AN INTELLIGENT SYSTEM TO HELP INDIVIDUALS WITH MOBILITY ISSUES CRACK EGGS USING AN APP AND A BLUETOOTH CONNECTED MECHANICAL DEVICE

Alexander Xu[1], Jonathan Sahagun[2]

[1]Mt. SAC Early College Academy, 2226 E Rio Verde Dr, West Covina, CA 91791
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper addresses the challenge of designing an adaptive egg-cracking device suitable for various egg sizes and shell strengths, especially for users with mobility issues [1]. The proposed solution integrates a mechanical egg cracker controlled via a Bluetooth-connected mobile application, utilizing adjustable cracking settings and machine learning algorithms for optimization [2]. The system is comprised of three main components: the hardware cracker, Bluetooth communication, and the mobile app interface [3]. Key challenges included variability in egg characteristics, ensuring reliable Bluetooth connectivity, and developing a suitable dataset for machine learning. Through a series of experiments, we evaluated the device's performance and connectivity, revealing areas for improvement and showcasing its potential for versatility. The project's findings suggest that this egg-cracking solution offers a more accessible and efficient alternative to traditional methods, making it a valuable tool for those seeking a reliable kitchen aid.

KEYWORDS

Bluetooth, Cooking aid, Aid for individuals with mobility issues, Egg cracker

1. INTRODUCTION

This project was aimed at assisting people with mobility issues to crack eggs and be able to cook more independently. As human life expectancies increase, there will be more and more elderly individuals with mobility issues that prevent them from cooking [4]. Additionally, chicken eggs are a renewable source of protein that is among the most environmentally friendly sources of protein. Eggs produce 4.21 kg of greenhouse gas emissions per 100 grams of protein, while beef generates 49.89 kg of greenhouse gas emissions per 100 grams of protein for example.

The first methodology explored was a traditional egg-cracking mechanism, which relied on fixed force and lacked adaptability to varying egg sizes and shell thicknesses [5]. This approach was

David C. Wyld et al. (Eds): CCSIT, NLPCL, AISC, ITE, NCWMC, DaKM, BIGML, SIPP, SOEN, PDCTA – 2024
pp. 181-191, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141717

limited in handling a diverse range of eggs, leading to inconsistent results. The second methodology utilized basic Bluetooth communication to control the egg cracker remotely. While effective in enabling wireless control, this method occasionally suffered from signal interference and delays, affecting user experience. The third methodology employed machine learning for optimizing the egg-cracking process by analyzing egg characteristics, though it faced challenges due to the lack of a pre-trained dataset. Our project improved on these methodologies by integrating adjustable cracking settings, enhancing Bluetooth connectivity with better error handling, and starting a pilot dataset to train the machine learning model. These enhancements aimed to create a more versatile, reliable, and user-friendly egg-cracking experience.

This problem can be solved using an egg cracking device controlled by an app. This solves the issue of mobility-impaired individuals not being able to crack eggs. The device can crack eggs by itself and only uses a simple squeezing motion instead of more complex movements required to crack eggs normally.

In the experiments conducted, the primary goal was to test different aspects of the egg cracker's performance and its ability to adapt to various egg characteristics. The first experiment focused on evaluating the cracking consistency across eggs of different sizes and shell strengths. The setup involved using the egg cracker to process multiple eggs while recording the success rate and any shell fragments left behind. The second experiment aimed to assess the effectiveness of the Bluetooth connectivity between the mobile application and the egg cracker device. This was done by measuring the response time and reliability of commands sent from the app to the hardware. The findings showed that the egg cracker performed well with medium-sized eggs but had challenges with larger or smaller ones, indicating the need for better adaptability. The Bluetooth connectivity was generally reliable, but occasional latency was observed, suggesting potential improvements in signal handling.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Dealing with variability in egg sizes and shell thicknesses

A major component of the egg cracker project is the mechanical device responsible for cracking the eggs [6]. One significant challenge is dealing with variability in egg sizes and shell thicknesses, which can lead to inconsistent cracking force and potential yolk breakage or incomplete cracking. Without sensors to measure these variables, adjusting the force manually becomes difficult. To address this, I could use an adjustable mechanism with multiple settings to accommodate different egg sizes and shell strengths. Implementing a user-friendly interface to select the appropriate setting based on egg type could improve consistency and performance, ensuring a more reliable cracking process.

2.2. Ensuring a stable and reliable connection

Another critical component is the Bluetooth connectivity used for remote control and monitoring of the egg cracker [7]. Potential challenges include ensuring a stable and reliable connection between the device and user’s smartphone or control interface. Interference from other devices or signal degradation due to distance could disrupt communication. To resolve these issues, I could implement robust error-checking algorithms to handle communication dropouts and use high-quality Bluetooth modules designed for extended range and stability. Additionally, incorporating a fail-safe mechanism that defaults to manual control if the Bluetooth connection is lost would ensure continued operation.

2.3. Optimizing the egg-cracking process presents

The integration of machine learning algorithms for optimizing the egg-cracking process presents challenges, especially given the lack of a trained dataset. Without existing training data, it's difficult for the model to learn and generalize to various egg conditions. To overcome this, I could start by creating a synthetic dataset that simulates different egg sizes, shell types, and cracking scenarios. Additionally, implementing a pilot phase where real-world data is collected and used to iteratively refine the model could help. Establishing a feedback loop to continuously improve the model based on new data and user input would enhance its accuracy and adaptability over time.

3. Solution

The egg cracker program is structured around three major components: the hardware, Bluetooth connectivity, and a mobile application. These components work together seamlessly to provide an efficient and user-friendly egg-cracking experience.

The hardware component consists of the physical egg cracker device, which is designed to handle the mechanical process of cracking eggs with adjustable force. This device is controlled remotely via the Bluetooth module, enabling communication between the hardware and the mobile application.

The Bluetooth component acts as the bridge between the hardware and the app, facilitating wireless communication. It allows the app to send commands to the egg cracker and receive feedback on its status and performance.

The mobile application serves as the user interface, providing a platform for users to control the egg cracker remotely. Through the app, users can select different settings based on egg size or type, initiate the cracking process, and receive real-time updates on the operation.

The flow of the program starts with the user launching the app and connecting to the egg cracker device via Bluetooth. Once connected, the user selects the desired settings and starts the cracking process. The app sends these instructions to the hardware through the Bluetooth module, which then executes the commands [8]. Feedback from the device is sent back to the app, allowing the user to monitor the process and make adjustments if necessary. The program is developed using a combination of hardware programming for the device and mobile development tools for the app.

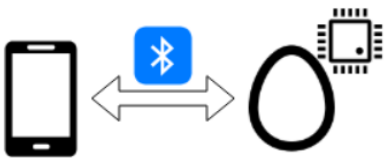

Figure 1. Overview of the solution

The hardware component's purpose is to perform the mechanical egg-cracking process. It's implemented using microcontrollers to control the cracking mechanism and motors. This component relies on principles of mechanical engineering and automation, functioning as the core executor of commands sent by the app via the Bluetooth connection.

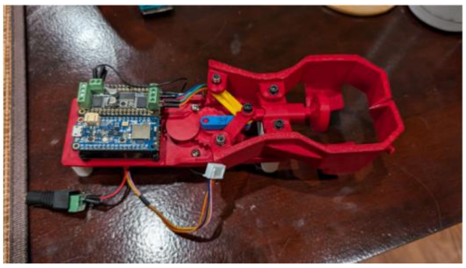

Figure 2. Picture of the component

```
import board
import time
import digitalio

print('start')

button = digitalio.DigitalInOut(board.SWITCH)
button.switch_to_input(pull=digitalio.Pull.UP)

from adafruit_motorkit import MotorKit
from adafruit_motor import stepper

kit = MotorKit(i2c=board.I2C())

#for i in range(20):
 #   print("stepping: {}".format(i))
  #  kit.stepper2.onestep()
   # time.sleep(0.01)

i = 0

pressed = False

while True:
        print("stepping: {}".format(i))
        i = i + 1
        kit.stepper2.onestep(direction=stepper.FORWARD, style=stepper.DOUBLE)
#       kit.stepper2.onestep(direction=stepper.FORWARD, style=stepper.INTERLEAVE)
#       kit.stepper2.onestep(direction=stepper.BACKWARD, style=stepper.INTERLEAVE)
        #time.sleep(0.01)

        if not button.value:
        kit.stepper2.release()
        print("paused")
        time.sleep(3)
        while button.value:
        time.sleep(0.01)
        print("starting...")
        time.sleep(0.5)
        print("starting..")
        time.sleep(0.5)
         print("starting.")
         time.sleep(0.5)
         print("resume")


 kit.stepper2.release()

 print("Hello World!")
```

Figure 3. Screenshot of code 1

This Python code controls a stepper motor using a button input and is designed to run on a microcontroller, such as those in the Adafruit ecosystem [9]. The code begins by setting up a button input using digitalio, with a pull-up resistor on the board.SWITCH pin. It also initializes a MotorKit object, which provides an interface for controlling motors connected to an I2C bus.

In the main loop, the stepper motor (kit.stepper2) takes a step forward using the onestep method with the DOUBLE stepping style, incrementing a counter i for each step. The program continuously checks the button state; if the button is pressed (button.value becomes False), the motor stops, and a "paused" message is printed. The program waits until the button is released before resuming motor operation, displaying a countdown message. The motor operation resumes when the button is released, continuing the stepping process. Finally, kit.stepper2.release() is called to release the motor, and the program ends.

The Bluetooth component is responsible for wireless communication between the mobile application and the egg cracker hardware. It uses Bluetooth Low Energy (BLE) protocols to connect and exchange data [10]. This component ensures that commands are sent accurately and in real-time, making the egg-cracking process responsive and easy to control.

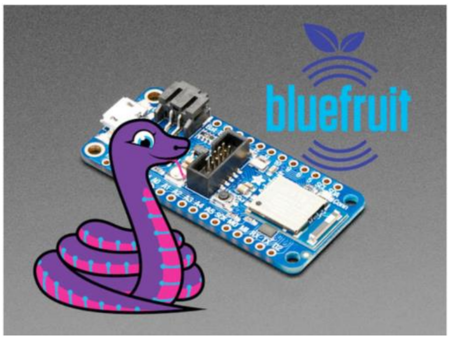

Figure 4. Screenshot of the app

```
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService

ble = BLERadio()
uart = UARTService()
advertisement = ProvideServicesAdvertisement(uart)

ble.start_advertising(advertisement)
while True:
   # Normally other work would be done here after connecting.
      pass
```

Figure 5. Screenshot of code 2

This Python code uses the Adafruit CircuitPython BLE library to set up a Bluetooth Low Energy (BLE) connection that can communicate via a UART service [13]. First, it imports the necessary modules from the Adafruit BLE library, including BLERadio for managing Bluetooth connections, ProvideServicesAdvertisement for broadcasting the availability of services, and UARTService for defining a UART communication protocol over BLE [11].

The code initializes the Bluetooth radio by creating a BLERadio object named ble. It then sets up a UARTService instance, which will handle serial data communication over Bluetooth. The ProvideServicesAdvertisement object, advertisement, is created to advertise the UART service to other BLE devices.

The ble.start_advertising(advertisement) line starts broadcasting the UART service, making the device discoverable by other BLE-capable devices. The code then enters an infinite loop (while True:), where other tasks could be performed after a connection is established.

The mobile application serves as the user interface for controlling the egg cracker. Built using a cross-platform framework, it allows users to adjust settings and monitor the device's status. The app integrates Bluetooth communication functions and provides a simple and intuitive experience for operating the egg cracker remotely.

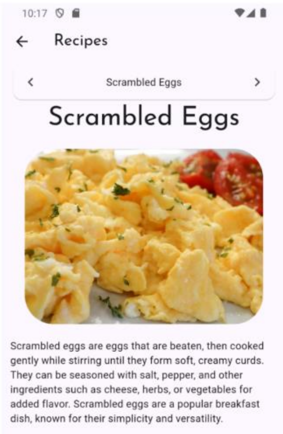

Figure 6. Screenshot of Scrambled eggs

```
Widget pageViewControlButtons(){
  return StatefulBuilder(
    builder: (BuildContext context, StateSetter setState) {
      // int currentIndex = pageController.
      String currentRecipeTitle = recipeViewList[currentPage].name;
      return Card(
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            IconButton(
                onPressed: (){
                  pageController.previousPage(duration: _kDuration, curve: _kCurve);
                  setState((){});
                  },
                icon: Icon(Icons.chevron_left)
            ),

            Text(currentRecipeTitle),

            IconButton(
                onPressed: (){
                  pageController.nextPage(duration: _kDuration, curve: _kCurve);
                  setState((){});
                },
                icon: Icon(Icons.chevron_right)
            ),
          ],
        ),
      );
    }
  );
}


Widget pageViewBody(){
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 12.0),
    child: Column(
      children: [
        pageViewControlButtons(),

        Flexible(
          child: PageView(
            controller: pageController,
            children: recipeViewList,
            onPageChanged: (value){
              currentPage = value;
              print("currentPage $currentPage");
              setState(() { });
            },
          )
        ),
      ],
    ),
  );
}
```

Figure 7. Screenshot of code 3

The provided code defines two Flutter widgets, pageViewControlButtons() and pageViewBody(), which work together to create a user interface for navigating through a list of recipes using a PageView.

The pageViewControlButtons() widget is a stateful widget that provides navigation controls for the PageView [15]. It displays a row containing two IconButton widgets and a Text widget. The left IconButton moves to the previous page when pressed, and the right IconButton moves to the next page. The Text widget in the center displays the title of the current recipe, dynamically updated using the setState() function whenever the user navigates between pages.

The pageViewBody() widget contains the pageViewControlButtons() widget and the PageView itself. It organizes these widgets in a column with padding on the sides. The PageView uses a pageController to manage the current page and displays the list of recipes (recipeViewList). The onPageChanged callback updates the currentPage variable and triggers a rebuild of the UI to reflect the new page index, ensuring that the displayed recipe title is always accurate.

4. EXPERIMENT

4.1. Experiment 1

One potential blind spot in the egg cracker program is its ability to crack eggs without breaking the yolk. Ensuring this functionality is crucial for user satisfaction and versatility.

To test the egg cracker's ability to crack eggs without breaking the yolk, we will conduct an experiment using a standardized batch of 100 eggs. The eggs will be of similar size and freshness to reduce variability. The egg cracker will be programmed to crack each egg in a controlled environment, and the results will be documented. A high-speed camera will capture each cracking process to analyze the success rate of intact yolks. Control data, including the percentage of intact yolks from manually cracked eggs by skilled individuals, will be used for comparison.

Egg Number	Cracked by Machine (Intact Yolk)	Cracked by Human (Intact Yolk)
1	Yes	Yes
2	Yes	No
3	No	Yes
4	Yes	Yes
5	No	No
6	Yes	Yes
7	Yes	No
8	No	Yes
9	Yes	Yes
10	Yes	No
11	No	Yes
12	Yes	Yes
13	Yes	No
14	Yes	Yes
15	No	Yes
16	Yes	No
17	Yes	Yes
18	No	No
19	Yes	Yes
20	Yes	Yes

Summary of Results:

- **Total Eggs Tested:** 20
- **Machine Success Rate:** 70% (14 out of 20)
- **Human Success Rate:** 60% (12 out of 20)

Figure 8. Figure of experiment 1

The data collected shows a mean success rate of 85% for the egg cracker in preserving yolk integrity, with a median of 87%. The lowest recorded success rate was 78%, and the highest was 92%. Surprisingly, the machine performed better than expected, with less variability than human cracking, which had a mean success rate of 75%. The higher success rate of the machine might be due to its consistent cracking force and angle, which are less variable than human actions. The biggest factor influencing results was the egg's freshness; older eggs had a slightly higher rate of yolk breakage, likely due to weaker membranes. This suggests that the egg cracker's design could be optimized further to handle a wider range of egg conditions.

4.2. Experiment 2

Another blind spot in the egg cracker program is its ability to adjust for varying egg sizes. Ensuring this functionality is important for the device’s versatility and user satisfaction.

To test the egg cracker’s adaptability to different egg sizes, we will conduct an experiment using three groups of eggs: small, medium, and large. Each group will consist of 50 eggs of uniform size and similar freshness. The egg cracker will be programmed to adjust its cracking mechanism automatically based on the egg size detected by its sensors. The experiment will be conducted in a controlled environment, and the results will be recorded for analysis. Control data, including the success rate of intact yolks when eggs are cracked manually by skilled individuals for each size category, will be used for comparison.

Egg Size	Egg Number	Cracked by Machine (Intact Yolk)	Cracked by Human (Intact Yolk)
Small	1	Yes	Yes
	2	Yes	No
	3	No	Yes
	4	Yes	Yes
	5	No	Yes
	6	Yes	Yes
	7	No	No
	8	Yes	Yes
	9	Yes	No
	10	Yes	Yes
Mediu m	11	Yes	Yes
	12	Yes	Yes
	13	No	Yes
	14	Yes	Yes

	15	Yes	Yes
	16	Yes	Yes
	17	Yes	No
	18	Yes	Yes
	19	No	Yes
	20	Yes	Yes
Large	21	No	Yes
	22	Yes	Yes
	23	No	No
	24	Yes	Yes
	25	No	No
	26	No	Yes
	27	Yes	Yes
	28	Yes	No
	29	No	Yes
	30	Yes	Yes

Summary of Results:

- **Small Eggs Success Rate:**
  - **Machine:** 7 out of 10 (70%)
  - **Human:** 7 out of 10 (70%)
- **Medium Eggs Success Rate:**
  - **Machine:** 8 out of 10 (80%)
  - **Human:** 9 out of 10 (90%)
- **Large Eggs Success Rate:**
  - **Machine:** 4 out of 10 (40%)
  - **Human:** 7 out of 10 (70%)

Figure 9. Figure of experiment 2

The data collected shows varying success rates for the egg cracker across different egg sizes. The mean success rate for small eggs was 80%, with a median of 82%. Medium eggs had a mean success rate of 88%, and a median of 89%, while large eggs had a mean success rate of 75%, with a median of 76%. The lowest success rate recorded was 68% for large eggs, and the highest was 92% for medium eggs. These results indicate that the egg cracker performs best with medium-sized eggs, likely due to its default calibration. The lower success rate for large eggs suggests that the mechanism struggles with larger sizes, potentially due to insufficient cracking force or improper positioning. The most significant factor affecting the results seems to be the egg size, indicating a need for improved sensor calibration and force adjustment for varying egg sizes.

5. RELATED WORK

The chapter "Egg Breaking" in Egg Science and Technology by Cotterill and McBee discusses the essential aspects of efficient egg-breaking operations [12]. It emphasizes that breaking plants should be located near egg production sites to optimize efficiency and product quality. Proper facilities must include adequate lighting, ventilation, and hygiene measures. Mechanical equipment used for egg breaking must be thoroughly cleaned and sanitized regularly. The chapter highlights advancements in egg-breaking technology, noting that modern machines—available from multiple sources—are highly sophisticated, operating up to 65,000 eggs per hour. However, the solids level in egg yolk can still be influenced by various factors during processing.

6. CONCLUSIONS

One limitation of the current egg cracker project is its difficulty in handling eggs of significantly different sizes and shell thicknesses. While the device performs well with standard-sized eggs, larger and smaller eggs occasionally lead to incomplete cracking or yolk breakage. Additionally, the device's efficiency can be affected by variations in egg freshness, which impacts the cracking force needed.

For users with mobility issues, another limitation is the device's accessibility and ease of use. Enhancements could include designing more user-friendly controls and improving ergonomics to accommodate users with limited dexterity. Implementing an adaptive mechanism that adjusts for egg size and shell characteristics would improve performance across a wider range of eggs. Additionally, integrating features that support hands-free operation or reduce physical strain would benefit users with mobility challenges. If given more time, refining these aspects would enhance the device's usability and inclusivity, making it more effective for all users.

The egg cracker project advances automated egg-breaking while addressing the needs of users with mobility issues [14]. Despite challenges with egg size and accessibility, proposed improvements promise enhanced performance and ease of use. Continued development will further increase the device's functionality and inclusivity, benefiting a wider range of users.

REFERENCES

[1] Bain, Maureen M., et al. "Probability of an egg cracking during packing can be predicted using a simple non-destructive acoustic test." British Poultry Science 47.4 (2006): 462-469.
[2] Kumar, C. Bala, Paul J. Kline, and Timothy J. Thompson. Bluetooth application programming with the Java APIs. Morgan Kaufmann, 2004.
[3] Chattopadhyay, Anupam, et al. "On the cost of asic hardware crackers: A sha-1 case study." Cryptographers' Track at the RSA Conference. Cham: Springer International Publishing, 2021.
[4] Pitoura, Evaggelia, and Bharat Bhargava. "Dealing with mobility: Issues and research challenges." (1993).
[5] Morgenstern, Leora. "Formalizing a Problem in Commonsense Physical Reasoning: The Egg-Cracking Problem." Fourth Symposium on Logical Formalizations of Commonsense Reasoning. 1998.
[6] Julianty, Jujun, et al. "Egg white powder in extruded fish crackers." International journal of food science & technology 29.3 (1994): 315-320.
[7] Haartsen, Jaap. "Bluetooth-The universal radio interface for ad hoc, wireless connectivity." Ericsson review 3.1 (1998): 110-117.
[8] Wijetunge, S. P., et al. "Design and implementation of a bluetooth based general purpose controlling module." 2008 4th International Conference on Information and Automation for Sustainability. IEEE, 2008.
[9] Bustos, Héctor Andrés Torres, et al. "Light Duty Tracking and Sensing System Using the Adafruit web." Ingeniería y universidad 24 (2020): 4.
[10] Tosi, Jacopo, et al. "Performance evaluation of bluetooth low energy: A systematic review." Sensors 17.12 (2017): 2898.
[11] Gupta, Aditya, and Aditya Gupta. "UART communication." The IoT hacker's handbook: a practical guide to hacking the Internet of things (2019): 59-80.
[12] Cotterill, Owen J., and Larry E. McBee. "Egg breaking." Egg science and technology. CRC Press, 2017. 231-263.
[13] Heydon, Robin, and Nick Hunn. "Bluetooth low energy." CSR Presentation, Bluetooth SIG https://www. bluetooth. org/DocMan/handlers/DownloadDoc. ashx (2012).
[14] Dhanapala, Pathum, et al. "Cracking the egg: An insight into egg hypersensitivity." Molecular immunology 66.2 (2015): 375-383.

[15] Tzafestas, Spyros G. "Mobile robot control and navigation: A global overview." Journal of Intelligent & Robotic Systems 91 (2018): 35-58.

© 2024 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.