POOPICKPRO: A SMART WAY FOR ENVIRONMENTAL RESPONSIBILITY AND DOG WASTE MANAGEMENT

Shilei Cao [1], Jonathan Sahagun [2]

[1] Arnold O. Beckman High School, 3588 Bryan Ave, Irvine, CA 92602
[2] California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

Dog poop is a concern that can be a huge detriment for environments, but are overlooked due to the smaller appearances it seems to create compared to the other harms. As the problem would simply be solved with the enforced responsibility of dog owners, we decided to create a marketable, smart robotic arm with a connected app that encourages dog owners to pick poop up while climbing up rankings on a mobile app. It solves the idea of insanitary rejection with the safe distance, and creates competition that will help owners pick up poop voluntarily. The main problems were design miscalculations, mechanical restrictions, and advanced code features [1]. In experimenting, tests were performed on the sensor's accuracy to detect poop and the reliability of the app's ranking system. In the end, the correct sensor that correlates with dog poop's elements, and a method of spaced time for accurate and fair poop pick up counts was used. This idea is convenient, and methodically incorporates the responsibility of dogs onto owners. The creations of such research has not only helped discover the impact small inconveniences can create, and the basic responsibilities that can resolve large issues.

KEYWORDS

Impactful, Methodical, Convenient, Smart Sensors

1. INTRODUCTION

In daily life, many communities are frustrated by the lack of care and responsibilities the outside environment receives. We may walk out to find ourselves accidentally stepping on gum that was carelessly spat on the ground, trash flying into our yards, and dog poop stinking up the entire neighborhood. These daily inconveniences turn our joy and patience into anger and disgust. Not only our feelings are neglected, but a majority of health and environmental hazards are causes of such neglect. A perfect example of such a hazard is the harm dog poop places on the environment. The chemicals in the excrement of dogs will enter bodies of water with sewers, depleting oxygen and growing harmful algae in water, harming the marine life's survival. It may even cause diseases that are harmful to human beings, if in contact [2]. 39% of dog owners also show to not pick up their dog's feces, even though they often display close contact with their pets that can endanger their rates of being affected with parasites [3]. These concerns may gradually increase if the majority of dog owners refuse to take care of their responsibilities of picking up dog poop. So, because of the extremely common amount of times we encounter these harmful excrements in life, we decided to solve this issue using our own strengths. Since we can't change how dogs choose to produce, we thought we would start with providing help for those who may

David C. Wyld et al. (Eds): NLCAI, AIFU, CCSEA, BIoT, SEA, SIPRO, BDML, CLOUD – 2025
pp. 293-302, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151022

need assistance in taking care of their dog. It would not only help environmental pollution be reduced by a significant amount (if popularized), but it could also become a popular product for business due to the smartness of the device that is more advanced against its competitors. Over time, our product will have a huge impact in a person's daily life, and also remove the problem of the stink dog owners try to avoid.

In the first methodology, the imposed solution was through an attempt to create a program that tried to solve the problem of the restrictions of the effectiveness of the human eye. This program created a system that detected dog poop in difficult areas, but it was inconvenient to carry, which we improved on with a mobile app.

The second method focused on enforcement through laws and camera technology, but with its expensive maintenance and limited enforcement, we changed the issues from punishment to encouragement, and a building of discipline [4]. The buildup of healthy habits is what will truly make people change instead of cheating the systems, instead of enforcement that will cause more conflict.

The last solution focused on transforming poop into an environmentally friendly fertilizer [5]. However, the collection would either require employment of volunteers or workers, or the discipline of owners would still have to be improved. We improvised on this by targeting the main cause of this problem and used smart technology to attract users. This methodology could potentially be a second step after the collection of poops is improved.

To solve this problem, we looked at all aspects of why some refuse to pick up dog poop. The answer is simple: waste is dirty, and nobody wants to touch waste. To reduce the harm, the easiest way would be to encourage more people to pick up after their dogs. However, since we can't change the smell and appearance of excrement, we thought that creating a device that helps do this task for you would be the best solution. This solution is the most achievable, convenient, and marketable. If we made it smarter to eliminate the issues with these arms, the product would be significantly more flawless. It would be more affordable than expensive chemical solutions, and can be much more convenient. You simply bring it, place a bag on the claw, pick up the waste with the claw, and throw it into the trash. After long-term implantation of this project into our environment, long term results will also be visible. With more marketing techniques, those who refuse to spend the money will eventually become convinced to invest a small amount of money into such a beneficial resource. Additionally, this product's way to test would minimize waste in resources that will add to our budget. To further perfect this solution, we thought of potential obstacles a pet owner may face while using this product. We decided to add sensors, technology, and applications that will not only help become more convenient, but its advancement will also win over numerous losing competitors. In the application, we wanted to add ways to promote environmental beautification even more, where we included friendly competition that promotes this behavior, and helpful mechanisms for problems in the devices [6]. This device would also be customizable to the comfort of a user's own liking, and possibly promote trends.

In section four, the two experiments focused on how well each key feature responded to our necessities. In the first experiment, there was a control group along with two groups with different independent variables. For the second experiment, we set it up with 2 different outcomes, and depending on which outcome the experiment reached, we declared its success. The important findings for the first experiment were about the accuracy of the sensor, which is set by the code of the borderlines, and the collection of data the sensor is assigned to have. We found a success for accurately detecting real dog poop, mainly because of a selection of a sensor that detects the correct elements of dog poop [7]. The second experiment focused on the correct code that spaced out the time between each collection count, therefore determining the reliability of the ranking system. We set it up with a connection of the sensor to the app, which should correctly determine the poop in the set gap of time. The most significant findings were the correct collection of dog poop, and the typical time it takes to complete the task. The fair count of data was caused by a correct connection from the app to the sensor, a correct code for the app's functionality, and a good restriction to prevent cheating.

2. CHALLENGES

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

2.1. Coding Errors

As with every complicated project comes, ours had a few errors as well. The first problem was a lot of errors with coding, and finding ways to code features that were unfamiliarized before. With limited experience, we could do a lot of research for different features, as well as connections to programs that are credible and already coded, like Google's Firebase for data collection. We could also add better design to make the app seem more comfortable for users, and design something that is more appealing. We could add security features to maximize fairness for users, and competition to establish habits.

2.2. The Mechanical Parts

Another issue in this robot was in the mechanical parts. We had found an 3-D printing arm model online, but the called parts weren't available [8]. When we tried to buy bolts to connect the different parts together, many of the parts weren't available in the checked stores. We could saw and modify our own pieces if they aren't available in stores. If we truly cannot find the part, we can replace them with low-budget alternatives. Additionally, many sensors or technology were very expensive to buy. We could do more research into popular sensor markets to find the most effective and affordable one, with comparisons on different potentials. We could also make sure our parts don't require a complementary technology to go with it, so it keeps its maximum budget.

2.3. Implement

Reality is different from expectations. We had plans in every field of design that were carefully composed, but were hard to instill with the design plan. The app designs sometimes couldn't completely be implemented into the code. We could first create the app and codes with the simplest features, then modify them later on. In another instance, the placements of the different materials onto the prototype were not as expected. The design for the arm was very logical in the first place, but some of them on the arm made it hard to easily place technology like sensors and lights onto the arm. We could remove a few features that could be replaced with real life items, like the nightlight idea with your phone's flashlight. This could be a great feature for accessibility, but compared to the essential parts of the arm, these ideas need to be cut off.

3. SOLUTION

The boron (or microcontroller) on the poop scooper is first connected to the firebase, set up with different device names that are defined by different words to simplify [9]. First, the sensor is able to detect the components contained in dog poop. When the sensor is on, it sends the number of detected values to a connected boron. The boron, which comes with network data that allows it to be used outdoors, sends the detected numbers from the sensor into the cloud system through the connected data antenna [10]. The database (connected to Particle) is programmed to only count values above a certain number limit, which is a certain amount more than what values would show in regular air. When poop is detected, the values will change drastically. Additionally, in the database on particles, it is programmed so that it only detects data every 5 minutes, which ensures fairness. Then, the firebase transmits the given, unusual data after each set number of data collections into the app through a code on Android Studios. The connection of the leaderboard and the data is transmitted through a code, and the leaderboard changes in live time for every data point collected in response. The system is then complete for the leaderboard and the users' competitions. This system incorporates three different program platforms, including Flutter (connected to Android Studios), Particle, and Google's Firebase. These programs are connected through the data network, which is why the boron needs a portable data wire. The boron and sensor are also all connected to the arm to ensure the most direct data.

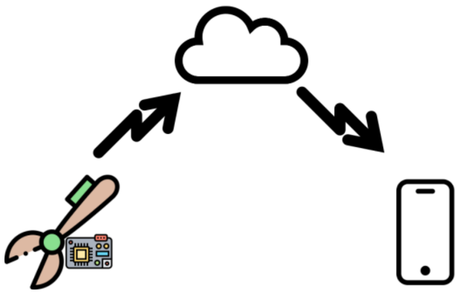

Figure 1. Overview of the solution

The first component is the boron, where we used a Boron LTE-M starter kit that is connected to an EstherSIM. The embedded SIM card is able to connect to the database and use network data portably. It serves as a translator that takes in the data from the sensor, and transfers it to the database in the cloud system, which then transcripts that into the app's leaderboard according to the given code.

Figure 2. Picture of the Boron

```
#include "Particle.h"

SYSTEM_MODE(AUTOMATIC);

SerialLogHandler logHandler(LOG_LEVEL_INFO);


int gasPin = A0;
int gasValue = 0;
float voltage = 0;

char messagestring[30];


void setup() {
  Serial.begin(9600);

  }

void loop() {
  gasValue=analogRead(gasPin);
  Serial.println(gasValue);

  voltage = (gasValue / 4095.0) * 3.3;
  Serial.println(voltage);

  if (voltage > 1.0) {
      sprintf(messagestring, "{\"data\":%d, \"voltage\": %0.2f, \"id\": \"goofy\"}", gasValue, voltage);
      Serial.println(messagestring);
      Particle.publish("firebase",messagestring, PRIVATE);
      delay(60*5*1000);
  }


  delay(5000);
}
```

Figure 3. Screenshot of code 1

This code is written for a Particle microcontroller and is designed to read data from a gas sensor connected to analog pin A0. The SYSTEM_MODE(AUTOMATIC) macro ensures that the device connects to the cloud automatically. A SerialLogHandler is set up for logging at the INFO level. In the setup() function, Serial.begin(9600) initializes serial communication for debugging purposes.

The loop() function continuously reads the analog value from the gas sensor using analogRead(gasPin) and stores it in gasValue. This value is then converted into voltage using the formula (gasValue / 4095.0) * 3.3, since the Particle microcontroller operates on a 3.3V system and has a 12-bit ADC (Analog-to-Digital Converter), meaning its readings range from 0 to 4095. The raw gas sensor value and the calculated voltage are printed to the serial monitor.

If the voltage exceeds 1.0V, a JSON-formatted string is created using sprintf(), containing the gas sensor data, voltage, and an identifier "goofy". This message is then published to a cloud event named "firebase" using Particle.publish(), ensuring private data transmission. After publishing, the program waits 5 minutes (60*5*1000 milliseconds) before proceeding to the next cycle, preventing excessive data transmission. The loop also includes a 5-second delay (delay(5000)) to slow down data collection when the voltage is below 1.0V. This setup is useful for monitoring gas levels and sending alerts when a threshold is exceeded.

The leaderboard serves as a way to help stimulate competition between users for a chance to win prizes. It will increase the discipline of owners, and a sense of responsibility.The leaderboard is coded to sort active users from most to least, where it is directly connected to the firebase. It will sort the data from the Firebase as scores according to what the code commands the lowest number of a valid score to be.

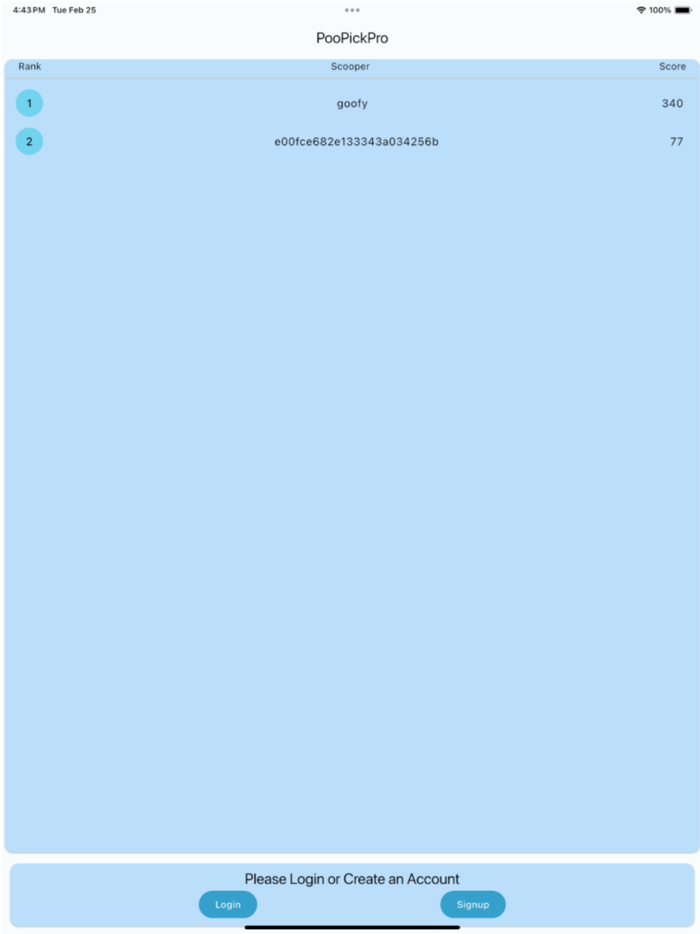

Figure 4. PooPickPro

```
Map<String, int> sortLeaderboard(Map<dynamic, dynamic> leaderboard){
 Map<String, int> result = {};


 leaderboard.forEach((key, value) {
   result[key] = leaderboard[key]["events"].length;
 });


 var sortedEntries = result.entries.toList()..sort((a, b) =>
b.value.compareTo(a.value));

 Map<String, int> sortedMap = Map.fromEntries(sortedEntries);

 print(sortedMap);

 return sortedMap;

}
```

Figure 5. Screenshot of code 2

This Dart function, sortLeaderboard, takes a map (leaderboard) with dynamic keys and values and returns a sorted map where keys are String and values are int. The function first initializes an empty map called result, which will store the processed data. It then iterates over the leaderboard map using .forEach(), extracting the length of the "events" list for each key and storing it in result. This means that the function assumes each value in the leaderboard map is another map containing a list under the "events" key.

After populating result, the function converts its entries into a list using .entries.toList(), then sorts this list in descending order based on the number of events (b.value.compareTo(a.value)). The sorted entries are then converted back into a Map<String, int> using Map.fromEntries(), ensuring the final map maintains the descending order. Finally, the function prints and returns the sorted map. This function is useful for ranking users or entities based on event participation.

The firebase collects basic user information for security and accounts, and it serves as a connection that translates the data from the cloud database into points for the user on the leaderboard. By sorting data points into variable names that are created based on different email registrations, the variables and their datas are kept in the Firebase so they are not lost.

Figure 6. Screenshot of sign in page

```
Widget leaderboardFutureBuilder(){
 DatabaseReference ref = FirebaseDatabase.instance.ref('device_data');

 return FutureBuilder(
   future: ref.get(),
   builder: (context, snapshot){
     if (snapshot.hasData){
       if(snapshot.data == null){
         return Text("Error");
       }
       if(snapshot.connectionState == ConnectionState.waiting){
         return Center(child: CircularProgressIndicator());
       }
       else {

         Map<String, int> leaderboard = sortLeaderboard(snapshot.data!.value
as Map<dynamic, dynamic>);
         return drawLeaderboard(leaderboard);
       }
     }
     return Center(child: CircularProgressIndicator());
   },
 );
}
```

Figure 7. Screenshot of code 3

The leaderboardFutureBuilder function in this Dart code returns a widget that asynchronously retrieves data from a Firebase Realtime Database and displays a sorted leaderboard. It starts by defining a DatabaseReference (ref) that points to the 'device_data' node in Firebase. The function then returns a FutureBuilder widget, which listens to the data retrieval process and updates the UI accordingly.

The FutureBuilder uses ref.get() as its future, meaning it waits for Firebase to return data before proceeding. Inside the builder function, the code checks if snapshot.hasData. If data exists but is null, it returns a simple "Error" text widget. If the connection state is waiting, meaning the data is still being retrieved, a CircularProgressIndicator is displayed in the center to indicate loading. Once data is available, it converts the snapshot's value into a map, sorts it using sortLeaderboard(), and then passes the sorted data to the drawLeaderboard() function, which likely generates the leaderboard UI. If the snapshot has no data or is still loading, a loading indicator remains on the screen. This function ensures an efficient way to fetch and display Firebase data in a leaderboard format.

4. EXPERIMENT

4.1. Experiment 1

A potential blind spot we wanted to test was the accuracy of the sensors. If this part of our scooper doesn't work out, the scooper's main feature of it being smart would be false.

We will set up the experiment with the data of the sensor in one end, and the combination of three different groups: the control group with no elements, the group with an unmentioned element in the list of detectable elements for the sensor, and the actual usage of the scooper with dog poop. We will collect a set amount of data points for each group, and compare using a dot chart that will see if the sensor actually works only on dog poop. This way, we can compare the data visually to not only see what values we will set the boundary of a poop detected as, but also see if the values do change.

4.2. Experiment 2

Another blind spot we will test is the fairness of the collect count for each user, since this will help the right person get the right rankings. We will test this by looking at how the system counts a successful collection.

The experiment will start with the database, the live app reaction with two blank accounts, and the scooper's sensor connected to the data collection. When we go out in real life and use the robotic arm, we see the sensor create a smell count every 5 minutes. When the app counts the poop, we will not only count it's real life accuracy to detect real poop, but also make sure it counts only once for every poop collected. We will make sure the numbers go up correctly, and will change the rankings for users accordingly. The data will be collected by a few different trials set at different days in order for multiple trials and multiple data points.

5. RELATED WORK

The first comparisons were to a trained model that acknowledged dog poop in environments challenging to find for the human eye [11]. It helps owners locate, which is similar to our poop scooper's sensors. The system is overall very effective, requires skill to build, and is very advanced in training. However, their models haven't been trained well for mobile phones yet, which we improved on by creating apps that have updated firebases for convenience. The group also didn't recognize that most mindsets behind not picking up poop is because of resistance towards feces, not a physical inability. We improved on that with a scooper that provides a more sanitary option.

This scholarly source tackles the infamous dog poop laws of New York, where the state imposed fines on any dog owner that is found to not have taken care of their dog feces [12]. The solution is very straightforward and effective, where the fear of charges force the owners to do their responsibilities. However, this is only limited to proof, and there is a lot of poop that is unrecognizable without cameras. Additionally, the complicated New York began to turn political in the early stages of this law, looking at whose issue this really came from. We manage to solve this problem from the roots, by encouragement and not force.

The research focuses on turning dog poop into compost, making dog poop into something that is more beneficial than disrupting to the environment [13]. This solution is relatively difficult, as it requires a collection of the feces from different owners, and ignores the more complicated collection processes that would follow this idea if the plan was taken into action. Additionally, it wouldn’t solve the basic thinking for dog owners, so regardless of how effective this plan is, the dog owners wouldn’t be affected to take action. With our project, we can, again, solve the problem from the roots with convenience, something they ignored.

6. CONCLUSIONS

Some mistakes with our project is the size of the scooper, which may be too big and heavy to carry around, defecting the idea of convenience. However, this aspect is very limited because of the materials and printers we were allowed to use. With the budget we have, we are only allowed to create so much, and that makes the machine easier to break. Additionally, the sensors and apps may be too costly for the scooper to be advertised cheaply, which may be troubling for this product to be vastly spread and operated. If we had more time, more code would be added onto the app for better features that were included in the prototype [14]. The app has a very basic function that is still very weak in terms of fairness. The overall rules of using the app are still very vague. Additionally, other features like light in the darkness, and temperature game features could be added to enhance the experience. In the future, we will be able to create cheaper, more advanced, and personalized designs with this prototype to be more appealing and convenient for users.

Overall, the scooper provides a wide variety of features that has the potential to improve a large group of dog owners [15]. This change could ultimately improve environmental problems and pollution, even if it’s only a small group. With a combination of programming, mechanics, and design, the scooper includes a variety of skills and technology that will help the community, starting with a small step.

REFERENCES

[1] Lee, Hyun Keun, et al. "Mechanical restriction versus human overreaction triggering congested traffic states." Physical review letters 92.23 (2004): 238702.

[2] Sinha, Vijaykumar, Mayank R. Patel, and Jigar V. Patel. "PET waste management by chemical recycling: a review." Journal of Polymers and the Environment 18.1 (2010): 8-25.

[3] Mandarino-Pereira, Arisa, et al. "Prevalence of parasites in soil and dog feces according to diagnostic tests." Veterinary parasitology 170.1-2 (2010): 176-181.

[4] Mancuso, Massimo, and Sebastiano Battiato. "An introduction to the digital still camera technology." ST Journal of System Research 2.2 (2001).

[5] Chen, Jiao, et al. "Environmentally friendly fertilizers: A review of materials used and their effects on the environment." Science of the total environment 613 (2018): 829-839.

[6] Cool, Colleen, and Hong Xie. "How can IR help mechanisms be more helpful to users?." Proceedings of the American society for information science and technology 41.1 (2004): 249-255.
[7] Crall, Jon. "" ScatSpotter" 2024--A Distributed Dog Poop Detection Dataset." arXiv preprint arXiv:2412.16473 (2024).
[8] Cabibihan, John-John, M. Khaleel Abubasha, and Nitish Thakor. "A method for 3-d printing patient-specific prosthetic arms with high accuracy shape and size." IEEE Access 6 (2018): 25029-25039.
[9] Chougale, Pankaj, et al. "Firebase-overview and usage." International Research Journal of Modernization in Engineering Technology and Science 3.12 (2021): 1178-1183.
[10] Melloy, G. F., P. R. Summon, and P. P. Podgursky. "Optimizing the boron effect." Metallurgical transactions 4 (1973): 2279-2289.
[11] Gross, Matthias. "Natural waste: canine companions and the lure of inattentively pooping in public." Environmental Sociology 1.1 (2015): 38-47.
[12] Samuelson, Pamela. "Google Book Search and the future of books in cyberspace." Minn. l. REV. 94 (2009): 1308.
[13] Wuthisuthimethavee, Suwit, et al. "Do Not Throw Pet Faeces Away: Composted Manures Obtained from Dog and Cat Faeces Contain High Nutrients and Effectively Cultivate Plants." Recycling 9.6 (2024): 123.
[14] Houde, Stephanie, and Charles Hill. "What do prototypes prototype?." Handbook of human-computer interaction. North-Holland, 1997. 367-381.
[15] Ruangpayoongsak, Niramon, Jakkrit Sumroengrit, and Monthian Leanglum. "A floating waste scooper robot on water surface." 2017 17th international conference on control, automation and systems (ICCAS). IEEE, 2017.