Available online at
https://jcrinn.com/
https://crinn.conferencehunter.com/

Journal of Computing Research and Innovation 10(2) 2025

Journal of Computing Research and Innovation

CNN Integrated Mobile Application: Food Image Recognition for Recipe Generation

Muhammad Imran Nor Azlan Shah[1], Norlina Mod Sabri[2*], Gloria Jennis Tan[3], Zhiping Zhang[4]

[1]*Zen Computer System, Persiaran Flora, Cyber 12, 63000 Cyberjaya, Selangor, Malaysia.*
[2,3]*Faculty of Computer and Mathematical Sciences, UiTM Terengganu Branch, Kuala Terengganu Campus, 21080 Kuala Terengganu, Terengganu, Malaysia.*
[4]*College of Computer and Mathematics, Xinyu University, Jiangxi, P. R. China.*

ARTICLE INFO

Article history:
Received 27 June 2025
Revised 31 July 2025
Accepted 6 August 2025
Published 1 September 2025

Keywords:
CNN
Mobile Application
Food Image
Recognition

DOI:
10.24191/jcrinn.v10i2.554

ABSTRACT

The rapidly advancing of the digital world has encouraged the use of mobile applications in almost every aspects of our everyday life. This includes transforming the way we obtain our meal, whether to order from food providers or simply cook for ourselves. The CNN Integrated Mobile Application for Food Recipe Generation is intended to improve users' culinary experiences by offering suggestions for recipes and intelligent ingredient recognition. This research explores the Convolutional Neural Networks (CNN) algorithm to tackle the problems of effective ingredient management and minimize food waste through smartphone application. Through the mobile application, the ingredient photos can be scanned by users or uploaded, after which the CNN model processes the images to precisely identify the ingredients. The application provides users with a wide range of meal alternatives that are customized to their available ingredients by retrieving relevant recipes from external databases such as the Spoonacular API based on the recognized ingredients. The research methodology consists of 3 main phases, which are Data Preprocessing, CNN Implementation and Performance Evaluation. In this research, the CNN algorithm has generated a good and acceptable performance with more than 96% accuracy. This research has shown how machine learning, mobile development, and user-centric design can be successfully combined to create a useful tool for contemporary culinary demands. The app encourages a move towards more sustainable and thoughtful eating habits by acting as an incentive for change at the community level. When communities embrace these ideas, the app plays a key role in tackling more significant social issues associated with food waste, supporting international initiatives that are detailed in the UN Sustainable Development Goals (SDGs) for a more sustainable and responsible society.

[2*] Corresponding author. *E-mail address*: norli097@uitm.edu.my
https://doi.org/10.24191/jcrinn.v10i2.554

1. INTRODUCTION

In today's fast-paced society, finding a balance between quick meal preparation and making healthy food choices can sometimes be challenging. The evolving urban lifestyle has increased the demand for solutions that streamline our daily food decisions, while ensuring a balance between efficiency and mindful health practices. Traditional methods, including manual seeking and searching is not sufficient anymore, thus requires a new revolutionary method in providing ease while actively seeking for good food. Furthermore, people nowadays frequently consume food prepared by third-party entities such as restaurants, catering services, and takeaways, often through online delivery apps for added convenience (Chopra et al., 2023). As a result, looking for cooking recipes and ingredient information seems to be a tedious task. It might be laborious and time-consuming for people to manually check the ingredients that are available, choose what to prepare and look up appropriate recipes especially for those with hectic schedules. Due to the lack of proper planning, consumers may find it difficult to use products before their expiration dates, resulting in a greater chance that they would be wasted. This scenario results in financial losses and food waste (Rodrigues et al., 2023). A more effective and technologically advanced solution that modernizes the process of ingredient identification and recipe recommendation is becoming more and more necessary considering these disadvantages (Dsouza et al., 2024).

Currently, recipe recommendation systems have become widespread in order to provide clients with customized recipe recommendations based on their dietary preferences, level of cooking proficiency, and availability of ingredients. However, the majority of recipe recommendation systems rely on textual inputs or predefined categories of ingredients, which might not accurately reflect what the user has in their pantry (Anitha et al., 2023). Based on these motivations, a mobile solution is proposed with the ultimate aim of enhancing the users' quality of life by providing a seamless and intelligent food recipe assistance. Recent studies suggest that interactive and personalized tools, such as mobile applications, can be more effective in supporting real-time decision-making (Kansaksiri et al., 2023). The proposed solution is to overcome the issues of global food loss, hectic schedules, and inadequate recipe recommendations by utilizing CNN algorithm. The objective of the research is to develop and explore the Convolutional Neural Network (CNN) algorithm for the mobile application in the food image recognition. This study proposes the implementation of CNN to revolutionize the identification of food ingredients, initially focusing on 5 classes of ingredients including egg, lettuce, tomato, carrot and onion. In the context of the food scanning app, image recognition technology serves as the backbone for accurate and efficient identification of ingredients from user-provided images. The selection of CNN algorithms is a strategic one, driven by their robustness in handling image data and their capacity for scalability. CNN is a cutting-edge technique that is frequently used to train computers to recognize and classify images to obtain a trained dataset (Anitha et al., 2023).

In this study, the mobile application will recommend recipes based on the image detected. This will make finding recipes quicker and simpler than the manual searching. The development of the application supports the proposed issues by using image recognition to swiftly identify ingredients, addressing the consequences of poor planning and over-preparation. It encourages home cooking by recommending recipes based on available ingredients, aligning with modern lifestyles. Utilizing CNNs capability, the mobile application will revolutionize how individuals manage food, offering a sustainable, time-efficient, and personalized solution to daily culinary challenges. This paper is arranged into 5 main sections, beginning with the Introduction section. The following sections cover the Literature Review, Methodolgy, Results and Discussion and finally the Conclusion and Recommendation.

https://doi.org/10.24191/jcrinn.v10i2.554

2. LITERATURE REVIEW

This section presents brief reviews about similar works of the research and CNN algorithm. In similar works, several latest deep learning related research projects are compiled and summarized. The review on CNN discussing its advantages in food image recognitions.

2.1 Similar works

This section presents the deep learning research projects which have involved food identification and classification. A research was iniated in China to curb obesity, by using Yolo to recognize fast food and display its information (Chen & Chiang, 2025). Another research has identified real time food items using Yolov5 and also display its information (Anand et al., 2024). Both research have showed improved performance of yolo in their projects. In UK, in order to reduce wastage, a food image recognition research has been conducted using CNN and the accuracy has been improved (Boyd et al., 2024).

A recipe recommendation system using YOLOv5 for ingredient detection and calorie calculation is proposed by Swain et al. (2023). Meanwhile, a similar work introduces a system that employs Convolutional Neural Networks (CNN) for personalized recipe recommendations based on user-provided ingredient images (Anitha et al., 2023). Both approaches offer their own strengths as the system using YOLOv5 includes consideration of nutritional value while CNN focuses on user experience and tailored recipe recommendations. CNN also addresses ingredient scarcity, simplifying the recipe suggestion process, and enhancing user experience. However, both methods also come with similar weaknesses in which accuracy of ingredient detection is highly dependent on image quality and data training.

In the research by Li (2023), the article proposes a system using CNN and Transformer for predicting food recipes from images. Strengths lie in CNN's feature extraction, but weaknesses include difficulty capturing fine details. Combining CNN and Transformer models can overcome these weaknesses, improving recipe prediction accuracy, and suggesting that knowledge distillation enhances model portability. Chopra et al. (2023) presents a solution for image-recipe association in Indian cuisine using machine learning. Strengths include accurate predictions for a specific cuisine. A potential weakness is the limitation to Indian cuisine, potentially hindering the retrieval of recipes from other cuisines.

Based on literatures, this research has been motivated by the performance of CNN in the previous food image recognition research projects. CNN has proven to be able to produce good performance in the food detection and classification problems. By linking input data to spatial features, the CNN algorithm can detect and extract key information from images, making it highly useful across various applications (Sri & Balakrishnan, 2024). Table 1 shows the similar works using deep learning algorithms and their results in the food image recognitions.

Table 1. Similar works using deep learning in food image recognitions

No	Title	Problem	Result	Reference
1	Deep learning-based automatic food identification with numeric label	Food identification and calorie management system is proposed due to obesity issue	Yolo classify and count objects within improved time complexity	Chen and Chiang (2025)
2	Food Recognition Using Deep Learning for Recipe and Restaurant Recommendation	Identify various food items in real time from an image or video of multi-object meals	YOLOV5 obtained average accuracy over 80%	Anand et al. (2024)

https://doi.org/10.24191/jcrinn.v10i2.554

3	Fine-Grained Food Image Recognition: A Study on Optimising Convolutional Neural Networks for Improved Performance	Increasing issue of food waste	Improved accuracy for DenseNet	Boyd et al. (2024)
4	Recipe Recommendation Using Image Classification	Recipe recommendation systems use textual inputs or pre-defined categories of ingredients, not accurately reflect what the user has in their pantry	The use of CNNs provides high accuracy in identifying ingredients	Anitha et al. (2023)
5	Ingredients to Recipe: A YOLO-based Object Detector and Recommendation System via Clustering Approach	Website lack real time analysis of ingredients used for cooking	YOLO is a lot faster algorithm than its competitors, with a maximum frame rate of 45 FPS.	Swain et al. (2023)
6	Ingredient Detection, Title, and Recipe Retrieval from Food Images	Simply looking at a picture of food on social media does not provide access to the cooking process.	Recipe generation model presents notable advancements over prior models in the field	Chopra et al. (2023)
7	Predict Food Recipe from Image Using CNN/Transformer & Knowledge Distillation for Portability	Chef cannot remember the detailed steps to all the dishes, and ordinary citizens do not have the time or energy to create their recipes.	The resulting model's complexity was minimized	Li (2023)

2.2 Convolutional neural network (CNN)

CNNs are an effective method for image classification in computer vision (Anitha et al., 2023). Unlike traditional image recognition algorithms, CNN excels in capturing intricate patterns and features within images, making them highly effective for identifying nuanced details in food ingredients. The hierarchical structure of CNN enables them to understand spatial relationships and contextual information, enhancing the precision of ingredient recognition. While CNN may be slower due to their deep architecture, the trade-off is increased accuracy, a crucial factor in the context of the app where precision is paramount. In contrast, certain image recognition algorithms might not have the intricacy and depth required to precisely identify minute details in food photos. Certain algorithms might process information more quickly, but at the expense of accuracy, which could result in incorrect recommendations and misclassifications. This comparison highlights CNN's applicability to the complex issue of culinary ingredient recognition. Classification of images using CNN is an optimal solution as its built-in convolutional layers reduce the high dimensionality of the image without losing its information (Navaprakash et al., 2025; Singh & Susan, 2023).

However, it's important to recognise CNN's shortcomings, particularly its high computational overhead, which causes processing times to increase. In spite of this limitation, the precision obtained by CNN is in perfect harmony with the app's goals of offering accurate ingredient identification and customised recipe suggestions. CNN's integration guarantees that accuracy comes before speed, a calculated move that improves both the app's general use and its ability to reduce food waste. Convolutional Neural Networks are very accurate networks for image classification and recognition in a short time (Kurian & Jacob, 2023). In conclusion, the food scanning app made a calculated decision to include the CNN algorithm because of its capacity to improve ingredient recognition and suggestion accuracy. The contrast with other algorithms highlights CNN's distinct benefits and establishes them as the best option for the app's complex image recognition tasks. Deep learning, especially convolutional neural networks (CNNs), has become the cornerstone of modern food image recognition systems. These models excel at automatically extracting features from images, enabling accurate classification and segmentation of food items (Gautam & Khanna, 2024).

https://doi.org/10.24191/jcrinn.v10i2.554

3. METHODOLOGY

The research methodolgy consists of 3 main phases, which are Data Preprocessing, CNN Implementation and Performance Evaluation phase.

3.1 Data preprocessing

The dataset for this research has been obtained from Roboflow.com, a well-known online website used for storing and sharing datasets. The dataset used in this study was specifically selected to align with the research focus on ingredient identification and was sourced from Roboflow's extensive library. Roboflow contains the required datasets that fill in the criteria and requirements needed to develop the app, mainly the availability of datasets of the 5 chosen classes; which are egg, tomato, lettuce, onion and carrot.

Resizing the images is the first step in the preprocessing stage of creating the CNN model. By giving the CNN a constant input size, uniform sizing of all the images in the dataset improves its performance. It is essential to resize every image to a predetermined size before training in order to expedite this procedure and prevent any problems. Since they lighten the computational load and speed up processing, smaller image sizes are especially advantageous for mobile applications with low processing power which improves user experience overall. Image resizing is integrated into the code of CNN model development through the utilization of the tf.keras.preprocessing.image_dataset_from_directory function. The image size is set to 150x150 pixels, which is common for CNN.

3.2 CNN implementation

In this research project, the mobile application employs the CNN algorithm to precisely detect and categorize various food products, specifically eggs, tomatoes, lettuce, carrots, and onions. The CNN model was put into practice using the TensorFlow and Keras libraries. The architecture makes use of many convolutional layers, max-pooling layers, and fully linked layers to extract and learn features from images. Convolutional layers extract features, max-pooling layers minimize spatial dimensions, and fully connected layers perform the final classification. Then, the model is compiled using the Adam optimizer, employing categorical cross-entropy as the loss function, and accuracy is chosen as the evaluation metric. Fig. 1 shows the code snippet for building the CNN model.

```
# Define the model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_HEIGHT, IMG_WIDTH, 3)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(5, activation='softmax')  # Assuming 5 classes: egg, tomato, lettuce, carrot, onion
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Fig.1. Code snippet of building the CNN model

Fig. 1 shows the CNN model is built using the Sequential class, which creates a linear stack for layer-by-layer definition. The CNN uses multiple layers to learn and extract information from input images. Convolutional layers extract features, max-pooling layers minimize spatial dimensions, and fully connected

https://doi.org/10.24191/jcrinn.v10i2.554

layers perform the final classification. The first Conv2D layer uses 32 filters, each with a 3x3 kernel size, and the second layer uses 64 filters, each with a 3x3 kernel size. To incorporate nonlinearity into the network and allow the model to learn complicated patterns, the ReLU (Rectified Linear Unit) activation function is used. The model's input shape is defined as (150, 150, 3), indicating that the input images have a 150x150 pixel resolution and three color channels (RGB).

The MaxPooling2D layers are used to downsample the feature maps by selecting the maximum value inside each 2x2 zone, reducing spatial dimensions while retaining the most important information. The Flatten layer, which comes after the pooling layers, transforms the 2D feature maps into 1D vectors that may be used as input for the fully connected layers. The ultimate classification procedure is carried out by dense layers, sometimes referred to as fully connected layers. With 512 neurons, the initial Dense layer is a strong layer for feature learning. The output Dense layer uses the softmax activation function to create a probability distribution over the five food classes, which are represented by the five neurons in the layer. Finally, the model is compiled using the Adam optimizer, employing categorical cross-entropy as the loss function, and accuracy is chosen as the evaluation metric.

Processed images and labels are used to train the model. During the training process, the model is fed training data, weights are adjusted by backpropagation, and model parameters are improved. Using the dataset supplied by the train_generator, the model is trained for a predefined number of epochs via the model.fit() method. The model receives batches of training data from the train_generator, a data generator. In order to guarantee that the model is trained on different portions of the dataset in each epoch, it creates batches of images and labels. The model's performance is evaluated using validation data at the end of each training period, which enables tracking of the model's evolution and overfitting detection. Finally the model is converted to TensorFlow Lite in order to be used on mobile devices. A suite of tools called TensorFlow Lite makes it easier to run TensorFlow models on embedded, mobile, and Internet of things devices.

3.3 Performance evaluation

Evaluating the performance of the classification algorithm involves a few different metrics to measure the algorithm’s effectiveness and accuracy. Accuracy, precision, recall and F1 are some of the metrics in performance measurements. The classified data obtained via the proposed classification algorithm would be measured in a confusion matrix. The confusion matrix contains the classifier’s measurement parameters with four essential properties. Fig. 2 shows the confusion matrix.

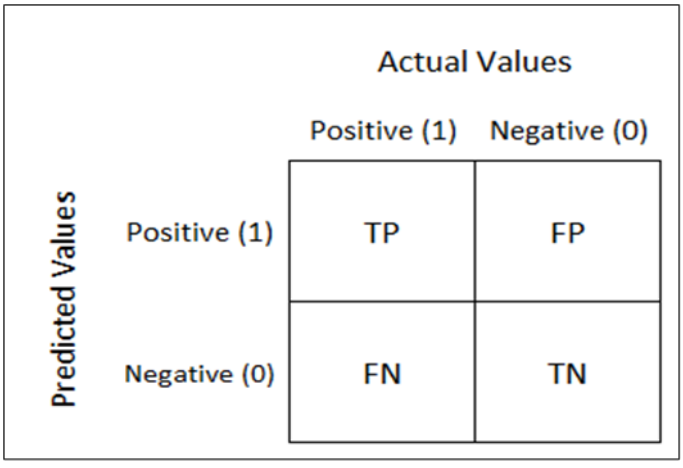

Fig. 2. Confusion matrix

https://doi.org/10.24191/jcrinn.v10i2.554

Based on Fig. 2, TP (True positive) represents the total number of cases that are predicted true and correctly predicted, TN (True negative) represents the total number of cases that are predicted false and correctly predicted, FP (False positive) is the total number of cases that are predicted true but a wrong prediction, while FN (False negative) represents the total number of cases that are predicted false and is a wrong prediction. Using the values obtained via the confusion matrix, the performance evaluation metrics could easily be calculated using the formula in Eq. (1) – (4). Eq. (1) represents the measurement of accuracy, while Eq. (2) represents the measurement of precision. Eq. (3) represent the recall evaluation and Eq. (4) represents the F1 score evaluation.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\frac{TP}{TP + FP} \tag{2}$$

$$\frac{TP}{TP + FN} \tag{3}$$

$$2 * \frac{accuracy * recall}{accuracy + recall} \tag{4}$$

3.4 System architecture

System architecture shows the multiple layers between users, devices, networks, and data. Fig. 3 shows the system architecture of the project. The system architecture of the CNN Integrated Mobile Application for Food Recipe Assistance is meticulously planned to provide accurate and efficient ingredient recognition and recipe recommendations. The primary application of the system for processing photos and recognizing ingredients is the Convolutional Neural Network (CNN) algorithm. To begin the process, a sizable dataset consisting of images of five distinct ingredients was obtained from Roboflow. This dataset goes through a data preprocessing step where photos are shrunk and standardized in order to preserve consistency and enhance the model's performance. The CNN algorithm is then fed the processed data. The model is retained for later usage after being assembled with the proper configurations during the training process. Rigorous testing is conducted to assess the model's accuracy and ensure it meets the required performance standards. This design makes extensive use of the user interface (UI), which provides a clear-cut and easy-to-use experience. The user interface (UI) enables users to submit photos from their gallery and uses the device's camera to take shots of ingredients. Moreover, users can manually look up recipes by utilizing the UI's input interface. The CNN model examines the image once an ingredient has been scanned or uploaded in order to identify the item. Utilizing this recognition, the system retrieves relevant recipe recommendations from an external API, the Spoonacular API, which offers a large recipe database using this recognition. The user interface then displays the recommended recipes to the user, guaranteeing a smooth and interesting experience. The user is then presented with the suggested recipes via the UI, ensuring a seamless and engaging experience. Additionally, the integration of TensorFlow Lite optimizes the CNN model for mobile devices, providing quick and responsive ingredient recognition. The project's comprehensive approach is demonstrated by the system architecture, which ensures both technical efficiency and user satisfaction through the clear delineation of components and their interconnections. This architecture's application

https://doi.org/10.24191/jcrinn.v10i2.554

demonstrates how cutting-edge technologies can be used to improve routine chores, making it a useful tool for both home cooks and food connoisseurs.

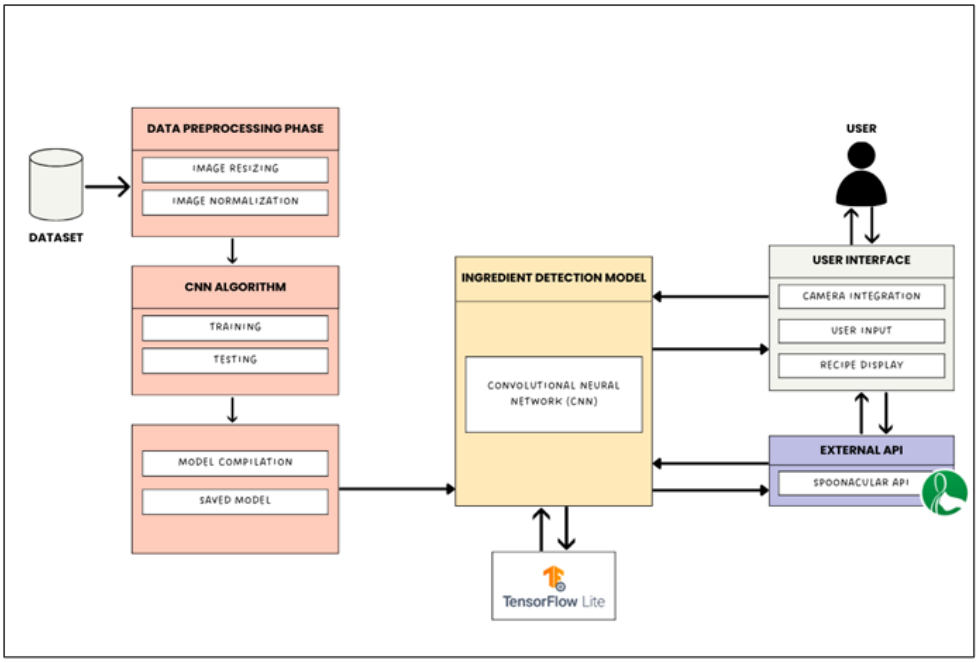

Fig. 3. System architecture of the project

3.5 Use case diagram

A use case diagram is constructed to identify the relationship between users and functionalities involved in the application. Fig. 4 shows the use case diagram designed for this application. The use case diagram outlines the interaction between the user and the application, highlighting the primary functions and processes involved. Based on Fig. 3, the process begins when the user Launches the Application. This initial step involves opening the app on their mobile device, which brings up the main interface of the food recipe assistant. From here, the user has several options for interacting with the system. The user can choose to search recipes manually, Scan Ingredients using Camera, or Upload Image. The search recipe manually option allows users to type in keywords or browse through a list of recipes based on their preferences. This is useful for users who already have a specific recipe in mind or want to explore new recipes. Alternatively, the Scan Ingredient using Camera option enables users to take a photo of an ingredient directly within the app. This image is then processed by the application to identify the ingredient using Convolutional Neural Network (CNN) technology. The upload image function allows users to select a pre-existing photo of an ingredient from their device's gallery. This flexibility ensures that users can easily provide the necessary input data for ingredient recognition. Once an image is provided through either scanning or uploading, the system proceeds to recognize ingredients using CNN. The CNN model analyzes the image to accurately identify the ingredient, leveraging its trained dataset and recognition capabilities. If the ingredient is successfully recognized, the application will Provide Recipe Recommendation based on the identified ingredient. This step involves querying the database for recipes that include the recognized ingredient, ensuring that users receive relevant and practical recipe suggestions. Users can then view the recipe, which provides detailed instructions and ingredient lists for preparing the recommended dishes. In the event of any issues, such as unrecognized ingredients or technical errors, the system includes an Error Handling process to manage these situations and ensure a smooth user experience.

https://doi.org/10.24191/jcrinn.v10i2.554

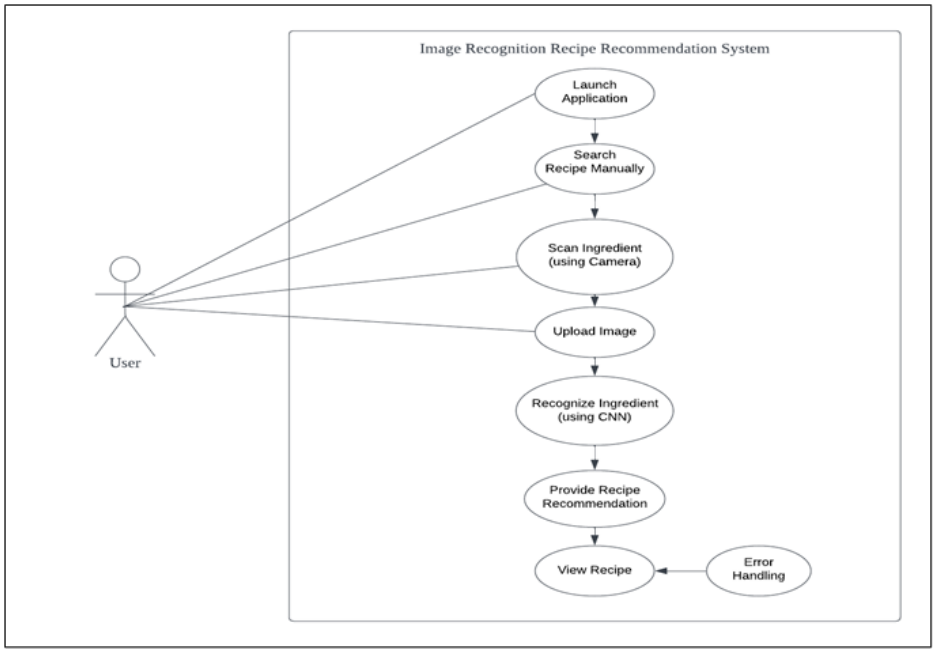

Fig. 4. Use case diagram for the apllication development

4. RESULTS AND DISCUSSION

This section covers the evaluation of CNN model using data split ratio, graphical analysis of the training and validation accuracy over epochs and the confusion matrix results. The user intefaces of the prototype are presented at the last part of the section.

4.1 Evaluation of CNN model

The evaluation of the CNN model involved testing its development using three different data splitting ratios to determine which configuration yielded the highest accuracy. The chosen splits were 70:30, 80:20, and 90:10, and each was tested over 30 epochs to maintain consistency. The goal was to identify the optimal ratio that would balance the training and validation data, ensuring that the model generalizes well to unseen data. Table 2 shows the data split ratio and the results.

Table 2. Data split ratio and results

Data Splitting Ratio	Test Accuracy	Model Accuracy
80:20	0.870	95.8%
90:10	0.917	96.8%
70:30	0.925	96.8%

Based on Table 2, these results indicate that the 70:30 split provided the highest test accuracy, closely followed by the 80:20 and 90:10 splits. However, the differences in accuracy between the splits were relatively small, suggesting that the model performs consistently well across different training-validation splits. This consistency is critical for ensuring the robustness of the CNN model in real-world applications. Fig. 5 shows the epochs for the testing using 70:30 split.

https://doi.org/10.24191/jcrinn.v10i2.554

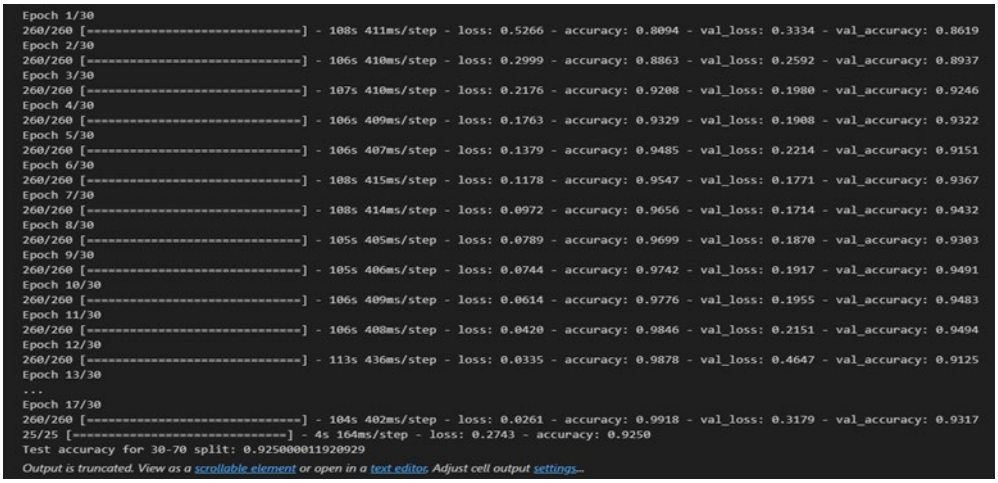

Fig. 5. Epochs for 70:30 split testing

Early stopping was used in the model-building process to prevent overfitting and guarantee effective training. Early stopping keeps track of how well the model is performing on the validation set and stops training when it no longer improves. This method helps prevent overfitting, which occurs when the model learns the noise in the training data rather than the underlying patterns and is especially helpful when training for a large number of epochs. An extra measure of protection was offered by the early stopping method, which made sure that the model training would end if more epochs did not result in performance improvements. This prevented overfitting and saved computing resources.

4.2 Analysis of results and graphical representation

Graphical analysis of the training and validation accuracy over epochs for each data split reveals valuable insights into the model's learning process. The training accuracy curves consistently approached high values, indicating that the model learned effectively from the training data. The validation accuracy curves, meanwhile, demonstrated the model's ability to generalize to unseen data. Fig. 6 shows the graph of 70:30 model split, which has shown the best results.

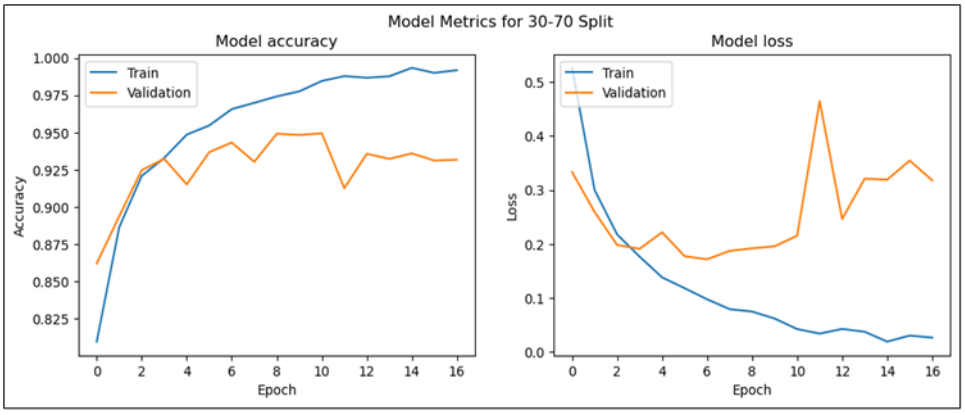

Fig. 6. Graph of 70:30 model split

https://doi.org/10.24191/jcrinn.v10i2.554

The graphs in Fig. 6 display the model metrics for a 70-30 training-validation split, showing trends in accuracy and loss over 16 epochs. In the accuracy graph, the training accuracy (blue line) steadily increases, approaching near-perfect accuracy as epochs progress, indicating that the model is learning effectively from the training data. The validation accuracy (orange line), while generally improving, shows some fluctuations, which may suggest slight overfitting as the model starts to memorize the training data rather than generalizing well. In the loss graph, the training loss (blue line) consistently decreases, reflecting the model's improving performance. The validation loss (orange line), however, shows a more variable trend with occasional spikes, further indicating potential overfitting issues. Despite these fluctuations, the overall trend suggests that the model is learning and improving, but adjustments in hyperparameters or additional regularization may be needed to enhance generalization to new data.

Fig. 7 shows the graphs which display the model metrics for a 80-20 training-validation split, showing trends in accuracy and loss over 12 epochs. In the accuracy graph, the training accuracy (blue line) shows a steady increase, indicating effective learning from the training data, and reaches high levels by the end of the epochs. The validation accuracy (orange line), while initially improving, shows some fluctuations, suggesting that the model is experiencing slight overfitting as it starts to memorize the training data rather than generalizing. In the loss graph, the training loss (blue line) consistently decreases, reflecting the model's improving performance. The validation loss (orange line) initially decreases but then fluctuates, indicating variability in how well the model generalizes to unseen data. Despite these fluctuations, the overall trend suggests that the model is learning well but may benefit from adjustments in hyperparameters or regularization techniques to enhance its generalization capabilities and reduce overfitting.

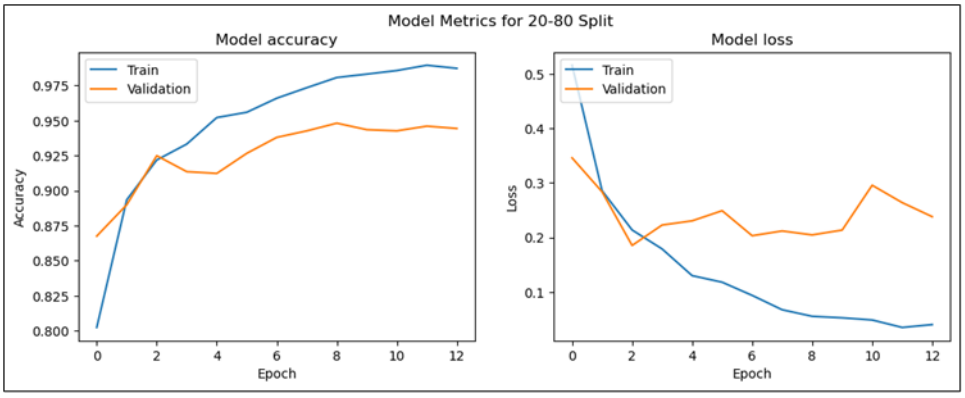

Fig. 7. Graph of 80:20 model split

Fig. 8 shows the graphs which display the model metrics for a 90-10 training-validation split, showing trends in accuracy and loss over 12 epochs. In the accuracy graph, the training accuracy (blue line) steadily increases, indicating effective learning from the training data and approaching near-perfect accuracy. The validation accuracy (orange line) also shows an overall upward trend, although it exhibits some fluctuations, suggesting minor overfitting but still maintaining high performance. In the loss graph, the training loss (blue line) consistently decreases, reflecting the model's improving ability to minimize errors during training. The validation loss (orange line), while initially decreasing, shows some variability, indicating fluctuations in the model's generalization to unseen data. Despite these fluctuations, both accuracy and loss trends suggest that the model is performing well, but further fine-tuning of hyperparameters or regularization techniques may be beneficial to enhance its stability and generalization.

https://doi.org/10.24191/jcrinn.v10i2.554

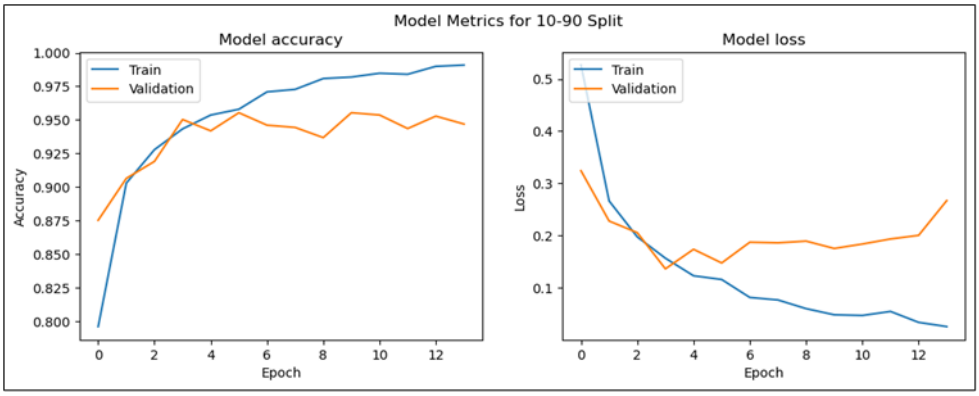

Fig. 8. Graph of 90:10 model split

Based on the overall results, the 70:30 split, the model showed a slight edge in test accuracy, suggesting that having more validation data helps in fine-tuning the model's hyperparameters and ensuring better generalization. The 80:20 split provided a balanced approach, yielding high test accuracy while maintaining substantial training data. The 90:10 split, while slightly less effective in test accuracy, still produced robust results, highlighting the model's capacity to learn well even with a smaller validation set.

In conclusion, the evaluation of the CNN model through different data splitting ratios and the implementation of early stopping have proven to be effective strategies for developing a robust ingredient recognition system. Due to the early stopping, the number of epochs conducted has been different in the data split ratio testing. The number of epochs in machine learning training differs when using early stopping because early stopping is a dynamic, adaptive mechanism designed to prevent overfitting and unnecessary training. In this research, the 70:30 split emerged as the most optimal configuration, providing the highest test accuracy. However, the consistent performance across different splits underscores the reliability of the CNN model in accurately identifying ingredients, paving the way for its successful deployment in the food recipe assistance application.

4.3 Confusion matrix results

The accuracy testing for the ingredient detection system involves the utilization of a confusion matrix reflecting the model's classification performance on 5 datasets with over 14,000 images. The matrix comprises four key metrics: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). TP represents instances where the model correctly identifies ingredients within the 5 classes, FP denotes instances where the model incorrectly identifies non-ingredients as ingredients, FN indicates instances where the model incorrectly identifies ingredients as non-ingredients, and TN signifies instances where the model correctly identifies non-ingredients. Fig. 9 show the results of the confusion matrix for 70:30 data split.

https://doi.org/10.24191/jcrinn.v10i2.554

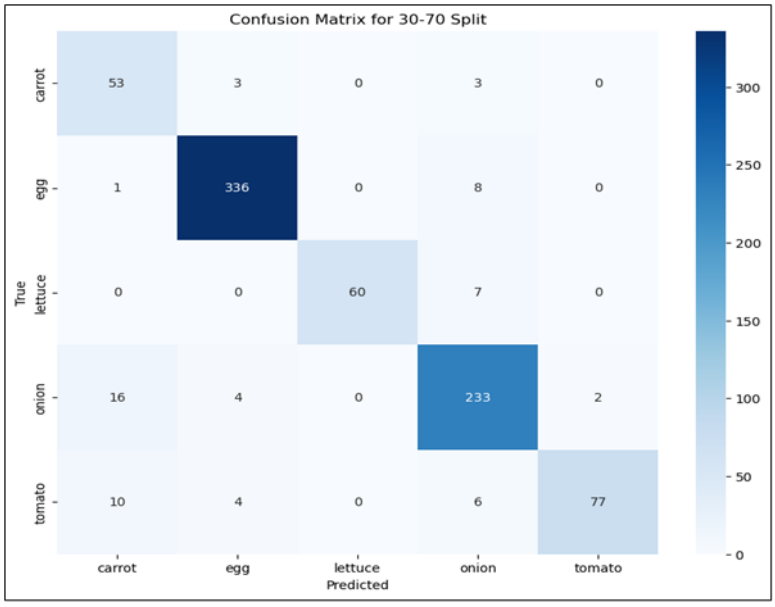

Fig. 9. Confusion matrix results

For the specific CNN model being evaluated, the confusion matrix in Fig. 9 shows the following values for different classes. The model correctly identified 53 instances of carrot but misclassified 3 as egg, 3 as onion, and 0 as tomato. For the egg class, the model accurately predicted 336 instances, while 1 was misclassified as carrot, 0 as lettuce, 8 as onion, and 0 as tomato. Lettuce was correctly identified in 60 instances, with 0 misclassified as carrot, egg, and 7 as onion. Onion had 233 correct predictions, but 16 were misclassified as carrot, 4 as egg, and 2 as tomato. Lastly, the tomato class had 77 correct predictions, with 10 misclassified as carrot, 4 as egg, and 6 as onion.

Based on these values, the accuracy, precision, recall, and F1 score are calculated to evaluate the performance of the CNN model. The overall accuracy of the CNN model is determined to be 0.92, indicating that 92% of the model's predictions are correct. For the egg class, precision is calculated as 96.6%, indicating that 96.6% of the samples identified as egg by the model were indeed egg. The recall for the egg class is also 96.6%, showing that the model accurately identifies egg 96.6% of the time when predicting a sample as egg. The F1 score for the egg class is 96.6%, providing a balanced measure of the model's performance for this class. These metrics collectively provide a comprehensive evaluation of the model's performance across different classes. Table 3 shows the results of the confusion matrix results.

Table 3. Calculation of accuracy, precision, recall, F1 score

	Calculation	Answer	Percentage
Accuracy	**(TP+TN) / (TP+TN+FP+FN)** (152+452) / (152+452+193+193)	0.968	96.8%
Precision	**TP / (TP+FP)** 152 / (152+193)	0.842	84.2%
Recall	**TP / (TP+FN)** 152 / (152+193)	0.970	97.0%
F1 Score	**2*(precision recall*) / (precision + recall)** *2*(0.441*0.441) / (0.441 + 0.441)	0.896	89.6%

https://doi.org/10.24191/jcrinn.v10i2.554

The results of the system are consistent with the calculations done manually, as shown in Fig. 9. The system's calculations for metrics like accuracy, precision, recall, and F1 score are verified by the consistency of the results. The accuracy and dependability of the implemented ingredient detection system are highlighted by the match between the values generated by the system and the calculations done manually. This validation ensures the validity of the evaluation metrics used in the project and increases confidence in the system's ability to accurately assess the Convolutional Neural Network model's performance.

4.4 System usability scale (SUS) results

Fig. 10 shows the histogram representing the distribution of SUS scores for the mobile applcation. The SUS assessment was conducted to evaluate the usability of the mobile application. A total of 20 respondents participated in the evaluation and the SUS scores ranged from approximately 55 to 100, with the average score recorded at 83.33**.** This average score indicates that the application demonstrates excellent usability, with strong acceptance and satisfactiona among users. The histogram reveals that the majority of users rated the system between 80 and 95, with the highest concentration of scores occurring in the 90–95 range. This suggests that most users found the system to be intuitive, efficient, and satisfying to use. Only one respondent provided a relatively lower score, which appears to be an outlier. This might indicate a unique usability issue or a different user expectation, but not a critical issue. Overall, the results suggest that the application is well-designed and user-friendly, with strong user satisfaction and minimal usability issues.

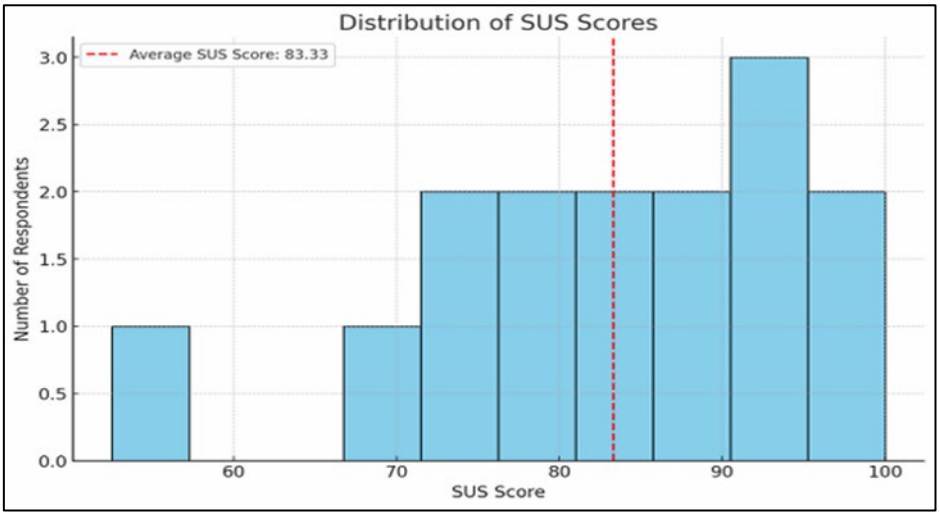

Fig. 10. SUS score results

4.5 User interface of the mobile application

Microsoft Visual Studio's Flutter is used to create the user interface for the "CNN Integrated Mobile Application for Food Recipe Assistance" prototype. The app's homepage's graphical user interface (GUI) is shown in Fig. 11. When the homepage loads, the majority of the white space is occupied by the camera screen. This bold design decision was made on purpose to draw attention to the app's core function, which is ingredient detection. Towards the bottom of the page are three primary buttons: an "image picker," a "camera icon," and a "search icon".

https://doi.org/10.24191/jcrinn.v10i2.554

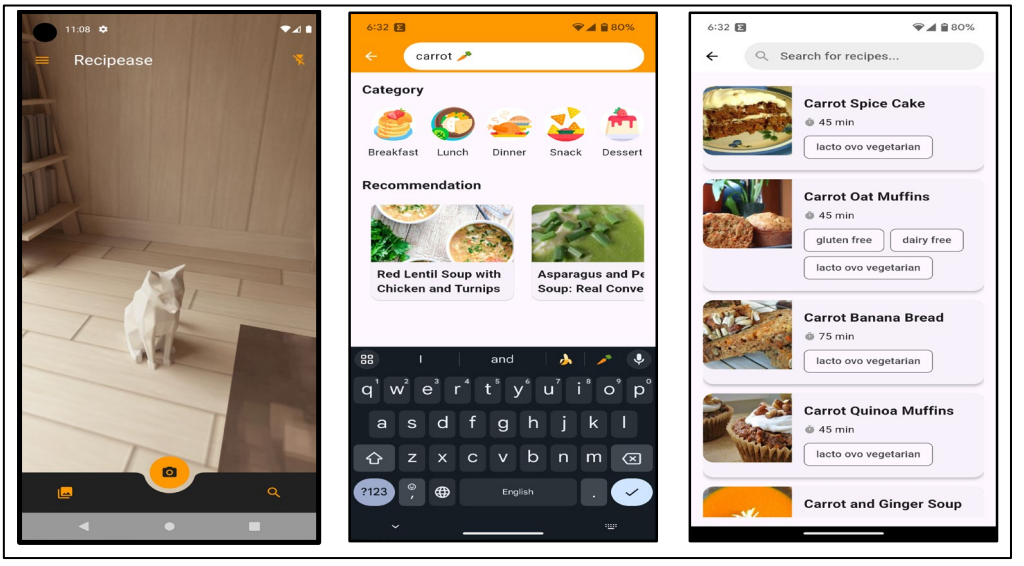

Fig. 11. Homepage and the search page of the application

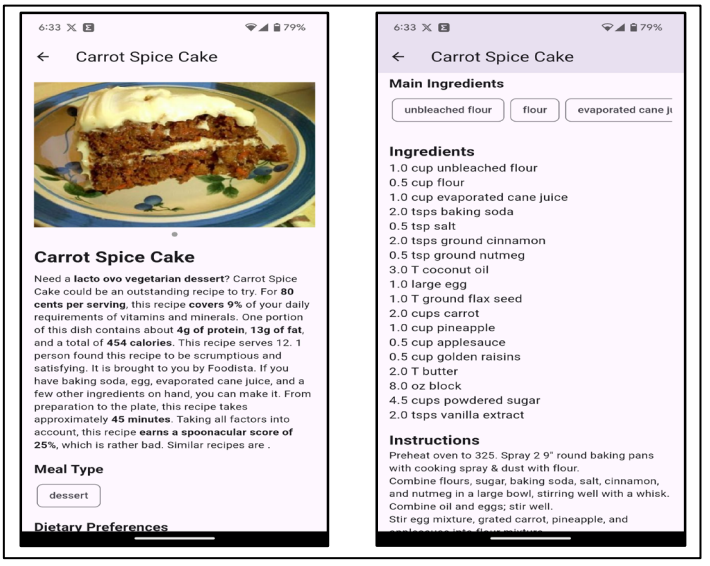

Fig. 12. Sample recipe based on menu

As shown also in Fig. 11, pressing the search icon brings up the search page and applies the Spoonacular API. Users can manually search for recipes on this page by entering keywords into the search bar. The keywords are not limited to name of recipe or ingredient used but even cuisine, dietary preferences and even meal types such as breakfast, lunch or dinner. This feature makes sure that even if consumers decide not to use the picture detection functionality, they can still find recipes. Fig.12 shows the sample of recipe details for Carrot Spice Cake when user clicks on the menu.

https://doi.org/10.24191/jcrinn.v10i2.554

In the meantime, the CNN model for ingredient detection is directly connected to the camera icon and image selection button. The software examines the image and goes to the statistics page when an image is selected or a snapshot is taken. Then, it shows the detected ingredient and its confidence level. An example of the statistics page with "carrot" selected as the detected ingredient is displayed in Fig. 10. Users can view the ingredient that the CNN model identified in detail on this page, along with the confidence level and the image that was utilized for identification. As illustrated also in Fig. 13, if an ingredient or item is entered that does not fall into one of the five predefined classes, the system correctly detects the lack of a recognized ingredient and displays a pop-up message indicating a detection error. This visual aid for error handling makes sure users are aware of any problems with their input and improves the user experience overall by giving quick, clear feedback.

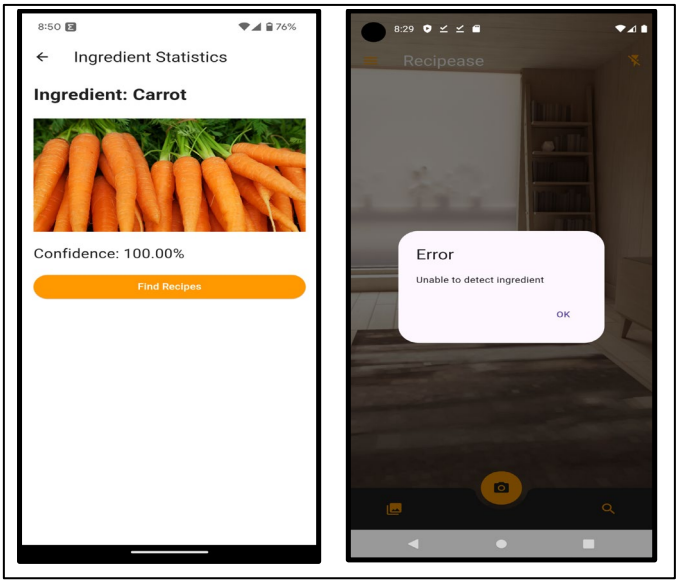

Fig. 13. Statistics page with carrot as input and pop-up message of error handling

5. CONCLUSION AND RECOMMENDATION

CNN Integrated Mobile Application for Food Recipe Assistance has effectively accomplished its goals by utilizing cutting-edge machine learning methods to improve component identification and offer customized recipe suggestions. The experiment showed how Convolutional Neural Networks (CNN) can be used effectively to reliably identify items from user-provided photos, which will help with meal preparation efficiency and cut down on food waste. The significance of the research is that it is in perfect harmony with the Sustainable Development Goal (SDG) 12 of the United Nations, which places a strong emphasis on responsible production and consumption. It promotes effective ingredient management, enabling consumers to maximise the amount of food in their pantries and drastically lowering the possibility of needless food waste. This improves everyday cooking's usefulness and supports the application's objective of encouraging responsible consumption. The software supports a more sustainable and thoughtful approach to home cooking by providing people with the tools to use ingredients effectively. The main contribution of the research project is the food ingredient image recognition using CNN that is implemented for mobile platform. Most of the current similar research project are using web based platform which has

https://doi.org/10.24191/jcrinn.v10i2.554

its limitation in mobile application accessibility, such as poor screen reader support or low contrast and small text.

This mobile application was developed and implemented using an organized methodology, guaranteeing a reliable user experience. Thorough testing and assessment established the good accuracy of 96.8% and dependability of the application, fulfilling the project's main objectives. Nevertheless, the project encountered many constraints that offer prospects for forthcoming enhancements. This initial project only focus on 5 common ingredients, which limits its usefulness for users with a variety of culinary requirements. Furthermore, relying on outside APIs to provide recipe recommendations could lead to dependency and possible restrictions on the consistency and availability of data. Future improvements should focus on broadening the dataset to include a wider variety of ingredients, improving the CNN model's training to increase accuracy, and adding more features like meal planning tools and nutritional data. Furthermore, investigating the incorporation of offline features and diminishing reliance on external APIs may enhance the robustness of the program and enhance user contentment. These enhancements will not only address the current limitations but also pave the way for a more comprehensive and versatile solution, solidifying the project's contribution to the culinary and technology fields.

6. ACKNOWLEDGEMENTS/FUNDING

The authors would like to acknowledge the support of UiTM Cawangan Terengganu for the continous support given to the research initiatives in the university.

7. CONFLICT OF INTEREST STATEMENT

The authors declare that this research was conducted in the absence of any conflict of interests.

8. AUTHORS' CONTRIBUTIONS

Muhammad Imran Nor Azlan Shah: Development and analysis, writing original draft; **Norlina Mohd Sabri**: Supervision and editing draft; **Gloria Jennis Tan**: Review manuscript; **Zhiping Zhang**: Formatting and final review.

9. REFERENCES

Anand, L., Tyagi, R., & Mehta, V. (2024). Food recognition using deep learning for recipe and restaurant recommendation. In Bhateja, V., Lin, H., Simic, M., Attique Khan, M., Garg, H. (Eds) Cyber Security and Intelligent Systems. *Lecture Notes in Networks and Systems (pp. 1056)*. Springer, Singapore. https://doi.org/10.1007/978-981-97-4892-1_23

Anitha, E., Nandhini, S., Palani, H. K., & Marshal, M. C. (2023). Recipe recommendation using image classification. In *Proceedings of the 2023 5th International Conference on Smart City Applications* (pp. 1023–1033). Springer International Publishing.

Boyd, L., Nnamoko, N., & Lopes, R. (2024). Fine-grained food image recognition: A study on optimising convolutional neural networks for improved performance. *Journal of Imaging, 10*(6), 126. https://doi.org/10.3390/jimaging10060126

Chen, Y. C., & Chiang, H. C. (2025). Deep learning-based automatic food identification with numeric label. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-025-20648-x

https://doi.org/10.24191/jcrinn.v10i2.554

Chopra, B., Jain, G., Mahendra, N., Sinha, S., & Natarajan, S. (2023). Ingredient detection, title and recipe retrieval from food images. In *International Conference on Ubiquitous and Future Networks, ICUFN* (pp. 288-293). IEEE. https://doi.org/10.1109/ICUFN57995.2023.10199735

Dsouza, H. J., Nayak, K. S., Karkera, K. K., Varghese, M., & Shreejith, K. B. (2024). Ingredient detection and recipe recommendation using deep learning. *International Journal of Advanced Research in Computer and Communication Engineering*, *13*(3), 630–635. https://ijarcce.com/wp-content/uploads/2024/04/IJARCCE.2024.133100.pdf

Gautam, G., & Khanna, A. (2024). Content based image retrieval system using cnn based deep learning models. *Procedia Computer Science, 235*(2023), 3131–3141. https://doi.org/10.1016/j.procs.2024.04.296

Kansaksiri, P., Panomkhet, P., & Tantisuwichwong, N. (2023). Smart cuisine: Generative recipe & ChatGPT powered nutrition assistance for sustainable cooking. *Procedia Computer Science, 225*, 2028–2036. https://doi.org/10.1016/j.procs.2023.10.193

Kurian, V., & Jacob, V. (2023). Importance of CNN in the classification of remote sensing images. In *Proceedings of the ACCTHPA 2023 - Conference on Advanced Computing and Communication Technologies for High Performance Applications* (pp. 1-4). IEEE. https://doi.org/10.1109/ACCTHPA57160.2023.10083375

Li, Z. B. (2023). Predict food recipe from image using CNN/transformer & knowledge distillation for portability. In *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering* (pp. 852–859). IEEE. https://doi.org/10.1109/ICSECE58870.2023.10263583

Navaprakash, N., Reddy, S. V., & Dakshinesh, S. (2025). Improving accuracy in text extraction from images using region-based convolutional neural networks algorithm compared to convolutional neural network algorithm. In *2025 International Conference on Artificial Intelligence and Data Engineering (AIDE)* (pp. 706-710). IEEE. https://doi.org/10.1109/AIDE64228.2025.10987308

Rodrigues, M. S., Fidalgo, F., & Oliveira, Â. (2023). RecipeIS—Recipe recommendation system based on recognition of food ingredients. *Applied Sciences (Switzerland), 13*(13), 1-19. https://doi.org/10.3390/app13137880

Singh, P. K., & Susan, S. (2023). Transfer learning using very deep pre-trained models for food image classification. In *2023 14th International Conference on Computing Communication and Networking Technologies* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCCNT56998.2023.10307479

Sri, B. R., & Balakrishnan, T. S. (2024). *Classification of pests in agricultural farms using convolutional neural network compared to artificial neural network*. 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1–4. https://doi.org/10.1109/ICCCNT61001.2024.10725825

Swain, M., Manyatha, A. R., Dinesh, A. S., Sampatrao, G. S., & Soni, M. (2023). Ingredients to recipe: A YOLO-based object detector and recommendation system via clustering approach. In *Proceedings of the 3rd International Conference on Artificial Intelligence and Smart Energy* (1397-1404). IEEE. https://doi.org/10.1109/ICAIS56108.2023.10073769

https://doi.org/10.24191/jcrinn.v10i2.554