Journal of Robotics Research (JRR)
Volume 2, Issue 1, Date: 30 - June - 2025

Design, Prototype, and Testing of an Automated Candy Sorting Machine

Tarek Mohammed [1,*], Cody Buzzard [2]
[1,2] Purdue University, Fort Wayne, IN, USA
Email: [1] mohammat@purdue.edu, [2] markmechanicaleng@gmail.com
*Corresponding Author

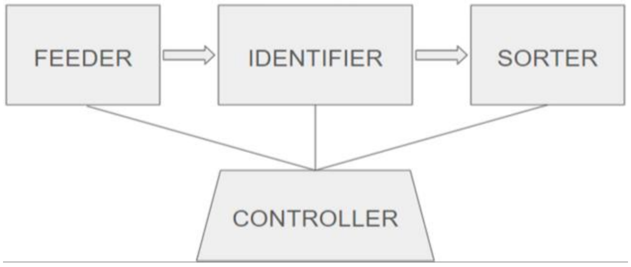
Abstract—**This paper represents a sorting device for chocolate producers to upgrade them to an automated, fast, and cost-effective packaging process from a previous hand packing operation. Higher packaging rates of processes food can be achieved by improving sorting efficiency. The function of the proposed mechatronic system is to take candy pieces from a large central hopper and sort them, by color, into individual bins. The hopper can hold a full-size package of Skittles and the sorter can sort at a rate of up to 100 candy pieces per minute with an accuracy of 84%. With the reciprocal motion, the servo motor can turn through 360° to dispense one skittle to the identification system and return for another. The operation starts with the pocket at 0° to pick up one skittle, rotates 90° to present that skittle to the identification sensor, rotates another 90° to dispense it to the sorting system before rotating back 180° to its starting position. An image sensor is used to determine the color of the candy being presented. When the two data registers are connected and synced with the same clock signals, the 10-bit data from the image sensor reading the colors of the skittles can then be transmitted from the peripheral register to the controller register.**

Keywords—Automated sorting device, Image sensors, Microcontroller, Servo motor, Mechatronics

I. Introduction

The automation of candy packaging has been a key challenge for the confectionery and chocolate industry. The purpose of the wrapping and packaging is not just a means of preserving the product, but also a means of drawing consumers appeals [1], [2]. Prior to automation, packaging numerous candy pieces was a laborious and error-prone task. Each piece was placed in the tray one at a time, which was labor intensive for the packaging of millions of candy pieces. There was also a risk of cross-contamination during manual handling. However, automation has made this process much more efficient. This technological advance allows companies to increase production efficiency, improve product quality and reduce operating costs, while ensuring food safety and the attractive presentation of chocolates, a crucial element in today's marketing [3], [4], [5].

The function of the proposed mechatronic system is to take Skittle pieces from a large central hopper and sort them, by color, into individual bins. The hopper is able to hold a full-size package of Skittles. To achieve this, a mechatronics system is required with several functional modules. A feeding system to take the Skittles, one by one, from the hopper and deliver them to the identifier and from the identifier to the sorter. An identification system with an optical sensor to determine the color of the Skittle [6], [7]. Lastly, a sorter to deliver the Skittle to the appropriate bin. Each of these subsystems are controlled by a central control module and powered by a standard wall power supply [8], [9].



(a)

(b)

Fig. 1. (a) Functional Modules of the Automated Sorting Device; (b) Photograph of the Prototype Sorting Device.

II. Design Configuration

A. Feeder

The task of delivering the Skittles from the hopper to the sensor and from the sensor to the sorter is accomplished by a rotating drum inside a close-fitting housing. The drum has a series of pockets around its circumference each sized to hold a single Skittle and with edges shaped to prevent Skittles jamming in the feed mechanism [10]. As the drum rotates, each pocket passes by the main feed tube leading to the larger hopper and a skittle is allowed to fall in. The drum rotates further, moving the Skittle around to be presented to the sensor, and once the sensor has taken its reading the drum continues around to dispense the Skittle out onto the sorting mechanism.

The rotating feeder drum is driven directly by a servo motor, and it is this motor that determines the main characteristics of the feed system. The speed of the feed system is determined by the speed at which the servo motor rotates the feed drum, and after testing it was determined that the ideal speed to turn without allowing the Skittles to bind in the mechanism was approximately 65° per second. With further development of the feed mechanism, the shape of the feed drum and housing could be optimized to increase this speed.

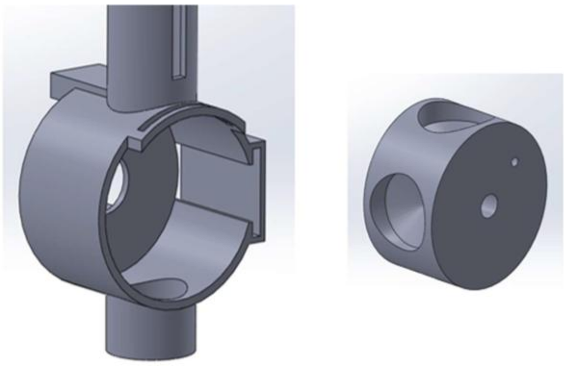


Fig. 2. Feed Mechanism Housing (Left) and Drum (Right)

Originally, the servo motor controlling the feed drum was meant to be one capable of full 360° rotation with a variable speed control. This would have allowed for the drum to operate in a continuous rotation, utilizing all four of the designed pockets for increased efficiency. However, due to supplier issues, a servo with a 180° range of motion had to be utilized. This meant that instead of a continuous rotation for the drum, an oscillating motion had to be used, limiting the speed at which the Skittles could be processed.

B. Identifier

The original sensor specified for the identification system was an Avago ADJD-S311- CR999, which is an integrated CMOS with on board RGB filtering, 10-bit per channel resolution, and an integrated LED for illumination. However, this sensor is of a retired design and was replaced with the Adafruit APDS9960. This sensor is largely similar but has both one significant upside and one significant downside. The benefit of it over the sensor originally specified is the inclusion of a built-in proximity sensor. The original design included a separate proximity sensor, located in the feed housing between the hopper and the identification sensor, that allowed the system to determine when the Skittles were properly loaded in the feed drum and ready for analysis. The inclusion of a proximity sensor with the color sensor allows for a simplification of both the mechanical design and the code.

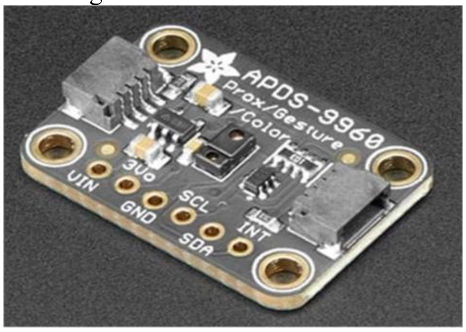


Fig. 3. Adafruit APDS9960 Digital Proximity, Ambient Light, and RGB Sens

However, the downside of the Adafruit sensor replacement is the lack of an integrated illumination LED. The sensor used to detect and evaluate the color of Skittle being presented to it is very sensitive and therefore very sensitive to the color and intensity of ambient light.
However, efforts made to block ambient light with the hardware of the feed mechanism housing reduced the effectiveness of the sensor. So, a balance had to be struck between blocking ambient light to preserve the integrity of the sensor's output while allowing enough light in for the sensor to function properly. In future versions, this could be addressed with an LED integrated into the design to provide light of a consistent and known color and intensity.

C. Sorter

The sorter subsystem is the simplest of the entire machine. It is a simple chute mounted to a servo motor and positioned such that when the servo rotates, the chute is positioned to guide the Skittles to the appropriate bin once they have been identified by the identification system. As with the feed system, the original design called for a servo capable of full 360° rotation. But due to supplier issues, a 180° servo had to be utilized. This results in a much slower, back and forth operation to the sorting mechanism. However, given the slower than anticipated operation of the feeding mechanism, an increase in speed in the sorting mechanism would not result in an increased speed of the machine.

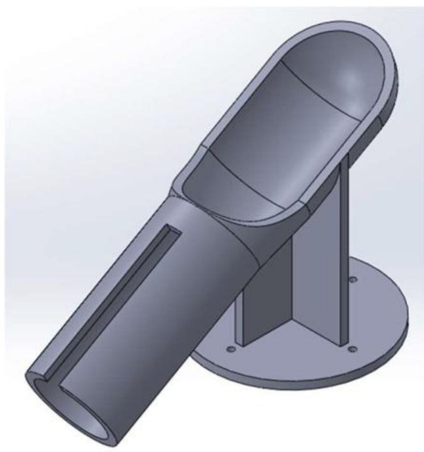


Fig. 4. Sorting mechanism Chute

D. Microcontroller

The controller used was an Arduino Uno and it was programmed to perform several tasks. The first is to control the servo controlling the feeding mechanism. Utilizing an incrementing for loop with an included delay, the Arduino limits the speed of the rotation drum to prevent the Skittles being fed in from binding as the drum moves back and forth. The second and main task of the Arduino is to receive the information from the identification system and determine which bin the Skittle needs to be deposited into. The Arduino communicates with the sensor via I2C and receives four data points when the sensor is polled. 3 16-bit integers corresponding to the intensity of the red, green, and blue light the sensor detects as well as a fourth 16-bit integer corresponding to the overall intensity of the light. These values are then compared to values obtained during calibration and the color of the Skittle in question is determined. The Arduino then directs the sorting mechanism to deposit the Skittle in the correct bin.

E. Power & Driving System

1. Motor Analysis

Given the overall lightweight nature of the driven systems in the Skittle sorter mechanism, the torque requirements for the motors were overall quite low. The moment of inertia for the rotating parts was determined by analysis within SolidWorks, and was found to be 0.052 lbs.*in^2 for the feed chute utilized in the sorting system and 0.035 lbs.*in^2 for the rotating drum used in the feeder system. It was determined, therefore, that the motor selection was to be determined by speed and range of motion offered.

Speed of the servo motors specifically was not a functional requirement of the system, but the speed of the servos chosen would govern the overall speed of the mechanism. It was determined early in the testing that the main bottleneck in the speed of the overall mechanism was in the feed mechanism and was not due to the speed at which the motor could turn the drum mechanism, but rather the speed at which the drum feed mechanism could run without jamming. And so, the speed of the motor had to be slowed significantly in the code. Therefore, a significant optimisation of the feed mechanism's hardware would be required before the motor's speed emerged as a limiting factor.

2. Motor Selection

The S35-STD GWS Continuous Rotation Servo was selected initially due to its ability to rotate continuously, which was necessary for the initial concept of operation for the feeder drum. The rotation is controlled by pulse width modulation (PWM) which changes the speed and angle of the output based on the change in pulse widths supplied by the microcontroller.

Fig. 5. Picture of original servo

General specifications

Speed @ 6V:	0.14 sec/60°
Stall torque @ 6V:	39.2 oz·in
Speed @ 4.8V:	0.16 sec/60°
Stall torque @ 4.8V:	35 oz·in
Hardware included?:	Y
Lead length:	11 in

Fig. 6. Original Servo specifications

3. Speed

The speed is 0.14 sec / 60 deg and would take 0.42 seconds to rotate the 180 degrees needed to move the Skittle from the inlet to the outlet of the feeder. This means the mechanism can theoretically discharge at a rate of 142 SPM.

$$Theoretical\ Discharge\ Rate = \frac{60\ seconds}{discharge\ speed} = \frac{60}{0.42} = 142\ SPM$$

However, due to supplier issues previously discussed, the initially specified servo was unavailable, and an alternative had to be chosen. The Tower Pro SG51R Micro Servo was selected as a replacement due to its small form factor and its availability.

Fig. 7. Picture of servo replacement

Operating Voltage:	4.8V
Stall Torque (@4.8v):	0.6kg/cm
Operation Speed (@4.8v):	0.1sec/60°
Weight: 5g	5g

Fig. 8. Servo Specifications

Given that this micro servo has a higher angular speed at a similar operating voltage, it could theoretically handle a higher throughput of skittles. However, as previously discussed, the main bottleneck in the system is in the hardware of the dispensing system, rather than in the motor speed. So, a significant amount of optimization would be required to take full advantage of the increased speed.

F. Sensing System

The sensor originally specified for this system was the Avago ADJD-S311-CR999 (Fig. 6). This is a four-channel sensor with a dedicated channel for red, green, and blue light with a fourth channel for clear light. It featured a resolution of 10 bits per channel and an integrated LED for the illumination of samples. It also featured on board analog to digital conversion to alleviate processing load on the main control module and eliminate the need for an external ADC and communicated with the main control board via a 2-wire serial interface.

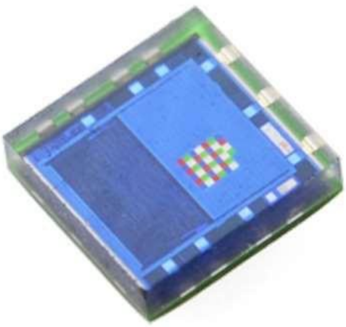


Fig. 9. Image Sensor

However, this particular sensor is no longer available from the manufacturer. Its suggested replacement is the Adafruit APDS9960 (Fig. 10). Based on a similar architecture, this sensor operates in largely the same way. It utilizes the same sensor, and therefore has the same four channel, 10 bit per channel color sensing, and communicates with the controller via a similar 12C connection. However, the notable differences between this sensor and the original specifications are consequential to the overall operation of the mechanism. This sensor has an integrated proximity detector, which allows for the elimination of the discrete component in the original design. However, the more impactful change is the omission of an integrated LED for automatic illumination. The sensor used to determine the color of the skittle being presented is sensitive enough that the ambient light outside of the machine has a significant effect on the reading produced.

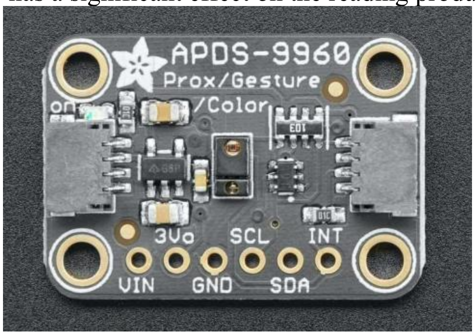


Fig. 10. Replacement Image Sensor

However, when the outer housing of the feed mechanism was redesigned in such a way as to exclude external light, there was not enough illumination to generate an accurate reading. A notable improvement to the reliability of the mechanism could be achieved with the inclusion of a separated, constant source of illumination within the identification mechanism.

G. Control System

The controller chosen was an Arduino Uno R3, a microcontroller based on an ATmega328P. It has 14 digital IO pins, 6 analog IO pins, is capable of serial communication via UART, I2C, or SPI protocols, and runs on nominal 5 volts. It has a storage capacity of 32 KB of flash memory, which was more than sufficient to hold the program written for the skittle sorter, which took up roughly 8 KB.

The Arduino Uno (AU) controls both the servo motors using the servo library using loops shown to tell the motor which position to move to. To turn the servo head 180 degrees, the code is used

```
void loop() {
 for (pos = 0; pos <= 180; pos += 1) { // goes
from 0 degrees to 180 degrees in steps of 1 degree
myservo.write(pos); // tell servo to go to
position in variable 'pos'
       delay(15);      // waits 15ms for the servo
to reach the position
 }
```

The AU converts the position in degrees into pulse widths and sends the pulses via the control wire to the servo. The servo's processor reads the pulse widths sent by the AU and moves the servo arm to a position based on the pulse width given. For example, a Futaba servo converts the pulse width in milliseconds to the different positions that it has to move, shown in Fig. 11.

The pulses have a 5V peak and 0V ground Fig. 12.

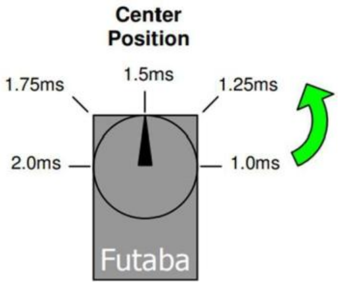


Fig. 11. Pulse width timing [1]

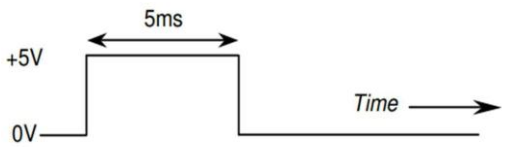


Fig. 12. Sample pulse width of 5ms sent by the AU vs Servo Position [1]

III. DATA CONVERSION AND RESULTS

A. *Image Sensor Data Transmission and Processing*

The ADAFRUIT APD9960 image sensor uses I2C to communicate to the AU and 2 pins are required to interface it. It is a serial bus system capable of addressing up to 1008 devices on only two wires. The downside is that it is half duplex which means it has to read and write using only 1 wire as a result.

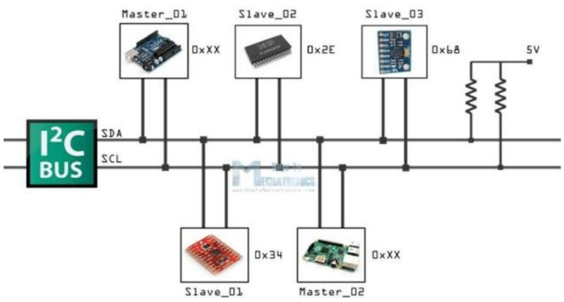


Fig. 13. Sample I2C Bus Connecting Controllers and Peripherals [2]

The two lines needed for the controller to communicate with the devices on the bus are Serial
Data (SDA) which carries the data signal and Serial Clock (SCL) which carries the clock pulses. The clock signal is set by the controller and synchronizes both data registers on the device and controller, so the bit information is stored in the right register.
When the two data registers are connected and synced with the same clock signals, the 10-bit data from the image sensor reading the colors of the skittles can then be transmitted from the peripheral register to the controller register.

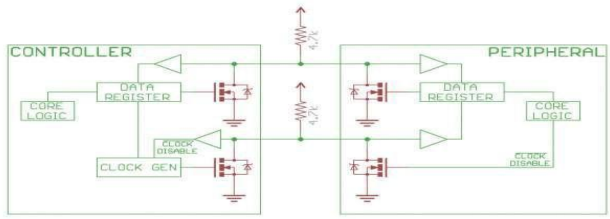


Fig. 14. Schematics for SDA and SCL lines linking the data registers and clock signals

This is done with a handshake, letting the peripheral know that the controller would like to read a 10-bit data with the bits 11110. Then the 10 bits label in the figure below (Fig 12) A0-A9 is read by the controller as 1's and 0's. The read/write (R/W) bit shown after bit A8 is transmitted lets the peripheral know if the controller is reading or writing from the register, 1 means the controller is reading and 0 means the controller is writing data. The acknowledge (ACK) bit shown after the R/W and A0 bits is for the peripheral to send a bit back, acknowledging that it has received the previous 8 bits. Once the controller sends 8 bits it waits for a response from the peripheral to acknowledge that it has received the data without error before continuing to send another 8 bits as shown below. This helps to reduce erroneous data being sent as the controller can try to resend/read the 8 bits of data if the signal gets interrupted or corrupted from environmental noise/conditions.

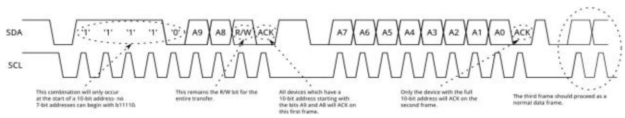


Fig. 15. 10 bit read/write protocol for I2C devices [3]

Next, the Arduino reads the values for the 3 colors and C (light value without processing) and stores it in the variables r,g,b,c to be compared with calibrated values with the following code. This is done with the built in i2c_readbyte function. The low and high refers to low and high sensitivity readings of color from the sensor.

```
// RGB color results, 10bit results

i2c_readbyte(I2C_ADDR,REG_DATA_RED_LO,&r_lo);
i2c_readbyte(I2C_ADDR,REG_DATA_RED_HI,&r_hi); r = r_lo + (r_hi<<8);
i2c_readbyte(I2C_ADDR,REG_DATA_GREEN_LO,&g_lo);
i2c_readbyte(I2C_ADDR,REG_DATA_GREEN_HI,&g_hi); g = g_lo + (g_hi<<8);
i2c_readbyte(I2C_ADDR,REG_DATA_BLUE_LO,&b_lo);
i2c_readbyte(I2C_ADDR,REG_DATA_BLUE_HI,&b_hi); b = b_lo + (b_hi<<8);
i2c_readbyte(I2C_ADDR,REG_DATA_CLEAR_LO,&c_lo);
i2c_readbyte(I2C_ADDR,REG_DATA_CLEAR_HI,&c_hi); c = c_lo + (c_hi<<8);
```

B. *Prototype Operation*

Overall, the Skittle sorting machine works as intended, albeit significantly slower than initially anticipated, averaging 25 Skittles per minute. This lack of speed was due primarily to the last-minute change in the servo discussed in section 1 above, necessitating a change in the operating principle of the feed mechanism. Instead of a continuous rotation a 360° servo would allow for, the use of the available 180° servo meant a reciprocal motion had to be utilized instead, at a significant cost in efficiency.

With the reciprocal motion, the servo must turn through 360° to dispense one skittle to the identification system and return for another. That is, it starts with the pocket at 0° to pick up one skittle, rotates 90° to present that skittle to the identification sensor, rotates another 90° to dispense it to the sorting system before rotating back 180° to its starting position. A continuous rotation servo, however, would only need to rotate in constant 90° steps, dispensing a skittle from the pocket on every quadrant of the drum with each step.
The speed of the machine was further reduced by the lack of uniformity in the overall shape and size of individual skittles. The wide variation in the length, circumference, and flatness of the individual skittles meant that the feed mechanism had to be run much slower than the maximum speed the servos would allow to prevent jamming.

Overall, the machine reaches its target accuracy when used in the ambient light in which it was calibrated. In optimal lighting conditions, the machine performed with an accuracy of 84% over 300 skittles. However, due to the limitations of the design as discussed above, the accuracy falls off quickly when the ambient light changes in any major way from that in which its calibration values were determined. Some simple but significant changes to the identification system, discussed in the next section, would need to be made to increase the reliability of the machine.

IV. Conclusions

The easiest opportunity for improvement would be a rebuild to include the 360° servos originally specified in the design. While the replacement 180° servos allowed the machine to function and its concept to be proven, they severely limit its overall performance. The target speed of 100 skittles per minute would be difficult to reach with the reciprocal motion of the feed system. But rebuilding the system to have the correct, continuous motion would drastically increase the speed.

The speed could further be improved with an optimization of the physical shape of the rotary feed drum and the housing it rotates inside of. As discussed, individual skittles can vary drastically in shape and size, so creating a system that can reliably feed them without jamming is difficult. However, with further iteration, it should be possible to shape the system in such a way as to allow the feed system to run much faster and hit the 100 skittles per minute goal.

While the accuracy of the machine meets its goal when used in optimal light, it leaves much to be desired in terms of repeatability. As discussed previously, the identification sensor is very sensitive to ambient, but the lack of an integrated light source means that any attempt to block out ambient light leaves too little light for the sensor to function. This could easily be solved with a small light integrated into the identification housing to provide light to the sensor or a known, stable color, temperature, and intensity. Then the housing itself could be redesigned to exclude more ambient light, giving the machine much less source of variation in its readings.

References

[1] *Servo Intro*. (n.d.). Basic X. Retrieved March 14, 2024, from https://web.archive.org/web/20170819051925/http://www.basicx.com/Products/robotbook/servo%20intro.pdf

[2] Codeavour. (2023, April 28). *I2C communication*. STEMpedia Education. Retrieved April 10, 2024, from https://ai.thestempedia.com/docs/evive/evive-technical-specifications/arduino-core-interface/i2c-communication/

[3] *I2C*. (n.d.). SparkFun Learn. https://learn.sparkfun.com/tutorials/i2c/all

[4] Willem. (2017, February 5). M&Ms and Skittles sorting machine – Willem Pennings. https://willempennings.nl/mms-and-skittles-sorting-machine/

[5] Skittles M&M's Sorting Machine - ivc wiki. (n.d.-b). https://beta.ivc.no/wiki/index.php/Skittles_M&M%27s_Sorting_Machine

[6] Otero-Herrera, A., Fuentes-Gaviria, L., Pérez-Cervera, C., & Andrade-Pizarro, R. (2025). Development of edible films based on sweet potato (Ipomoea batatas) starch and their application in candy packaging. *International Journal of Biological Macromolecules*, 140031.

[7] Nita, N., Ramli, A. R., Budiardjo, G. T. H. W., & Muhpidah, M. (2024). Effect of packaging type on the quality of red ginger soft-candy. In *BIO Web of Conferences* (Vol. 96, p. 01033). EDP Sciences.

[8] K. Kirkpatrick, R. Ashley, M. Hoffer, and T. Mohammed, "A Robust Environmental Chamber for Testing the Durability of Outdoor Pumps and Coatings," J. Robot. Res., vol. 2, no. 1, 2025.

[9] J. A. Amparado, "The Capability of AI to Detect Problems in Electrical Systems," J. Electr. Electron. Res., vol. 2, no. 1, 2025.

[10] P. Ray, R. Rawal, B. Jajal, and A. Z. Rizvi, "Prediction of Heart Diseases Using Cooperative AI Approach," J. Electr. Electron. Res., vol. 01, no. 01, 2024.