ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

FPGA-based object tracking, obstacle avoidance, and voice-activated trolley

---

Team #23

JIARUN HU
(jiarunh2@illinois.edu)
YANG ZHOU
(yangz15@illinois.edu)
YIHANG HE
(yihangh2@illinois.edu)
HAOMIN WANG
(haominw3@illinois.edu)

TA: Tielong Cai
Sponsor: Said Mikki

May 22, 2023

Abstract

Nowadays, there is a growing trend in the development of electric vehicles, with a focus on equipping them with intelligent systems. However, existing Systems on Chip (SOCs) based on non-real-time operating systems fall short in meeting the real-time and safety requirements of in-vehicle systems. To address this issue, we propose a low-energy and high-performance system design using a Field Programmable Gate Array (FPGA). Our system comprises a trolley with object tracking, obstacle avoidance, and voice-activated capabilities. The control subsystem, based on Arduino, communicates with the FPGA to receive signals from the object tracking and voice-activated subsystems. The object tracking subsystem uses an FPGA-based image processing unit to track objects, while the obstacle avoidance subsystem employs ultrasonic sensors. The voice-activated subsystem utilizes an FPGA-based acoustic processing unit. Our design incorporates pipeline, unroll, and dataflow techniques to optimize performance. Verification through simulation and testing ensures the correct functionality of the subsystems. The proposed FPGA-based system offers low energy consumption, real-time performance, and enhanced intelligence for our smart trolley.

Key words: intelligent system, Field Programmable Gate Array (FPGA), object-tracking, obstacle-avoidance, voice-activated, convolutional neural network (CNN).

Contents

1 Introduction

1.1 Problem

Nowadays, the development of electric vehicles has become a trend. At the same time more and more new energy vehicle startups like to equip their cars with intelligent systems. However, existing Systems on Chip (SOCs) are always based on non-real-time operating systems and cannot meet the real-time and safety of the in-vehicle system. Common systems which are based on Central Processing Unit(CPU) and Graphics Processing Unit(GPU) tend to have high energy consumption [1], which will have a negative impact on the endurance of the vehicle. Therefore, it is necessary to design a system with low energy consumption and high real-time performance.

1.2 Solution

In order to achieve low energy consumption and high real-time performance, our solution is to design a specific system to control our trolley based on Field Programmable Gate Array (FPGA). The latency of some FPGA-based designs can be deterministic. Besides, FPGAs are typically more energy efficient than other platforms except for Application Specific Integrated Circuits (ASICs), which usually have higher Non-Recurring Engineering (NRE) costs [1]. For neural network inference accelerators, FPGA implementations can be up to 10x more energy efficient than GPU ones [2].

1.3 Functionality

Our trolley has 3 functions: object tracking, obstacle avoidance and voice activated. The trolley has voice activated mode and object tracking mode, and users can switch the mode by a voice command. In voice activated mode, the movement of the trolley is controlled by single word commands manually by users and the object tracking function is disabled. In the object tracking mode, the voiced commands that control the movement of the trolley is disabled, instead, the trolley will track an object with a single color and avoid colliding with the obstacles in the path.

1.4 Subsystem Overview

1.4.1 Top Level Diagram

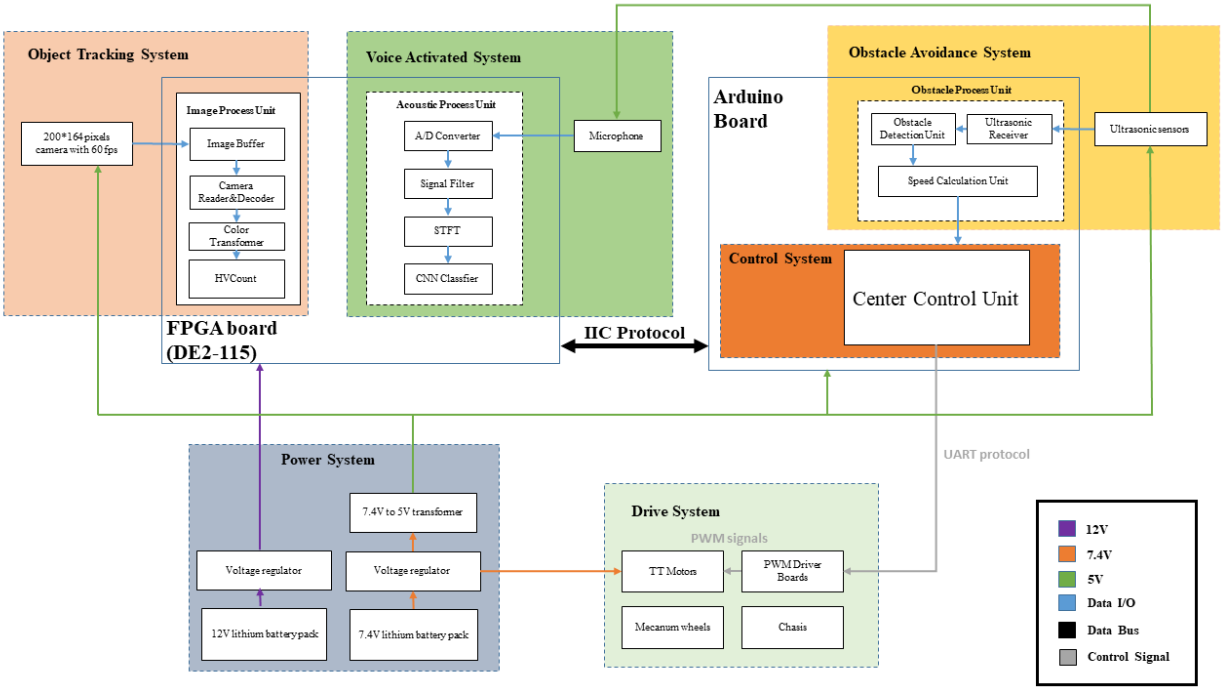

Figure 1: Top Level Diagram of System

1.4.2 Subsystem Descriptions

1. **The Object Tracking Subsystem**:

   The Object tracking subsystem is specifically designed to track a balloon with a single color. When we move such a balloon in front of our trolley, the object tracking system calculates proper x_error and h_error values and sends them to the control subsystem.

   This subsystem comprises a camera and processing unit in FPGA, which consists of a camera reader, decoder, color space transformer, and HVcount Module. The camera captures images with 200x164 pixels at 60 fps, saving them into the Block Random Access Memory (BRAM) of the FPGA. The color space transformer reads pictures from BRAM, firstly converts RGB to YCBCR color space, then uses a binary threshold module to convert it to a black and white image to which the HVcount module can apply for identifying the location of the object effectively. Finally, HV-count determines the x_error and h_error and sends them to our control subsystem for processing.

2. **The Obstacle Avoidance subsystem**:

   The Obstacle avoidance subsystem is specially designed to help the trolley bypass

obstacles in its way. It comprises ultrasonic sensors and an obstacle processing unit implemented on our Arduino Uno board.

The ultrasonic sensor detects the distance between the trolley and an obstacle. If the distance is lower than a certain threshold, the obstacle avoidance subsystem will detect it and adjust its speed and direction accordingly. The control subsystem processes these new speeds of the wheels and adjusts the trolley's direction to bypass the obstacle in its path.

3. **The Voice-Activated subsystem**:

   The Voice-Activated subsystem is specifically designed to recognize natural language instructions with one or two words and send them to the control subsystem. It consists of a microphone and an acoustic processing unit implemented in our FPGA board.

   The acoustic processing unit contains an A/D converter, a short-term Fourier transform (STFT) module, and a convolutional neural network (CNN) classifier. All of these units are designed in hardware design language (HDL) and integrated into our FPGA board.

   First, our microphone captures the user's voice and sends it to the A/D converter to convert it into a digital signal. The STFT module then processes this digital signal and turns it into a spectrum. Finally, the CNN classifier in figure 1 classifies this processed spectrum and recognizes the type of instruction it represents. These recognized instructions are then sent to the control subsystem for further processing.

4. **The Control Subsystem**:

   The control subsystem is based on Arduino and communicates with FPGA using IIC (Inter-Integrated Circuit) protocol to receive signals from the object tracking subsystem and voice activated System. We have designed two modes for our control system: tracking mode and voice manual control mode, which can be switched through voice instructions.

   In tracking mode, the control subsystem receives two data from the object tracking subsystem - x_error and h_error. x_error represents the offset of the object's center point along the x-axis from the screen center point, while h_error represents the difference between the size of the target object and the predetermined size. Our control subsystem then calculates the corresponding speeds for each wheel of the trolley and sends the signal to drive subsystem via Universal Asynchronous Receiver/Transmitter (UART) protocol.

   In Voice manual control mode, the Voice Activated System sends speed and direction signals to the control subsystem via IIC protocol, which are then translated into two corresponding speeds for each wheel. Again, it sends the signal to drive subsystem via Universal Asynchronous Receiver/Transmitter (UART) protocol.

   Regardless of the mode, the control system always processes the signal from the outside and sends the control signals to drive subsystem.

5. **The Drive subsystem**:

   The Drive subsystem comprises four Mecanum wheels, four TT motors, two Pulse Width Modulation (PWM) driver boards, and a chassis. The Mecanum wheels enable the car to change direction by setting the speed of each wheel independently without needing a servo.

   To achieve this, the PWM driver board receives speed signals from the control subsystem through UART protocol, generating four corresponding PWM signals that can be used to control each of the four TT motors separately. This setup provides precise and efficient movement control in different directions.

6. **The Power subsystem**:

   The Power subsystem is made up of two lithium battery packs, 7.4V and 12V respectively, as well as a 7.4V to 5V transformer and voltage regulation circuits. The 12V battery pack is connected to the regulation circuit and then to the FPGA board. On the other hand, the 7.4V battery packs are connected to the 7.4V to 5V transformer and regulation circuit before being connected to the drive subsystem and control subsystem. This helps ensure reliable and stable power delivery throughout the robotic car system.

2 Design

2.1 The Object Tracking Subsystem

2.1.1 Design Procedure

The object tracking pipeline is shown in figure 2.

The Object Tracking subsystem utilizes the OV2640 Image Sensor to capture images in front of the trolley and then sends these images to the FPGA via the SCCB protocol. To accomplish this, we first designed a camera init module within our FPGA board that could initialize the image sensor to Ultra Extended Graphics Array(UXGA) mode and set the output size of our sensor to 200x164 pixels. We also utilized a clock_pll to generate the necessary 12MHz clock for our image sensor xclk port and a 4MHz clock for the SCCB protocol to transfer data.

Once the initialization was complete, we design a camera reader that would read line and frame pixels and store them in an image cache implemented by on-chip memory within the FPGA. After this, a color space transformer was employed to transfer the RGB image into YCbCr, which helps avoid lighting variations, before finally transforming the image into black and white[3]. The HVCount module would count the black pixels to determine the size and location of the tracking object and compare it with the set target, ultimately resulting in determining x_error and h_error, which indicate the horizontal location and size of the region of black pixels correspondingly. These two pieces of information are then sent to our control system in Arduino through the IIC (Inter-Integrated Circuit) protocol. To implement this protocol, we set our FPGA as the slave and our Arduino as the master, and designed an IIC slave module within the FPGA.

We apply equations (2.1) when converting RGB images to YCbCr

$$\begin{aligned} Y &= (0.183R + 0.614G) + (0.062B + 16) \\ CB &= -(0.101R + 0.338G) + (0.439B + 128) \\ CR &= -(0.399G + 0.040B) + (0.439R + 128) \end{aligned} \tag{2.1}$$

We use a set of threshold values of Y, Cb and Cr to identify our target. If the the perceived YCbCr value is in the threshold, it will be converted to black pixel, otherwise it will be converted to white pixel. Therefore, we need to set the threshold values to the range of YCbCr values of our target to ensure the subsystem can identify our target accurately. We take several photos in figure 3 of our target and get the maximum and minimum YCbCr values of each photo in figure 4 to get the range of YCbCr values of our target, then we use range of YCbCr values of our target as the threshold value.

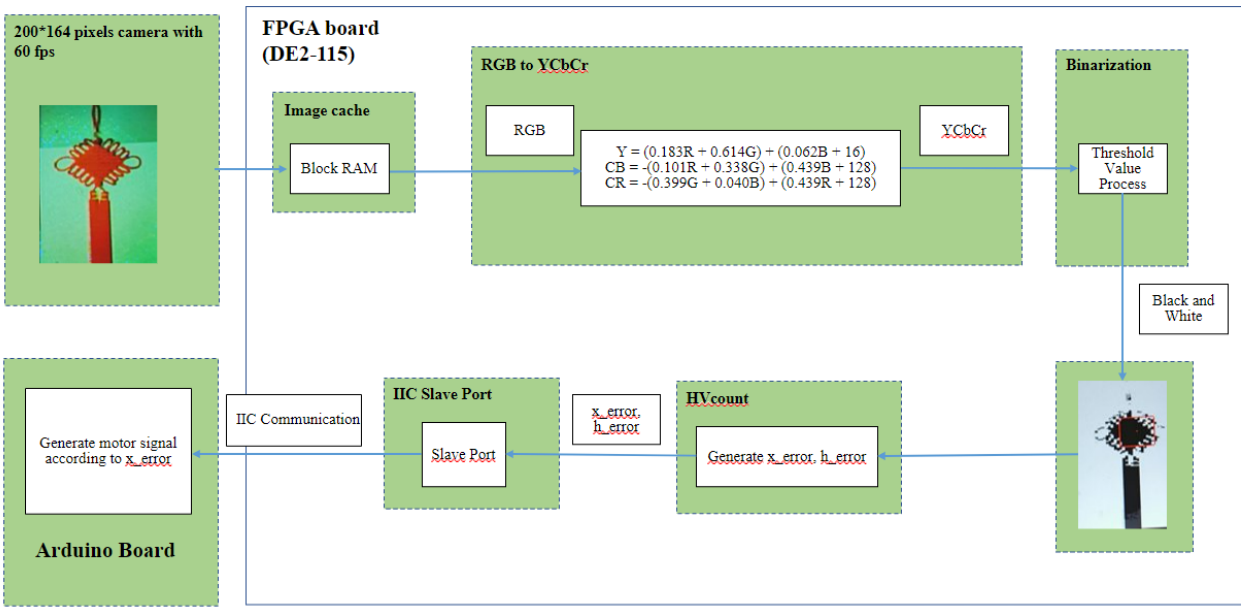

Figure 2: Object Tracking Pipeline

Figure 3: Photos of Target

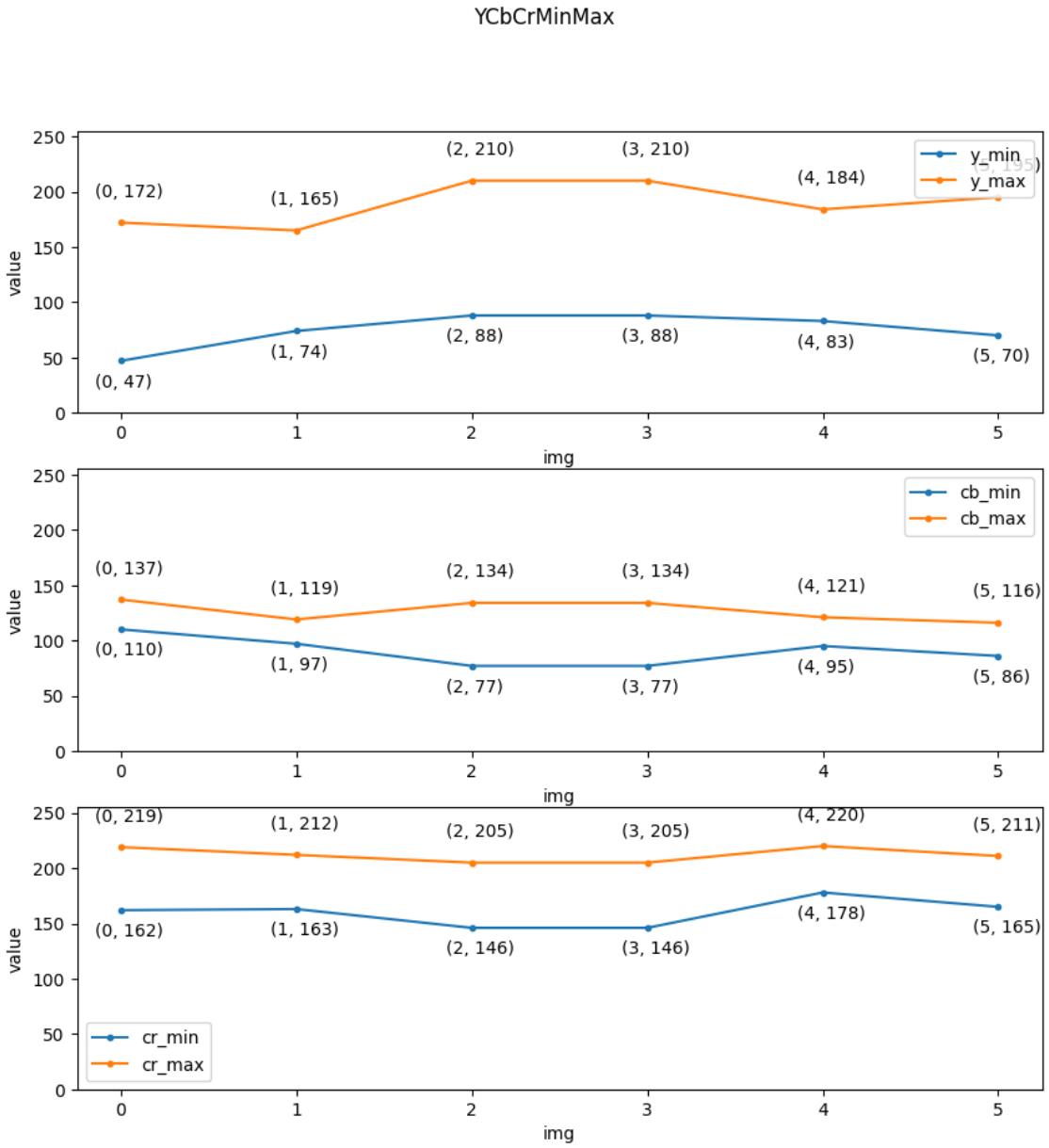

Figure 4: Maximun and Minimum YCbCr values of Photos

2.1.2 Design Details: Camera

For our object tracking subsystem, we require a camera capable of identifying objects that operate on 5V DC voltage supplied by our FPGA board with I2C protocol support. After researching, we considered OV2640 and OV7670 models. [4] [5]

Regarding resolution, the OV7640 has a maximum resolution of 640x480 pixels, and the OV2640 can capture images up to 2 megapixels (1600x1200). Nevertheless, our current usage only requires 200 x 164 pixels for identifying objects, as the FPGA BRAM is not enough for high-resolution applications.

Both cameras use Serial Camera Control Bus(SCCB) protocol for configuration and I2C bus for image data transfer. The OV2640 supports multiple data formats including JPEG, BMP, and YUV, while the OV7670 supports RGB565, RGB555, YUV422/420, and grayscale. However, since we only use the SCCB protocol for our purpose, it is not important.

In addition, the OV2640 has a parallel output mode that can achieve faster data rates compared to the OV7670 model.

Considering all factors, we have decided to select the OV2640 model due to its transmission speed for real-time object tracking.

2.2 The Voice Activated Subsystem

2.2.1 Design Procedure

The Voice-Activated subsystem assumes the responsibility of recognizing specific natural language instructions and subsequently executing corresponding actions. After we review recent research, it accomplishes this task through the sequential operation of various components, including the Analog to Digital (A/D) converter, signal filter, short-time Fourier transformer (STFT) module, and a CNN classifier[1]. And since our DE-2115 board has a WM8731 audio chip, we use it as our A/D converter directly by implementing a WM8731 controller. And for the CNN classifier, we trained it on our computer, quantized it from 32bit to 8bit for saving memory, and implemented it on our FPGA in Verilog. Alternatively, we have tried to use nios ii, but the latency requirements cannot at satisfied. And we also tried various configurations of the CNN model as table 1 to balance the logical block use and accuracy.

2.2.2 Design Details: Generate Spectrum for CNN

In this part, the microphone serves as the input device for capturing audio, producing an analog signal. To facilitate the conversion of the analog signal into digital form, we employ the wm8731 chip as our A/D converter. The communication between the wm8731 chip and the FPGA is established using the IIC protocol, with the FPGA acting as the master and generating the necessary clock signals for the wm8731 chip. The configuration of the wm8731 registers involves setting the sample rate to 44.1 kHz and the data width to 16 bits.

Due to the disparate calculation speed of the FPGA and the data flow rate of the wm8731 chip (where the FPGA is faster), a FIFO with a capacity of 1024 words is employed as a data buffer. The FIFO operates by caching the incoming data, and once it reaches full capacity, the oldest data is popped from the FIFO and directed toward the FFT module for processing.

To optimize the speed of the FFT calculation, the module is set to operate in streaming mode. This mode allows simultaneous calculation and data input, thereby enhancing the overall processing efficiency. Furthermore, the data type utilized in this context is block floating point. In a block-floating point FFT implementation, each value possesses an

independent mantissa while sharing a common exponent within each data block. This approach ensures precision like that of floating-point data, while simultaneously conserving logic elements akin to fixed-point data representation, reducing the complexity of calculation at the same time. [6]

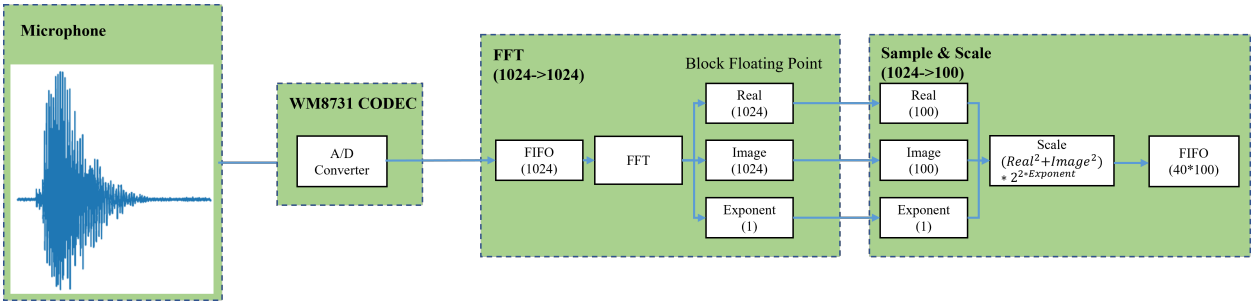

Figure 5: First part of the Voice Activated Subsystem, generate spectrum for CNN classifier

There is a sample and scale module after FFT. It catches the formal 512 data from FFT module due the symmetry of FFT calculation and resize it to 100 by accumulating adjacent 5 points. Repeat this procedure 40 times to get a spectrum with size 40x100. This process can be considered as STFT, which has a formula:

$$X(n,w) = \sum_{m=-\infty}^{\infty} x(m)\omega(n-m)e^{-j\omega m} \tag{2.2}$$

The $x(m)$ is the input signal from the A/D convector, and the $\omega(m)$ is a window function. In our design, we use a rectangular window. Here, our window length $N = 1024$

$$\omega(n) = R\_n(n) = \begin{cases} 1, 0 \leq n \leq N-1 \\ 0, otherwise \end{cases} \tag{2.3}$$

2.2.3 Design Details: CNN Classifier

The CNN classifier is comprised of two convolution layers and one fully connected layer, serving the purpose of classifying the type of the generated spectrum. In our experimental framework, we have explored various sizes of CNN models to strike a balance between the utilization of logical elements and achieving high accuracy.

After considering the overall logical block count in DE-2115, we have ultimately opted for the third configuration. Moreover, to enhance computational efficiency and reduce chip memory usage, we have employed quantization techniques to convert our model from a 32-bit floating-point representation to an 8-bit integer representation. This quantization process not only offers the advantage of minimizing additional clock latency arising from floating-point arithmetic but also facilitates faster calculations. Furthermore, in our hardware design, we have leveraged pipeline and unroll designs to maximize the degree of parallelism, aiming to optimize the utilization of available resources.

Model Architecture	Parameter Numbers	Accuracy	Logical Elements Use
Conv1(40*100→16*18*48) Conv2(16,*18*48→32*7*22) Linear(32*7*22→3)	27984	95.2%	More than 4×10^5
Conv1(40*100→16*18*48) Conv2(16*18*48→8*7*22) Linear(8*7*22→3)	4,632	90.7%	2.2×10^5
Conv1(40*100→8*18*48) Conv2(8*18*48→1*7*22) Linear(1*7*22→3)	862	82.1%	108390

Table 1: Design alternatives of CNN model

- **Pipeline Design**
  Within our convolution layer, a deliberate configuration has been implemented, involving the allocation of 40 x 5 buffers. This strategic arrangement ensures that during the loading process of spectrum data, a segment of data equivalent to the 5x5 convolution kernel size is loaded at each interval. As a result, the calculation module is not compelled to await the complete loading of all data before commencing its computations. Instead, it promptly initiates calculations on the data
- **Unroll Design**
  In our Convolution layer ii, in addition to employing a pipeline design, we have also implemented an unroll technique. This approach facilitates parallel computation for each input channel and output channel, as opposed to the sequential calculation performed individually. Consequently, for a 7x1 size, the total computation time is reduced significantly by a factor of 7, resulting in notable time savings.
- **Dataflow Design**
  The dataflow technique in our system operates on a task-level pipeline architecture. By exploiting this approach, we can efficiently transfer data from the preceding module to the subsequent module even before the completion of calculations in the former. This strategy maximizes the degree of parallelism, allowing for simultaneous data loading and computation. As a result, the overall latency is effectively reduced, enhancing the overall performance of the system.

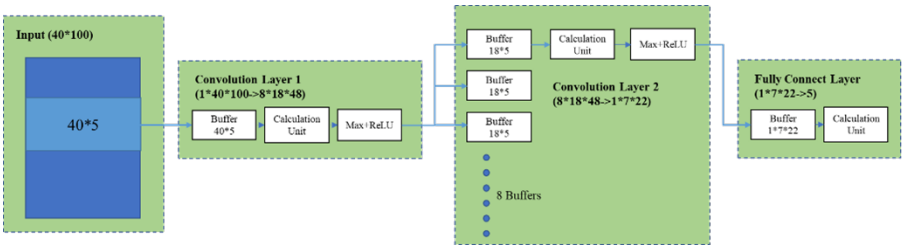

Figure 6: Hardware Design of CNN Classifier

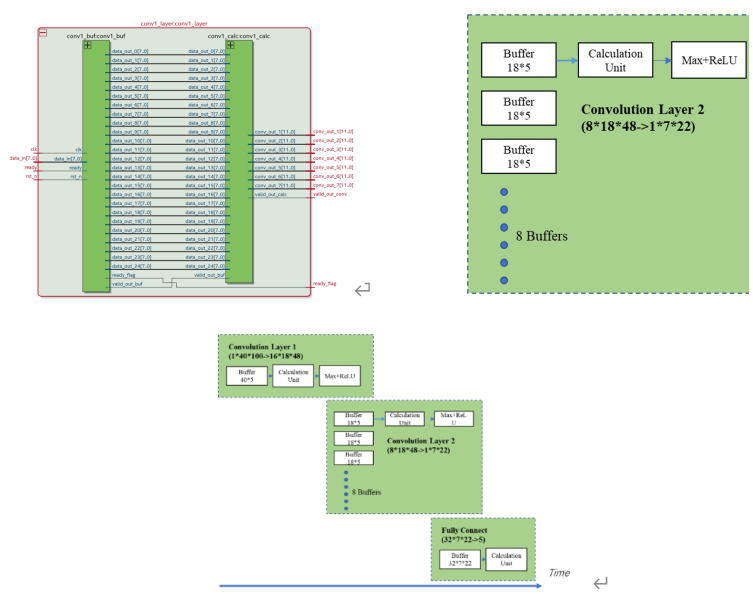

Figure 7: Pipeline, Unroll and Dataflow Design

2.3 The Obstacle Avoidance Subsystem

2.3.1 Design Procedure

The obstacle avoidance algorithm is simple yet effective for our purposes. In the event an obstacle is detected, the trolley will move backward briefly before turning left. If the obstacle persists, this cycle will repeat until the obstacle is bypassed.

2.3.2 Design Details: Sensor

We choose to use an ultrasonic sensor for our obstacle avoidance subsystem, and after considering several options - HC-SR04, JSN-SR04T, and US-100, we have decided to use the HC-SR04 sensor. When making our decision, we took into consideration several factors. These included the operating range, sensing angle, accuracy, power consumption, working voltage, and price of each sensor. [7] [8] [9]

In terms of operating range, HC-SR04 has a range of 2cm to 400cm, while JSN-SR04T has a range of 20cm to 600cm, and US-100 has a range of 2cm to 450cm. Given that our purpose is obstacle avoidance, JSN-SR04T was not suitable as its range could be lower

than 20cm.

Regarding the sensing angle, JSN-SR04T has the widest angle at 75 degrees, while HC-SR04 and US-100 both have a narrower angle of approximately 15 degrees. For our purpose of obstacle avoidance, a 15-degree angle is sufficient.

HC-SR04 and US-100 both offer accurate distance measurements with up to 0.3cm resolution, while JSN-SR04T has a lower resolution of around 1cm. Though 1cm is workable, 0.3cm is preferable.

US-100 has the lowest power consumption among the three sensors, while HC-SR04 has the highest. However, since power consumption is not a primary consideration for our project, this factor was relatively less important in our decision process.

In terms of working voltage, HC-SR04 and US-100 both operate on 5V DC, while JSN-SR04T operates on a range of 3.3V - 5V DC. Since we are using Arduino which is capable of supplying 5V DC, all sensors are compatible.

Lastly, price was also a factor in our decision. HC-SR04 is the most affordable option, with prices ranging from $1-$2 USD depending on quantity purchased. JSN-SR04T is typically the most expensive, with prices ranging from $8-$15 USD, while US-100 falls in between with prices ranging from $3-$5 USD.

Based on our analysis, we determined that HC-SR04 offers a good balance of range, accuracy, and price, making it a reliable choice for obstacle avoidance projects.

2.4 The Control Subsystem

The control subsystem of our autonomous trolley is based on Arduino, which offers several advantages over other platforms. Firstly, Arduinos are highly versatile and offer numerous options for integrating with various sensors and actuators required in an autonomous trolley operation, enabling the system to perform the three main functions effectively.

Secondly, Arduino's robustness, low power consumption, and real-time processing capability make it an ideal microcontroller for controlling autonomous trolleys. Its embedded processing system can read signals from multiple sensors, allowing the trolley to track objects and avoid obstacles in real-time as it moves.

Thirdly, Arduino is designed to be energy-efficient, utilizing microcontroller technology optimized for low power consumption, making it possible to run for extended periods using minimal energy. Furthermore, Arduino boards come with power-saving modes such as sleep mode, which reduces power consumption when not in use, improving its lifespan.

Overall, these features make Arduino a cost-effective, reliable, and flexible platform for building autonomous trolleys that operate smoothly, efficiently, and with minimal energy usage, making them ideal for our project.

2.5 The Drive Subsystem

We decided to use Mecanum wheels due to their advantages over normal wheels. These special wheels allow for omnidirectional movement, meaning the robotic car can move in any direction without needing to change its orientation. Normal wheels, in contrast, are only capable of moving forward and backward.

Furthermore, the individual motor control of each Mecanum wheel allows for better maneuverability, enabling the car to perform more complex movements such as sideways translation or rotation. Traditional wheels, on the other hand, are generally limited to simple movements like turning left or right.

Despite these benefits, Mecanum wheels do have some disadvantages. They tend to be more expensive than traditional wheels and also require more complex control mechanisms. Additionally, because Mecanum wheels have rollers on their circumference, they provide less traction, which in turn can result in reduced speed and acceleration.

Nonetheless, for our purpose of object tracking, the reduced traction is acceptable given the improved maneuverability offered by Mecanum wheels.

3 Requirements and Verification

3.1 Object Tracking Subsystem

Initially, we tested the RGB to YCbCr module, which converts the color space. We confirmed the module's functionality by comparing its output with the results generated using Python code on a computer. This testing guaranteed that the module works correctly.

Afterward, we tested the Binarization module and the HVcount module, which generates x_error and h_error. We use a Video Graphics Array (VGA) screen to display the image from the camera and the image after the process of HVcount module to verify if the algorithm works correctly. Figure 2. also gives the verification result in the 2 images. The full-color image is the original image from the camera and the black-and-white image is the one after the Binarization module. In addition, there is a red rectangular region in the black-and-white image, which is the result of the HVcount module. We can ensure the Binarization module correctly convert the pixel with the target color to black and the HVcount module successfully identifies the main body of the target.

Furthermore, we tested if the IIC protocol is executed correctly when the FPGA sends x_error and h_error to the Arduino. When the object tracking subsystem is running, we connect the FPGA and Arduino connect wires that transfer SCL and SDA signals to the oscilloscope. We compared the output of the oscilloscope and the standard waveform of the IIC protocol as the figure 8 to check if the IIC protocol works correctly.

Finally, the object tracking system was proven to be highly effective during real-world testing. The object tracking algorithm can effectively detect and track the target object, maintaining an average distance of 15cm from the expected position. Moreover, thanks to its high responsiveness, the algorithm can quickly adjust when the object changes direction, ensuring that the trolley stays on track.

Figure 8: IIC standard waveform

3.2 Obstacle Avoidance Subsystem

To further test the accuracy of the HC-SR04 ultrasonic sensor for obstacle detection, we conducted multiple trials with different obstacle sizes. The tests involved placing obstacles of varying dimensions along the path of the trolley and recording the sensor's readings as it moved toward them.

In our tests, we found that the ultrasonic sensor was able to accurately detect obstacles of various sizes, shapes, and textures with consistently low distance error values of less than 0.3 cm. Even when faced with smaller objects such as pebbles or uneven surfaces

like carpeting, the sensor was able to provide precise distance measurements ensuring the safety and efficiency of the trolley's operation.

Moreover, the low power consumption of the sensor allowed us to run these tests multiple times without worrying about draining the battery, making it a reliable and cost-effective solution for our obstacle detection requirements.

Overall, our testing showed that the HC-SR04 ultrasonic sensor was an excellent choice for our application, providing highly accurate obstacle detection capabilities with minimal energy requirements.

Using the ultrasonic sensor, the obstacle avoidance subsystem could identify nearby obstacles and modify the trolley's trajectory accordingly. When an obstacle appeared within 25cm, the trolley would move back by 10cm and turn left approximately 30 degrees to avoid it before proceeding to follow the target object. This maneuver proved highly effective, as the system successfully avoided all obstacles encountered during the testing phase. Additionally, the system consistently maintained a speed of around 25 centimeters per second while being able to adjust and avoid obstacles in less than 0.5 seconds, demonstrating high efficiency and reliability.

3.3 Voice-activated Subsystem

3.3.1 A/D converter verification

To meticulously observe the operational aspects of the WM8731 device, we employ Signaltap as a means of monitoring the clock signal and output. In order to enable its functionality within the I2S slave mode, we meticulously configure the pertinent register settings. Notably, the "lrc" signal serves as an indicator for the Left and Right channels, while the "mclk" represents a clock signal generated via a phase-locked loop (PLL) embedded within the FPGA device specifically for the WM8731 module. Furthermore, the "bclk" signal assumes the role of facilitating data transfer, precisely determined by the formula $f \times W \times 2$, where in f denotes the sample rate and W represents the data length. For the present scenario, our sample rate f is set to 44.1kHz, while the data length W is configured as 16 bits. By carefully scrutinizing the signal emanating from the A/D converter and diligently comparing it with the specifications outlined in the WM8731 documentation, we are able to confidently ascertain the precise adherence to the expected timing characteristics.

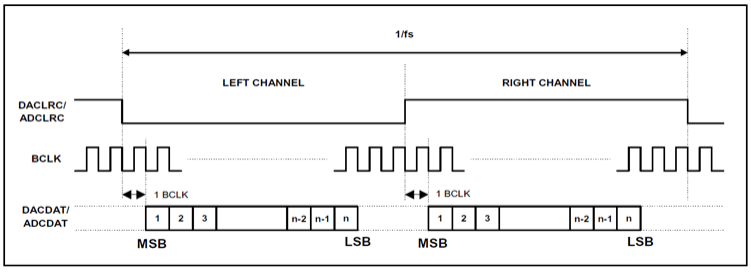

Figure 9: I2S Mode Timing of WM8731[10]

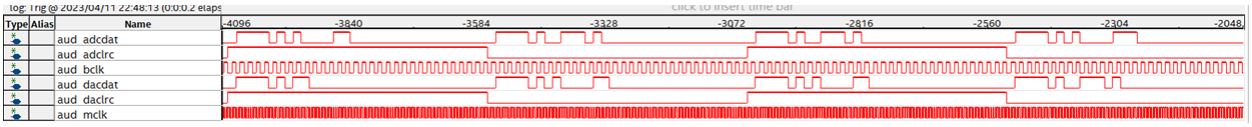

Figure 10: Signal of WM8731 controller, grasped by Signaltap

3.3.2 FFT Verification

During the simulation phase, we conducted comprehensive testing to validate the high-level requirements of our voice-activated subsystem. Specifically, we focused on simulating the waveform of the FFT and CNN classifier components.In order to validate the functionality of the FFT module, we initially employ the modelsim simulation environment. Subsequently, we proceed to program our FPGA device and utilize Signaltap, a debugging tool, to capture and analyze the relevant signals. Within both the simulation and JTAG testing, we diligently inspect vital parameters including "sink_sop," "sink_eop," "sink_valid," "sink_ready," "source_sop," "source_eop," and "source_valid." In figure 12, it can be observed that the end of the input and the start of the output signals are closely aligned. The processing of 1024 input and output data points requires 2048 cycles. With a clock signal frequency of 50 MHz, each iteration of the FFT module takes approximately 0.04096 seconds. The output signals are represented in block floating-point form, which matches the results obtained from our MATLAB simulations.

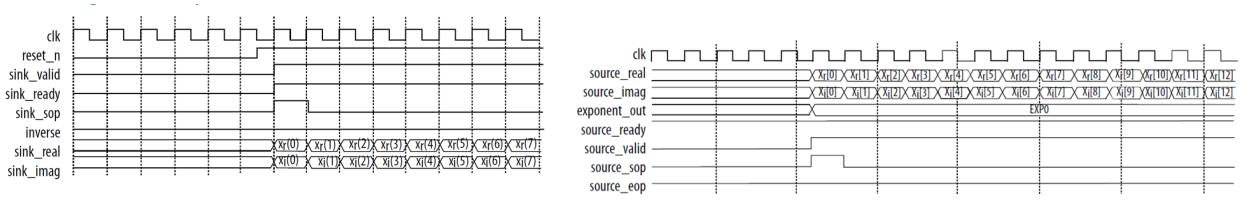

Figure 11: Expected FFT Streaming Data Flow Output[11]

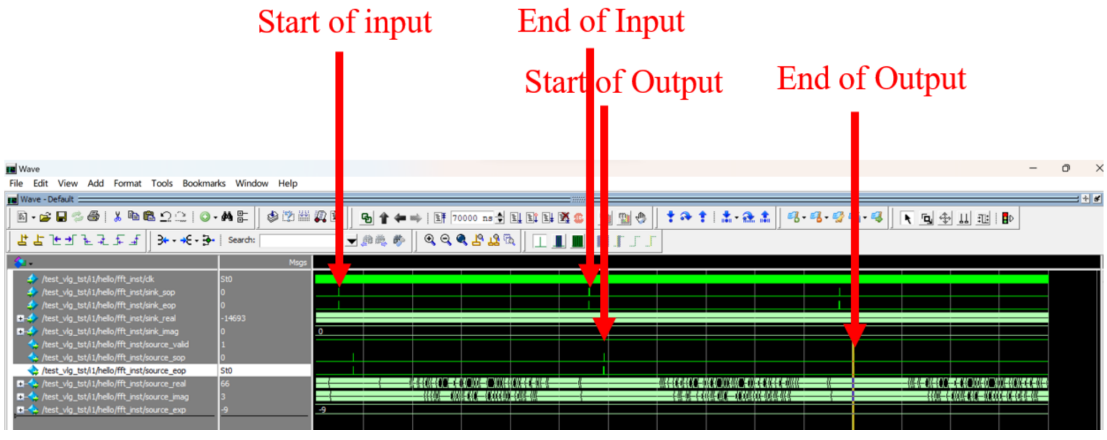

Figure 12: Simulation of FFT

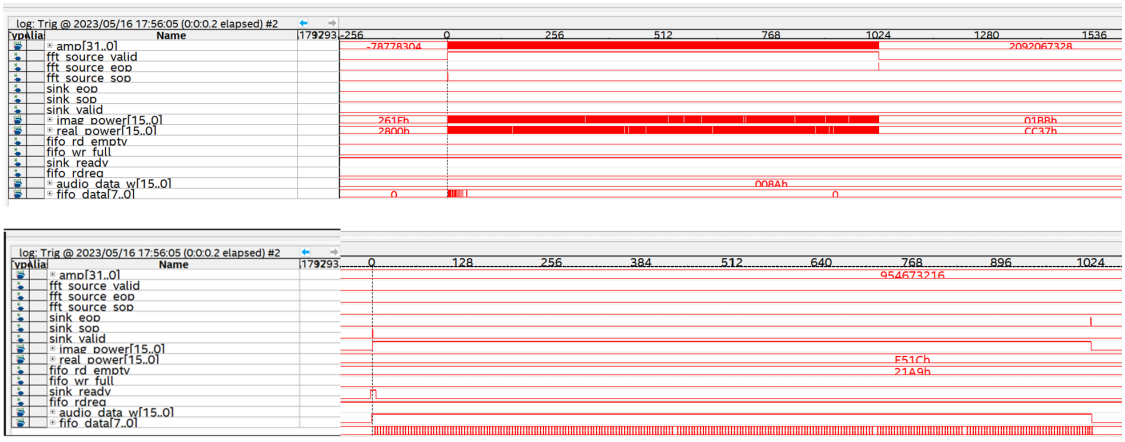

Figure 13: Output and Input Data Flow grasped by Signaltap

3.3.3 CNN Verification

Figure 8 depicts our simulation results for the CNN classifier. Leveraging dataflow, pipeline, and unroll strategies to optimize latency, our inference speed is remarkably fast. The calculation is completed shortly after the input data is processed. The 40x100 input data requires 4000 cycles to finish processing, and the calculation is completed at 4835 cycles, equivalent to 0.0967 seconds. By combining both parts, a single iteration of our voice-activated subsystem operates with a latency of approximately 0.13766 seconds, demonstrating exceptional real-time performance. Hence, our voice-activated subsystem successfully meets the high-level requirement for real-time performance.

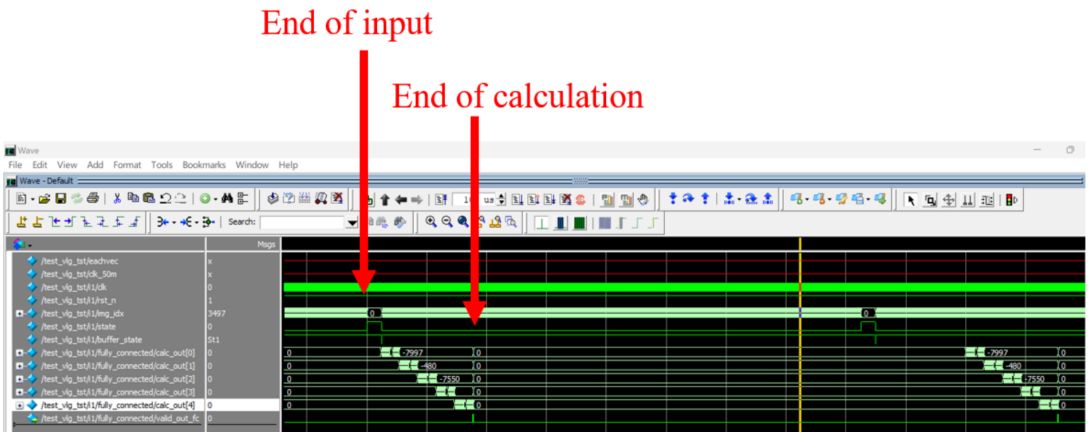

Figure 14: Simulation of CNN classifier

Furthermore, we conducted an assessment of our project's energy consumption. The voice-activated and object tracking subsystems reside on the FPGA, with a total power dissipation of 773.93 mW shown in figure 9. Additionally, we effectively utilized 95% of the logic elements available on the FPGA. Consequently, our project maximizes the utilization of FPGA resources while adhering to our low energy consumption requirement.

Power Analyzer Status	Successful - Tue May 16 18:43:31 2023
Quartus Prime Version	18.1.0 Build 625 09/1...8 SJ Standard Edition
Revision Name	object_detect
Top-level Entity Name	object_detect
Family	Cyclone IV E
Device	EP4CE115F29C7
Power Models	Final
Total Thermal Power Dissipation	773.93 mW
Core Dynamic Thermal Power Dissipation	592.21 mW
Core Static Thermal Power Dissipation	114.25 mW
I/O Thermal Power Dissipation	67.47 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Flow Status	Successful - Tue May 16 18:43:31 2023
Quartus Prime Version	18.1.0 Build 625 09/1...8 SJ Standard Edition
Revision Name	object_detect
Top-level Entity Name	object_detect
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	108,814 / 114,480 (95 %)
Total registers	17937
Total pins	69 / 529 (13 %)
Total virtual pins	0
Total memory bits	1,274,460 / 3,981,312 (32 %)
Embedded Multiplier 9-bit elements	40 / 532 (8 %)
Total PLLs	1 / 4 (25 %)

Figure 15: Power dissipation of FPGA

3.4 Power Subsystem

We require the 7.4v and 12v batteries to supply stable 7.4 V±5% and 12 V±5% power. In verification, we use the oscilloscope to test the performance. For the 12v battery, the mean voltage is 12.379v and in a range [12.31, 12.469], which satisfies our 12 V±5% requirement. Therefore, our power subsystem meets our design requirements.

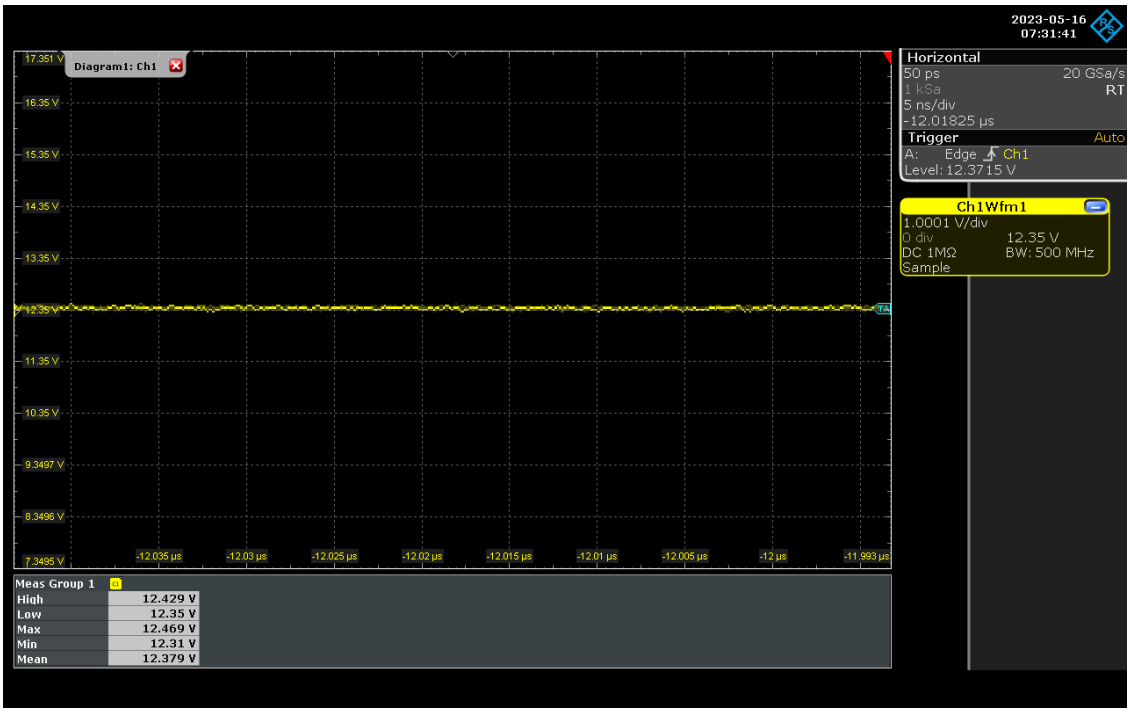

Figure 16: 12v battery on oscilloscope

4 Cost

4.1 Labor:

For each team member, we have estimated a salary of 20 RMB per hour, with each person working 15 hours per week. Since the senior design project spans 9 weeks, the total labor cost can be calculated as follows:

$$Salary = 4 * 20 * 15 * 9 = 10800\,RMB$$

4.2 Components:

The table below presents the components used in our project, including their prices per unit, quantities, and total prices:

Part	Item	Price per unit	Amount	Total Price
Control and Drive Subsystem	Arduino & car	558	1	558
Power Subsystem	12v battery	45.37	1	45.37
Voice-Activated Subsystem	Mini-microphone	138	2	276
Object Tracking Subsystem	Ov2640 camera	133	3	399
Obstacle Avoidance Subsystem	Ultrasonic sensor	25	2	50
Grand Tota				1328.37

4.3 Sum of Total:

By combining the labor cost and component cost, we can determine the total cost of our project:

$$Total\,Cost = Salary + Components = 10800 + 1328.37 = 12128.37\,RMB$$

Considering both the labor and component expenses, the total cost of our project is 12128.37 RMB. These costs were carefully calculated and managed to ensure the successful completion of our project within the allocated budget.

5 Conclusion

5.1 Accomplishments

Throughout the design, implementation, and testing of the object tracking subsystem, significant milestones and accomplishments have been achieved. Key achievements include successful image capture and storage in FPGA BRAM without errors, high-speed image processing with negligible time delay, and reliable tracking algorithm that effectively communicates image processing results to the Arduino for accurate motor control and trolley direction adjustment. Overall, the object tracking subsystem demonstrates effective functionality, successfully realizing all intended functions.

The Obstacle Avoidance Subsystem equipped with the HC-SR04 ultrasonic sensor accurately detected obstacles of various sizes, shapes, and textures with distance error values under 0.3 cm. It operated efficiently on small objects and uneven surfaces, ensuring trolley safety. The sensor's low power consumption enabled multiple tests without battery concerns, offering a reliable and cost-effective obstacle detection solution. The system's obstacle avoidance subsystem used the sensor to identify obstacles within 25 cm, prompting the trolley to move back 10 cm and turn left 30 degrees to avoid them. The maneuver was successful, maintaining a speed of 25 cm/s and reacting in under 0.5 seconds, proving efficiency and reliability.

Regarding our Voice Activated Subsystem, the A/D converter exhibits reliable performance. Through simulation verification, the FFT module and CNN classifier accurately produce the expected outcomes for input analog signals. By employing application and Pipeline techniques, as well as incorporating unroll and dataflow design strategies, we have managed to minimize the latency of the subsystem. Consequently, we have successfully developed a real-time system with low power consumption. Nevertheless, during the implementation on our trolley, we encountered precision limitations that failed to meet the specified requirements. Additionally, the subsystem occasionally produced erroneous results in relation to our instructions.

In terms of the comprehensive functionality of the trolley, the integration of all subsystems exhibits effective synergy, enabling seamless data sharing through the utilization of IIC and UART protocols. The control subsystem adeptly receives and appropriately processes the received data, effectively governing the trolley's movement. Moreover, our trolley successfully meets the stringent criteria of low latency, modularity, and low power consumption at a high level. The sole area that necessitates improvement pertains to the precision requirement of the Voice Activated Subsystem.

5.2 Uncertainties and Future Work

The object tracking subsystem in our project encounters challenges related to target loss, primarily due to limitations imposed by the frame rate and pixel count of the camera. These constraints are ultimately governed by the size of the FPGA BRAM (Block RAM), which can result in the system losing track of the target. To overcome these challenges,

one potential solution is to explore hardware upgrades or reconfiguration. This may involve increasing the available memory, enhancing processing speed, or optimizing related resources to improve image capture and analysis. Furthermore, the integration of additional sensor data or information from external sources, such as ultrasonic sensors, could potentially enhance the accuracy and reliability of the object tracking subsystem.

On the other hand, the voice-activated subsystem faces a predicament of low prediction accuracy. This limitation stems from the resource constraints imposed by the FPGA, preventing the implementation of a deep convolutional neural network. Additionally, the training set used may differ from the real FFT output obtained on the FPGA, necessitating further verification of the combined FFT and CNN classifier module. In future endeavors, it would be beneficial to explore methods for designing a less resource-intensive voice-activated subsystem while concurrently improving prediction accuracy.

5.3 Ethical Considerations

We rigorously adhere to the guidelines outlined in the IEEE Code of Ethics [12].

We uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities

The project entails gathering and storing confidential information, including voice recordings or images. It's crucial to maintain confidentiality and implement suitable measures to safeguard it against unauthorized access or release.[13] We meticulously stored and encrypted the data and deleted them after completing the tests.

In terms of safety and regulatory standards, we adhere to relevant regulations, industry standards, and campus policies. Referring to the ISO 13849-1:2015 Safety of machinery our trolley is designed with safety in mind to prevent accidents or injuries.[14] We ensure that the trolley and its components are designed and manufactured in a safe manner and that appropriate safety features are included, such as emergency stop and obstacle detection sensors. We make sure that the trolley doesn't have any sharp edges or protrusions that could injure people.

We use Lithium-ion batteries in our trolley. Overcharging a Lithium-ion battery can cause it to overheat, which can result in a fire or explosion.[15] To avoid this, we doesn't leave the battery plugged in once it's fully charged. High temperatures and humidity can also damage Lithium-ion batteries and increase the risk of a fire or explosion. So we store them in a cool, dry place, away from direct sunlight.

We treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others.

We strive to ensure this code is upheld by colleagues and co-workers.

References

[1] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Understanding the Limitations of Existing Energy-Efficient Design Approaches for Deep Neural Networks," en,

[2] D. Castells-Rufas, V. Ngo, J. Borrego-Carazo, *et al.*, "A Survey of FPGA-Based Vision Systems for Autonomous Cars," *IEEE Access*, vol. 10, pp. 132 525–132 563, 2022, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3230282.

[3] X. Lu, D. Ren, and S. Yu, "FPGA-based real-time object tracking for mobile robot," in *2010 International Conference on Audio, Language and Image Processing*, Nov. 2010, pp. 1657–1662. DOI: 10.1109/ICALIP.2010.5685091.

[4] *OV2640 datasheet — OmniVision technologies - datasheetspdf.com*. [Online]. Available: https://datasheetspdf.com/datasheet/OV2640.html (visited on 05/13/2023).

[5] *OV7670 Datasheet — OmniVision Technologies - Datasheetspdf.com*. [Online]. Available: https://datasheetspdf.com/datasheet/OV7670.html (visited on 05/13/2023).

[6] K. Kalliojarvi and J. Astola, "Roundoff errors in block-floating-point systems," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 783–790, Apr. 1996, Conference Name: IEEE Transactions on Signal Processing, ISSN: 1941-0476. DOI: 10.1109/78.492531.

[7] "US-100 ultrasonic sensor module temperature compensation distance measuring manufacturers and suppliers china - pricelist - kuongshun electronic," Kuongshun Electronic Limited. (), [Online]. Available: https://www.kuongshun-ks.com/uno/uno-sensor/us-100-ultrasonic-sensor-module-temperature-compen.html (visited on 05/13/2023).

[8] "JSN-SR04t v3.0 waterproof ultrasonic range finder," ProtoSupplies. (), [Online]. Available: https://protosupplies.com/product/jsn-sr04t-v3-0-waterproof-ultrasonic-range-finder/ (visited on 05/13/2023).

[9] P. Marian. "HC-SR04: Datasheet, specs, and more — ElectroSchematics," ElectroSchematics.com. (Jul. 1, 2013), [Online]. Available: https://www.electroschematics.com/hc-sr04-datasheet/ (visited on 05/13/2023).

[10] *WM8731 Datasheet*. Wolfson Microelectronics, PLC. [Online]. Available: http://datasheet.elcodis.com/pdf2/88/68/886817/wm8731.pdf (visited on 05/20/2023).

[11] *FFT IP Core: User Guide*. Intel, Co. [Online]. Available: https://cdrdv2.intel.com/v1/dl/getContent/667064?fileName=ug_fft-683374-667064.pdf.

[12] "IEEE code of ethics." (), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 05/21/2023).

[13] 14:00-17:00, *ISO/IEC 27001:2013*, en, Dec. 2020. [Online]. Available: https://www.iso.org/standard/54534.html (visited on 03/24/2023).

[14] *ISO - ISO 13849-1:2015 - Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design*. [Online]. Available: https://www.iso.org/standard/69883.html (visited on 03/24/2023).

[15] *Safe Practice for Lead Acid and Lithium Batteries*. [Online]. Available: https://courses.engr.illinois.edu/ece445zjui/documents/GeneralBatterySafety.pdf.

Appendix A Requirement and Verification Table

Table 2: System Requirements and Verifications

Requirements	Verification	Verification status (Y or N)
Power Subsystem 7.4V+-5V power supply 12V+-5V power supply	Use a voltage meter to check if the voltage output is equal to the specified value. Use an oscilloscope to check if the voltage output is stable.	Y
Drive subsystem Movement Control The Arduino should send signals to motors to control the speed of motors. The speed of motors varies by the signal and the maximum value of signal is 2500.	Connect Arduino to computer and monitor the port output. Use the controller to set the speed of motors. The output signal should be controlled by a controller. And the maximum value should be 2500.	Y
Object Tracking subsystem Tracking subsystem IIC protocol module	Connect our FPGA to a screen with VGA and check the result. Use modelsim HDL simulator to simulate whether the IIC protocol in FPGA can work properly. Use an oscilloscope to test whether GPIO in the FPGA board works properly. Connect Arduino and test car movement.	Y
Obstacle Avoidance subsystem Ultrasonic Sensor can detect obstacle. The Obstacle Avoidance subsystem should generate suitable output	Put an obstacle around 15cm in front of the trolley, if the trolley enters the Obstacle Avoidance mode, the buzzer will beep 4 times. Put an obstacle on the way of the trolley, test if the trolley starts turning at around 50cm and see if the trolley collides with the obstacles in this process.	Y

Requirements	Verification	Verification status (Y or N)
Voice-Activated subsystem A/D converter FFT module CNN classifier	Use Signaltap to monitor the output of A/D convertor module. Use modelsim simulation and Signaltap together to test FFT module. Test CNN can classify spectrum from FFT module correctly.	Y
Control subsystem IIC protocol and UART protocol Mode switch Conflict handling	Connect Arduino to PC to test values. Send voice command, the control system can switch between tracking Put an obstacle between the object and trolley, trolley should avoid the obstacle first and then track the object.	Y