Software Impacts 18 (2023) 100585

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

FridgeSnap: A software for recipe suggestion based on food image classification

Liam Boyd, Nonso Nnamoko *

Department of Computer Science, Edge Hill University, St Helens Road, Ormskirk. L39 4QP, United Kingdom

ARTICLE INFO

Keywords:
Image classification
Food waste
Android application
Deep learning
Recipe recommender

ABSTRACT

Food waste is a global issue, affecting both consumers and producers, with heavy economical and environmental impact. This paper presents FridgeSnap, a tool based on image classification that has the potential to help reduce the global food waste. The tool receives as input, images of singular food items taken from users' electronic device such as mobile phone or tablet, then identifies the constituent food item through an underlying deep learning model before suggesting possible recipes that can be made with the food item. The application was built within the Android Studio IDE using Java programming languages and XML, thus works on Android devices. A potentially shippable version of the FridgeSnap including source code and the android application package (APK) file, is freely available on Github.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2023-411
Permanent link to Reproducible Capsule	
Legal Code License	MIT License
Code versioning system used	GIT
Software code languages, tools, and services used	Python, Java and XML
Compilation requirements, operating environments & dependencies	Python ≥ v3.8, numpy ≥ v1.21, pandas ≥ v1.4, scikit-learn ≥ v1.1, tensorflow-gpu ≥ v2.1.0, cudatoolkit ≥ v10.1.243, CUDNN ≥ v7.6.5, tensorflow ≥ v2.1.0
If available Link to developer documentation/manual	README available at: www.github.com/liiiiamb/FridgeSnap#readme
Support email for questions	liam_boyd2@outlook.com

1. Introduction

Food waste is a pressing concern that requires innovative solutions. Despite efforts to address the issue at both consumer and producer levels, existing approaches have proven ineffective. Therefore, a fresh and user-friendly approach is needed to engage consumers. Current strategies primarily focus on incentivising recycling of food waste, like the 'Nanjing Green Account scheme', where consumers receive electronic points for correctly recycling their food waste [1]. However, these incentives primarily encourage recycling rather than utilising the food items themselves. In this paper, we present FridgeSnap, a mobile application designed to leverage leftover or mispurchased food items that consumers may have, aiming to provide a practical solution to reduce food waste. The tool uses advanced image recognition based on multi-class deep learning classification model to provide tailored suggestions on what users can cook with their excess food items.

FridgeSnap caters primarily to consumers who often purchase 'multipacks' or large quantities of a specific food item, such as potatoes or carrots, but ends up using only a portion of what they have purchased. The functionality of FridgeSnap revolves around employing image classification techniques on food items captured with the camera on users' mobile device. Specifically, the captured image goes through a classification phase where an underlying pre-trained deep learning model is used to identify the food item and subsequently added to a list that may contain single or multiple food items. Then, FridgeSnap generates recipes that utilise the food items in the list, upon user request. This means that consumers no longer need to discard items they have either mistakenly purchased or do not intend to use entirely. Instead, they can rely on FridgeSnap to provide them with recipe recommendations that make efficient use of those items. This innovative solution empowers

* Corresponding author.
E-mail addresses: 24514471@edgehill.ac.uk (L. Boyd), nnamokon@edgehill.ac.uk (N. Nnamoko).

https://doi.org/10.1016/j.simpa.2023.100585
Received 14 September 2023; Received in revised form 20 September 2023; Accepted 20 September 2023

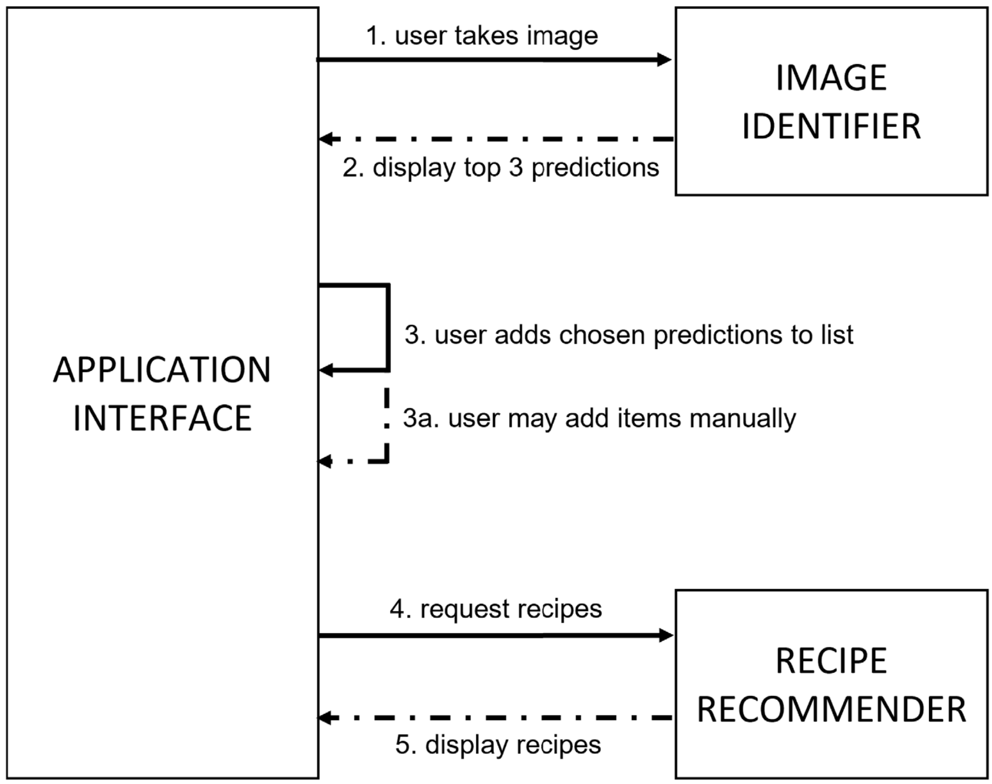


Fig. 1. FridgeSnap overview of components.

consumers to avoid unnecessary waste and make the most of their food purchases, enhancing sustainability and reducing overall food waste.

The current version of FridgeSnap uses a multi-class classification model trained to recognise 20 food items including *bacon*, *bananas*, *bread*, *broccoli*, *butter*, *carrots*, *cheese*, *chicken*, *cucumber*, *eggs*, *fish*, *lettuce*, *milk*, *onions*, *peppers*, *potatoes*, *sausages*, *spinach*, *tomatoes*, and *yogurts*. In contrast, most of the existing approaches in this field rely on manual entry of food items in textual form which can be time consuming, e.g., SuperCook,[1] BigOven,[2] and Epicurious.[3] This makes FridgeSnap both efficient and innovative for minimising user effort and food waste.

2. Application architecture

Fig. 1 illustrates a general overview of the architecture and components of the FridgeSnap application which consists of three main aspects:

(a) the **Application Interface**, for allowing the user to interact with the software. This includes capturing images with the user's device camera, identifying the food item in the image, and making recipe suggestions.
(b) the **Image Identifier**, which uses a pre-trained deep learning model to identify the food items captured on users' device.
(c) the **Recipe Recommender**, used for generating tailored recipes based on the identified food items.

3. Application development & methodology

FridgeSnap was developed within the AndroidStudio IDE, using `Java` programming language for the back-end, and `XML` for the front-end i.e., **Application Interface**. A pre-trained deep learning model (i.e., **Image Identifier**) that was developed using Python programming language forms the backbone of FridgeSnap for identifying food images. The **Recipe Recommender** retrieves recipes for the identified images. A detailed development and methods are presented in Section 3.1 for the Application Interface, Section 3.2 for the Image Identifier, and Section 3.3 for the Recipe Recommender.

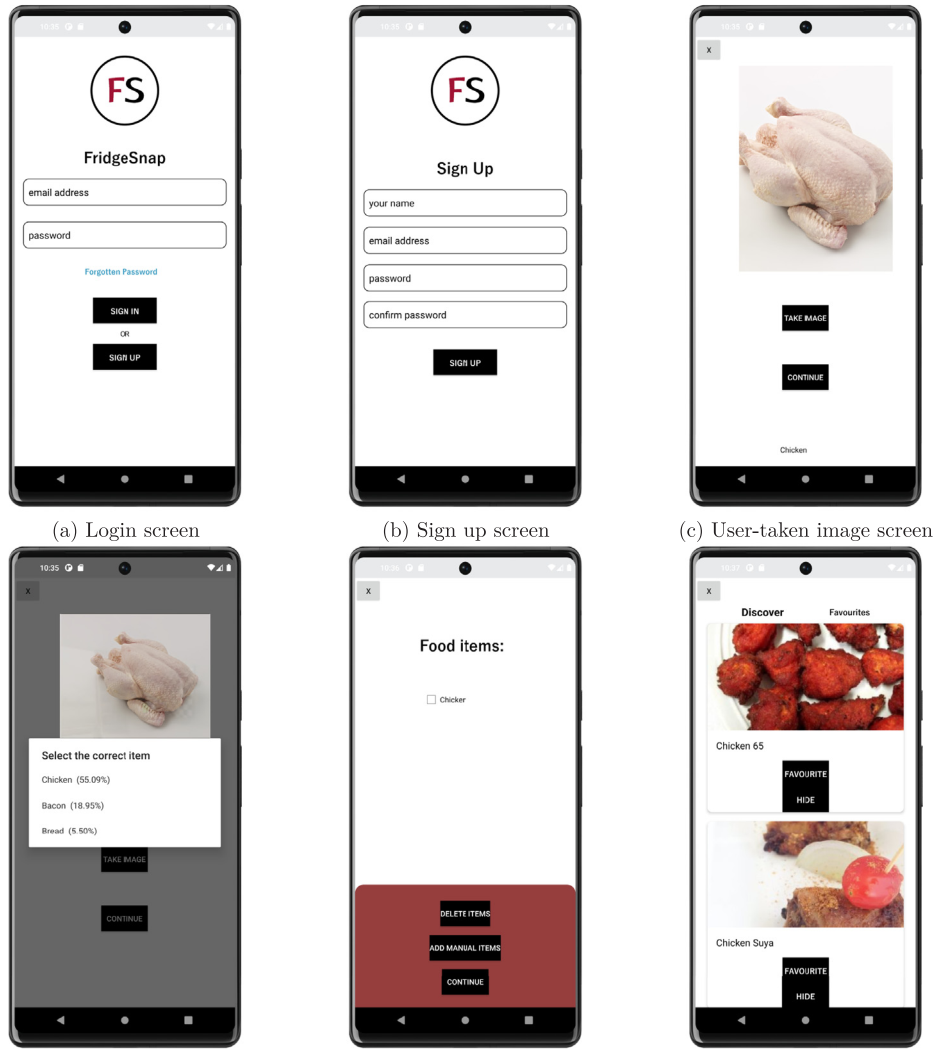
3.1. Application interface

The application interface was developed following research into similar applications and an understanding of relevant functionalities. Low-fidelity and high-fidelity wireframes were initially developed to outline blueprint for the application screens including object placement and interaction within the application. The low-fidelity wireframe was developed with `Miro`.[4] while the high-fidelity wireframe was developed with `ProtoPie`[5]

The application interface was developed in `AndroidStudio` which provides a pre-set emulator to view and test various functionalities of the application. The front-end view of the application was built using `XML` and the back-end was developed using `Java`. All buttons within the application were developed using the `AndroidX AppCompatButton` library, which allowed for all buttons within the application to keep a consistent and uniform styling. Each screen was developed using either a `LinearLayout`, a `ConstraintLayout`, or a `RelativeLayout` depending on the objects included within a screen and the order in which objects must be displayed.

The back-end of the application included the usage of Android camera intent, which allowed for the devices' camera to be used during the section of the application where the user takes an image of an item. This aspect of the application rely on a pre-developed deep learning model discussed in Section 3.2, to process images taken by the user. An example of the image screens can be seen in Fig. 2.

[1] www.supercook.com/#/desktop
[2] www.bigoven.com
[3] www.epicurious.com
[4] www.miro.com
[5] www.protopie.io



(a) Login screen (b) Sign up screen (c) User-taken image screen

(d) Image prediction screen (e) Food item list (f) Recipe recommendation screen

Fig. 2. Example FridgeSnap application screens.

User-taken images are taken and processed through the inclusion of a button, which activates the device camera (when clicked) to allow the user to take an image. Once an image is taken, it is displayed on the screen within an `ImageView` area as shown in Fig. 2(c). Further implementations within the application, such as the list of food items, within the general food item list, are displayed as checkbox items (see Fig. 2(e)). This allows for items to be removed, if desired. Additionally, recipes displayed within the recipe discover page of the application are displayed within a `CardView`, with information regarding each recipe stored within separate cards in the layout as shown in Fig. 2(f).

3.2. Image identifier

The image identifier is based on `DenseNet` [2], a pre-trained convolutional neural network architecture preferred to other architectures (i.e., `VGG16`, `MobileNet`, and `InceptionNet`) following an extensive testing phase that showed its superiority on the training data in terms of accuracy and loss function. The current version of the image identifier model was trained to identify 20 food item groups. The training utilised a dataset consisting of 40,000 images, with $\sim 2,000$ images belonging to each food item group. The dataset is publicly available on Mendeley Data repository entitled 'Multi-class food image dataset' [3]

The model was training over 50 epochs using 224×224 image resolution to retain a high quality. `Adagrad` optimiser was preferred over `SGD`, `Adam`, and `RMSProp`, during training to determine the most accurate parameter value for the training data used. The final, most accurate version of the image identifier model, resulted in a test accuracy of 78%, and a loss value of 0.0135, with an overall runtime of 6,689 s.

3.3. Recipe recommender

FridgeSnap uses `Spoonacular`[6] Application Programming Interface (API) to retrieve a list of approximately 10 recipes per food item entered. Each recipe card contains the title of recipes, and an image of the recipe. The recipe titles are further utilised to carry out a google search of each recipe.

In order for `Spoonacular` to be adopted into the application, the API key must first be retrieved. This API key is inserted into the application within the `DiscoverFragment` page, which is where the recipe searching takes place within the application. Details of how the recipes are called within the application is shown in Fig. 3. Specifically, recipes related to a given food item list are retrieved and added to a defined card layout for further processing.

[6] www.spoonacular.com

```
private void loadRecipes() {
   for (int i = 0; i < predictionList.size(); i++) { //initializes for loop - set to run for the
       amount of items within the predictionList
      recipeService.getRecipes(API_KEY, predictionList.get(i)).enqueue(new
          Callback<RecipeResponse>() {
         @Override
         public void onResponse(Call<RecipeResponse> call, Response<RecipeResponse> response) {
            RecipeResponse recipeResponse = response.body();
            if (recipeResponse != null) {
               List<Recipe> recipes = recipeResponse.getResults();
               for (Recipe recipe : recipes) {
                  addRecipeCard(recipe);
               }
            }
         }

         @Override
         public void onFailure(Call<RecipeResponse> call, Throwable t) {

         }
      });
   }
}
```

Fig. 3. Load Recipes code snippet.

4. Application demonstration

Upon launching the application, the user is presented with a sign-in page (see Fig. 2(a)). On this page, the user has the ability to enter log-in details and then sign into the application by clicking Sign In. On this page, the user further has the options to reset their password by clicking Forgotten Password or to create an account by clicking Sign Up (see Fig. 2(b)).

After the user has successfully signed-in or signed-up to the application, they are taken to a screen where they can take an image of their desired items (see Fig. 2(c)). To take an image, the user must click Take Image, and then use the activated camera to capture an image of their desired item. After taking an image, a prediction box, which can be seen as Fig. 2(d), will show with the top three predictions of what the deep learning model thinks the item is. After the user chooses the correct item from the list, they can either continue to add items by clicking Take Image again or they can continue by clicking Continue. Once the user continues from the image screen, they are shown a list of all items they have taken an image of as shown in Fig. 2(e). Each listed item has a checkbox next to it which allows the user to remove the item, by selecting the item in the list and clicking Delete Items if desired. The user has the ability to also add items manually on this page by clicking on Add Manual Items and then entering the name of the item and clicking Submit followed by Continue. Once all items have been added, this will populate the general food item list with all items and the user is able to proceed by clicking Continue, which carries out a search of recipes and displays the obtained information in a CardView shown in Fig. 2(f). On this screen, the user has the ability to:

(a) select favourite recipes by clicking Favourite on any recipe card
(b) hide recipes by clicking the Hide button on a recipe card
(c) view recipes by clicking anywhere on a recipe card.

At any time during the run-through of the application, the user is able to exit the application and log-out by clicking on the X button at the top left of many screens within the application. A video demonstrating the usage and functionality of the application is available on Youtube entitled FridgeSnap Demonstration.[7] The software source code including the APK file and documentation is also available on Github.[8]

[7] www.youtu.be/YhYpCkvr_So
[8] www.github.com/liiiiamb/FridgeSnap

4.1. User experience evaluation

The user experience (UX) on the application was tested without real users and this was carried out in two methods:

(a) **Feature testing**, which includes testing each individual aspect of the application, such as text validation, and the interaction between objects on each screen.
(b) **Action runtime testing**, responsible for determining the time it takes to carry out a certain action within the application.

With regards to the feature testing of the application, 51 tests were carried out overall on each individual aspect of the application, such as text validation testing on the landing page, and if functionalities such as the ability to delete items from the overall food item list work as expected. Identified feature drawbacks were fixed in the current version of FridgeSnap, including deviations from the original design to ensure optimum UX.

Action runtime testing was carried out using CogTool[9] which allows for automatic user interface design evaluation with a predictive human performance model. The purpose of this testing phase is to understand any negative UX impact that the application may have on the user and CogTool has been shown in previous research [4] to produce a valid cognitive representation of a 'real' skilled user. Table 1 highlights the time it takes to carry out certain tasks within the application. For example, the results indicates that manual entry of food item(s) onto the application takes 8.4 s more than simply taking an image of the item(s).

5. Potential impact of application

FridgeSnap has the potential to contribute significantly to research, particularly in image recognition and food-related fields, while also addressing a pressing societal issue by reducing food waste, promoting sustainable practices, and improving the economic well-being of individuals and households. Details of potential impact on research, economy and environment are as follows:

[9] www.cogtool.org

Table 1
CogTool results.

Task	Average time taken to carry out task
Sign up	4.7 s
Reset password	7.1 s
Sign in	1.7 s
Take image of item	5.4 s
Add item manually	13.8 s
Remove item from list	13.6 s
Search for recipes	13.6 s
Open favourite recipes	15.4 s
Open discovered recipes from favourites	16.9 s

Research: To the best of our knowledge, the dataset developed for this research is the first to include a collection of individual food items that can be used for multi-class food image classification tasks. Unlike existing datasets such as Food-101 [5] that primarily feature images of full meals and Food-5K [6] that categorise food items into broad groups, this dataset focuses on singular food items, such as `potatoes` and `carrots`. This uniqueness and specificity makes the dataset a new benchmark for evaluating and advancing food recognition algorithms, potentially leading to significant improvements in multi-class image classification techniques, with applications extending beyond FridgeSnap's mission to combat food waste. The availability of this dataset [3], coupled with the open-source code of FridgeSnap, promotes accessibility and fosters collaboration/innovation within the research community.

Environment and sustainability: Food waste is a significant contributor to environmental problems, including soil contamination, water pollution, greenhouse gas emissions, and reduced crop yields, and annual food waste is estimated to be around 7.2 million tonnes [7]. Preventing food from going to waste is one of the easiest and most powerful recommendations to save money and lower climate change footprint [8,9]. By helping people to identify and utilise food items efficiently, FridgeSnap indirectly contributes to a reduction in the amount of food discarded by consumers, thus reducing the environmental footprint associated with food waste. This aligns with sustainable development goals related to food security and environmental conservation.

Economy and education: The economic impact of food waste per family is estimated to be around £680 a year [10]. FridgeSnap helps people make better use of the food they already have, which can lead to decreased grocery bills. This translates into economic savings and improved financial well-being for individuals and households. Additionally, FridgeSnap's feature of suggesting recipes based on available food items can educate users about cooking and meal planning, thus encouraging culinary creativity, and potentially leading to healthier eating habits.

6. Limitations and future work

While FridgeSnap offers valuable assistance in identifying food items from images, a notable limitation of the current version is its inability to determine the quantity of the identified food items. Enhancing the tool to estimate food quantities could greatly improve its utility especially to filter recipe suggestions based on the available food items, thus offering a more comprehensive and tailored cooking experience. This potential enhancement would not only reduce food waste further but also facilitate more precise and efficient meal planning.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was conducted as part of a final year undergraduate project at Edge Hill University. Special thanks to the Computer Science Department for providing resources to carry out the research.

References

[1] C. Li, Y. Huang, M. Harder, Incentives for food waste diversion: Exploration of a long term successful Chinese city residential scheme, J. Clean. Prod. (2017) 491–499.

[2] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, 2016, arXiv:1608.06993.

[3] L. Boyd, N. Nnamoko, Multi-class food image dataset, 2023, http://dx.doi.org/10.17632/y2k2crmzx8.1, https://data.mendeley.com/v1/datasets/y2k2crmzx8/draft.

[4] F. Arshad, N. Nnamoko, J. Wilson, B. Roy, M. Taylor, Improving healthcare system usability without real users, Int. J. Healthcare Inform. Syst. Inform. 10 (1) (2015) 67–81, http://dx.doi.org/10.4018/IJHISI.2015010104, http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJHISI.2015010104.

[5] D. Becker, Food 101, 2017, https://www.kaggle.com/datasets/dansbecker/food-101.

[6] A. Antonov, Food-5K image dataset, 2019, https://www.kaggle.com/datasets/trolukovich/food5k-image-dataset.

[7] N. Nnamoko, J. Barrowclough, J. Procter, Solid waste image classification using deep convolutional neural network, Infrastructures 7 (4) (2022) http://dx.doi.org/10.3390/infrastructures7040047, https://www.mdpi.com/2412-3811/7/4/47.

[8] C. Castells-Somoza, Fighting Food Waste: Current and Upcoming Policies and Regulations in Spain, Springer Fachmedien Wiesbaden, Wiesbaden, 2023, pp. 305–318.

[9] Environmental Protection Agency, Preventing wasted food at home, 2023, https://www.epa.gov/recycle/preventing-wasted-food-home.

[10] E. Graham-Rowe, D.C. Jessop, P. Sparks, Identifying motivations and barriers to minimising household food waste, in: Resources, Conservation and Recycling, 2014, pp. 15–23.