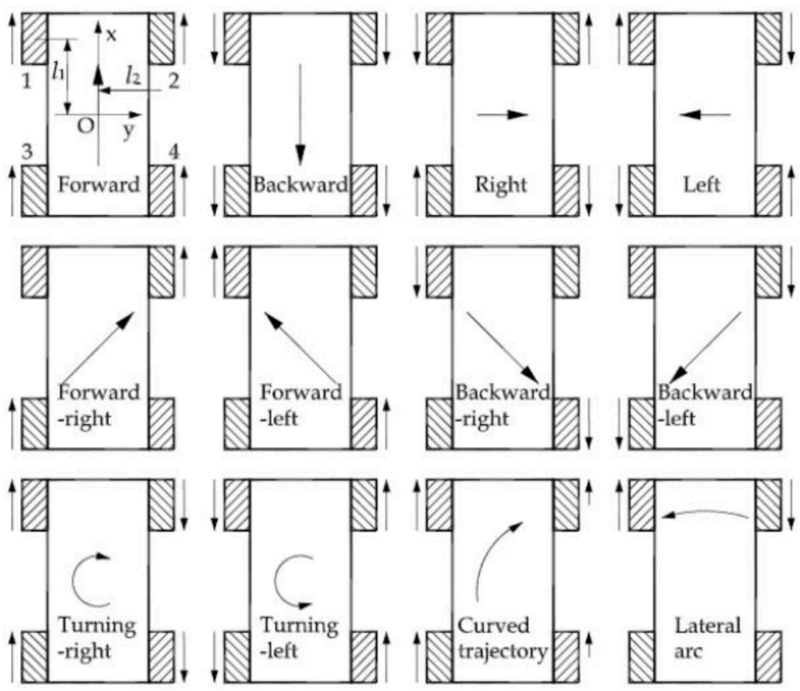

Figure 1-2: Mecanum Wheels Moving Directions [2]

5) Capacity of container

X-Bot is designed to be portable. The dimensions are 720 * 360 * 390 (L*W*H, mm). Three shuttlecock containers are fixed on the chassis of the Bot and 7 shuttlecocks could be sorted in order in each container.

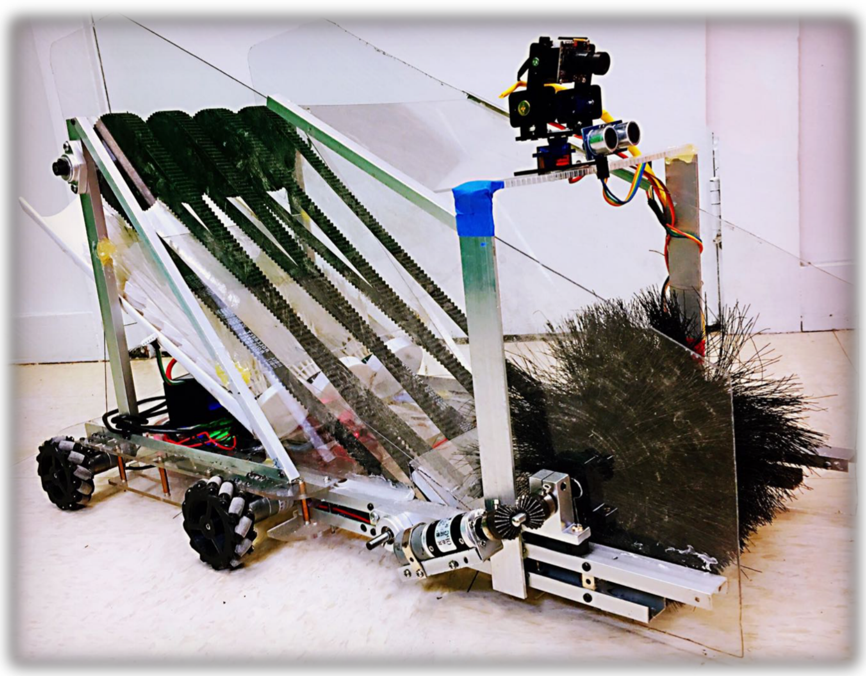

Figure 1-3: X-Bot

6) Camera

A camera is needed if we would like to make the Bot see, collect, transmit and sort the shuttlecocks in order. In this project, we choose a 480 * 640 camera to do shuttlecock detection.

7) Ultrasonic Module

In semi-automatic working mode, an ultrasonic module is essential to avoid obstacles when the robot goes forward.

II. DESIGN

2.1 Overview

Several functions are part of Bot. With the Bot, we can collect, transmit and sort shuttlecocks on the floor in professional gymnasiums. It is easy to control the Bot via cell phone or a laptop. A GUI and a program are developed in Python. All the components are visible on the Bot. We did some calculations and many tests before the Bot was finally formed.

2.1.1 Experiments & Components

In the beginning, we drew a draft of X-Bot without exact dimensions. According to the functions mentioned above, we designed several main parts: a Raspberry Pi used to control the whole system, driver boards used to control the brush and belts in the pulley system, a brush used to sweep shuttlecocks onto belts in the pulley system, a pulley system used to transmit shuttlecocks into containers, battery packs added to supply power for the boards and motors for wheels, a smart vehicle with a chassis and four wheels, a mechanical frame proposed as a stand of X-Bot and many components adopted to connect the whole system (such as bearings, gears, screws, wire etc.), a camera used to help the Bot see shuttlecocks, and an ultrasonic module to avoid obstacles.

1) Vehicle chassis

A smart vehicle with a two-layer chassis is needed. In consideration of the size of the Bot (the diameter of the shuttlecock container is 70mm and there are three containers), we chose plastic as the raw material of the chassis and designed the chassis to 300*400 (W*L, mm) to make the Bot light and easy to move. The upper layer is used as the foundation of mechanical system. The lower layer is used as the base of hardware, including a main board (Raspberry Pi 3 B+), a driver board for the brush, a driver board for the pulley system and battery packs. It is difficult and time exhausting to purchase raw materials. We chose transparent plastic boards as raw material and installed wheels, boards, and a battery pack on the chassis ourselves in the workshop at home.

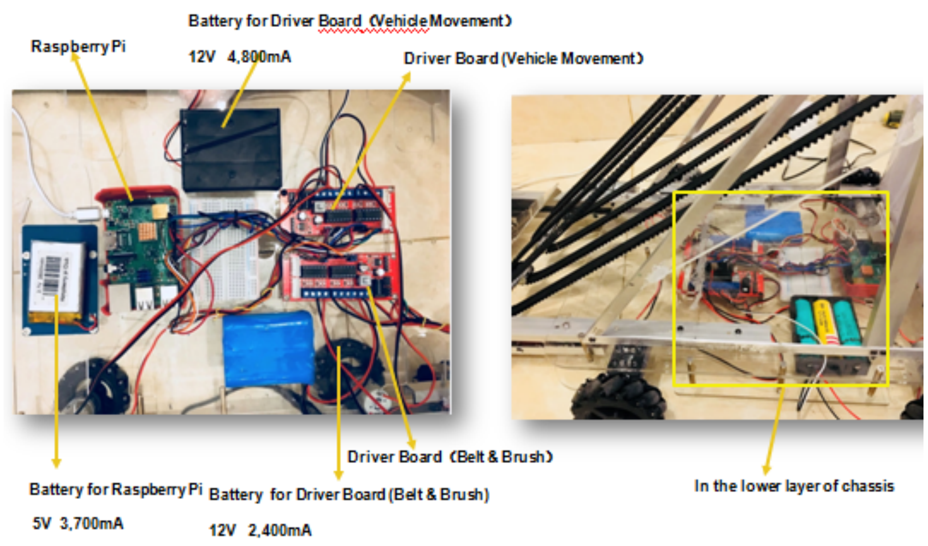

Figure 2-1: Components on the second chassis in the beginning of the project

2) Mechanical frame design

The second important thing was to set up the mechanical frame. Aluminum alloy is the first choice to make the stand of X-Bot because of its light weight (the density of aluminum is 2.7 g/cm³) and easy to be fixed on the chassis. According to our calculations, if the angle between the belt in the pulley system and the first layer of the chassis is 30 ~ 45 degrees, it is easy for shuttlecocks to be transmitted on the belts and fall into the container in order. We bought aluminum alloy strips online and calculated the length of each edge of the frame. We cut them into the dimensions we needed and installed the frame on the chassis by using drill to punch stiletto, using screwdriver to fix screws. The size of the Bot is 720 * 360 * 390 (L*W*H, mm).

Figure 2-2: Tools

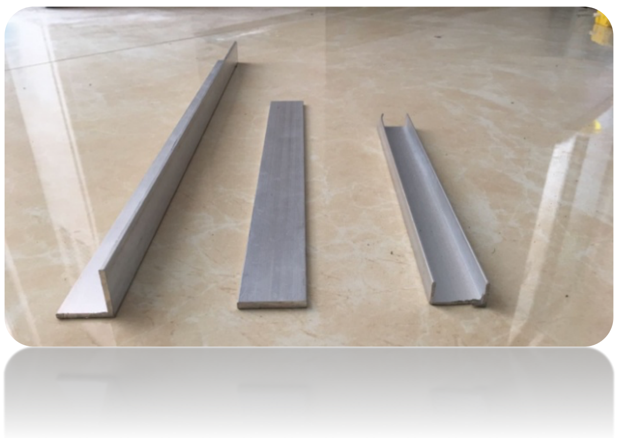

Figure 2-3: Aluminum Alloy

3) Brush and axles selection

It took several weeks to get a perfect brush. At first, we bought a finished product online. It was tough to use the finished product because the diameter of the brush is about 100mm. It was hard to sweep shuttlecocks onto belts. After many tests, we contacted with the manufacture to customize a brush with a diameter of 200mm to sweep shuttlecocks easily. The brush could rotate both clock-wise and counter-clockwise controlled by programming. It is controlled by the micro-controller.

Figure 2-4: Brush Testing

In order to find auxiliary products online easily and to make structure be stably. we choose axles with diameter of 6mm.

We can see Figure 2-5 of the final structure of the brush. With the brush, the Bot can sweep shuttlecocks on to the belts.

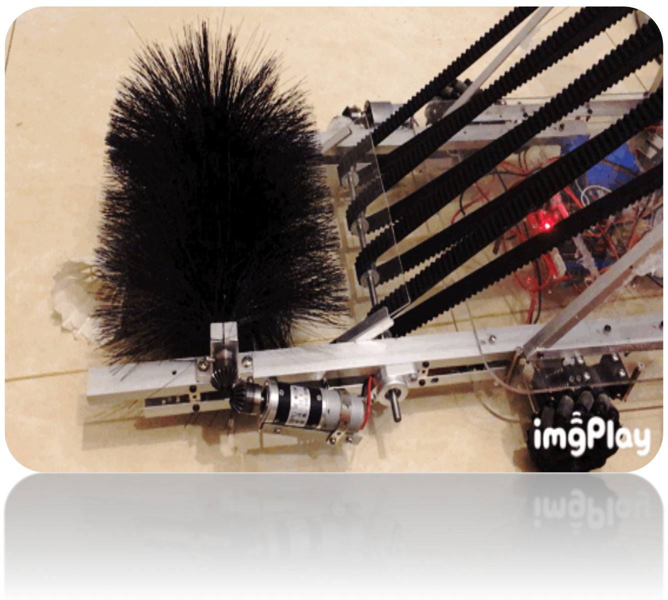

Figure 2-5: Brush and umbrella gears in X-Bot

4) Belts selection

For the pulley system (Figure 2-6), the selection of motor, gears, and belts and their installation manner are very important. We tested four kinds of materials for the belt: cotton rope, rubber rope, rubber belt with steel wire and rubber belt. It is impossible to transmit shuttlecocks by using ropes because the connection area is too little to hold shuttlecocks. For rubber belt with steel wire, there is no finished cycle belt. We sewed the belts by hand. A pulley system can work. Since we wanted to improve the presence of the system, we finally bought a finished product in the market to get same length of each belt. The pulley system is controlled by Raspberry Pi 3 B+.

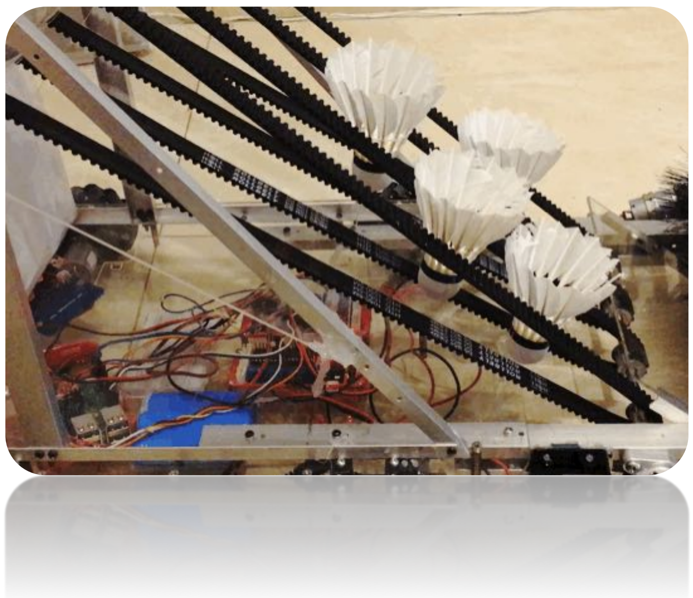

Figure 2-6: Belts in the Pulley System

5) Wheels design for pully system

Initially, we chose smooth plastic wheels for the pulley system. During the experiments, we found that it is not easy for shuttlecocks to fall into the container. Then we designed the wheels by using software and printed them out using a 3D printer.

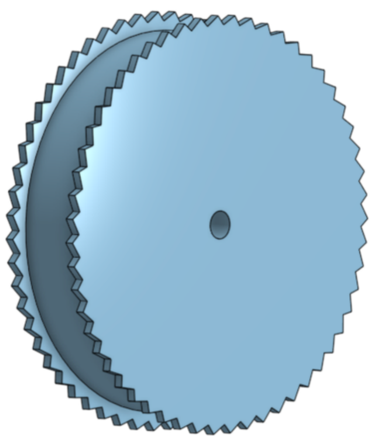

Figure 2-7: New Design of the Wheel in Pulley System

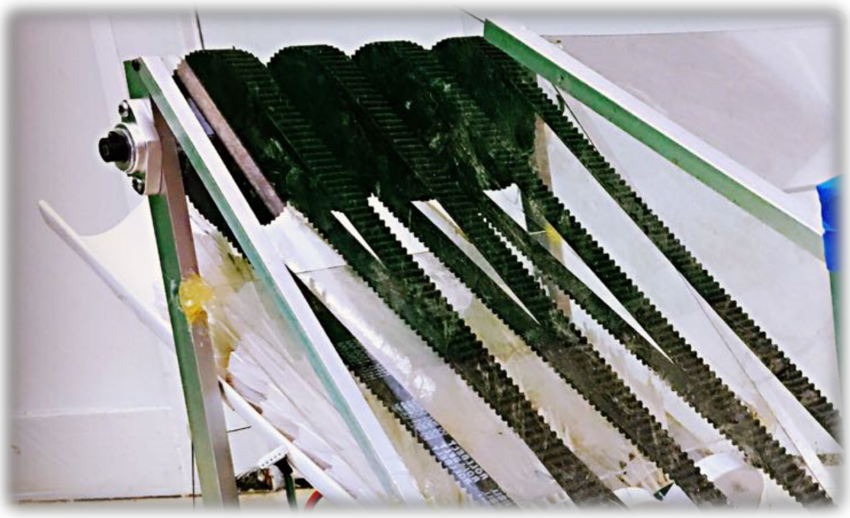

Figure 2-8: Wheels in the Pulley System

X-Bot has a chassis with a main board, driver boards and battery packs fixed on the second layer. There are four Mecanum wheels to help the Bot to move around. A brush is installed in the front of the Bot to sweep shuttlecocks onto the convey belts in the pulley system. The pulley system consists of several gears, pulleys and transmitting belts.

6) Gears selection

The Bot consists of a chassis with two layers, four Mecanum wheels driven by four motors to move around, and a brush with a motor to collect shuttlecocks. An axle is used to connect with belts to transmit shuttlecocks into containers. A mechanical frame and parts in the Bot are used to connect different structures. Two pairs of umbrella gears are used to connect the motors and axles. The axle of the brush and the axle of pulley system at the bottom are installed horizontally. The motor for the brush and the motor for pulley system are installed vertically. Gears are used to connect the axle and the motor. If we choose common gears, it is difficult to fix the position of the motors and the width of the Bot would be bigger. We looked for a good way to connect the axle and motor. For both the brush and pulley systems, we chose a pair of umbrella gears to connect the axle and motor. The rotation speed of the motor is 100r/min. The advantages of using umbrella gears are: connect components in both horizontal and vertical directions, space saving and better presentation.

7) Container selection

Shuttlecock containers are made from standard containers and the container stands are made of paper. Metal could be the raw material for optimization.

8) Camera selection

A 480 * 640 (480P) camera[3] (Figure 2-9) is installed on top of brush to help the Bot see shuttlecocks on the floor. The camera can see objects in the range of 40cm to 90cm (the distance is between the camera to the object). We choose the greatest distance when the camera sees many shuttlecocks at one time.

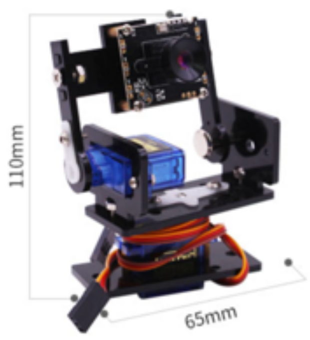

Figure 2-9: Camera[3]

9) Distance calculate between the Bot and shuttlecock

We calculate the distance between Bot and shuttlecock by using triangle similarity (Figure 2-10):

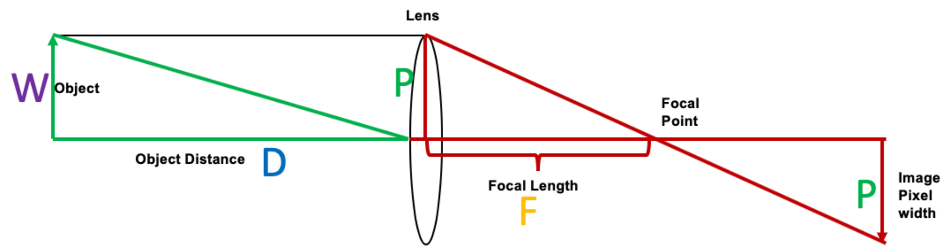

Figure 2-10: Distance from camera to object

W = Actual object width
D = Distance from camera to object
P = Object pixels width in image
F = Focal length

Formula: D = (W x F) / P

The following figure is the relationship between the distance (from camera to object) and pixel width in image (Figure 2-11). The camera effective zone is between 40cm and 90cm from the Bot to shuttlecock.

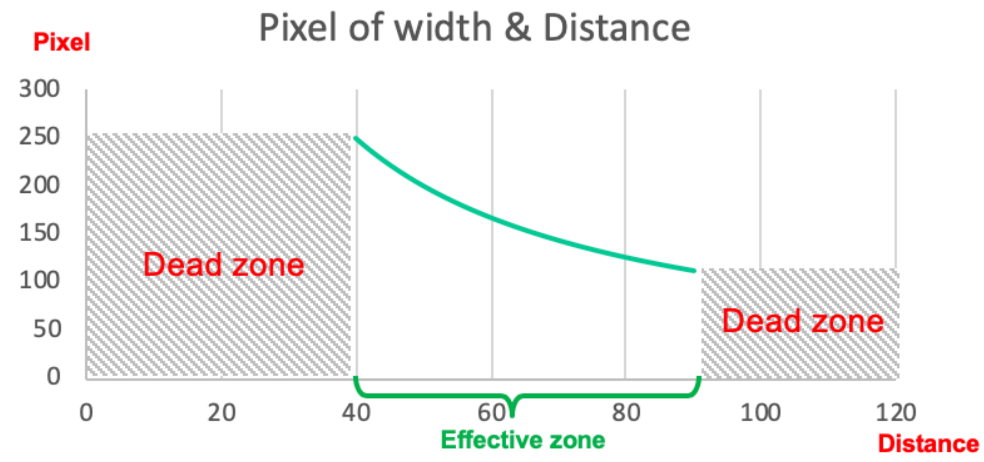

Figure 2-11: The relationship between pixel width and distance

The following figure is that relationship between Bot running time and distance (Figure 2-12). The car speed rate is 50cm/s. The maximum running time is 2 seconds in every movement.

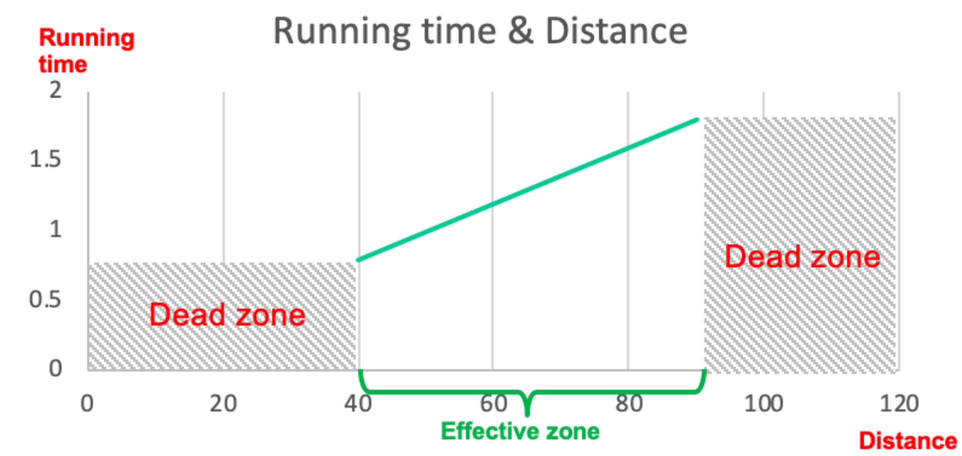

Figure 2-12: The relationship between running time and distance

10) Ultrasonic Design

An ultrasonic module is right under the camera to detect obstacles in front of the Bot. If the distance from the module to the obstacle is less than 30cm, the Bot goes backward. If the distance is equal to or greater than 30cm the Bot goes forward. We measured the distance by using the following formula:

Distance = ((ECHO - TRIG) x 340m/s) / 2

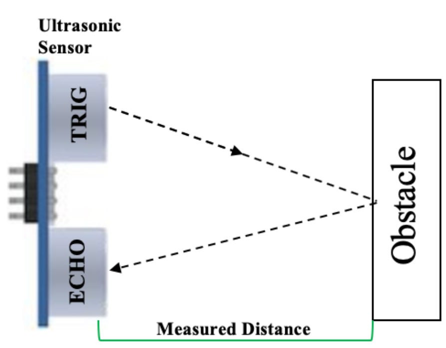

Figure 2-13: Ultrasonic Module

2.2 Working Mechanism

2.2.1 Block Diagram

In the Bot, there are several modules: the cell phone and PC used separately as operating devices, the power supply module used to supply power to the main board, driver boards, the brush and pulley system, and the camera module used to generate a video stream and send data to microprocessor. A Wi-Fi module is used as a connection device to connect the Bot, the microcontroller and camera are used to detect shuttlecocks and an ultrasonic module is used to avoid obstacles.

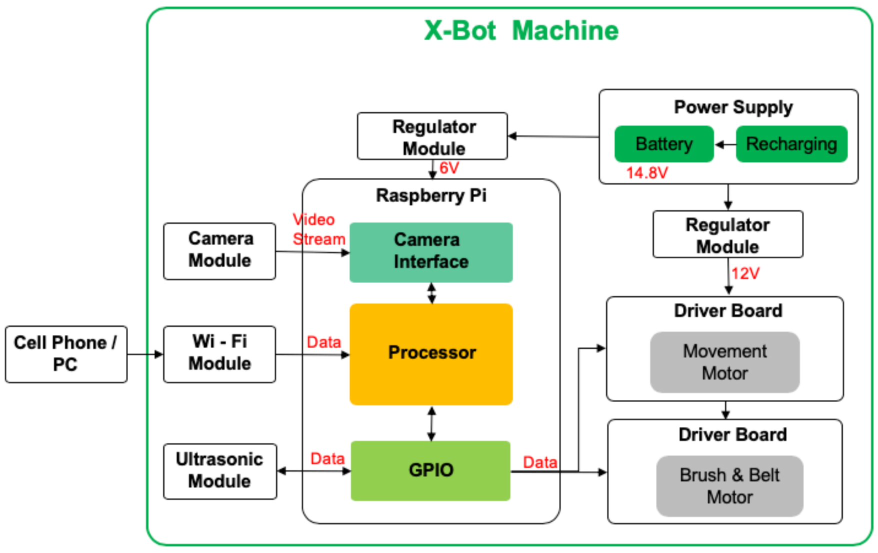

Figure 2-14: X-Bot Block Diagram

There are three boards used in the Bot. The Raspberry Pi 3 B+ is used as the main board to encode, decode and perform data analysis and processing. One driver board is used to control the axles and motors of the brush and pulley system. Another driver board is used to control the movement of the Bot. The LiPo battery is rechargeable.

2.2.2 State Diagram

The Bot is designed to be controlled by a person. Initially, there is a person to power on the Raspberry Pi and log into the control GUI in a mobile phone or laptop to operate the Bot. The Bot communicates with devices via Wi-Fi. Image identification has been added. There is a camera installed on the top of the brush. The camera is used to capture images of shuttlecocks and send data to the Raspberry Pi. An ultrasonic module is used to detect the distance between the module and obstacles. After data analysis, the micro-controller sends data to the brush. The brush starts working if a shuttlecock is observed. Then shuttlecocks are swept onto the conveyor belts in the pulley system and transported into containers. Because of the structural design, shuttlecocks fall into containers in order. After all the containers are full, we can take out sorted shuttlecocks in order. We can reboot the Bot and make it work cycle by cycle.

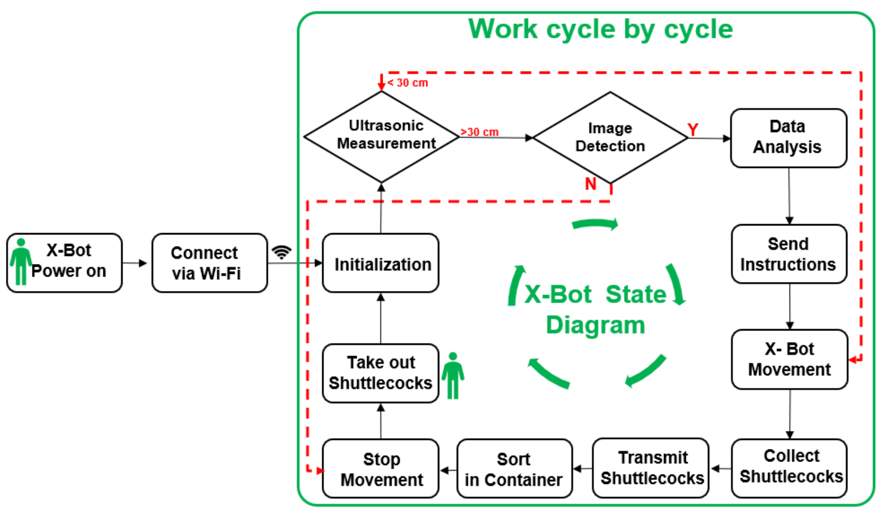

Figure 2-15: X-Bot State Diagram

2.3 Battery

The battery is very important to the system. We not only make sure the Bot can run continuously 1 hour, but also want to make sure that the battery is as safe as possible.

2.3.1 Consumption of Bot

If we want the Bot work for one hour continuously, the total consumption in an hour is (Table):

No.	Components	Q'ty	Max Current per Unit	Voltage	Power
Total		11	2,311mA		38.08W
1	Raspberry Pi	1	500mA	6V	3W
2	Camera	1	160mA	6V	0.96W
3	Ultrasonic	1	15mA	6V	0.09W
Sum		3	675mA	6V	4.05W
4	Driver Board	2	36mA	12V	0.86W
5	Motor for wheels	4	400mA	12V	19.2W
6	Motor for brush	1	600mA	12V	7.2W
7	Motor for Pully system	1	600mA	12V	7.2W
Sum		8	1,636mA	12V	34.03w

Remark: The standard voltage of the battery for Raspberry Pi is 5V [1]. But we have to increase the voltage to 6V if we want to use USB camera to work stably on Raspberry Pi.

Table 2-1: Consumption of Bot

2.3.2 Battery selection

We selected a rechargeable lithium battery unit to supply power for the Bot. The voltage is 14.8V and the current is 5,000mA. The total energy it can supply in an hour is:

No.	Battery	Q'ty	Current	Voltage	Power
1	LiPo battery (4 cell)	1	5,000mA	14.8V	74W

Table 2-2: Battery capacity

The working hour of the Bot is: 74W*h / 38.08W ≈ 1.94h

It means the Bot can work about 1.94hr after every charging.

2.3.3 Safety for Battery

LiPo cells are affected by the same problems as other lithium-ion cells. This means that overcharge, over-discharge, over-temperature, short circuit, crush and nail penetration may all result in a catastrophic failure, including the pouch rupturing, the electrolyte leaking, and fire. [4]

All Li-ion cells expand at high levels of state of charge (SOC) or over-charge, due to slight vaporization of the electrolyte. This may result in delamination, and thus bad contact of the internal layers of the cell, which in turn brings diminished reliability and overall cycle life of the cell. [5] This is very noticeable for LiPos, which can visibly inflate due to lack of a hard case to contain their expansion.

For security, we selected the following battery and charger (Figure 2-16):

Figure 2-16: LiPo Battery [6] and Charger [7]

2.4 Image detection

In the mode that the Bot can see shuttlecocks, we designed a program using an OpenCV library [8] and the Python language. When there are no obstacles and a shuttlecock is detected by the camera, a person can start the Bot by using the button 'Start' on the GUI. In this stage of preparation, we took many photos of shuttlecocks in the lab, trained the Bot to recognize shuttlecocks and made the Bot see, collect, transmit and sort the shuttlecocks that are in one line on the floor into containers.

2.4.1 Dataset Preparation

In order to train the Bot to see shuttlecocks, we use the LBP feature extraction in OpenCV. The main algorithm is KNN (K Nearest Neighbors). In the first step, a dataset of shuttlecock is required. We took many photos in the lab as a positive image dataset and downloaded some images from ImageNet as a negative image dataset.

Figure 2-17: Sample Datasets

2.4.2 LBP (Local Binary Patterns) brief introduction

We used Local Binary Patterns (LBP) as the feature extraction method for image recognition. This description captures fine-grained image details. The LBP methodology has its roots in 2D texture analysis. The basic idea of LBP is to summarize the local structure in an image by comparing each pixel with its neighborhood. A pixel is taken as the center and threshold for its neighbors. If the intensity of the center pixel is greater than its neighbor, then denote it with 0. If the center pixel is lesser than or equal its neighbor, then denote it with 1. For example (Figure 2-18), with 8 surrounding pixels there are 2^8 possible combinations, called Local Binary Patterns.[9]
The generated rules are shown below:

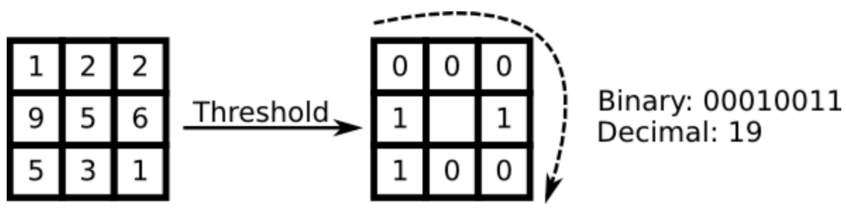

Figure 2-18: Generate Example[9]

In order to make LBP encode details in different scales, the operator was extended to use a variable neighborhood [8]. The idea is to align an arbitrary number of neighbors on a circle with a variable radius, which enables to capture the following neighborhood(Figure 2-19):

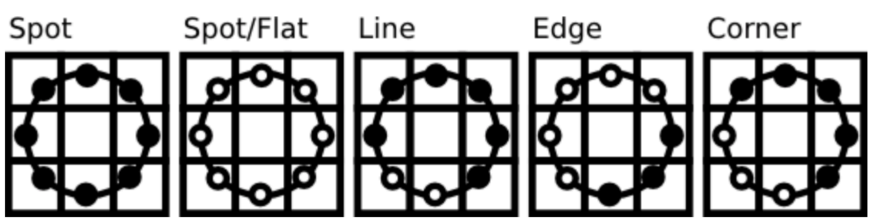

Figure 2-19: Generate Example [9]

For a given Point (x_c, y_c) the position of the neighbor $(x_p, y_p), p \in P$ can be calculated by:

$$x_p = x_c + R\cos\left(\frac{2\pi p}{P}\right)$$
$$y_p = y_c - R\sin\left(\frac{2\pi p}{P}\right)$$

Where R is the radius of the circle and P is the number of sample points. [9]

But the number of LBP values generated is 2^P by using this method. To solve this problem, a useful extension to the original operator is the so-called uniform pattern [10] which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor. This idea is motivated by the fact that some binary patterns occur more commonly in texture images than others. A local binary pattern is called uniform if the binary pattern contains at most two 0-1 or 1-0 transitions. [11]

After several experiments, we set the number of nearest neighbors is K = 10 as a result of best performance in detection.

2.4.3 Training Parameters in OpenCV

We have to define some parameters when we train a new model in OpenCV.

The NumPos, NumNeg, and NumStage are defined by us. In addition, OpenCV suggests the definition and values of other parameters[12].

No.	Name of Parameters	Value
1	NumPos (No. of Positive Images)	354
2	Image size	40*40
3	NumNeg (No. of Negative Images)	1027
4	NumStages (No. of Stages)	6
5	MinHitRate (Minimum Hitting Rate)	0.999
6	MaxFalseAlarmRate (Maximum False Alarm Rate)	0.2
7	WeightTrimRate (Weight Trim Rate)	0.95

Table 2-3: Name of parameters

2.4.3 Testing Result

We add bounding box to each detected shuttlecock using the camera. Because the X-Bot could move around under the control of a person, almost all the shuttlecocks on the floor could be recognized.

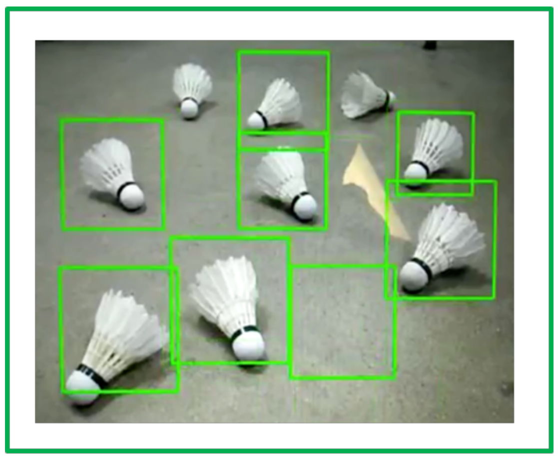

Figure 2-20: Testing Result

2.5 Programming for control

Python is used as the programming language [13]. There are two control modes in the Bot: controlled by person and semi-automatic.

2.5.1 Controlled by person

We developed a GUI for the Raspberry Pi using Python to control Bot movement and the brush and pulley system. We can control the Bot by using our cell phone or laptop. The Bot can build the Wi-Fi connection to transmit the control signals. There are several buttons shown on the GUI: forward, backward, forward-left, forward-right, backward-left, backward-right, brush, pulley system and so on. With the operating system, even a child can control the Bot easily.

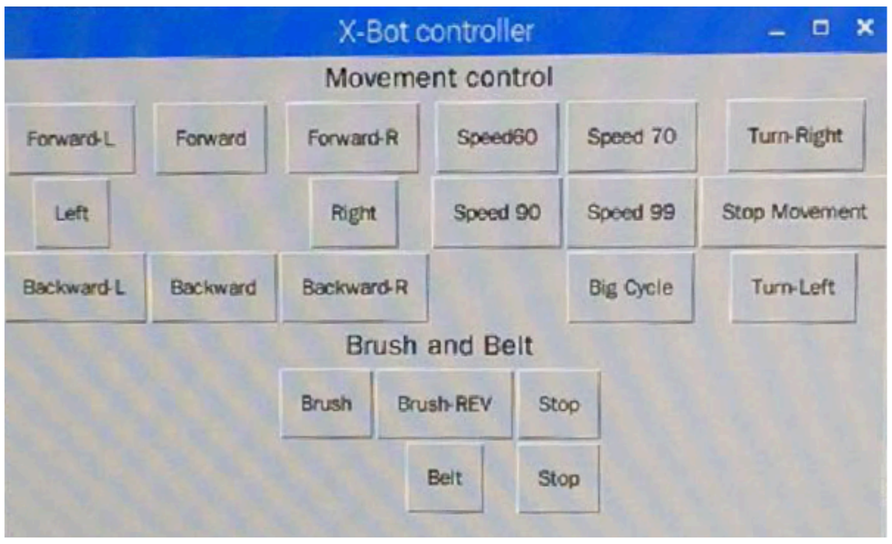

Figure 2-21: Straightforward

2.5.2 Semi-automatic

We developed a GUI with two buttons for the Raspberry Pi to run and to stop the Bot. There are two logical methods for the Bot: Collect straightforward and Collect Turn-Left/Right.

1) Move straightforward:
First, the Bot starts to detect objects. If the Bot detects objects, it will move forward. After collecting 3 times, it will transmit shuttlecocks into the containers. Then the Bot will perform detection again.

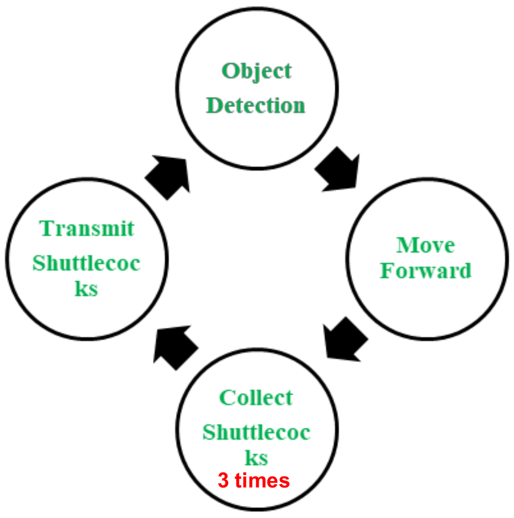

Figure 2-22: Straightforward

2) Collect Turn-Left/Right
The Bot starts object detection. If the Bot detects objects, it will move forward. After collecting three times, it will transmit shuttlecocks into the containers. Then the Bot will rotate a fixed angle to detect objects. If the Bot could not find any objects, it will stop moving.

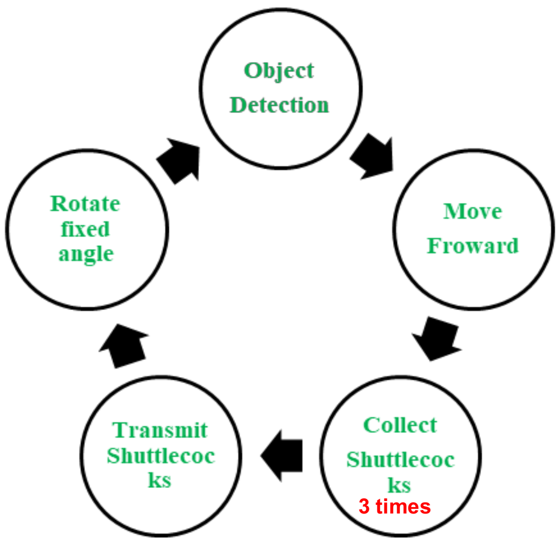

Figure 2-23: Turn-Left/Right

III. WORK PARTITIONING & SCHEDULING

Professor Tessier is our advisor. There are two members in the X-Bot team. Xi (Daniel) Wang is the team leader. He is responsible for software design and monitor the whole working process. Li (Charlotte) Wang did hardware design and website design. We began all the work in the end of this June (after PDR) and finished the project in the middle of December.

IV. CONCLUSION

We made a physical portable Bot to detect, collect, transmit and sort shuttlecocks into containers in order. A rechargeable battery unit is used to supply power to the Bot. After every charging, the Bot can run continuously in about 2 hours. All three containers can hold 21 shuttlecocks combined.

The accuracy for shuttlecock detection by using OpenCV is shown as the following curve:

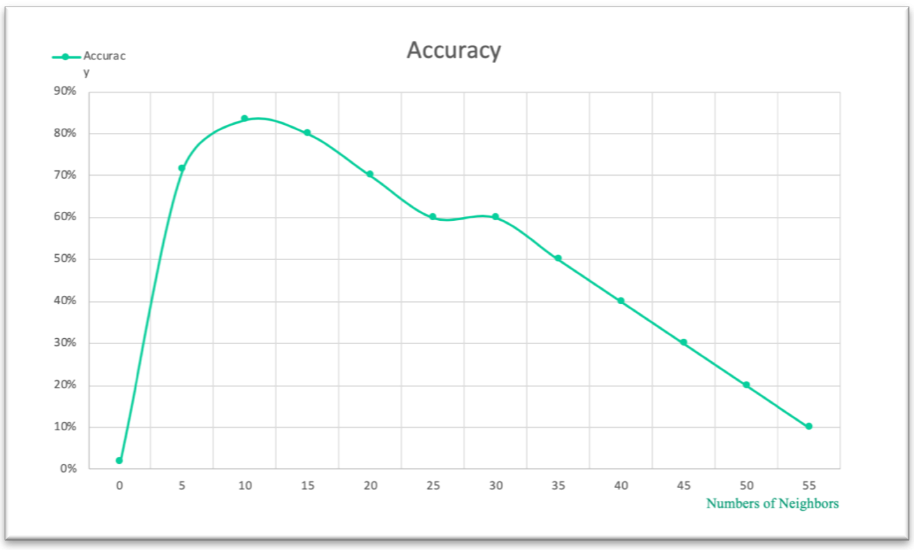

Figure 2-24: Accuracy

We completed object detection function by using Python and OpenCV. There are two working modes in X-Bot. One mode allows the Bot to be controlled by laptop/handphone. Another mode allows the X-Bot to recognize shuttlecocks and collect them into containers.

The Bot's performance is good in the mdoe of going straightforward during detection. The testing result of object detection is shown as Figure 2-25:

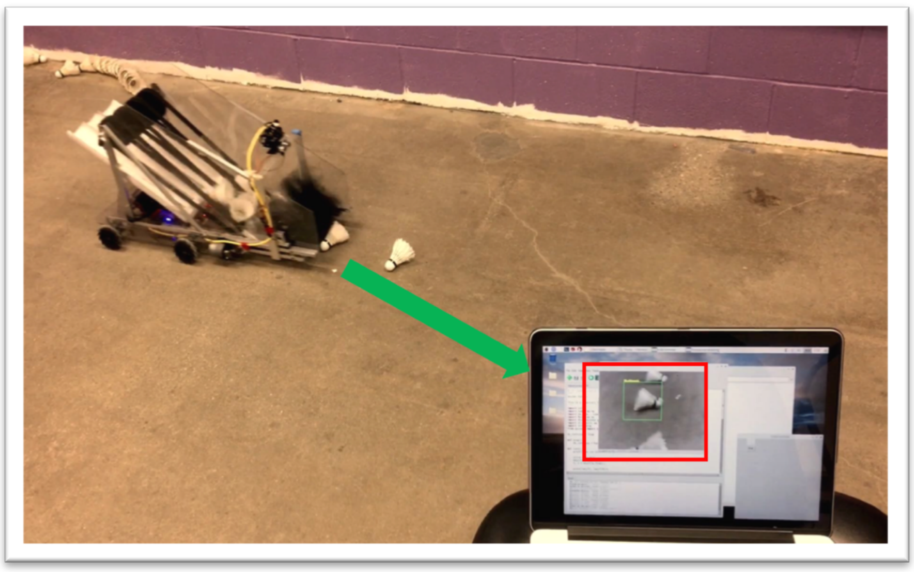

Figure 2-25: Collect straightforward

During the project, we found the computation capability of Raspberry Pi 3 B+ is not good at processing real time data delivered from the camera.

In the future, we will try to add a GPU to do graphic processing and implement Faster R-CNN (Regression Convolutional Neural Networks) to improve detection accuracy.

The teamwork has been excellent. Both Daniel and Charlotte have good communication. We have meetings with Professor Tessier every week to ensure every target could be completed well.

V. ACKNOWLEDGEMENT

Thanks a lot for Professor Tessier's great support! He has given us many strong suggestions and helped us to optimize the project.

Thanks for Professor Hollot, Professor Eslami and Professor Xia! As evaluators of our project, all of them have given us many suggestions to define the scope of our project.

VI. REFERENCE

[1] Raspberry Pi Document: http://www.raspberrypi.org

[2] https://www.researchgate.net/figure/Vehicle-motion-according-to-the-direction-and-angular-speed-of-the-wheels-6_fig4_233867057

[3] https://item.taobao.com/item.htm?spm=a1z10.5-c-s.w4002-14871668706.19.3f996b84jkwo2T&id=561420113647

[4] FAA Battery Incident Chart, includes incidents of Lithium-Polymer-Air ignition after puncturing. Ex: Entry for 11-Dec-2007

[5] Vetter, J.; Novák, P.; Wagner, M.R.; Veit, C. (9 September 2005). "Ageing mechanisms in lithium-ion batteries". Journal of Power Sources.

[6] http://www.turnigy.com/

[7] https://www.skyrc.com/

[8] https://opencv.org

[9]https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms

[10] Barkan et. al "Fast High Dimensional Vector Multiplication Face Recognition." Proceedings of ICCV 2013

[11] https://en.wikipedia.org/wiki/Local_binary_patterns

[12]https://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html?highlight=numstages#viola2004

[13] Programming In Python 3: A Complete Introduction To The Python Language 2nd