BananaPi: a Tangible Communicative Device

Matthew Co
mc15184@bristol.ac.uk
University of Bristol

Jiayi Hong
sn18225@bristol.ac.uk
University of Bristol

Gizat Makhanov
tq18441@bristol.ac.uk
University of Bristol

Dameli Ryspayeva
dr15197@bristol.ac.uk
University of Bristol

Yunchang Zhou
ef18272@bristol.ac.uk
University of Bristol

ABSTRACT

BananaPi is a task management system that organizes grocery list between flatmates. It mixes old-fashioned ”pen and paper” and contemporary ”fully online” styles of task scheduling. The system uses fridge magnets to represent food items, computer vision to identify and map the magnets to the corresponding products, and cloud services to transfer the extracted list of groceries to mobile devices of all flatmates. In the following paper, we present our first prototype and examine it in detail by discussing the strengths, faults, scalability, possible improvements and applications of our system.

CCS CONCEPTS

•**Human-centered computing** → **Graphics input devices;**

KEYWORDS

Interactive Devices; RaspberryPi; Computer Vision; Communication;

INTRODUCTION

According to the preliminary study that we conducted to identify the common trends among university students who share food with their friends or flatmates, there are 4 main ways students use to organise

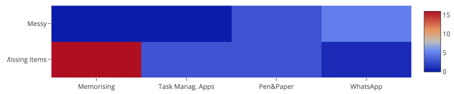

Figure 1: Main problems encountered by using a certain method of communication.

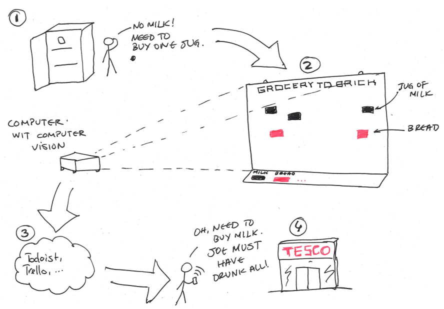

Figure 2: Representation of Our System.

their grocery lists and there are 2 main problems that they encounter when using their preferred method.

The 2D-histogram (fig. 1) indicates that most students prefer not to organise a list of groceries but to memorise everything they need to buy which leads to them often forgetting what needs to be bought. The second most popular way is to use WhatsApp or similar apps, however, as shown in the fig. 1, most find it to be too messy for organising grocery lists as information often gets lost among other messages. Interviewees whose preferred method is to write down the grocery list on paper also see this way of organising both messy and not necessarily preventing from forgetting to include something they need in the to-buy list.

To address the aforementioned problems we propose a solution that we believe can significantly reduce the chances of people forgetting to note down what needs to be bought as well as we present a device that can connect several generations and enable a better information exchange between them.

User Case

Jane, Ann, and Erin rent a three-bedroom apartment and share one fridge. One day Jane went to have cereal with milk for breakfast and used up all milk. She then decides to let Ann and Erin know that the milk is gone and ask somebody to buy a jug on their way home.

She approaches the fridge and picks white magnet, that represents 0.5 L of milk, and sticks it to the fridge's front door.

A Raspberry Pi with a camera module and Computer Vision software, that takes a photo of the fridge every 5 minutes, then sends notifications to both Ann and Erin as a TO-DO task, that they need to buy milk when it sees changes (i.e. a newly placed magnet).

RELATED WORK

The "Bit Planner" project [1] inspired us to look into the ways we could possibly mix the human-computer interaction and high-tech functionality to manage workload. Bit planner users have a physical calendar board made out of Lego bricks only on which they can place their Lego blocks that represent their projects or tasks. They can then synchronise it with their online calendars by simply taking a photo of the whole Lego board which will then update the calendars' of all collaborators. However, the need to take photos of the board and the fact that one needs to remember which colours of the Lego blocks correspond to which projects could be perceived as drawbacks of the system which brings us to the differences between our devices. Our system addresses these issues by making use of food-shaped fridge magnets and Computer Vision to automate the updating process and remove any ambiguity as well as applies the idea to a different scenario - organising to-do lists.

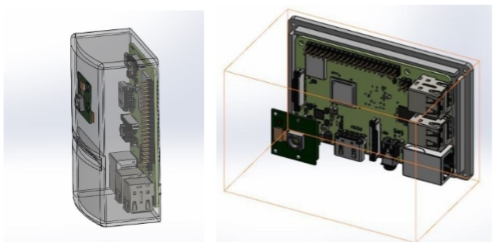

Figure 3: The device's case

Figure 4: BananaPi's First Prototype.

Table 1: Comparison of COCO trained models

Model name	*Speed (ms)*	*Accuracy*
ssd_mobilenet_v2_coco	31	22
ssdlite_mobilenet_v2_coco	27	22
faster_rcnn_inception_v2_coco	58	22
faster_rcnn_nas	1833	43

DESIGN

Our first prototype of the BananaPi has a small 3D-printed case inside which there is a RaspberryPi computer together with a RaspberryPi camera module. Together they enable image detection with the help of the Commputer Vision object detection algorithm running on the computer that was trained to detect the 3D food magnets we made. We designed a case in such a way that one does not need to take out the RaspberryPi in order to debug or start the program as the case has special holes for all the necessary cables.

Hardware

Software

The system uses Raspbian Stretch OS version 4.14, a Debian-based operating system for Raspberry Pi.

Tensorflow is an open-source machine-learning framework that provides object detection API. We used Tensorflow version 1.12. There are multiple object detection models in the market, such as SSD, R-CNN, and YOLO. Each of these models have various level of complexity, therefore because of the limitations on the hardware of Raspberry Pi, we use SSD. The model is fast, however, the main drawback is itffs accuracy in detecting objects. The comparison in efficiency of each models is shown in Table 1.

As with the models, there are several datasets with images, COCO being the largest one. Our chosen model SSD is pretrained on the COCO dataset.

The software implementation consists of three main components: image capture, object detection, and communication (Figure 5).

The image capture component captures a frame from the Raspberry Pi camera. It then uses Tensorflow and models to detect predefined set of products. If there it finds a product in the image, it will create a task in Wunderlist task management application via Wunderlist API. The system waits a predefined time and starts the detection process over.

APPLICATIONS

We designed a prototype for groups managing a grocery list. However, our system can be used for different purposes. Essentially, anything that one can represent as a to-do list (e.g. group work or timetable management), our device can operate on. Even if we consider a visa application process which requires one to collect all the necessary documents, our system can help with this. However, some alternations would need to be made such as redefining the mapping between magnets and tasks.

Figure 5: Steps to Object Detection

FEEDBACK FROM USERS

During the demo day we asked people to assess the usability of our device and share their opinions on what they thought could be improved for them to seriously consider integrating it in their day-to-day life.

The results showed that everyone we talked to strongly agreed that the device is very straightforward to use, the fact that it is small and therefore does not take a lot of space was noted by many of our users to be a big positive and most people agreed that BananaPi makes a step towards bridging the gap between technology-proficient people and not so by catering to the preferences of both due its tangible and online nature.
All our interviewees strongly agreed that the functionality of the current prototype was not sufficient for a daily life but many agreed that with a wider range of functions and the ability to place it farther away from the fridge they would consider using it.

POSSIBLE IMPROVEMENTS

Our first prototype has certain hardware restrictions that greatly affect its functionality:

(1) The RaspberryPi is not powerful enough to cope well with more advanced Computer Vision algorithms that have more complex databases. Due to this we had to resort to using lighter versions of Computer Vision object detection algorithms which resulted in the BananaPi being able to recognise only a small number of food items.
(2) The RaspberryPi camera module also proved to be not suitable for such device as when placed farther than one meter away from the fridge the magnets become unrecognizable due to the low resolution of the camera.

BananaPi v2

Having analysed the feedback we received during the demo day, for the next version of the BananaPi we plan to do the following:

(1) Replacing the RaspberryPi 3B+ computer with a more powerful one such as Odroid-XU4 or even Huawei HiKey 960. This will enable us to run more advanced object detection algotithms with more sophisticated databases which will therefore make our device recognise more food items.
(2) Using any better camera that can take good pictures when placed several meters away from the object.
(3) Integrating motion detection and night vision to trigger recording of the board only when someone is next to the fridge.

(4) Adding a ”remove” functionality that allows users to delete an item from the list by removing a magnet from the fridge.
(5) Implementing a two-way communication (fridge < – > phone) rather than a one-way (fridge – > phone) information exchange.
To do this, there are two cases that should be considered: somehow informing users who placed a magnet on the fridge that the to-buy list has been updated, and informing them that a magnet needs to be removed if someone ticks off the food item in the grocery list. We think to implement the former by adding speakers to the device that will produce sound when the computer sends a command to do so (i.e. after updating the to-buy list) and the latter by adding light lamps that will light up when something has been bought so that people at home then could know from looking at the flashing light of the device that they need to check their phones to see what has been bought and to remove the corresponding fridge magnets.
(6) Last but not the least is providing the ability for users to see system error messages in case if something goes wrong by adding a display to the device.

Ambitious Improvements

The expressive power of our interactive device is limited in the following way: when users place a magnet on the fridge, a predefined message is sent notifying their roommates about what should be bought, meaning that the users cannot create custom messages. For example, they cannot specify what kind of milk to buy, of which brand and where to buy it from etc. To enable them to do so, we can make use of virtual languages such as Ruru [2] which enables programmers to code robots’ actions without actually typing anything but by using graphical symbols only. So, we could employ similar technology to make our device recognise more complex messages. So, for instance, by placing a milk magnet and a magnet of a farmer next to each other, the user should be able to define a message ”buy milk from the farm” which the device could understand and translate to a text message that will then be added to the to-buy list of all flatmates.

Merits

From the verbal feedback that we received during the demo day, we can sum up the following advantages of our system:

(1) Usability. It is easy and very straight-forward to use meaning that even people who are not technology-advanced (e.g. the elderly) can use BananaPi.
(2) Size. The BananaPi can fit into one’s palm so it does not take a lot of space.
(3) Speed. It enables fast communication. Currently, the system takes 3-5 seconds to recongnise a magnet and updating of the to-do list happens with a delay of 1-4 seconds, while placing

a magnet on the fridge should not take longer than 3 seconds. Therefore in the worst case contacting flatmates should not take longer than 12 seconds in the worst case.

Learning Outcomes

Throughout the project, we encountered not only software related but also hardware problems that challenged us to get outside of our comfort zone. For the hardware part, we learned how to fully set up a functioning raspberry pi system and install various software on it as well as how to 3D model, 3D print and laser cut anything that we needed. We consider these to be very useful skills as we essentially have built a functioning device from scratch. For the software part, implementing a new communication way forced us to learn new Computer Vision techniques and think about the community and how we could possibly make the lives of people easier and more fun. We met problems that we eventually could not resolve so easily such as running out of memory and failing because of that to train a strong model. However, we learnt from our mistakes and came up with solutions to the aforementioned problems that can be integrated in the future versions of BananaPi.

CONCLUSION

In this paper, we have introduced BananaPi, an interactive and efficient management system designed to organise grocery lists between flatmates. With simple interaction with physical products, it frees users from time-consuming typing and memorising. Conceptually, BananaPi offers a new method for visual communication and it's use can be extended in many areas, such as project management and education.

REFERENCES

[1] Bit Planner. (2018). Bit Planner. [online] Available at: http://www.bit-planner.com/

[2] J. P. Diprose, ”End user robot programming via visual languages,” 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Pittsburgh, PA, 2011, pp. 229-230.