Smart Pet Food Dispenser

Stella Zhou and Tan Huynh, *Students, University of California, Irvine,* Fadi Kurdahi, *Professor, University of California, Irvine*

***Abstract*—Recently, pet obesity has become a widespread problem in the United States. Our project aims to combat pet obesity and help pet owners or rescues dispense the ideal amount of food by developing a smart pet food dispenser. The smart pet food dispenser takes photos of the pet detected by the motion sensor and send the data to Amazon Rekognition. The machine learning model from the cloud will identify the specific dog or cat breed, then using that information, the food dispenser will dispense the recommended amount according to the breed's size and veterinarian's suggestion or according to the user's input. We built a desktop application that is implemented into the Raspberry Pi to allow for user input. Throughout the process, there was some difficulty obtaining enough torque to turn the dispenser, but eventually through trial and error, we eventually got the product to produce the expected results. After two quarters of development and building, we completed the project as it distinguishes between dog or cat breeds accurately and properly dispenses food based on the information grabbed from our data or the user's input.**

Index terms – *Amazon Rekognition, ultrasonic ranger, machine learning, smart pet food dispenser.*

I. Introduction

IN the United States, pet obesity is an epidemic with nearly 51% of pets classified as overweight [1]. Dogs and cats are classified as overweight if they are 10% to 15% above their ideal weight. Various veterinarian associations recognize that pet obesity is becoming a more prevalent issue and encourage veterinarians to discuss obesity with their clients because there are numerous health concerns associated, such as shortened lifespans, heart and respiratory disease, diabetes, etc. [2][3]. There are many factors that contribute to pet obesity, but the two main factors of pet obesity are over feeding and giving too many treats daily [4]. Current treatments to address pet obesity include exercise with a therapeutic diet, underwater treadmill therapy for extremely overweight pets who cannot walk properly, medication using dirlotapide which increases circulation levels and reduces appetite, calorie restriction for small animals, or incorporating a more balanced diet [3]. These methods have been proven to achieve weight loss in pets, however, it was found that pets tend to have rebound weight gain and to further prevent that, it is important to maintain a healthy diet and continue exercising the pet.

Our project aims to combat the pet obesity epidemic by building a smart pet food dispenser. The goal of the smart pet food dispenser is to prevent overfeeding with portion control and other various features that would help with common bad eating habits, such as eating too fast. The Smart Pet Food Dispenser can be incorporated as early prevention or to maintain a balanced diet because it will only dispense the recommended amount based on the pet's size which can contribute to preventing weight gain rebounds. Our project is an improvement upon current automatic pet food dispensers which primarily focus on dispensing dry food at a specific time every day. Most automatic pet feeders do not account for pets who have bad eating habits or pet owners who overestimate the food dispensed.

Our project uses a Raspberry Pi as the microprocessor to connect the software portions to the hardware portions. We have completed our project as our device distinguishes between dog and cat breeds, and it seamlessly connects with the desktop application that was developed. In addition, other features were implemented such as the refill indicators and text message alerts.

II. Materials Used

Below is the list of the materials used for the project:

1) *Raspberry Pi*

The Raspberry Pi was used as the microprocessor to connect the hardware and software portion.

2) *USB Pi-compatible camera*

The USB Pi-compatible camera was used to take photos of the pet in view to send to the machine learning model.

3) *2 wall dry-food dispenser*

The 2 wall-mounted dry-food dispenser was used to hold a pet's dry food.

4) *2 NEMA-17 Stepper Motors*

The 2 stepper motors were used to dispense food by rotating the dry-food dispenser's latch.

5) *2 A4988 Stepper Motor Drivers*

The 2 stepper motor driver boards were used as a link to communicate between the Pi and the motor.

6) *12V DC Power Supply*

The DC power supply was used to power the two stepper motor.

7) *Breadboard*

The breadboard was used to connect the raspberry pi to the dispenser parts.

8) *Jumper cables*

The jumper cables were used on the breadboard to connect GPIO to hardware accessories.

9) *3 ultrasonic rangers*

The ultrasonic ranger was used to measure the distance between the lid and the dry-food for the refill indicator feature.

10) *Resistors*

The resistors were used to regulate the flow of electrical current.

11) *Capacitor*

The capacitor was used to decouple the DC power supply.

12) *Wooden Box*

The wooden box was used to make our project be more user friendly and aesthetic by having the breadboard and the wires inside.

13) *2 6.35mm D-shafts*

The D shafts were used to connect the stepper motor to the wheel inside of the dispenser to rotate it.

14) *2 5mm to 6.35mm Couplers*

The coupler connected the D shafts to the stepper motor.

15) *Super Glue*

The glue was used to hold material together.

16) *Duct tape*

The duct tape was used to hold material together.

17) *Drill*

The drill was used to create a hole in the dry-food wall dispenser for the DC motor.

III. High-level Hardware and Software System

A. *Hardware*

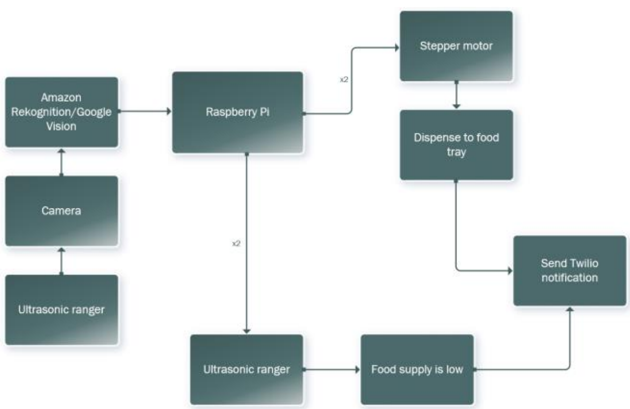

Fig. 1. Block Diagram of Hardware Interaction

The hardware components of the pet food dispenser depends on a Raspberry Pi 400 to serve as the microprocessor to interlink the hardware to the software portion which handles the logic. An ultrasonic ranger is used to detect motion which enables a USB Pi-compatible camera that captures an image of the pet and send it through the Pi and into Rekognition which processes what animal the pet is and if it’s a dog, what breed it is. The steppers are used to dispense food to the tray by rotating to a certain degree which depends on the recommended intake amount of food for the specific pet. We initially had problems finding enough torque to spin the wheel in the dispenser, but after testing with the 5V stepper motor, we substituted the 5V stepper with a 12V stepper motor which provided enough torque to properly spin the wheel. Since the motor requires more power than the 5V that the Pi can provide, a separate 12V power supply is required to provide sufficient power for the motor to rotate. An ultrasonic ranger also needs to be mounted to the lid of the food container, pointed down to determine when the food supply is running low and needs to be refilled. If the food has been dispensed or if the amount of food is low, a notification gets sent to the owner’s phone through Twilio to let them know. All of these components are connected to the GPIO pins on a breadboard as shown in Figures 2 and 3. Shown in Figure 2 is the internal setup of our project; we have our breadboard attached to the bottom of the box, and the Raspberry Pi is placed on the inside of the box cover. Super glued to the front, we have our stepper motors that are connected to 2 5mm to 6.35mm couplers which are connected to 2 6.35mm D-shafts which act as a form of connection to the wheel of the dispenser so that it may spin to dispense food. To allow the D-shafts and wiring to come out of the box, we drilled holes where necessary.

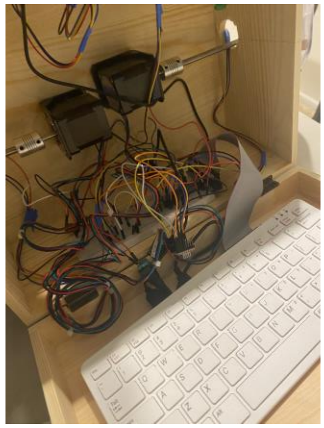

Fig. 2. Internal Physical Circuit

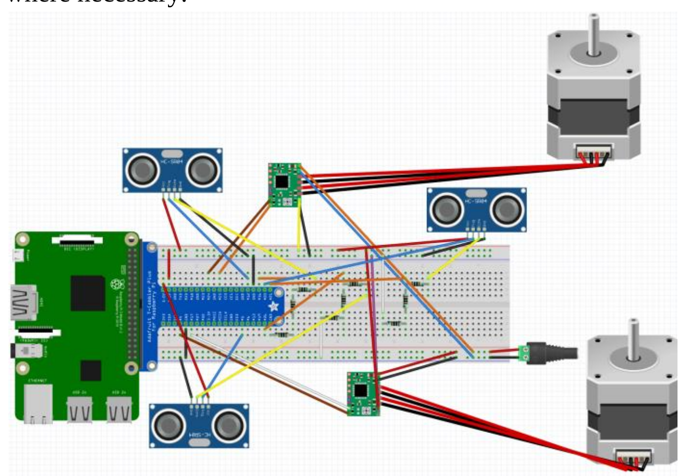

Fig. 3. Digital Circuit

B. *Software*

The overarching language that we use to program our backend software portion is Python due to its vast compatibility with the libraries that we use such as AWS SDK, GPIO, and Twilio. To pass the image sent from the camera to AWS Rekognition for label processing, AWS SDK is needed to write code to upload the image into an S3 bucket and have Rekognition read from the image to return its predicted labels. Python’s GPIO library will also be essential for us to link all the hardware elements connected to the GPIO pins to a controllable interface that allows us to receive data from hardware as well as setting the specifications of what we want it to do. Twilio is useful because it provides us with an SMS API which will be used to send text messages to the pet owner, letting them know when food has been dispensed or when the dispenser needs to be refilled. We also designed a localized graphical user interface using PySimpleGUI which will simplify the setup process and make it easy for the user to configure how the dispenser behaves as shown in Figure 4. Within the GUI, the user can calibrate the motion sensor, run the machine-learning algorithm to determine pet type, and alter specific parameters on how the food should be dispensed.

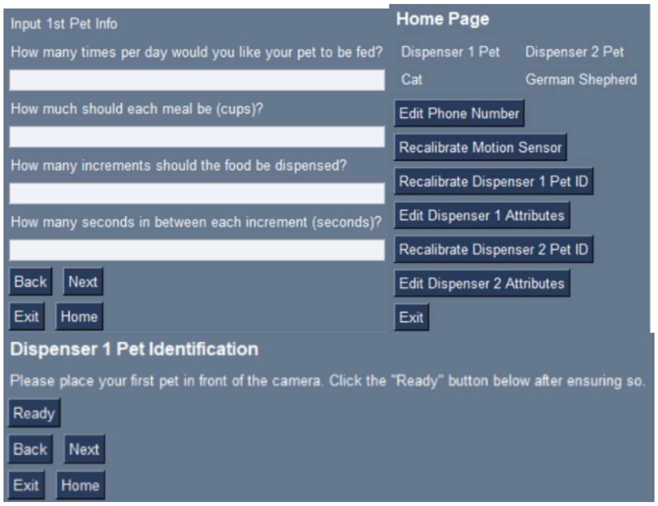

Fig. 4. Some Pages from the User Application

IV. Methods

We first researched what features we wanted to include in our project and the models which could be utilized in our software to properly determine the breed of a pet. After we completed our research, we built the backend of the project which consisted of training the machine learning model in Amazon Rekognition and testing the individual components that would be used such as the camera, ultrasonic ranger, etc. with the Raspberry Pi.

The basic structure of our project will have the Raspberry Pi connected to a breadboard, and the breadboard will have devices inside or around the food dispensers. The Pi-compatible camera is placed in between each dry-food dispenser and will always be initialized as 'ON' to take photos when the pet is in view. When the pet is in view of the camera, it will take five photos of the pet and send the image through Amazon Rekognition. Amazon Rekognition will determine the breed of the pet for each photo, and this information is sent to the Raspberry Pi where it will average the confidence output. With this information, the Raspberry Pi will determine which pet is labelled for the first or second dispenser, then the Pi will tell the Raspberry Pi how much the stepper motor should rotate the pet food dispenser latch based on the breed's size. After the pet is fed, a mobile notification will be sent to the owner. Inside both dry-food dispensers are ultrasonic rangers which are used to determine if the dispenser needs to be refilled. If so, another mobile notification will be sent to the owner specifying that the device needs to be refilled. After building and connecting the devices, we rigorously tested the dispenser to ensure that it produced the expected results and to ensure safety of the owner and the pet.

V. Results

After training the machine-learning model on AWS Rekognition using the Stanford Dogs dataset, we achieved an F1 rating for 86.4% which is derived from the precision score of 87.3% and recall score of 86.5%. To divide the training set and the testing set from the 20,310 images of different dog breeds, we used an 80:20 split so 16,418 images were to be used for training while 4,162 images are reserved for testing. Overall, we are happy with the results can be seen in Figure 6.

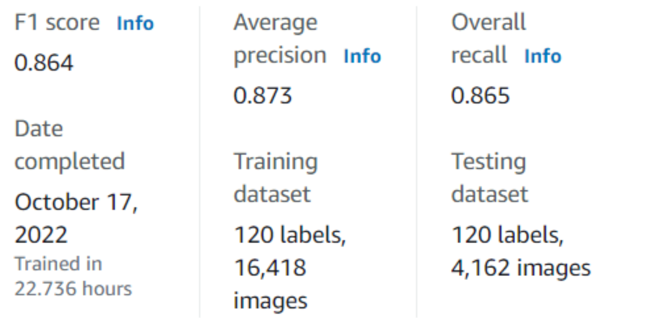

Fig. 6. Overall evaluation results for the model and evaluation results for each label in the test dataset

To detect whether a pet is a dog or a cat, we simply use the base model of Rekognition without the need for any custom labels training since the base model is already capable of differentiating between animals. Figure 7 shows that both pets are successfully recognized by Rekognition. Even though this model returns extraneous labels that will not be used, we simply filter out only the labels we need in the backend.

Fig. 7. Pet Detection

Since base Rekognition, is not sophisticated enough to differentiate between dog breeds which required us to train our own model using the Custom Labels feature. Figure 8 shows that our model successfully determines the dog to be a golden retriever with 99% confidence.

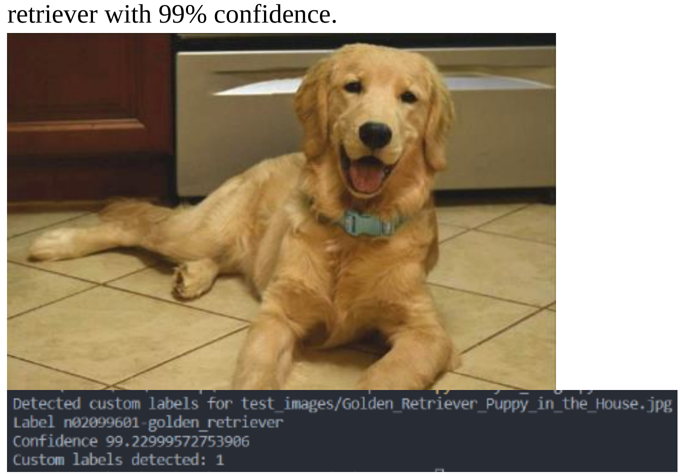

Fig. 8. Dog Breed Identification

Although there were initial concerns on how fast Rekognition would be to respond to an image with the labels, testing has shown that it is near instant, so there were no bottlenecking issues from slow responses.

Based on the user inputs from the desktop application, we saw expected results. The camera took five frames of the pet in front of the camera when the user was prompted to. These five photos were sent to the Amazon Rekognition cloud, and from the models, it determined the breed and type of pet which is then displayed on the home page. At a specified time, when motion is detected in front of the device, the camera will take another five photos, and it will again send these photos to the model, and based on its output, the backend will determine which labelled pet it was. Once the backend identifies the labelled pet, the device dispenses the proper amount of food, and depending on the user input, it can also incorporate incremental dispensing. There are some issues with the torque of the dispenser where depending on the type of food, it may not be able to fully push the food out of the dispenser, but most typical dog and cat kibbles will work with our project.

VI. Summary

A. Procedure

During our first quarter, we researched features that we wanted to incorporate in our project. After we discussed with our advisor and finalized our project's idea, we began coding the software portion and training our machine learning model to identify the breed of the pet based on a photo. We tested a few individual components that we intend to use and already have which require the Raspberry Pi, such as the ultrasonic ranger.

During our second quarter, we completed developing our backend and built our project's hardware components. Before assembling our project and wiring our breadboard, we used a software called Fritzing to help us design and build our prototype circuit. It ensured that we did not short circuit or build our breadboard improperly. After testing our circuit on Fritzing, we replicated the setup and wiring for our product. Then, we connected our circuit to our Raspberry Pi and tested the connection and overall system to see if our main loop worked properly. After thorough testing and exchanging parts (I.e. changed 5V stepper motor to 12V stepper motor and added stepper motor drivers), we placed the circuit inside of a wooden box to keep everything clean. We drilled holes on the side of the wooden box to allow wires to leave the box when needed, such as wiring for the ultrasonic rangers inside the dry-food dispensers. We also created a desktop application and tested its functionality with the circuit and dispensers.

B. Results

Overall, we followed our expected timeline and goals. The current results have been successful and the machine learning model have produced accurate outputs. As shown in section V, the pet and breed detection have high confidence percentages, and the simulated results match the expected results. After performing end-to-end testing, we determined that our dispenser system (as shown in Figure 9) is fully functional

Fig. 9. View of the Finished Dispenser

VII. Conclusion

We have achieved all of our goals for the senior design project. We developed and trained our machine learning model to our satisfaction because it correctly determines a pet's breed with high confidence percentages. We were initially concerned about bottlenecking Amazon Rekognition, however, our testing showed that it is reliable and sends results almost instantly. Throughout our building process, we were afraid of incompatible parts, but after discussing with our advisor on what should be replaced, we chose parts that were appropriate for our project. After completing our project, we thoroughly tested the safety of our project, and it produced the expected results where it dispensed based on motion detection and sent the owner a text message after food has been dispensed. In addition, we achieved a stretch goal which was to create a desktop application to allow the user to input how much they would like their pet to be fed or implement incremental feeding. Some potential ways to extend the project include utilizing an LCD display which will display useful information to the user (refill indicator, the pet has been fed, etc.) or implement a functionality where the device can measure if the pet has finished its food or not. This functionality can be useful because there are some health risk correlations to pets who do not consistently finish their foods (allergies, etc.). For example, a cat does not exhibit symptoms that they are in pain, thus, if there are signs that a cat does not finish their food, it could indicate a health risk.

Acknowledgment

We, Tan Huynh and Stella Zhou, the authors of this report, would like to thank our advisor Professor Fadi Kurdahi for providing guidance on the developing the project idea, suggesting hardware, and overall being extremely supportive. In addition, we would also like to thank the EECS 159 Shark Tank and Professor Kleinfelder for funding our project and providing equipment resources, and the 159 teaching assistants (TAs) for giving advice during the weekly lab checks.

REFERENCES

[1] "Veterinary emerging topics (VET) report - Banfield Pet Hospital," Jan-2020. [Online]. Available: https://www.banfieldexchange.com/-/media/Project/Banfield/Main/en/Exchange/Vet-report-overview/PDF/2020VETReportOverweightPetsUpdated.pdf?rev=1b9f2845688940678c05ec0c8a0b705f&hash=5F6811FEC787B429A327C8CFA787AD54. [Accessed: 09-Dec-2022].

[2] A. J. German, "Insights from 10 years of clinical research on Pet obesity: What can we ...," 26-Mar-2015. [Online]. Available: https://vetcenter.purina.es/sites/default/files/article-pdf/Article%20574_Insights%20from%2010%20Years%20of%20Clinical%20Research%20on%20Pet%20Obesity-%20What%20Can%20We%20Do%20Better.pdf. [Accessed: 09-Dec-2022].

[3] J. Shmalberg, "Treatment of obesity in cats and dogs," *Today's Veterinary Practice*, 17-Feb-2022. [Online]. Available: https://todaysveterinarypractice.com/nutrition/nutrition-notes-treatment-of-obesity/. [Accessed: 09-Dec-2022].

[4] J. Lynden, T. Hollands, and J. Ogden, "Animal obesity: What insights can a one health approach offer when it ...," *British Veterinary Association*, 25-Jul-2022. [Online]. Available: https://bvajournals.onlinelibrary.wiley.com/doi/10.1002/vetr.1904?af=R. [Accessed: 09-Dec-2022].

[5] "Automatic pet feeder using Arduino," *Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects*. [Online]. Available: https://circuitdigest.com/microcontroller-projects/automatic-pet-feeder-using-arduino. [Accessed: 09-Dec-2022].

[6] T. Sangvanloy and K. Sookhanaphibarn, "Automatic pet food dispenser by using internet of things (IOT): Semantic scholar," *undefined*, 01-Mar-2020. [Online]. Available: https://www.semanticscholar.org/paper/Automatic-Pet-Food-Dispenser-by-using-Internet-of-Sangvanloy-Sookhanaphibarn/d54e513726226a5d03feee4a575f33d82e690bb3. [Accessed: 09-Dec-2022].

[7] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, J. Clune, C. Packer, M. S. Palmer, and A. Swanson, "Automatically identifying, counting, and describing wild animals ... - PNAS," *PNAS*, 05-Jun-2018. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.1719367115. [Accessed: 10-Dec-2022]

Stella Zhou is a senior Computer Engineering undergraduate student at the University of California, Irvine. Some of her interests include back-end development, big data, UX/UI, and machine learning.

Tan Huynh was born in Da Nang, Vietnam in 2000. Currently, he is a senior undergraduate student at the University of California, Irvine where he is pursuing a computer engineering degree at the Henry Samueli School of Engineering. Some of his interests include backend software development, cloud services, API development, infrastructure-as-code, and Internet of Things.

Fadi Kurdahi received a B.S. degree from the American University of Beirut, Lebanon in 1981 and a M.S. degree from the University of Southern California in 1982, both in electrical engineering. He received his Ph.D. in computer engineering from the University of Southern California in 1987. His interests include VLSI system design and design automation of digital systems.