Feast In: A Machine Learning Image Recognition Model of Recipe and Lifestyle Applications

Evelyn *Toh Lee* Ann[1*], *Ng Sze* Hao[1], *Goh Wei* Wei[1], *Khor Chun* Hee[2]

[1] School of Computer Science and Engineering, Taylor's University, Subang Jaya, Selangor, Malaysia.
[2] Taylor's College, Subang Jaya, Selangor, Malaysia

Abstract. With the increase of individuals having an interest in the culinary world, the demand for recipe and lifestyle applications have increased. As we adapt to the changes around us during these trying times, many have also taken an interest in home-cooking. However, it may be challenging, especially for beginners to brainstorm recipes for cooking as they may not be equipped with the proper ingredients to do so. In this paper, we propose Feast In, a platform for web and mobile devices which aims to meet a user's needs for home-cooking. The platform focuses on three unique features which make Feast In more than just the average recipe platform. Firstly, an improved search algorithm which goes beyond searching for keywords would help users narrow down recipes which they can use in their kitchen. Next, customization features which would create a personalized experience, specifically towards recipes results. This would provide individuals who may face allergies or dietary restrictions an improved experience as they would not have to browse through recipes which do not meet their needs. Lastly, the search-by-image function which utilizes image recognition and machine learning technologies. Users will be able to upload an image of food that they have come across and Feast In will return a list of results which matches the image uploaded. By conducting this research, we were able to propose a unique lifestyle and recipe application which would aid users in searching for the perfect recipe.

1 Introduction

Subang Jaya, Selangor, Malaysia with the influx of individuals choosing to cook at home during the Movement Control Order (MCO), there is a distinct demand in recipe and cooking tools. Thus, 'Feast In' is a cloud-based application that is designed to offer home-cooking solutions to those who are unsure of what to cook. There are other similar platforms which offer similar services; however, we have identified several flaws and

[*] Corresponding author: evelyntohleeann@sd.taylors.edu.my

opportunities that can be implemented to improve overall user experience. The limitations of existing solutions are as follows:

1.1 Inaccurate results

Similar platforms in the market offer search functionalities whereby users are allowed to search-by-ingredient. However, this has proven to return inaccurate results as users can input any type of word or character they wish, even if it is not a relevant term. This can be improved by implementing a search algorithm which allows users to select from a dropdown list of ingredients instead of allowing users to enter any word of their liking. This would increase the accuracy of the search function as it eliminates the potential for error during user input.

1.2 Limited user experience

A large majority of recipe platforms are designed using static concepts which only allow users to view a recipe. Most platforms lack user interactivity as they do not promote interactive functions such as customisation. Feast In offers a unique user experience because users can personalise any recipe on the platform based on their preference. This allows users who do not have a specific ingredient to replace them with items they can get readily.

1.3 Image Recognition Is Not Optimized

Existing platforms do not utilize the advantages of image recognition as they rely heavily on text input. This is an opportunity for the next generation of recipe platforms which can benefit from the use of image recognition. Feast In offers a search-by-image function which allows the user to upload an image to the image recognition model which returns a list of similar recipes using model trained by machine learning.

1.4 Negligence of dietary needs

Users who suffer from allergies or those with personal/religious beliefs and allergies may face difficulties when they are browsing through recipes online. Most recipe platforms do not pay attention to this and are unaware that there are recipes that may not cater to individual needs because of their dietary needs.

1.5 Poor user interface

Many cloud-based applications tend to crowd their landing pages with unnecessary information such as advertisements, promotions and invitation to mailing lists. This may agitate users who are in a hurry as they would have to close multiple pop-ups before arriving at the landing page. This gives a bad first impression to the users as it is not aesthetically pleasing to the eye.

1.6 Lack of data analytics tools

There is a lack of data analytics and statistics tracking available currently in the market, particularly for recipe platforms. This serves as an opportunity for this project to develop comprehensive analytical tools to analyse and study user behaviour on recipe platforms.

Hence, the identified research gap is implementing machine learning algorithms [1] to process the image recognition model for the recipe search. The objectives of the project are to develop a cloud-based application which allows users to view and share recipes of their own with the online community. Secondly, the project aims to develop an improved search algorithm which can return results with precision. Thirdly, the project focuses on developing a user interactive focused platform which optimizes customisation to allow for flexibility and personalization. Lastly, an image recognition model would be developed to identify recipes based on images using machine learning.

The rest of the article are as follows. Section 2 includes the literature review of image recognition, machine learning and search algorithms. Related works were also used for further research and algorithms were proposed for the image recognition and search models. Section 3 discusses the proposed system of Feast In. It is detailed with the module description, system architecture as well as languages used for implementation. Section 4 discusses the strengths, weaknesses, opportunities and threats of the system. The article is concluded in section 5 with research limitation and future research.

2 Literature Review

This section of the paper will review the literature on the image recognition and machine learning and related works.

2.1 Image recognition and machine learning

Image recognition is a technology which is used to identify, recognise, or classify an item based on the image itself [2]. Image recognition technology utilises machine learning whereby the image recognition model is trained repeatedly until it can produce accurate results. The Feast In project will implement image recognition technologies for image classification [3] to allow users of the platform to search by an image uploaded. The aim of image classification would be to ensure that the images uploaded will be able to be correctly placed in its respective groups or sectors.

As mobile devices have increased in popularity over the last decade, there are many applications that have been developed to recognise items from an image itself [4]. An example would be Google Photos which is able to categorise images taken by a mobile device into various categories which would help users sort their images into specific albums. Instead of sorting images into general categories, Feast In aims to categorise the images into a specific food category altogether. This means if an image uploaded to the platform is a ‘Chicken Peperoni Pizza’, it will be compared against the learned model and output the food category ‘Pizza’, displaying recipes which match the food category.

Abu [2] has recommended building an image recognition model on top of the TensorFlow framework using Python for the image recognition model. The model will take in images as training data and the results of the accuracy rates of the model will be compared and improved as development occurs. The author also recommends using deep neural network (DNN) techniques especially when developing image recognition technology. Figure 1 shows the DNN process which was used by the author to train the image classification model [2]. The dataset will undergo training with the DNN and the process will reiterate until all of the images are recognised by the system.

2.2 Related Work

2.2.1 Feature Fusion

Feature Fusion [5] is an automatic image recognition system for food categories using technologies such as BoF (bag-of-features), gradient histogram, colour histogram and Multiple Kernel Learning (MKL) [6]. The system managed to garner a 62.52% accuracy rate for the classification for the 85 different food categories [7]. Their proposed image recognition system can handle various types of food with a high accuracy rate using the MKL method. An unknown food image can be classified into one of the 85 pre-defined categories.

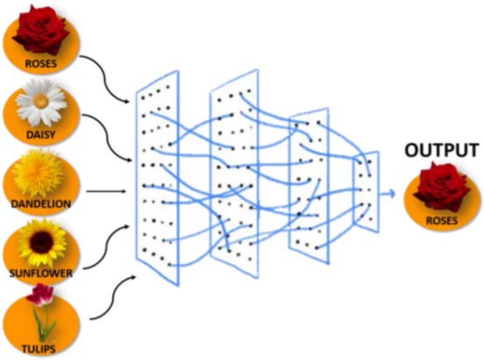

Fig. 1. DNN used for image classification

2.2.2 Nutrinet

NutriNet is an automatic food image recognition system which aims to present a new approach for food and drink image recognition and detection. They have approached the problem using technologies such as deep learning neural networks. Deep learning allows models composed of multiple layers to learn features which represents the input data with them. The models have improved results in a variety of fields such as computer vision [8-9].

The way the system works is that a user captures an image of food or drinks using a camera and will upload it to the system. The system will perform the image recognition using the deep convolutional neural networks on the uploaded image. The result is the category of a food or drink.

2.2.3 Cuisine Classification

A real-time calorie extraction and cuisine classification through food-image recognition was developed to implement dietary adjustments. TensorFlow [10] is used to build a machine-learning process to recognize the food images. The system managed to achieve an accuracy rate of 55.3% to recognize Korean, Chinese and Italian food [11].

Food taken by the user is uploaded to the trained model which will process the image and output recognized food names in an array. It also returns the nutrient data which consists of the calorie count as well as the nutrients according to the name of the food. The data is then sent back to the user.

TensorFlow [10] was utilised to build a machine learning system to train the datasets provided. Machine learning is used to train a computer to perform a task based on example data sets and past experience. It was used to train the image recognition system which managed to achieve a 55.3% accuracy.

2.2.4 Allrecipes.com

Allrecipes.com [12] is the authentic and biggest food-focused social network service. The concept of Allrecipes is “Created for cooks by cooks; where everyone plays a part in helping cooks discover and share the joy of home cooking". On the website, recipes are posted by members of the community and then copyedited by staff. In addition to sharing recipes, members may also review and rate recipes, as well as adding photos of dishes they completed using the recipe. However, Allrecipes' search feature [12] stands out from other platforms as it supports the addition and exclusion of specific ingredients. Users can type search terms to include or exclude and the system would search their database to match the criteria. However, this feature is poorly optimized as users are able to enter any words they want, and the system will still search for it. This serves as a good reference when developing Feast In as the concepts can be integrated while developing the search algorithm.

2.2.5 Upwork

Upwork is a worldwide freelancing platform that allows businesses and individuals to connect with each other and conduct business [13]. Businesses can hire freelancers by posting our job requirements, whereas freelancers can get jobs by searching for suitable jobs on the news feed and applying to the job. Although Upwork is not related to recipe platforms, Upwork's filtering option for filtering jobs is well designed as they provide various options for filtering. Clicking of any filter options changes the job news feed immediately. This feature will serve as a good reference when developing Feast In's filter and sort feature as we can refer to this while adding our own design to have our own theme.

2.3 Algorithm

2.3.1 Algorithm for image recognition model

The image recognition model would be trained using training set. This data will be trained using a pretrained model which are already trained on datasets and use the pre-learned weights to finetune the model with new data. The reason a pretrained model is used is because training a model from scratch would take a long time and would utilise high-end GPUs.

Even though the datasets are large in number, the classes would be limited to a smaller number to train the image recognition model on the specific classes. As the model can identify the images from a smaller set of data, we would be increasing the number of classes to train the model as well. The training process is depicted in Figure 2.

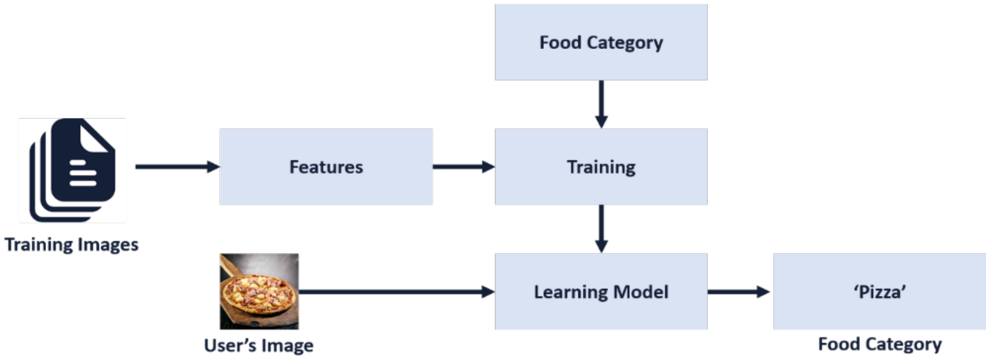

Fig. 2. Proposed design for dataset training.

Figure 2 depicts the proposed design of how training the datasets would take place. Firstly, features will be extracted from the images. Next, training of the datasets would take place with their respective food category. Once the training has been completed, it is tested with test images which will go through the learning model. The learning model will process the image and will match the image with its food category. This process is repeated until the accuracy of the model is up to par. Once the training model has finished training all the datasets, it can finally be used to take in original images which are uploaded by the user.

The image recognition model will then be implemented with the Feast In platform. The process of the image recognition model is depicted in Figure 3. First, the user will access the web or mobile application. They will then proceed to the search recipes page and click on the image recognition icon. A new window will appear to allow the user to input an image file. The image will then go through the learned model. The learned model will process the image and output a food category based on the image. The food category would be sent to the database to gather recipes which match the food category. Once the matches have been found, the list of results is shown to the user.

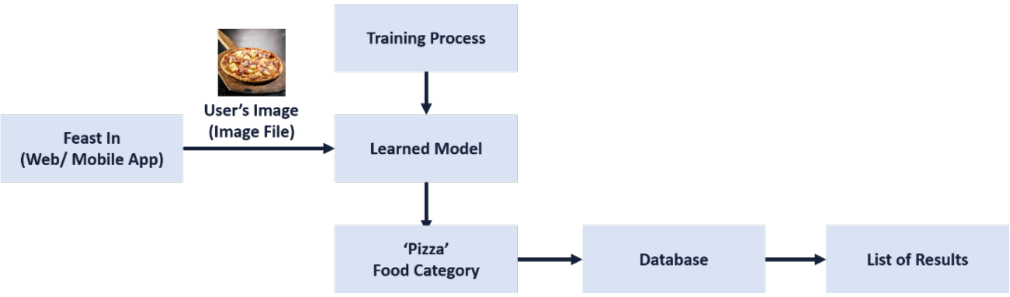

Fig. 3. Proposed image recognition model.

2.3.2 Algorithm for image search

Search bars can be found on most websites, but this does not mean they are properly implemented. Poorly coded websites which do not filter inputs produce bad search results and leaves the website vulnerable to SQL injections [14]. An SQL injection is extremely harmful as attackers may have unauthorised access to confidential data and the ability to modify or delete these data. Attackers perform SQL injection attacks by entering special characters in the search bar which can turn into a script when the search is executed [14].

Since the application will be using MongoDB [15], it may be vulnerable to NoSQL injection. To prevent a NoSQL injection attack from occurring in our system, inputs will be validated and sanitized by using application code. Moreover, MongoDB's built-in feature for secure querying will also be used. However, this still does not solve the inaccurate result issue. To prevent our system from generating inaccurate results, our system will search through the database each time a character is entered and display a dropdown list of ingredient or recipes that match that keyword. If there are no values that match the user input, a message will be shown to indicate that there are no results.

As the application will contain large amounts of recipe data, filter and sort functions are present to aid users in searching for a recipe. However, due to the large number of recipes, filtering and sorting recipes may have an impact on performance if the action is performed by the client's device. For instance, a search algorithm resulted in 1,000 recipes, and when users filter out the recipe from the client-side, their device would have to process the details of all 1,000 recipes, filter out the unwanted recipes, and then display them again.

This process is inefficient especially since the application is scalable with large number of recipes. To solve this issue, filtering and sorting recipes will be processed on the server side where the server is capable of processing huge requests in shorter amount of time due to its dedicated hardware. Whenever users filter or sort recipes, the client device makes an API call to the server to make the changes. With this method, the client device only needs to make an API call and make changes when it receives data from the server.

3 Proposed System

This section of the paper discusses the proposed system for Feast In. It includes the module description, proposed system architecture as well as the implementation languages and tools used for development of the system.

3.1 Module Description

The project contains a total of seven (7) modules. The first module is the Login and Registration Module. This module manages the users on the platform. New users will be prompted to register to the platform to use additional features. Existing users will be able to login to the platform using their credentials, a username and a corresponding password. Those who are admins can login and will be redirected to the admin dashboard.

The second module, the Administration Dashboard Module which allows administrators to access the admin dashboard. The admin dashboard allows admins to manage content, view statistics and feedback as well as admin logs. This helps the admins run the site effectively.

The third module, the Search Module is divided into three types of categories, to search-by-ingredient, to search-by-image or to search-by-recipes. Searching-by-ingredient will allow users to add ingredients to their search and generate a list of recipes which match their input. Searching-by-image will utilise image recognition technologies to output a list of recipes which matches the user's input. The user can also choose to search for recipes by entering the name of the recipe.

The fourth module is the Filter and Sort Module. This module allows user to further narrow down their search results by using the filter and sort functions. The filter function allows users to filter out unwanted criteria such as dietary needs and duration. The sort function allows users to sort by popularity and rating.

The fifth module, the Recipe Customization Module allows users who are logged in to customise an already existing recipe on the platform. Users will be able to edit and change parts of the recipe that may not be appropriate for their style of cooking and replace it to their own liking.

The sixth module is the Image Recognition module which allows users to upload an image that they do not know the recipe of. The uploaded image will then be sent to the image recognition model which is trained using pre-trained images based on machine learning. The image recognition model will then identify the food category of which it thinks the image belongs to and output a list of results that closely matches the image uploaded.

Lastly, the Rating and Review module allows users who are logged in to rate and review an uploaded recipe. The ratings of the recipe will be accounted during the sorting of recipes. This will also help ensure the quality of the recipes on the platform as lower-rated recipes would not be advertised when recipes are being sorted.

3.2 System Architecture

The system architecture for Feast In is a three-tier architecture which consists of the client tier, the business logic tier as well as the database tier (Figure 4).

The client tier of the system architecture is the front-end interface which allows user to interact with the system. ReactJS would be used for web application development and React Native would be used for the mobile application development. Both React and React Native are JS libraries which are used to build UIs.

The business logic tier of the system architecture is the middle-tier of the system architecture. APIs created in the business logic tier will be used to aid communication between the client tier and the database tier. To make requests to the database, both the web application and mobile application are required to make a call to the specific REST API. The API will then process the request and send the data back to the client.

The data layer is protected from direct access from the client as it consists of the database. MongoDB is selected as the database for our system. The business logic tier will request data from the data layer once it has processed a request from the client tier. When the business logic tier receives data from the database tier, it processes the data once again before making changes to the client tier. This layer also handles third-party APIs like Google's TensorFlow API. In the data tier, we have MongoDB as our NoSQL database server where we store data used by both web application and mobile application.

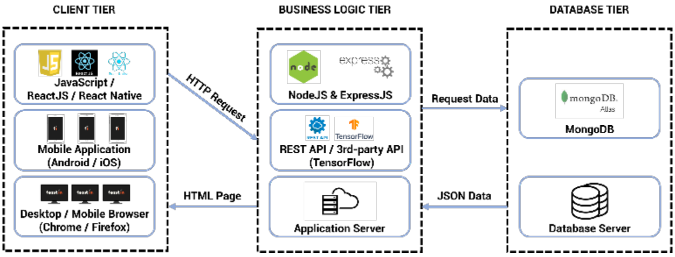
CLIENT TIER
BUSINESS LOGIC TIER
DATABASE TIER
JavaScript / ReactJS / React Native
Mobile Application (Android / iOS)
Desktop / Mobile Browser (Chrome / Firefox)
HTTP Request
HTML Page
NodeJS & ExpressJS
REST API / 3rd-party API (TensorFlow)
Application Server
Request Data
JSON Data
MongoDB
Database Server

Fig. 4. Feast in system architecture.

3.3 Implementation Language

Developing applications require a good understanding of the functionalities of a programming language and what it offers. This also include any libraries or frameworks associated with that language and the support for that language. The MERN stack was chosen for React [16] and its uniqueness of not interacting with the DOM [17] directly which boosts the performance of the application, resulting in a fast and robust application. As the application was to serve as a mobile application, a native application was proposed for its better performance and ability to run more smoothly as well as its ease of access to built-in mobile features. React Native was chosen as it allows us to develop our application for both platforms at the same time while only using a single programming language, JavaScript (JS) [18]. Moreover, React Native has similar usage as React, thus making the learning curve less challenging.

JavaScript was initially a client-side scripting language that was primarily used to add interactivity to a website [19]. However, it has evolved over the years and now lives on both the client-side and the server-side. On the client-side, JavaScript can be used to add interactivity to a website like validating the details of a form and responding to a button being clicked while also allowing developers to take advantage of browser built-in APIs (application programming interface) such as geolocation capabilities. On the server-side, JavaScript when used by NodeJS has the ability to make requests to a database, create files, and respond to HTTP requests [20-21]. JavaScript itself is one of the most popular languages. According to Stack Overflow in 2019, JavaScript was voted for the top programming, scripting, and mark-up language for seven consecutive years with over 67.8% votes. JavaScript, being a popular language, has vast amounts of libraries and frameworks that developers can used in their application, and because of that, documentation and support for JavaScript can be found easily.

3.4 Tools

3.4.1 InVision Studio

InVision Studio [22] is a powerful and new designing tool that was created to allow creative professionals to design high-quality screens with ease. It also works as a sharable prototype software that allows interactions and animations. Design layouts of Feast In's web and mobile application is done in InVision Studio. By using the same design file, prototyping of both platforms is also done in InVision Studio.

3.4.2 Visual Studio Code

To code the web and mobile applications, Visual Studio Code [23] (or VS Code) was chosen as the code editor. Visual Studio code is an open-source code editor and has gained popularity as it is simple to use but also incredibly powerful. Some of its features include syntax highlighting, auto indentation, snippets as well as support for various types of programming languages.

VS Code also has a large repository of package managers and built-in extensions which allows developers to code more efficiently. Additionally, debugging with VS Code is made simple as they have included an interactive debugger. This allows developers to execute commands, inspect variables and view call stacks all within the console. It also supports Git which grants the developer the ability to work on the code without leaving the editor [24]. Lastly, VS Code incorporates built-in support for web technologies such as React, HTML, CSS and JSON.

3.4.3 MongoDB Atlas

MongoDB Atlas [25] is a fully-managed worldwide cloud database service beneficial for modern applications developed by the same group of people who built MongoDB. Atlas, MongoDB Atlas for short, administers the complex process of deploying, healing, and managing of an individual's deployments on major cloud service providers like AWS, GCP, and Azure. Atlas guarantees scalability, availability, and compliance with the latest privacy standards and data security through proven practices and best-in-class automation [26].

3.4.3 Github

GitHub [27] offers version control using Git and hosting for software development. Git provides the source code management (SCM) and distributed version control functionality. In addition to providing features from Git, GitHub has its own features which include access control and collaboration features such as feature requests, bug tracking, wikis, and task management for every project. GitHub's basic service is free of charge and they are more commonly used for hosting of open-source projects [28].

3.4.3 Postman

Postman is a collaboration platform for the development of APIs. It simplifies every step that is required to build an API and streamlines collaboration so that developers are able to develop better APIs and at a faster pace. It is also capable of transforming an API into several different programming languages such as JSON [29] and Python [30]. The Postman platform delivers everything a developer would need for the entire lifecycle of an API from its main runtime services to client interfaces [31].

4 Proposed system

This section of the paper analyses the strengths, weaknesses, opportunities and threats of the system.

4.1 Strengths, weaknesses, opportunities and threats (SWOT) analysis

Strengths of the recipe and lifestyle platform includes an improved search function compared to other recipe applications which allows user to select from a list of ingredients instead of any word of their liking. Feast In also gives the ability for users to customize recipes to their own liking. This allows for personalization to suit the users. The platform also image recognition features which allow users to search for a recipe based on an image uploaded by the user.

Weaknesses of the platform includes that it would always require an Internet connection. As Feast In is an online web and mobile application, users are required to have an active Internet connection to use the platform. In addition, as the platform begins deployment, it might not be able to get traffic onto the site immediately, thus the platform may lack recipes. Granted, there are existing recipes uploaded by the administrators, but it would take some time before the platform is populated by recipes from users.

Many opportunities are available for the platform, including the possibility of multilingual support. Since Feast In aims to be an international platform for users all over the world, this is an opportunity for us to implement multilingual support which will translate the text into various languages. The multilingual support would also attract more traffic onto the platform. Feast In would also like to integrate Artificial Intelligence (AI) for recommendation of recipes based on a user's recent viewing history.

The main threat of the platform would be that there are many other recipe platforms which are well-established with large communities. Users may be hesitant to switch to Feast In as it is relatively new.

5 Conclusion

This study aims to develop an image recognition model which would be used in conjunction with the platform to aid users in identifying a recipe from an image uploaded by the user. The image recognition model will be trained using training images using machine learning. This aids users in identifying a certain recipe from a visual image. An advanced search algorithm will also be developed which aids a user in searching for recipes while preventing users from entering random characters which may obstruct their search results. This study also aims to create a customisation feature which allows users to customise a recipe to their own liking by changing the recipe fields.

Several limitations identified include the lack of recipes on the platform. As the platform is being deployed, it may not gather as much traffic as it is a new platform. This causes a lack of content as Feast In has yet to gain a dedicated audience. Granted, the platform would have several original recipes but as a whole it would rely on the public to upload recipes onto the platform.

Further studies will be conducted on recipe recommendation based on a user's activity using Artificial Intelligence (AI). This would allow the platform to suggest new recipes to the user based on their previous browsing activity which helps promote the personalisation of the platform.

The authors wish to express their utmost gratitude to Taylors' University Research & Enterprise Knowledge Transfer and Commercialization department, Professor Edwin Chung Chin Yau for support in this project. Besides that, the author would like to acknowledge their supervisors, Dr. Goh Wei Wei and Mr Khor Chun Hee for guiding throughout the project.

References

1. G. Bonaccorso, *Machine learning algorithms*, (Packt Publishing Ltd., Birmingham, 2017)
2. M.A. Abu, N.H. Indra, A.H.A. Rahman, N.A. Sapiee, I. Ahmad, (2019). Int. J. Eng. Res. Technol. **12**, 563 (2019).
3. Z.M. Chen, X.S. Wei, P. Wang, Y. Guo, Proceed. IEEE Conf. Comp. Vis. Patt. Rec. 5177 (2019).
4. H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, S. Cagnoni, Proceed. 2nd Int. Work. Multi. Assist. Diet. Manag. 41 (2016).
5. Z. Zhang, X. Zhang, C. Peng, X. Xue, J. Sun, (2018). Proceed. Euro. Conf. Comp. Vis. 269 (2018).
6. Y. Gu, J. Chanussot, X. Jia, J.A. Benediktsson, IEEE Trans. Geosci. Rem. Sens. **55**, 6547 (2017).
7. H. Hoashi, T. Joutou, K. Yanai, Proceed. 2010 IEEE Int. Symp. Multi. 296 (2010).
8. J. Ramírez, J.M. Górriz, A. Ortiz, F.J. Martínez-Murcia, F. Segovia, D. Salas-Gonzalez, J. Neurosci. Method. **302**, 47 (2018)
9. S. Mezgec, B.K. Seljak, Nutri. **9**, 1 (2017).
10. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Kudlur, Proceed. 12th Symp. Opera. Sys. Des. Imple. 265 (2016).
11. H.J. Suh, K.H. Lee, K. H. Int. J. Grid Distrib. Comp. **11**, 69 (2018)
12. Allrecipes. (2020). Retrieved 23 September 2020, from https://www.allrecipes.com/
13. Upwork (2020). Retrieved 23 September 2020, from https://www.upwork.com/
14. Z.S. Alwan, M.F. Younis, Int. J. Comp. Sci. Mob. Comp. 6, 5 (2017).
15. B. Hou, K. Qian, L. Li, Y. Shi, L. Tao, J. Liu, Proceed 3rd Int. Conf. Cyber Sec. Cloud Comp. 75 (2016).
16. V. Subramanian, *Pro MERN Stack,* (SpringerLink, India, 2017)
17. R.J. Walters, J.A. Laird, S.D. Knott, *U.S. Patent No. 9,460,292*. (Washington, DC: U.S. Patent and Trademark Office, 2016).
18. B. Eisenman, *Learning react native: Building native mobile apps with JavaScript*. (O'Reilly Media, Newton, 2015)
19. D. Crockford, *JavaScript: The Good Parts: The Good Parts*, (O'Reilly Media, Newton, 2008).
20. S. Tilkov, S. Vinoski, IEEE Internet Comp. **14**, 80 (2010)
21. R. Ferguson, *Introduction to JavaScript*. *In Beginning JavaScript*, (A press, Berkeley, 2019)
22. Invision Studio (2020). Retrieved 24 September 2020, from https://www.invisionapp.com/studio

23. Code, V. (2020). Visual Studio Code - Code Editing. Redefined. Retrieved 24 September 2020, from https://code.visualstudio.com/
24. T. Kahlert, K. Giza, Mikrosoft. 1 (2016).
25. Managed MongoDB Hosting | Database-as-a-Service. (2020). Retrieved 24 September 2020, from https://www.mongodb.com/cloud/atlas
26. Bialek, B. *The journey from a relational to a document-based database for fis balance sheet management. in the impact of digital transformation and fintech on the finance professional,* (Palgrave Macmillan, Cham, 2019).
27. Build software better, together. (2020). Retrieved 24 September 2020, from https://github.com/
28. J.D. Blischak, E.R. Davenport, G. Wilson, PLoS Comp. Bio. **12**, e1004668 (2016).
29. D. Lee, Proceed. Balis. Mark. Conf. 1 (2011)
30. B.N. Miller, D.L. Ranum, J. Anderson, *Python programming in context* (Jones & Bartlett Learning, Burlington, 2019)
31. C.S. Alm, *Web Application for customer-, account-and card register: the enfuce portal* (Semantic Scholar, New York, 2019)