Intelligent Multi-Fruit Recognition and Maturity Assessment Platform: Design, Development, Validation & Exploitation

Jihen REZGUI*, Lamia FOURATI†, Thomas TREPANIER*,
David GENOIS*, François Dufour-Martel *, Antoine Labelle*
* Laboratoire Recherche Informatique Maisonneuve (LRIMa), Montreal, Canada
† Digital Research Center of Sfax (CRNS)
Laboratory of Technology and Smart Systems (LT2S), Tunisia
jrezgui@cmaisonneuve.qc.ca, lamiachaari1@gmail.com, thomtrepanier@gmail.com,
david.genois13@gmail.com, francois-dm@hotmail.com, antoinelab01@gmail.com

***Abstract*—The emergence of the artificial intelligence and the Internet-of-Things (IoT) based technologies, reshaped several domains including "smart farming". Indeed, the utilization of machine learning and its associated algorithms has been increasing in the smart farming due to its ability to enhance production and assess the quality in an accurate and cost-effective way. In this context, this paper proposed an intelligent multi-fruit recognition and maturity assessment platform (IMFRMAP) based on smart image recognition that implements Convolutional Neural Networks (CNN) to analyse in real-time pictures of fruits taken with smart cameras that could be embedded in a cell phone or in a robot and to determine the fruit maturity state. Accordingly, we explained and highlighted the overall workflow of the platform as well as the different neural networks used and tested. Beside that, we proposed and detailed two applications exploiting and integrating IMFRMAP proposed platform. The first application is built around a human support robot (HSR) capable of delivering to an elderly person or to a person requiring living support the requested mature fruit from a fruit basket containing heterogeneous fruits. The second developed and tested application is a mobile application that emulates a smart fridge. The virtual fridge assesses the maturity of the fruits inside it and sends notifications and alerts to the user before the expiration of the validity of the fruit. The obtained results proved the utility, the accuracy and efficiency of the IMFRMAP for the two sketched applications.**

I. Introduction

A. General Context

B. Motivations

As the climate crisis continues to deteriorate, the need for digital based solutions increases every day. One major factor of climate change in developed countries is food waste, from its production to its consumption. IMFRMAP is designed to help the last one, since few global and effective solutions have been developed and since consumers often want to help with this crisis, but do not know where to start or how to contribute. As a matter of fact, 63% of the food Canadians wasted in 2017 could have been eaten [2] which amounts to a 17 billion dollars lost and to a 9.8 million tonnes of CO_2 released in the atmosphere. Global action is needed, fast, and with IMFRMAP, we aim at guiding consumers worldwide in their quest to reduce their carbon footprint.
Furthermore, the 2020 worldwide health crisis showed us that our most vulnerable population, the elderly, and other people requiring living support are ill-equipped to subsist to their alimentary needs in times like these. Therefore, we expanded IMFRMAP's architecture to provide personal support through the Human Support Robot (HSR). This will allow people in need to have an aiding robot delivering them ripe or fresh food safely and quickly thanks to IMFRMAP's capabilities.

C. Main Contributions

Our contributions in this paper can be summarized as follows: (1) We introduce our Intelligent Multi-Fruit Recognition and Maturity Assessment Platform called "IMFRMAP" which can detect and isolate multiple fruits in real time and then evaluate their level of maturity using deep learning techniques; (2) We introduce and explain the IMFRMAP's architecture's main components: the Robot, the Server, the Mobile Application and the Neural Networks; (3) We propose two main ways to exploit IMFRMAP's architecture in real-world scenarios: a human support robot able to pick up and deliver requested food items at the right maturity level to a mobility-restricted individual, and a virtual fridge meant to help consumers avoid food waste by assisting them in their daily fruit and vegetable usage and (4) We demonstrate the effectiveness and improvements of our platform with IMFRMAP's functionalities and present our various approaches to the neural network component.

D. Paper Organisation

Section II provides a detailed overview of related work similar to our platform. Section III contains a thorough explanation of IMFRMAP's design and architecture as well as its different components. Section IV demonstrates our simulation results. Section V details our ideas to exploit IMFRMAP, and Section **??** concludes our paper.

II. Related Works

In this section, we discuss relevant literature that studied and contributed to the issue related to fruit recognition and maturity assessment. Accordingly, in the first subsection, we present contributions that considered artificial intelligence mechanisms toward fruit recognition and maturity assessment, and in the second subsection we review related works that focused on automated robot using fruit recognition and maturity assessment frameworks.

A. Fruit Recognition and Maturity Assessment

1) Tomato Recognition and Maturity Assessment: Authors in [8] proposed an improved maturity tomato detection model based on the DenseNet deep neural network architecture. A structured sparse operation is suggested to reduce the stored data amount and to enhance the accuracy of feature propagation. Besides that, they introduced the Focal loss function to enhance the accuracy of the tomato detection system. To assess the growth of tomato plants and its maturity in a chamber by detecting the presence of flowers and fruits, authors in [9] developed a system providing tomato fruit maturity grading. A Regional-based Convolutional Neural Network (R-CNN) and a Single Shot Detector (SDD) were used for flowers and fruits detection. Tomato fruits maturity assessments were implemented using the Artificial Neural Network (ANN), K-Nearest Neighbors (KNN), and the Support Vector Machine (SVM). Authors in [10] provided a solution detects defect tomato fruit. They tested three deep learning models (VGG16, InceptionV3, and the ResNet50) to detect defects in the generated image dataset of tomato fruits (A total of 1200 images were considered in the training and testing). According to the authors the VGG16 model outperformed the other two models in the defect detection in the tomato fruit.

2) Multi-Fruit Recognition and Maturity Assessment: The machine learning approaches in correlation within image processing play a great role to provide intelligence for an automation system that differentiate the fruits according to their type, maturity, variety and state. In this context, authors in [11] reviewed research papers (published between 2010 and 2019) that focused on fruits identification, classification and grading. A deep learning framework dedicated to multi-class fruits detection based on improved Faster R-CNN was proposed in [12]. The framework includes (1) fruits image library creation, (2) data augmentation, (3) improved Faster RCNN model generation, and (4) performance evaluation modules.

B. Automated Robot using Fruit Recognition and Maturity Assessment Frameworks

Nowadays, automated robot (AuRo) are very solicited for yield estimation, harvesting, disease control, sorting and grading fruits.Therefore, during the last decades, several fruit detection schemes based on machine vision approaches were designed and developed for AuRo. In this context, authors in [13] reviewed the research progress and recent application of vision technology and harvesting robots in fruit picking with focus on the digital image processing technology and deep learning-based algorithms used in fruit recognition and localization. In this scope, authors in [14] proposed a machine vision framework based on deep learning for date fruit harvesting robots in an orchard environment. The framework includes three classification models used to classify date fruit bunches in real time according to their type, maturity, and harvesting decision. In the classification tasks, deep convolutional neural networks are utilized with transfer learning and fine-tuning on pre-trained models and two pre-trained CNN models (AlexNet and VGG-16) were investigated. To enhance the vision system robustness, the authors created a rich image dataset including more than 8000 images of five date types in different prematurity and maturity stages. According to the authors, the obtained results showed that a pre-trained CNN lead to a robust date fruit classification without the images pre-processing. In the same context, authors in [15] suggested an intelligent fruit detection method ensuring real time and high accuracy for AuRo. The proposed method is based on an improved multi-task cascaded convolutional network and on an improved augmented method that is based on image fusion to improve the detector performance. The performance of the system have been evaluated by considering apple fruit as a case study. Authors in [16] developed a harvesting system based on the IoT technology and smart image recognition. The crop maturity is accessed via object detection by training neural network models. Mature crops are harvested using robotic arms. Keras was deployed in order to construct a multilayer perceptron machine learning model and to predict robotic arm position and movements and a MobileNet CNN was used for image feature extraction model.

In [17], different approach strategies for harvesting robot were compared in terms of cycle times and failure rate. The studied approach strategies are: (1) in-field assessment by human observers, (2) image analysis using advanced algorithms or remote human observers, (3) multiple approach directions until the fruit is successfully reached. The authors evaluated the performance of the different approach strategies for a sweet pepper harvesting in greenhouse and laboratory conditions.

Fruit sorting is another challenging issue in agriculture domain, sorting robot could be used to overcome this issue. Authors in [18] presented a system for four fruit sorting (red and green tomatoes and red and green grapes). The proposed system uses a sorting robot that is based on size and color in a packaging system.

III. Proposed IMFRMAP: Design & Architecture

In IMFRMAP's current state, it is made of four vital components: (A) The neural networks, (B) the server, (C) the mobile application and (D) the human support robot (HSR). Its general purpose is meant to be applied to many challenges. We will detail the functioning of the different components and explain our choices in their regards. Figure 1 sums the sub-components and relationship of the platform:

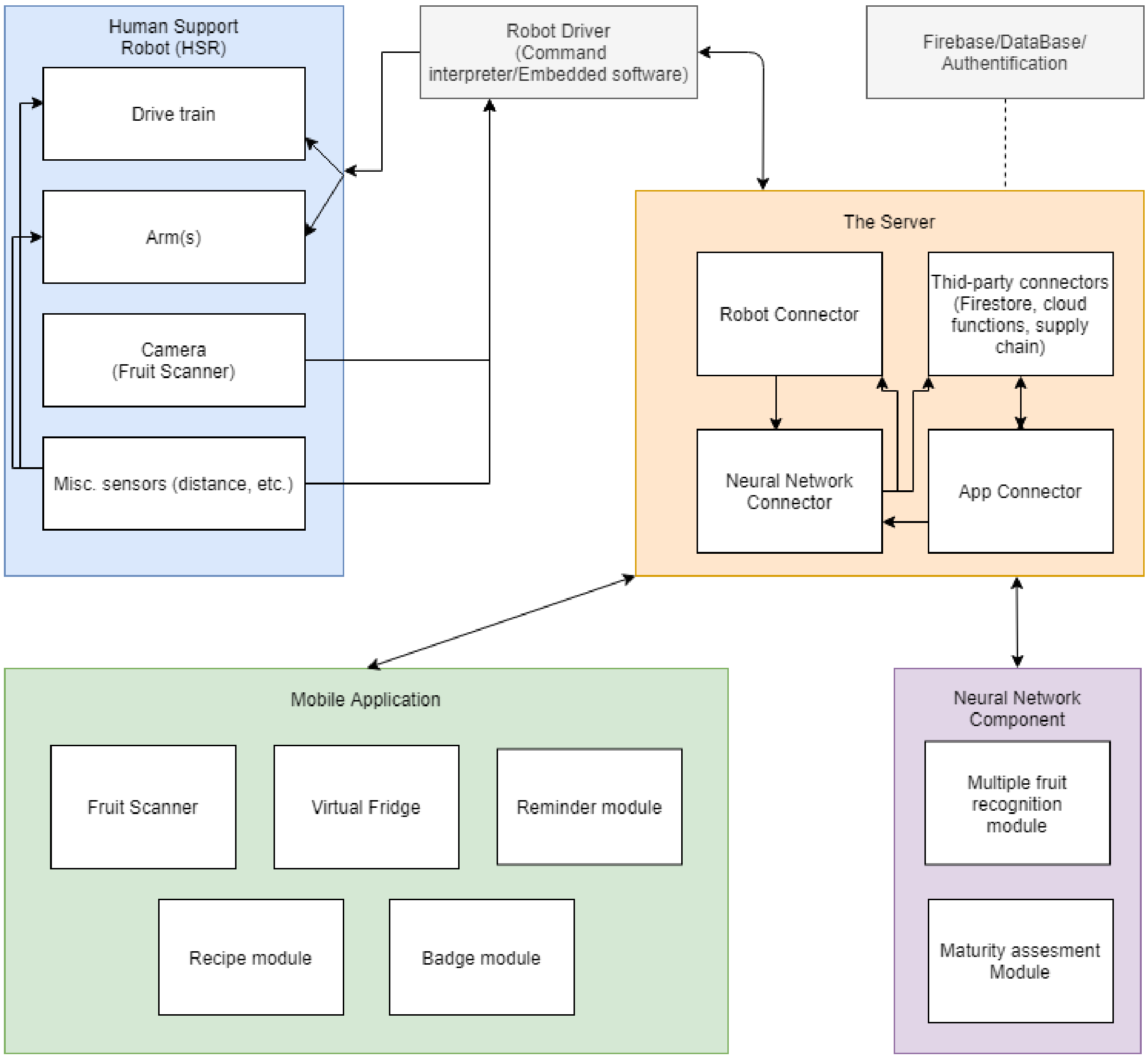


Fig. 1: IMFRMAP's overall architecture

A. Neural Network Component

IMFRMAP's smart camera capabilities such as fruit detection and maturity assessment are powered by deep learning algorithms. These algorithms are able to learn to recognize and detect patterns in a wide range of data which is great to generalize the concept of "fruit" and "maturity" for a computer, as it will learn to understand relationships between segments of data and their interpretation.
In our case, we use convolutional neural networks (CNN) which is a type of deep learning algorithm specialized in image classification. This type of NN is well suited to assess a large amount of data consisting of images and learning to label them by developing an understanding of what aspect of the image will impact its label.
As an example, we taught a CNN to label avocado with the exact number of days that have passed since it was bought. This result can then be used to assess the avocado's maturity and help customers buy an avocado that will be ripe at the right time.

1) Multi-Fruit Detection: IMFRMAP's real-time multi-fruit recognition feature is enabled by another deep learning technique which is called "object detection". In contrary to image classification which is an algorithm that will label an image as a certain object, an object detection NN will try to identify and tell where the object is in the image instead which is necessary when dealing with a lot of fruits in a single image, as we cannot label the image as one fruit in particular. Figure 2 shows the difference:

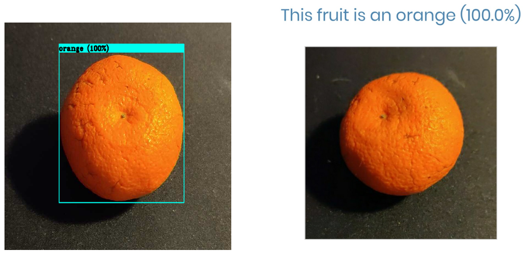


Fig. 2: The difference between object detection (left picture) and recognition (right picture).

In our case, we decided to use the object detection NN architecture named YoloV4 [3] which is one of the state-of-the-art architecture in real-time object detection as of now based on the Darknet Framework [4]. Since our platform has to be running in real-time, YoloV4 is a good balance between performance and average precision by improving on its predecessor, YoloV3 [5]. Here is an example of YoloV4 performing in real-time on a smartphone:

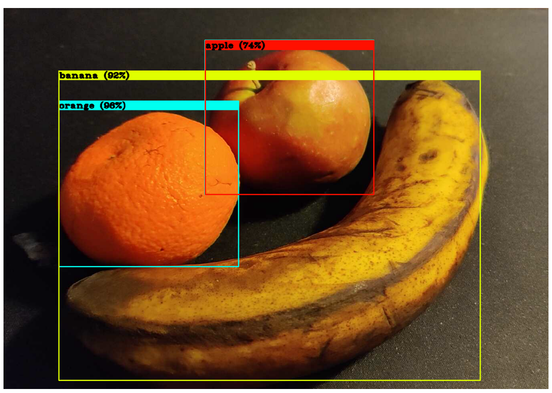


Fig. 3: YoloV4 detecting an apple, a banana and an orange in a single frame on a smartphone

2) Maturity Assessment: The maturity assessment feature of IMFRMAP is the central piece of the platform. It allows us to meet a customer's needs perfectly by determining how many days a supported fruit has before it rot. This is especially useful both while shopping to ensure that the customer will get the best fruit available to him or her, and while having it at home to know exactly how much time he or she has before having to throw the fruit away, giving them a better grasp on when to consume it.

This maturity assessment is done using a convolutional neural network (CNN), which is a type of deep learning algorithm specialized in image classification. These algorithms are able to learn to detect patterns in data and to use that information to label the image with the appropriate class. As of now, our maturity assessment CNN is able to classify an apple, a banana and an orange as either "Fresh" or "Rotten", and is capable of labelling an avocado with the exact number of days that have passed since it was bought and therefore determine how many days are left before the avocado is rotten.

Following the workflow of the platform, each fruit detected by the YoloV4 NN is then cropped from the original image to isolate it and its data from the rest of the picture. Then, depending on which fruit was detected, we send its picture to the appropriate CNN which was specifically trained to assess the maturity of the fruit. In fact, we trained a separate maturity assessment CNN for each fruit as we knew that this classification was complicated to grasp for a CNN. Therefore, each fruit has its own dedicated CNN which learned to assess the maturity of this fruit in particular. Usually, NN are good at generalizing concepts and you want your NN to understand as many of them as possible, but we knew that the key pieces of data used to label a rotten avocado and a rotten apple were not the same, and didn't want our NN to think that those two different concepts were the same.

In the end, this does not slow down our platform since each CNN is preloaded and ready to use, and we simply ensure the accuracy of the CNN. These multiple CNNs also allow us to have different labels depending on the fruit. As you saw, avocados are labelled daily, but apples, bananas and oranges are simply labelled "Fresh" or "Rotten".

We tried and tested multiple CNN architectures for this task such as the LeNet, AlexNet, SimpleCNN, SqueezeNet, which ended up getting outperformed, and the VGG16 and ResNet which were too large and slow to run for a real-time application. Section IV-A demonstrates our results according to each architecture.

3) Expanding the Maturity Assessment capabilities: As of now, we are only able to assess the maturity of four food items because of the lack of data on this matter. However, we can easily increase the number of products we can detect or recognize because such data is much more abundant. For instance, we can detect broccoli and carrots through our multi-fruit detection component, but we cannot assess their maturity because of the lack of data to train such a neural network.

In this regard, to expand the capability of the maturity assessment component, we collected data about the average life expectancy of a wide variety of food items which we will use to guide the consumer or the HSR instead. This will not be as precise as a neural network computation, but in the case that we encounter a fruit or a vegetable for which we cannot assess its maturity, we will be able to estimate it through this average.

The data we collected accounts for different storage environments such as a fridge, freezer or at room temperature. Through the IMFRMAP's mobile application, the user will be able to enter where and how long the fruit was stored and that information will be used to estimate the current maturity of the product.

Table II shows an example of the data we collected.

4) DataSets: As mentionned in III-A2, we trained our CNNs on fruit images datasets to teach the neural networks to classify an apple, banana or orange as either "Fresh" or "Rotten". To achieve this task, we used the "Fruit fresh and rotten for classification" dataset [6]. This dataset contains 10,901 samples of the aforementioned fruits as either fresh or rotten. However, many pictures in this dataset were inadequate

for training, so we had to clean it up and remove this unwanted data.
We ended up with a dataset of 9503 samples for the CNNs to train and test on. Figure 4 shows such example of this data.

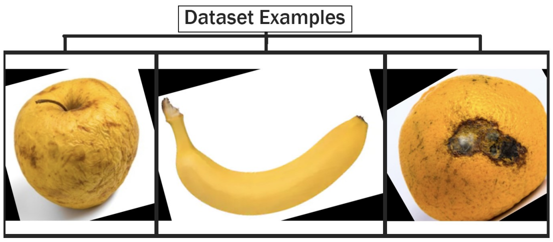


Fig. 4: Piece of the dataset used to train the maturity assessment CNNs on apple, oranges and bananas. (From left to right: Apple-Rotten, Banana-Fresh, Orange-Rotten

As for the avocado classification task, we bought 28 avocados and manually took daily pictures of them in real-world scenarios over the course of 12 days. However, we also wanted our platform to be able to assess the maturity of an opened avocado, so we cut up 8 of them and also took daily pictures over 12 days while they were stored in the fridge as opposed to closed ones which were stored at room temperature.
We then labelled each avocado picture with the day it was taken which end up being the classes our CNNs have to determine. We also separated the closed avocado dataset from the opened avocado one, because we wanted to train a separate neural network for each. Figure 5 shows examples of the picture we took and of their labels.

Fig. 5: Piece of the dataset used to train the maturity assessment CNNs on avocados. (From left to right and top to bottom: Close Day 1, Close Day 12, Open Day 3, Open Day 10

B. Server

The backend server stands as the core of our project's flow. It serves the purpose of allowing the neural networks and the application to communicate together and gives us more flexibility as to how we will handle the data.
Its middleman position allows us to filter the requests which then open multiple possibilities to us that we will detail further in this part.
No matter if the data comes from the app or a robot, the results are sent to the backend who will register them in the database with the relevant timestamps and send the results of the analysis back to the application. The end user gets to know what fruit have been captured, its freshness and the neural networks' degree of certainty of the answer once the results are received. The stored results in the database allows us to have a better grasp of the neural networks' evolution and, through the history, double-check the results to correct its behavior.

1) Firebase: Our backend implements Firebase[?] for user data storage and authentication. The approach we have taken gives us the opportunity to evolve the platform effortlessly.
Firestore[?]'s no-SQL solution makes it a lot easier to handle data straight from within the app. Each user has their own document with all of their app data within to quickly load and parse data they will need while browsing through the app to avoid complex queries that would take a significant amount of time to complete. Storing data in this format also makes it directly usable within the app.
Firebase authentication allows us to externalize the authentication module for effectivity and security purposes. A user can create an account from a given email, and delete their account along all their data whenever they wish to.

C. Mobile app

Other than the Neural Network component which is the main tool our platform uses to identify fruits and their maturity level, the platform's application is built on a modern hybrid mobile app framework (React Native [?]). This component will be referred to as the frontend component. The frontend and neural network components are linked together by the server-side application as mentioned previously. It will be explained in more detail in the next bullet point.
The front end side of the platform is the first component of the application users interact with. Therefore, it plays a critical role in the user adoption rate of the platform. This is the basis of our core design goal to provide an intuitive user-friendly environment where the end user of the platform can scan fruits and then receive information based on the results. We intend to keep the core experience available to all as soon as they enter the app without having to create an account to hook our users before walling features behind more clicks.
The mobile app contains several modules which will be explained down below. The first module users see is the fruit scanner as it is at the core of the platform. There is also an account system to unlock the virtual fridge module and the reminder module. These two modules will be the center point of the logged-in user's experience whether they have a fruit to analyzes or a pre-made meal to manually keep track of. Any fruit processed through the platform can ultimately be added to the virtual fridge.

1) Fruit Scanner: The fruit scanner within the app is used to take a picture from within the app, or upload an existing one to the server in order to receive results on its current status. Currently, the user needs to center the fruit. However, as mentionned in III-A1 and III-A2, with the introduction of the multiple-fruit detection component, the AI will automatically detect and crop different fruits within the picture to assess their maturity individually.

The identification of the fruit and its maturity could be used to provide the user with various information on their fruit to make better-informed decisions. Examples of information which can be given to the consumers to help them make informed decisions in the short-term and long-term are how to potentially identify if a fruit is good by analyzing the texture or how to conserve it more efficiently. It is important to keep in mind that identifying a fruit's maturity precisely only with an image is, to the best of our knowledge, not possible. Many aspects of a fruit's current state are invisible to the naked eye and require sensors which consumers typically do not have [?] as explained in [?]. Figure 6 shows an example of such an analysis by the neural network.

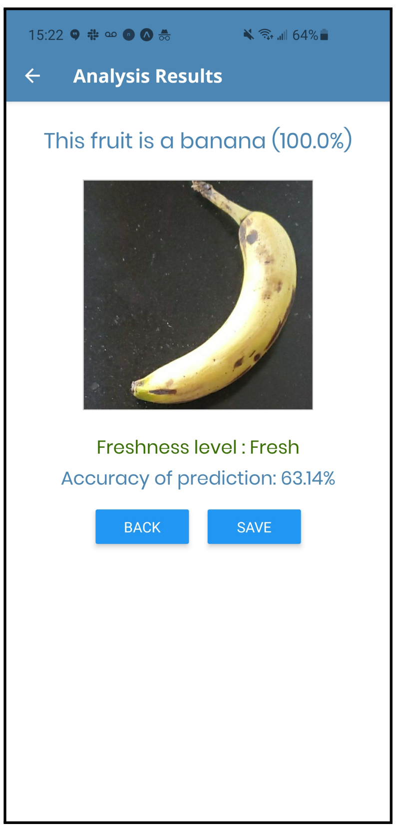


Fig. 6: The result of the Fruit Scanner analysis on a banana

Once a fruit was successfully analyzed, the user can add it to his virtual fridge directly to make it more convenient for him. Our goal is to provide a central point for a user to manage his items. All the possible use cases developed will lead to items being insertable in a user's virtual fridge.

2) Virtual Fridge: The virtual fridge (Fig. 7) is a module in which a user can manage all their products. We aim to reduce food waste by providing the user with this central point in their item management. For simplicity, we will refer to all items that can be added to the virtual fridge such as fruits or canned goods as fridge items.

A fridge item can be created after scanning a fruit through the scanner, or it can be manually entered in the case of canned goods, pre-made meals or other items not supported by the scanner. In the future, we plan to incorporate the robot module's choices to the virtual fridge directly. This will speed up the process by a lot as a user's whole grocery order could be added to the virtual fridge automatically.

A fridge item consists of a few details such as the known or projected expiry date, user entered notes, list of reminders and any other valuable information that was either entered or collected through the existing modules. To help recreate a physical setting, users can create categories to group their goods in order to separate them how they are in their kitchen. Our goal is to reduce food waste through various means. By having all items at hands, users are less likely to forget certain goods. Moreover, the reminder system provides the possibility to get a warning at a given date. It can be manual to plan a week's meals, or it can be automated.

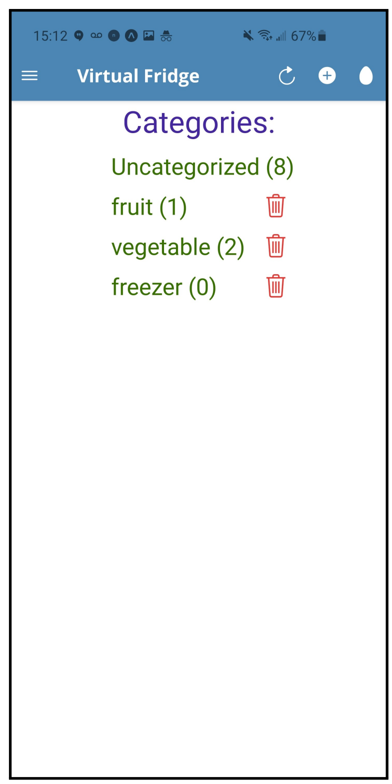


Fig. 7: An example of a virtual fridge layout

3) Reminders Module: A module closely related to the virtual fridge is the reminder module. It consists of notifying the user about a fridge item on a given date (see Fig. 8). The reminders are sent through notifications on a user's connected devices. Reminders can be created manually or automatically. The automatic reminder system creates reminders when an

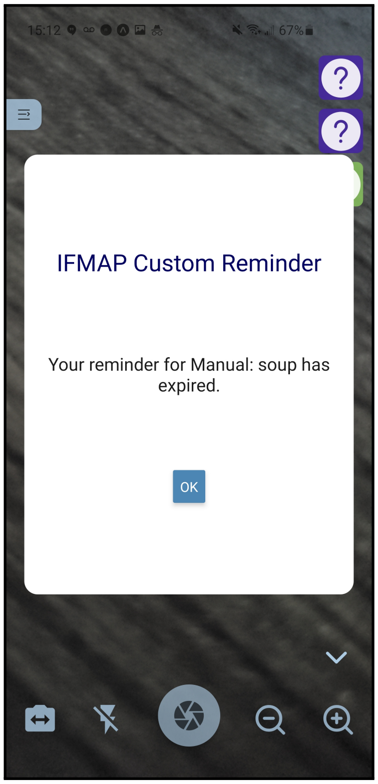


Fig. 8: An example of a reminder alert in the mobile application

item is created. Currently, the default is 20 percent of the time between now and its expiry date, but it can be tweaked by the user. This allows us to make sure nothing is forgotten about.
Ultimately, our goal is to use the data collected and our neural network component's prediction to create reminders at optimal dates automatically. This could vary depending on the type of fruit's average lifespan, or the current fruit's status.
The accumulated information, paired with the reminders, could allow us to suggest recipes from our database to our users that match their fruit's current status. By providing the user with options to use their goods, we lower the odds of them throwing it away because they think it cannot be used anymore.

4) Smart Recipe Proposal Module: One of the other modules of the app is the recipe module. This is our next development goal when it comes to the frontend. What we are going to develop is a section within the app that contains lots of recipes using fruits the app can analyze, and even more. This would allow us to give suggestions of meal to the user to decrease the odds of them throwing away a fruit they could have still used for something.
The goal is to identify the quantity and maturity of items inside the virtual fridge to find the most relevant recipe for the given user. This would decrease the chances of suggesting recipes the users cannot make at home, making us fully leverage the information provided by the virtual fridge.

5) Badges and user retention: Part of our goal, as stated before, is to allow users to feel like IMFRMAP can be their central point to manage their products. To keep the user coming back, we are experimenting with a badge module (Fig. 9). This module will award achievement-like badges to keep the user engaged in the platform. Some badges can be earned through scanning items or creating reminders, other badges can be rewarded for special contributions or support such as making a contribution on any of the repositories.

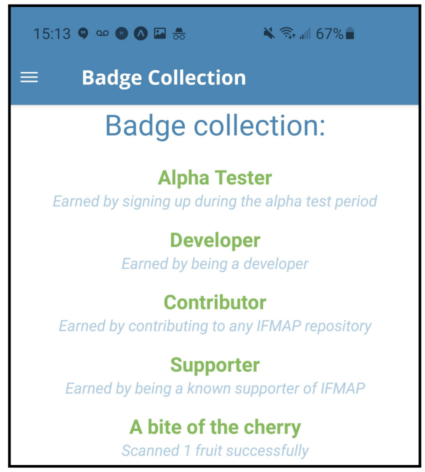


Fig. 9: The list of the currently available badges in IMFRMAP's mobile application

Moreover, progress is constantly tracked which gives us the ability to analyze how our platform is used and to tweak the platform accordingly based on our findings. We expect that understanding the daily habits of our audience is key to making a platform that fits their needs.

D. Robot

The robot component is used in the first application of IMFRMAP's architecture. It is meant to help mobility-restricted people by picking up and delivering an ensemble of food items to them. What puts this robot apart is its capability to exploit IMFRMAP's tools such as the fruit scanner and the virtual fridge to automate and improve the process.

1) Drive Train: The drive drain is the component that will allow the robot to move around either the warehouse or the house of the user. This robot will be grounded, therefore the use of all-terrain wheels would be appropriate.

2) Picking mechanism: The picking mechanism is the component of the robot that will allow it both to pick up food items in front of him and store them in a container on itself. The use of non-skid arms would be preferred to avoid letting food items slide off. A small basket or container on the robot itself would be used to store items during the picking phase and then easily finalize the order process by delivering all the items at once to the user.

3) Smart Camera: The robot's on-board camera has two smart purposes: (1) Recognise the food basket and approach it safely using distance through lens algorithms and (2) Help the robot to pick up the right item at the right maturity level using IMFRMAP's neural network component.
First of all, the camera will use IMFRMAP's multi-fruit detection neural network to detect when a fruit is in front of the robot (Fig. 10).

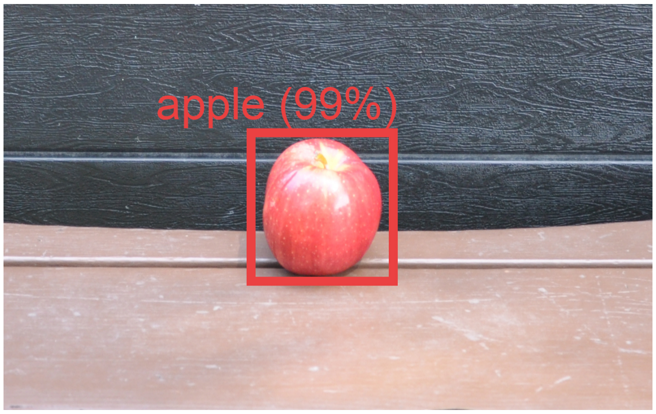


Fig. 10: Example of the view of the Robot

Then, the robot will have to approach the fruit and stop at the right distance. To achieve this, we compared two distance computing equations to decide the most accurate:
Equation 1:

$$D = \frac{f * r_h * i_h}{o_h * s_h} \tag{1}$$

Where

- D is the distance from the object
- f is the focal length of the camera used in mm
- r_h is the height of the object in real life in mm
- i_h is the height of the image in pixels
- o_h is the height of the object in the image in pixels
- s_h is the height from which the picture was taken in mm.

The second equation comes from the following:

$$\frac{s_w * o_h}{i_w * f} = \frac{r_h^1}{D} \tag{2}$$

Where

- s_w is the sensor width of the camera in mm
- i_w is the total image's width in pixels
- o_h is the object height in the image in pixels
- f is the focal length of the camera used in mm
- r_h^1 is the height of the object in real life in m
- D is the distance between the camera and the object

Therefore, we isolate D to compute the distance between the camera and the object.
Equation 2:

$$D = \frac{r_h^1 * f * i_w}{s_w * o_h} \tag{3}$$

The real-world accuracy of these equations is presented in Section IV-C.

Second, once the robot is in proximity of the items, it will use IMFRMAP's maturity assessment component to pick the right fruit desired in the order. As an example, if the user wants an old banana to put in a banana bread, the robot will look at the banana basket and pick one that would be too old to be appetising to eat.
This novel feature lets the robot take decisions that would normally be up to the user to take, hence improving the autonomy of the robot.

4) Robot Driver: The robot driver is the brain of the robot. It is where it will communicate with the server and where decisions as to its movement will be made. Figure 11 illustrates the decision process once the server sends an order to the robot.

IV. IMFRMAP Validation: Experimental Results and Discussion

A. Deep Learning Training

As explained in Section III-A2, we trained different convolutional neural networks to evaluate the maturity of some fruits and vegetables. The hyperparameters of our the neural networks are provided in Table I.

For apples, bananas and oranges, our CNNs only classify the images between "Fresh" and "Rotten". For this task, we tried to train two different CNN architectures, AlexNet [20] and LeNet [19]. For apples and oranges, AlexNet gave better results than LeNet, while LeNet turned out to surpass AlexNet for bananas. We think that the reason for this might be that AlexNet architecture's greater complexity is an advantage for the maturity assessment of apples and orange, but increase the risk of over-fitting for easier tasks such as the maturity assessment of bananas. Figure 12 shows the evolution of the accuracy on validation data during the training, for the best architecture in each case.

TABLE I: The hyperparameters of the trained CNNs

Parameter	Value
Image size	224 x 224 pixels
Number of channels	3
Seed (Randomizer shuffle	123
Batch size	50

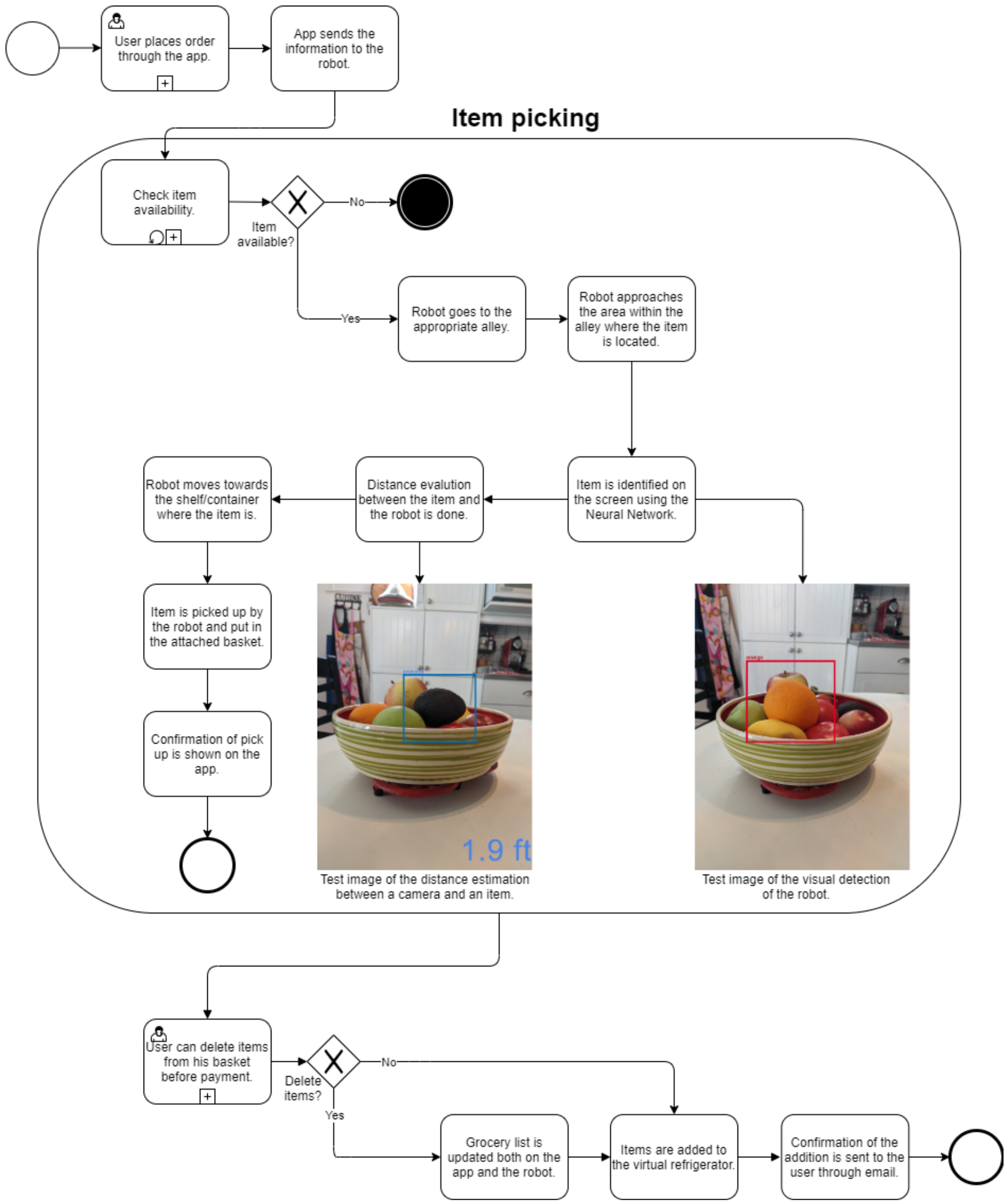


Fig. 11: Decision algorithm of the robot once it receives an order

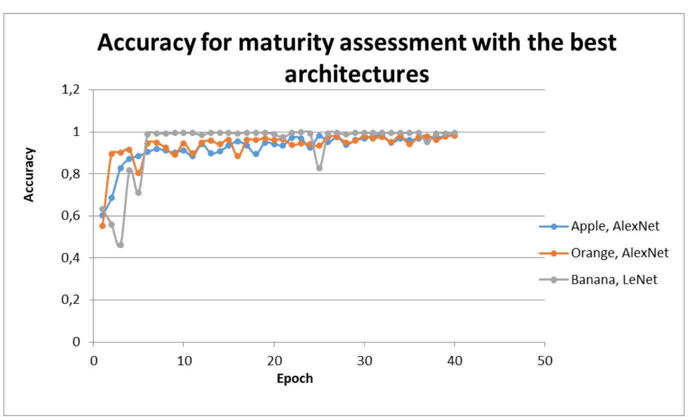


Fig. 12: Evolution of CNN's accuracy for maturity assessment of apple, banana and oranges

For the case of avocados, our dataset of dated images of avocados covering a 12 days period allows us to evaluate the maturity more precisely. Again, we trained both AlexNet and LeNet for this task. Figure 13 shows the resulting accuracy's.

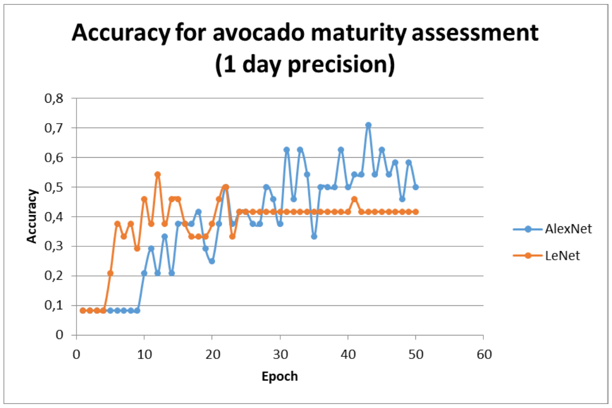


Fig. 13: Evolution of CNN's accuracy for maturity assessment of avocados to the nearest day

We can observe that, in this case, AlexNet obtain far better results than LeNet. Note that a plausible explanation for the flatness of LeNet's curve over the last 25 epochs is that the maximal capacity of the architecture is reached and the neural network can't learn further.

The maximal accuracy reached by AlexNet is $70,83\%$. This is relatively good given that the task of evaluating the maturity to the nearest day is much more difficult than a simple classification between "Fresh" and "Rotten", but it would still be desirable to have more accurate predictions. A possible solution is to renounce to a very high precision and to predict longer time ranges than a single day. Thus, we tried to gathered our data into ranges of three consecutive days and to train CNNs for this slightly easier task. The results are shown in Figure 14.

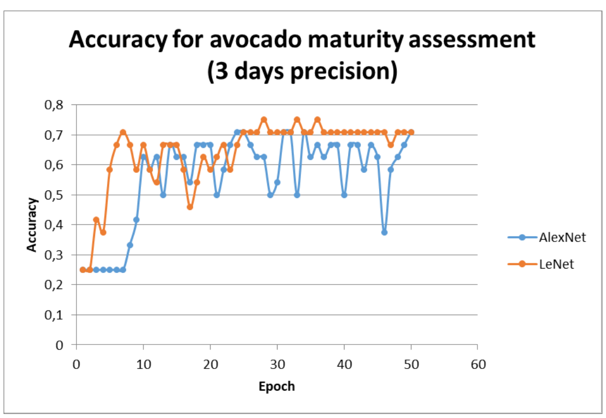


Fig. 14: Evolution of CNN's accuracy for maturity assessment of avocados to the nearest 3 days

For the 3 days precision, LeNet surpasses AlexNet and reach an accuracy of $75,00\%$. As before, this can be explained by the higher complexity of AlexNet's architecture making it more prone to over-fitting. The maximal accuracy is a little higher than for maturity assessment to the nearest day, but the difference is not major (an improvement of 4.17% only). In light of these results, hence, we conclude that it is still preferable to keep a precision of one day for IMFRMAP platform at the cost of a slightly lower accuracy.

Finally, we compared the training time of AlexNet between a CPU (**Intel Core i7-7700**) and a GPU (**Nvidia GeForce GTX 1080 8GB**). We expected the GPU to perform significantly faster than the CPU, since GPUs can perform parallel computation, a task that greatly improves the speed of backpropagation through the CNNs. Figure 15 demonstrates that this was the case, however, it also shows that the CPU training time for an epoch was not constant, especially at the end, while it is roughly constant for the GPU.

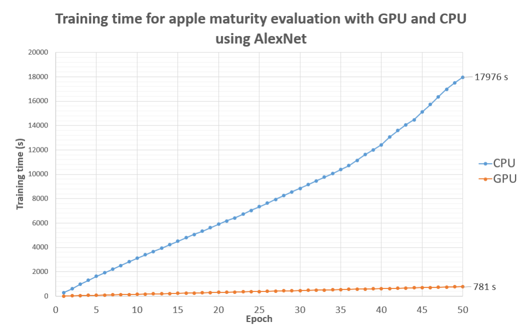


Fig. 15: Comparison of a CPU and a GPU training time in relation to the epoch

B. Expanding the Maturity Assessment capabilities

As of now, IMFRMAP only supports three fruits and one vegetable. However, we would like to support a greater range of food to better help the HSR and consumers. As explained in Section III-A3, if our multi-fruit detection tool detects fruits or vegetables that are not supported by our maturity assessment NN, then we will use the average life expectancy of the product to tell the HSR or user how long it will last. Table II shows data we collected on StillTasty.com [1] that details the average lifespan of our current food items to give an example.

TABLE II: The average fruit lifespan of IMFRMAP's supported fruits

Fruit/ Storage condition	Room temperature	Fridge	Freezer
Whole Apple	5-7 days	1-2 months	10-12 months
Cut up Apple	1-2 days	3-4 days	10-12 months
Whole Banana	2-5 days	5-7 days (ripe)	2-3 months
Cut up Banana	N/A	3-4 days	2-3 months
Whole Orange	5-7 days	3-4 weeks	10-12 months
Cut up Orange	N/A	3-4 days	10-12 months
Whole Avocado	4-7 days	3-5 days (ripe)	3-6 months
Cut up Avocado	N/A	3-4 days	3-6 months

C. Distance Computing Equations

As mentioned in Section III-D3, we compared two equations used to compute the distance between a camera and the object in front of it. To test these equations, we took pictures in real-world scenarios and compared their output (theoretical distance) to the real (experimental) distance of the object pictured. Through this comparison, we were able to determine which equation was the most accurate on average. Here are some of the pictures we used for the experiment. We used a

Fig. 16: Examples of pictures used. From left to right and top to bottom: Kiwi Above (KA), Apple Front (AF), and Mango Above (MA)

total of six scenarios, two for each fruit taken. Each fruit was taken from the front (suffix F) and from above (suffix A). The six scenarios are as follows: a kiwi from the front (KF), a kiwi from above (KA), an apple from the front (AF), an apple from above (AA), a mango from the front (MF), and a mango from above (MA). The settings for our experiment are shown in Table III.

TABLE III: Camera's parameters and fruit's dimensions used in the experiment

	KF	**KA**	**AF**	**AA**	**MF**	**MA**
f (mm)	22	22	22	22	22	22
r_h (mm)	50	50	65	65	88	88
i_h (px)	2848	2848	2848	2848	2848	2848
o_h (px)	193	224	241	310	358	402
s_h (mm)	15,8	15,8	15,8	15,8	15,8	15,8
s_w (mm)	23,6	23,6	23,6	23,6	23,6	23,6
i_w (px)	4288	4288	4288	4288	4288	4288
r_h^1 (m)	0,05	0,05	0,065	0,065	0,088	0,088
o_h (mm)	1,062	1,233	1,326	1,706	1,970	2,213

To find the distance with the second equation, we needed the pixel size ratio for the camera we used. We started by finding the image sensor dimensions which were 23.6 x 15.8 mm. The get the ratio, we divided the width of the sensor by the width of the picture taken (or height) and it gave us 23.6/4288 so approximately 5,5 x 10-3 mm/px. Now that we had the size of a pixel on our sensor, we simply needed to multiply it by the height of the object on the image taken in pixels. Finally, we obtained the real size of the object on the sensor when the picture is taken.

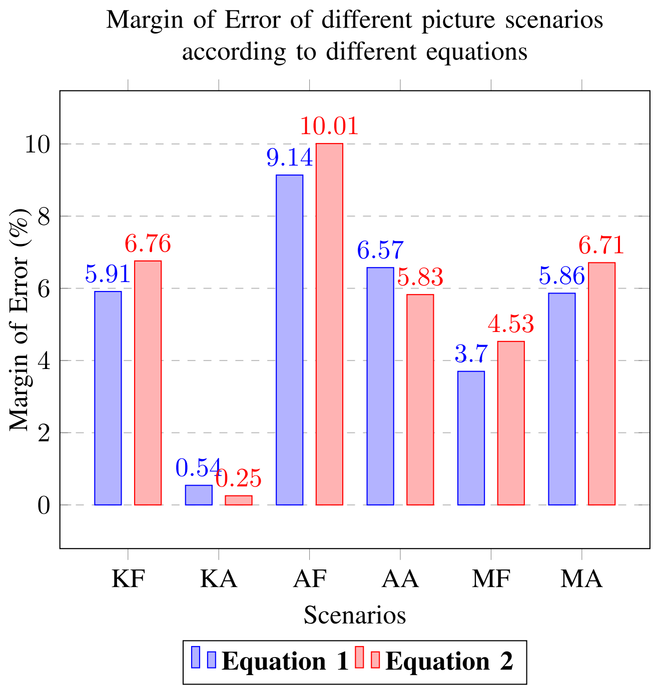
The camera we used had a focal length range from 18 to 105 mm. Even though you can look up an image's details, we decided not to use a zoom on the first calculations for a focal length of 18mm and then a small zoom for 22 mm.

It is worth mentioning that the physical measurements were not exact and are approximations of what we could measure with a measuring band. With that in mind, the formulas could happen to be a tiny bit more precise than shown here. The results of our computations are shown in Table IV.

TABLE IV: Comparison of the theoretical distance (T_d) and the experimental distance (E_d) between the camera and the fruit

	KF	**KA**	**AF**	**AA**	**MF**	**MA**
Equation 1						
T_d (m)	1,027	0,885	1,070	0,831	0,975	0,868
E_d (m)	0,970	0,890	0,980	0,890	0,940	0,820
Equation 2						
T_d (m)	1,036	0,892	1,078	0,838	0,983	0,875
E_d (m)	0,970	0,890	0,980	0,890	0,940	0,820

To better illustrate the disparities between the two equations, we plotted their margin of error on the following graph:



As we can see, the first equation has a lower margin of error four out of six times which would indicate that it is more accurate on average. However, we can also see that the difference between the two equations is very negligible, that is below 1%. This would account for a difference of 1 cm in real-world scenarios, which will often not cause any issue. Therefore, both equations could be used equally, but equation 1 could be prioritised for a better accuracy on average.

V. IMFRMAP Exploitation

IMFRMAP's design aims at detecting and computing the freshest or ripest fruit and vegetables. This competency can be used in many fashions. We have devised two principal uses of the design that are the most helpful and innovative. However, this design and the idea are purposefully general so that they could be applied in many fields and on many platforms depending on one's challenges and needs.

A. Application 1: Human Support Robot

As the aging population becomes a greater part of society each day, the need for help for these people grows more than ever before. This task is currently assigned to other human care workers, but their numbers is often not enough to provide help to the growing amount of people in need. Our Human Support Robot aims at relieving these care workers the recurrent task of bringing fresh food to these people.
The Human Support Robot or HSR is an autonomous system which could help order, pick-up and deliver fruits and vegetables according to one's needs by using IMFRMAP's Fruit Scanner technology to detect items such as fruit or vegetable and assess their maturity. This technology could be used in mobility-restricted people's household to bring them fresh food items through the IMFRMAP's app ordering system. We seek to provide relief to the human assistants in these households by creating an safe alternative for this task. Figure 17 illustrates a typical use cycle of the HSR. However, the design can be iterated upon to adapt to one's needs.

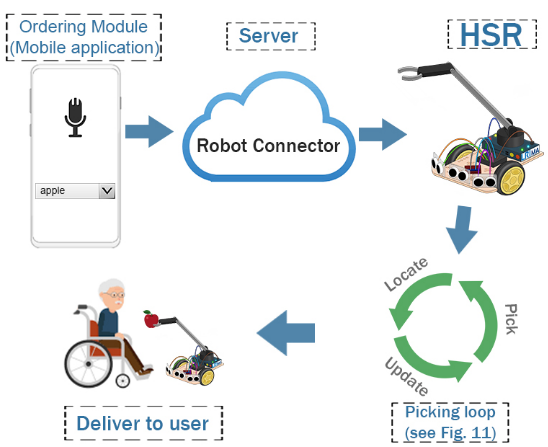


Fig. 17: A typical use cycle of the HSR. Robot drawing from ALIVE [7]

B. Application 2: Virtual Fridge

While IMFRMAP's fruit scanner is integrated in the HSR, this second application focuses on bringing it to the public to help consumers reduce their food waste. In fact, the IMFRMAP's Virtual Fridge is a component of our mobile application aimed at using our Fruit Scanner capabilities to help consumers better use and eat the food they buy. This is achieved through the numerous components that were explained in III-C, but it is all based around the use of a virtual fridge to store and represent the food we buy. This following typical use case presents the main features and explains the significance of their role.
First off, the user will take pictures of their food and analyse them using the Fruit Scanner as shown in Fig. 18 and 19. Then, they can save the fruit in the fridge by pressing the "Save" button.

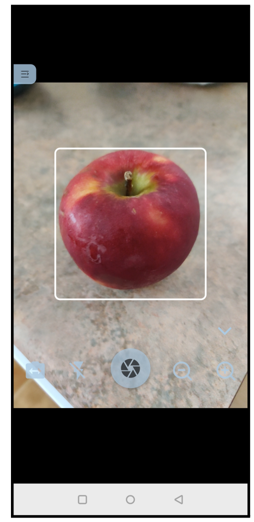


Fig. 18: A. Picture the apple

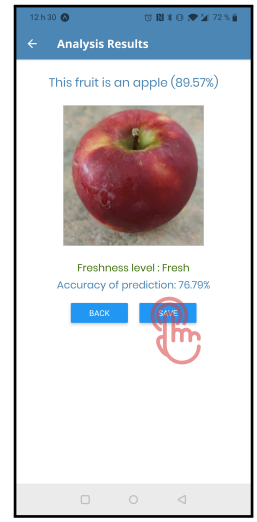


Fig. 19: B. Result of the analysis

Now, the Virtual Fridge only has one category ("Uncategorized") at the beginning and every picture will be sent in it by default. However, the user can add as many of them as they want by pressing the "Plus" icon in the top-right corner as demonstrated in 20. Figure 21 shows an example of different useful categories where the apple pictured as been moved to the relevant "Fruits" category. These categories help the user sort through and find their food items quickly and easily instead of searching through their fridge.

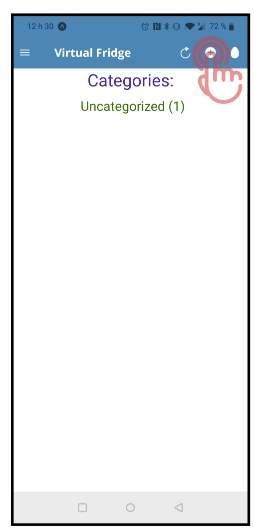


Fig. 20: C. Initial Virtual Fridge Layout

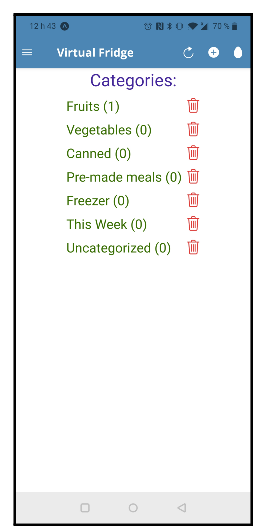


Fig. 21: D. Example of a Virtual Fridge Layout

Next, the user can now input useful information about the apple such as a note (Fig. 22 for its use and a reminder as to when they should eat it (Fig. 23 and 24. This reminder module is an essential part of our strategy to help users consume their food before their expiration date. As of now, we are able to set an automatic reminder for avocados with the information our neural network gives us, and in the future we will expand the number of automatic reminders we can set up through the use of the average maturity data we collected as mentioned in Section III-A3 and showed in Table II.

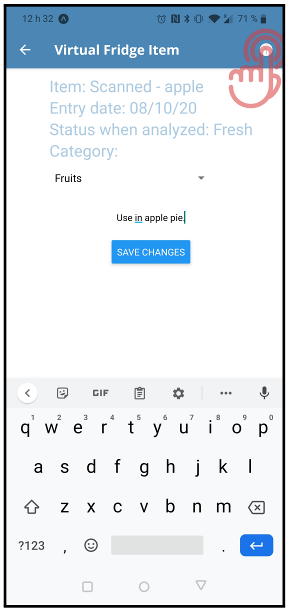


Fig. 22: E. Adding a note to the fruit

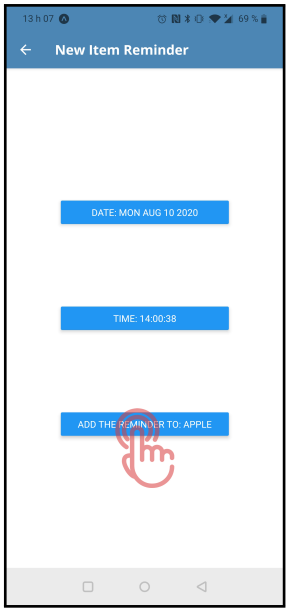


Fig. 23: F. Setting up the reminder

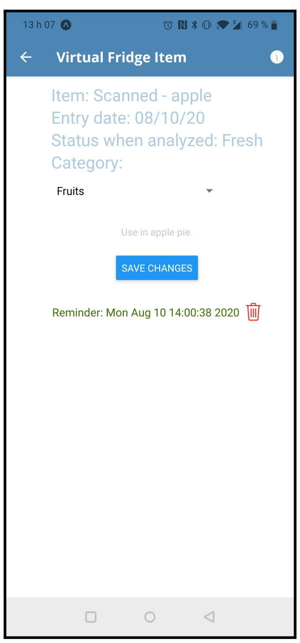


Fig. 24: G. Confirmation of the reminder

Finally, the apple is stored in the virtual fridge and ready to be used. We want to make the addition of food items to the Virtual Fridge as fast and efficient as possible to help users stay motivated to use this platform. Other features to keep the motivation of our users is the Smart Recipe Module which will be implemented in the future. This module, as explained, will allow users to search for recipes using the food they have in their Virtual Fridge. This module could have a neural network attached to it to propose recipes instantly that would fit the user's taste. Imagine simply asking your phone what to eat tonight and it giving you a recipe that you will love using ingredients you have at home.

Another big component to the retention of users is the badge system, which is a system that attributes badges to users for their involvement in our platform. These badges could be publicly shown and used to reward users for using our mobile application.

VI. Conclusion and Future Works

Acknowledgements

We would like to thank Amasten Ameziani and Hervé Claden for their valuables comments and participation in this project.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] ”Still Tasty: Your Ultimate Shelf Life Guide”. Stilltasty.com, 2009. https://www.stilltasty.com/ [last visited on 19-08-20]
[2] “Food Waste in Canada”. Lovefoodhatewaste.com, 2018. https://lovefoodhatewaste.ca/about/food-waste/ [last visited on 19-08-20]
[3] YoloV4: A. Bochkovskiy, C.-Y. Wang & H.-Y. Mark Liao ”YOLOv4: Optimal Speed and Accuracy of Object Detection”, 2020.
[4] J. Redmon ”Darknet: Open Source Neural Networks in C”, http://pjreddie.com/darknet/, 2013-2016.
[5] J. Redmon & A. Farhadi ”YOLOv3: An Incremental Improvement”, 2018.
[6] Fresh Rotten Dataset: https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification
[7] J. Rezgui, É. Gagné & G. Blain “Autonomous Learning Intelligent Vehicles Engineering (ALIVE 1.0) ”, accepted in IEEE ISNCC 2020.
[8] Liu, J., Pi, J., & Xia, L. (2020). A novel and high precision tomato maturity recognition algorithm based on multi-level deep residual network. Multimedia Tools and Applications, 79(13), 9403-9417.
[9] de Luna, R. G., Dadios, E. P., Bandala, A. A., & Vicerra, R. R. P. (2020). Tomato Growth Stage Monitoring for Smart Farm Using Deep Transfer Learning with Machine Learning-based Maturity Grading. Agrivita, 42(1), 24.
[10] de Luna, R. G., Dadios, E. P., Bandala, A. A., & Vicerra, R. R. P. (2019, November). Tomato Fruit Image Dataset for Deep Transfer Learning-based Defect Detection. In 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM) (pp. 356-361). IEEE.
[11] Behera, S. K., Rath, A. K., Mahapatra, A., & Sethy, P. K. (2020). Identification, classification & grading of fruits using machine learning & computer intelligence: a review. Journal of Ambient Intelligence and Humanized Computing, 1-11.
[12] Wan, S., & Goudos, S. (2020). Faster R-CNN for multi-class fruit detection using a robotic vision system. Computer Networks, 168, 107036.
[13] Tang, Y. C., Wang, C., Luo, L., & Zou, X. (2020). Recognition and localization methods for vision-based fruit picking robots: a review. Frontiers in Plant Science, 11, 510.
[14] Altaheri, H., Alsulaiman, M., & Muhammad, G. (2019). Date fruit classification for robotic harvesting in a natural environment using deep learning, IEEE Access 7 (2019) 117115e117133.

[15] Zhang, L., Gui, G., Khattak, A. M., Wang, M., Gao, W., & Jia, J. (2019). Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot. IEEE Access, 7, 56028-56038.
[16] Horng, G. J., Liu, M. X., & Chen, C. C. (2019). The Smart Image Recognition Mechanism for Crop Harvesting System in Intelligent Agriculture. IEEE Sensors Journal, 20(5), 2766-2781.
[17] Ringdahl, O., Kurtser, P., & Edan, Y. (2019). Evaluation of approach strategies for harvesting robots: Case study of sweet pepper harvesting. Journal of Intelligent & Robotic Systems, 95(1), 149-164.
[18] Dewi, T., Risma, P., & Oktarina, Y. (2020). Fruit sorting robot based on color and size for an agricultural product packaging system. Bulletin of Electrical Engineering and Informatics, 9(4), 1438-1445.
[19] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE.
[20] Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks, Advances in neural information processing systems.