# Automated Cat Feeder

## Smart, Connected, and Programmable

Clifton Francis
Dept. of Computer Engineering
San Jose State University
San Jose, USA
clifton.francis@sjsu.edu

Neville Mascarenhas
Dept. of Computer Engineering
San Jose State University
San Jose, USA
neville.mascarenhas@sjsu.edu

Sampath Lakkaraju
Dept. of Computer Engineering
San Jose State University
San Jose, USA
sampath.lakkaraju@sjsu.edu

Niyatkumar Patel
Dept. of Computer Engineering
San Jose State University
San Jose, USA
niyatkumar.patel@sjsu.edu

Under the guidance of Prof. Rakesh Ranjan

**Problem statements:**
Just like humans, obesity and diabetes can also be problems for cats. A cat's body fat has a direct correlation to its diet and the level of activity. Pet cats usually do not have high physical activity since their feed is provided by the owners and do not require to hunt. Controlling the diet of the cat is extremely important to make sure the cat is healthy. To achieve this, you need to either invest time for feeding the cat yourself or invest money to make sure someone dose the job, to avoid both a cat feeder is used. Most of the cat feeders available in the market is that they dispense the food out with a certain amount turns of a DC motor. If you are lucky, the pet will get the exact amount of food you want to feed them. However, most of the time, the food gets stuck. Either you will have to go in and manually feed cat or the cat does not eat, which defeats the purpose of an automatic cat feeder.

**Solution:**
The aim of this project is to develop a cat feeder which can be operated remotely using a web application. This feeder will be able to measure the weight of the feed that is being dispensed and retrieve the cats weight while it is feeding to monitor the cat's health. The product will be able to operate irrespective of the connectivity to the internet and make sure your cat is fed while you're occupied.

***keywords-*** Feeder (automated cat feeder), AWS (Amazon web services), Python, Mongo DB, Node.js, React, GUI, Motor driver Integrated Circuit - L293DNE, Analog to Digital Converter and Amplifier - HX711, motor (DC motor), Load cells.

### I. Introduction

A cat feeder dispenses a specified amount of food at a specified time with minimal effort from owner. An automated feeder usually has a motor controlled by a circuit which dispenses feed at a set time. The feeder does not have any mechanism which detects if the feed is dropped or not.  The feeder we have developed will provide both the functionalities along with remotely using internet and a web application. The feeder has additional functionalities such as weighing the feed dispensed and the weight of the cat. These can be viewed through the web application and verify whether the feed is dispensed properly along with the regular check of the weight of the cat, this data can be referred to modify the quantity of feed that being dispensed by the feeder. The feeder can dispense feed irrespective of its connectivity, thus providing owners a dependable system.

## II. TECHNOLOGIES USED:

- Python:
  A general purpose object-oriented, and high-level programming language [1]. Python 2.7 is used to aggregate the data coming from and going to the sensors and convert them into usable data.
- Mongo DB:
  A NoSQL database, used for developing the web application alongside the front end.
- JavaScript:
  JavaScript is an untyped and interpreted programming language. JavaScript being the language of the web we can make the webpages behave the way it is intended to [2].
- React.js.:
  React is a JavaScript library used for building user interface [3]. React libraries were used in building the GUI.
- Node.js.:
  Node.js is a JavaScript runtime built on Chrome's JavaScript engine. NodeJS is lightweight and efficient by using event-driven, non-blocking I/O model [4].

## III. COMPONENT USED:

- A Raspberry Pi 3 model B.
- A DC motor:
  A device which converts Direct current electrical power to mechanical power. The DC motor used for the feed dispense is of rating 4.5 V and 500mA.
- Load cells (1 Kg, 10Kg):
  A load cell is a transducer which is used to create electrical signals whose magnitude is directly proportional to the force being measured on it [5].
- HX711:
  An analog to digital converter that is used to amplify and convert analog signals. It is used to convert the analog signal from the load cell to the Pi [6]
- L293DNE:
  An integrated circuit that is designed to provide bi directional currents of up to 1 A and voltages from 4.5V to 36. It is used to run the motor using Pi and power source [7].
- Three C batteries for 4.5 V power source (1.5V*3).
- A breadboard to connect components together.
- Our prototype used PetMate Automatic Cat Feeder frame.

## IV. Architecture:

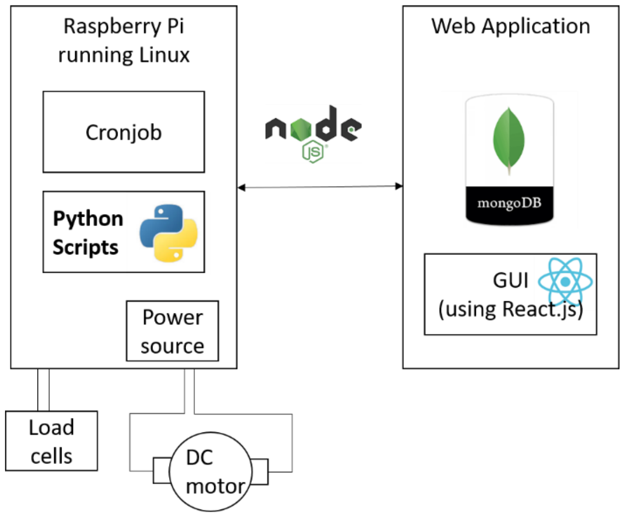

Figure 1:Architecture of Cat feeder

## V. IMPLEMENTATION

The prototype built uses a Raspberry Pi to control the functions of the feeder. The Pi is connected to web application consisting of Mongo DB and a simple GUI.

**The Hardware:**
A Raspberry Pi was used for controlling the feeder and store data offline. The feed is dispensed by a 4.5V 500mA DC motor, a 1KG and a 10KG load cell to weigh the feed and cat respectively.

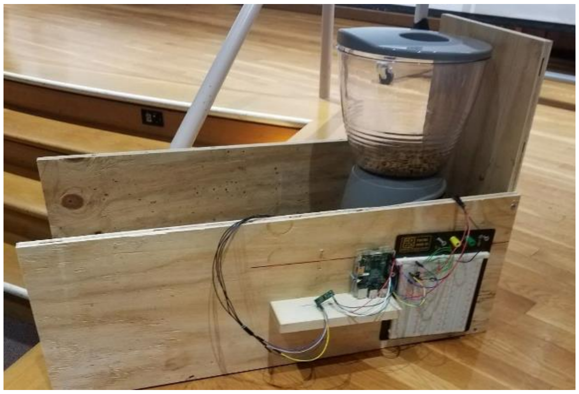

Figure 2: Cat Bistro

The Raspberry Pi can supply 5 volts but the current delivered by the Pi is in milliamperes which is not sufficient to rotate the motor. If the motor is directly connected to the Pi there are high chances that the motor may fry the Pi to gain current. To avoid this, the

DC motor is controlled by the Pi using an IC(L293DNE) which draws power from a separate power source and control information from the Pi. The Pi sends control signals to the IC, which in turn utilizes the power source to amplify the signals to a level where the DC motor can be operated. A load cell is a transducer which is used to create electrical signals whose magnitude is directly proportional to the force being measured on it [5]. The power of the signal is not high and measurements cannot be measured accurately if the load cell is connected directly to the Pi. To overcome this, each load cell is connected to an analog to digital converter (HX711) [6] which converts the analog signal from load cell to digital which is then provided to the same Raspberry Pi.

**The Software:**

The Raspberry Pi has Linux running on it to cater the needs. The Pi uses separate python scripts, "catBistro.py" to control the DC motor [8] and "scale_calibrations.py" and "hx711_python_2.py" scripts to calibrate, retrieve data from the load cells and convert them into a usable format [9]. The data is then sent to the webserver. For remote modification of the timing and quantity of the feed a Mongo DB is setup. The feeder can be controlled using a web application hosted. Both the services are hosted on AWS. Any changes in the web application will be stored in the Mongo DB. The Raspberry Pi continuously pings the Mongo DB, whenever there is a change in the records the Raspberry Pi pulls in the new data using Node.js and modify data stored in Raspberry Pi. A scheduled cronjob runs the scripts which controls the motor and the feed is dispensed accordingly. The web application consists of a NoSQL database (MongoDB) and a simple GUI for the front end which is built using React.js libraries. APIs are used to interact within the database GUI, Raspberry Pi. The "feedcat" API is used for storing the data into database from the GUI. The user uses this to change the settings of the feeder. The "getfeedreport" API fetches the feed data and projects it to the user in form of a report when requested. The "getData" API is used to fetch the entire data stored, which can be used for analytics and reporting.

## VI. Links

- Git hub
- Video

## VII. Challenges

- Working with hardware in general is a challenge. We had issues using a specific type of load cell and spent hours trying to figure out why they weren't working. Thankfully we had another type of load cell as a backup. Receiving the digital output from HX711 converter and converting the output to correct and usable units for weights using python also took a lot of time to debug.
- Making the Pi continuously ping the DB for checking and collecting the new information was a tough task.
- Making sure the back and front-end systems are highly available was a challenge. We used AWS and local storage for uninterrupted feed dispensed.
- The overall design of the cat feeder took quite a bit of time and work. Making sure that we built the feeder in such a way that the cat sits on the weight sensor while feeding was a big challenge. We verified that our design works with real cats.

## VIII. Future scope

- After collecting enough amount of data, we would like to perform data analytics to provide suggestions on the feed quantity and timings to the user.
- A mobile application can be developed for ease of use.
- Feeder design should be made to take a much smaller footprint.
- Feeder can be made more accurate and more informative with sensors to detect how much food is in the food hopper. Perhaps we can provide real time feeding cat patterns by detecting moments that cat enters the cat feeder even if it is not feeding time. This will tell us that the cat is most likely hungry.
- We can minimize cost by converting the Pi to a specific circuit which caters the required functionalities. This can be outsourced for production.

## IX. Conclusion

The developed feeder can dispensed feed using a DC motor controlled by the Pi. The dispensed feed will be weighed by a load cell to check whether the feed dispensed is of right quantity and stored for further use. This eliminates the possibility of improper feed dispensed. In addition, the cat's weight will be measured and stored for cat health monitoring. The feeder is connected to a web application though which it can be monitored and controlled. This provides user the advantage of scheduling the cat's feed any time, any amount and anywhere in the world.

## REFERENCES

[1] stesmo, "Python (programming language)," [Online]. Available: https://en.wikipedia.org/wiki/Python_(programming_language).

[2] R. g. guy, "JavaScript," [Online]. Available: https://en.wikipedia.org/wiki/JavaScript.

[3] stesmo, "React (JavaScript library)," [Online]. Available: https://en.wikipedia.org/wiki/React_(JavaScript_library).

[4] "node JS," [Online]. Available: https://nodejs.org/en/.

[5] Hooperbloob, "https://en.wikipedia.org/wiki/Load_cell," 8 December 2017. [Online].

[6] A. Semicondoctors, "HX711 -24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales".

[7] T. instruments, "L293x Data sheet," [Online]. Available: http://www.ti.com/lit/ds/symlink/l293.pdf.

[8] S. Monk, "Controlling a DC Motor," [Online]. Available: https://learn.adafruit.com/adafruit-raspberry-pi-lesson-9-controlling-a-dc-motor/overview.

[9] elix, "Build a digital Raspberry Pi Scale," [Online]. Available: https://tutorials-raspberrypi.com/digital-raspberry-pi-scale-weight-sensor-hx711/.

[10 "wikipedia.org," Wikimedia Foundation, [Onlin Available: https://www.wikipedia.org/.

[11 "License for HX711 code from the following repository:," [Online]. Availab https://github.com/gandalf15/HX711.

[12 " GPIO pins for Raspberry Pi Model B," [Onlin Available: https://www.raspberrypi-spy.co.uk/w content/uploads/2012/06/Raspberry-Pi-GPIO-Layout-Model-B-Plus-rotated-2700x900.png ( kB).

[13 " Using load cells and hx711 with Raspberry Pi [Online]. Available: https://tutoria raspberrypi.com/digital-raspberry-pi-scale-weigh sensor-hx711/.