JCRINN
Journal of Computing Research and Innovation
e-ISSN : 2600-8793

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

Journal of Computing Research and Innovation 11(2) 2026

Journal of Computing Research and Innovation

# Mobile Food Image Classification with Allergen Information using EfficientNetB0

Intan Nur Syahamah Tajul Ariffin[1], Hana Fakhira Almarzuki[2*], Khyrina Airin Fariza Abu Samah[3], Tajul Rosli Razak[4], Nur Aina Khadijah Adnan[5], Hafizatul Hanin Hamzah[6]

[1,2,4,6]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia.*
[3]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM) Melaka Branch, Jasin Campus, 77300 Merlimau, Melaka, Malaysia.*
[5]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM) Johor Branch, Segamat Campus, 85000 Segamat, Johor, Malaysia.*

ARTICLE INFO

*Article history:*
Received 6 March 2026
Revised 23 July 2026
Accepted 11 August 2026
Online first
Published 1 September 2026

*Keywords:*
Deep Learning
EfficientNetB0
Mobile Application
Food Image
Allergen Info
Image Recognition

*DOI:*
10.24191/jcrinn.v11i2.612

ABSTRACT

Food allergens represent a significant public health concern because they can trigger severe and potentially life-threatening reactions in sensitive individuals. Seven allergens focus on this study, which consist of tree nuts, shellfish, milk, peanuts, soybeans, eggs, and wheat. However, restaurant menus in Malaysia and many other regions often lack explicit warnings about allergenic ingredients. This study addresses this critical gap by developing a comprehensive mobile application for real-time food image classification that integrates EfficientNetB0 with personalized allergen information and safety alerts. The model was trained on 7,556 food images across eight food categories based on Kaggle/recipe datasets and achieved 96% test accuracy. The native mobile app integrates camera capture and gallery upload, provides instant classification, and confidence-scored predictions using a 70% threshold, and displays user-specific allergen symptom warnings through food-allergen mapping based on clinical sources and classification history. Experimental evaluation demonstrated promising classification performance and functional reliability of the mobile application. This mobile-deployed solution offers a practical approach for supporting safer food choices and improving public health awareness, while aligning with UN Sustainable Development Goal 3.

## 1. INTRODUCTION

Food allergens represent a significant public health concern because they can trigger severe and potentially life-threatening reactions in sensitive individuals. A food allergen is a substance in food that causes the body to react negatively to it (Masilamani et al., 2012). Meanwhile, a food allergy refers to the immune

[2*] Corresponding author. *E-mail:* hanafakhira@uitm.edu.my
https://doi.org/10.24191/jcrinn.v11i2.612

response to proteins found in food, which can be classified as immunoglobulin E (IgE) mediated (Iweala et al., 2018). As reported by Anvari et al. (2019), these reactions may range from mild discomfort to severe symptoms involving the neurological, cardiovascular, gastrointestinal, or respiratory systems, and in severe cases may lead to anaphylaxis requiring immediate attention.

Based on Kumar et al. (2012), peanuts, milk, tree nuts, eggs, shellfish, wheat, soybeans and fish are the common food allergens. Exposure to these allergens can trigger multiple symptoms simultaneously in sensitive individuals, including difficulty breathing, swelling of the tongue, mouth or face, skin reactions (an itchy rash and hives), vomiting, diarrhea and stomach pain. Based on these risks, food allergies can be life threatening, such as anaphylaxis requiring immediate medical attention , including a sudden rash, sudden swelling of mouth, lips, throat or tongue, shortness of breath, difficulty swallowing, fainting and unconsciousness (West et al., 2024).

Nowadays, many countries, including Malaysia, the United States and the European Union, require allergen-related information on product labelling to include the keyword “Contains” on the packaging to warn individuals about allergies (AdminBarakah, 2023). However, this support is often unavailable in restaurant settings or for freshly prepared food. Tan et al. (2023) claimed that restaurants in Malaysia generally do not identify or provide allergen information, which means customers frequently need to ask restaurant staff before placing an order. This creates a practical safety gap, especially for individuals who must make quick food choices in environments where ingredient and allergen information is not explicitly disclosed.

Although previous studies have explored food recognition, nutritional analysis, and allergy-related support systems, many existing solutions focus on only one main function, such as food classification, barcode-based allergen lookup, or nutritional recommendation, rather than integrating these capabilities into a practical mobile application. Some prior systems are desktop-based, while others do not provide user-specific allergen profiling or immediate allergy-related guidance during food selection. As a result, there remains a practical gap in mobile solutions that can classify food images in real time and provide personalized allergen information to support safer dining decisions.

To address this gap, this study proposes a mobile food image classification application that uses EfficientNetB0 and integrates personalized allergen information and safety alerts. The proposed system classifies food categories from captured or uploaded images and then presents associated allergen information through food-allergen mapping, rather than directly detecting allergens from the image itself. This approach enables users to receive relevant guidance more conveniently in restaurants and other unlabeled food environments, where allergen information is often not explicitly available.

The contributions of this study are threefold. First, the design and implementation of an EfficientNetB0-based image recognition–based food classification application that can accurately predict food names from captured or uploaded images (96% test accuracy); (ii) the integration of an allergen display module that links each classified food item with its common food allergens and relevant clinical information, including typical symptoms of allergic reactions matched to user profiles; and (iii) the provision of an educational interface that helps users better understand and avoid high-risk foods in their daily lives, enabling anytime/anywhere safety checks.

Beyond its technical contribution, the proposed system also offers practical public health value. By improving access to allergen-related information in unlabeled dining contexts, the application may help reduce accidental allergen exposure and support self-management among individuals with food allergies. Its mobile implementation also improves accessibility by allowing users to access guidance anytime and anywhere, which is particularly useful in real-world restaurant environments. In this way, the study supports broader health and well-being goals and aligns with Sustainable Development Goal (SDG) 3 by developing accessible food safety technology.

https://doi.org/10.24191/jcrinn.v11i2.612

## 2. LITERATURE REVIEW

This section reviews previous studies on image processing in food recognition, related works on food classification and allergen-support systems, and the suitability of EfficientNetB0 for the proposed application.

### 2.1 Image processing in food recognition

Food recognition has recently gained more attention in image processing and computer vision. This is because of its relevance to human health, dietary monitoring, and food-related decision support. Current food recognition systems commonly use convolutional neural networks (CNNs) to learn deep visual features from food images and support automatic identification through mobile or camera-based systems (Jiang et al., 2020). Despite these advances, food image classification remains challenging because food images often exhibit substantial variation in texture, appearance, colour, portion size, background, and lighting conditions. These factors can reduce classification consistency, especially in real-world environments where image quality is less controlled.

Recent progress in machine learning has improved the ability of image classification and object recognition models to identify food items more accurately. However, classification accuracy alone is insufficient for allergy-related decision support, as users also require contextual information such as allergen content, symptom relevance, and personalized warnings. This indicates that a practical food allergy support system should combine reliable food classification with user-centred allergen information delivery through image classification and object recognition approaches (Tran et al., 2021).

### 2.2 Similar work

Table 1 summarises and compares previous studies related to food image recognition and allergen-support applications. The reviewed works apply a range of deep learning techniques, including Google Teachable Machine, ResNet50, MobileNetV2, EfficientNetB0, ResNet-50V2, YOLOv8, Vision Transformer (ViT), InceptionV3, and VGG16. Although these studies demonstrate that deep learning can achieve strong performance in food classification, their objectives and system capabilities vary considerably.

For example, Allertify combines barcode scanning and image recognition to identify possible allergens in food products (Leba et al., 2024). However, its image recognition performance is relatively limited, which affects its reliability as a comprehensive food-allergy support tool. Khor et al. (2022) developed a Malaysian food allergen detection system using transfer learning and feature extraction techniques, but it was implemented on a desktop platform rather than a mobile environment. This reduces its practicality for real-time use in restaurants or other dining settings.

Chaitanya et al. (2022) proposed a CNN-based food classification system using InceptionV3 and Food-101-related data. Although the model achieved strong accuracy, the study focused primarily on nutritional analysis rather than allergy-specific decision support. Lam & Perera (2024) introduced a mobile application that uses YOLOv8-based hierarchical classification for Korean food recognition, which improved recognition under certain dataset conditions. Nevertheless, the system did not provide personalized allergen warnings or symptom information. Vishal et al. (2024) compared YOLOv8, Vision Transformer, and EfficientNet for Indian food classification and allergen prediction, and reported that EfficientNet achieved the best performance among the tested models. Similarly, Yudeepanao et al. (2024) compared MobileNetV2, EfficientNetB0, and ResNet-50V2 for Thai food classification and found that EfficientNetB0 produced the highest accuracy.

Overall, the literature shows that many deep learning models are effective for food recognition. However, most existing systems stop at classification and do not integrate personalized allergen profiling,

https://doi.org/10.24191/jcrinn.v11i2.612

symptom information, and practical mobile deployment into a single application. This indicates a clear gap in the current research. Therefore, the present study adopts EfficientNetB0 because it offers a favourable balance between classification performance and computational efficiency, making it suitable for mobile implementation while supporting real-time allergen-related guidance.

Table 1. Similar work

| References | Summary | Technique | Dataset | Result (%): Accuracy/F1 |
|---|---|---|---|---|
| Leba et al. (2024) | Allertify is an Android-based mobile application for identifying potential allergens in food products. It uses barcode scanning and image recognition to detect allergens in packaged food products and real food. The user interface was developed using Thunkable, while the back-end model was built with Google Teachable Machine. | Front-end was developed using Thunkable for the user interface, while Node.js was used for data processing. MySQL and Thunkable Data Source were used to store the data. For image recognition, Allertify used Google Teachable Machine, which is based on CNN and imports the TensorFlow.js library. | Gathering and uploading the dataset of food images for machine learning.<br>The dataset is split into two with 85% of the dataset for training and 15% for testing. | Barcode scanning produces a 100% success rate with an average processing time of 4 seconds.<br>Image recognition gets 50% accuracy with success rate 91% and 17 seconds of average processing. |
| Khor et al. (2022) | This work is designed to help anyone with allergies to prevent the food allergen on Malaysian food. The system targets three major food allergens such as peanut, cow' milk and shellfish. The system will predict the name of the food label and will categorize it as "Safe", "Potentially Unsafe" and "Unsafe". | Study applied fine-tuning and feature extraction for image transfer. The test data were evaluated using ResNet50, InceptionV3, and VGG16, and VGG16 was identified as the most suitable CNN architecture. The graphical user interface was developed using PyQt. | Overall, there are 5284 images of Malaysian food dataset representing 36 different food types.<br>Ingredient are obtained from the web and stored in a CSV file.<br>Dataset is split into train, validation and test. | Reaching an accuracy (VGG16) of 80.56% in classifying 36 varieties of Malaysian food. |
| Chaitanya et al. (2022) | This research proposed CNN model to identify and classify the food images. This application aims to identify various food images that also provide nutritional value to offer a beneficial healthy lifestyle. | Study used a pre-trained InceptionV3 model for transfer learning, Python Scrapy and Scrapy Spider for data collection, and Python 3.7.13 in Google Colab to develop and test the CNN model. | The image dataset is from Food-101, which provides 750 training images and 250 images for testing.<br>Used web scraper to obtain the information of restaurants, food's origin, nutritional details and recipe. | Accuracy for 20 classes: 97%<br>Accuracy for 25 classes: 96.52% |
| Lam and Perera (2024) | This research is aimed at people who like to eat Korean food but have a calorie restriction because of the limitations on health issues, personal preferences and | Offers a YOLOv8-based method (flat classification) to detect the food item.<br>To address the issues of the imbalance dataset and limitation on clear visualization | Image datasets of Korean food are taken from Roboflow.<br>The food is grouped into 4 categories including main dish, rice, soup and side dishes. | Average accuracy hierarchical classification: 88.50%, higher than flat classification, 85.32%. The flat classification is low because the data was not well |

https://doi.org/10.24191/jcrinn.v11i2.612

| | | | | |
|---|---|---|---|---|
| | religion. The mobile application provides images to capture the food items on the actual food. | between different food items, the research has proposed the Hierarchical Classification. | Dataset is split using an 8:1:1 ratio into training, validation and testing. | organized, causing difficult image detection. |
| Vishal et al. (2024) | The study compared YOLOv8, ViT, and EfficientNet for food classification and allergen prediction, and EfficientNet achieved the highest accuracy at 92.28%. | Compare three models such as YOLOv8, Vision Transformer(ViT) and EfficientNet to detect the food allergen. | Dataset contains 6,271 Indian food images across 20 classes from Kaggle, split into 80% training, 10% validation, and 10% testing. Food names and allergen details were also sourced from Kaggle. | YOLOv8: 92.05%<br>Vision Transformer (ViT): 52.04%<br>EfficientNet: 92.28% |
| Yudeepanao et al. (2024) | The study compared MobileNetV2, EfficientNetB0, and ResNet-50V2 for food detection, with the aim of improving food image identification for allergy sufferers and foreign tourists in Thailand. | MobileNetV2, EfficientNetB0 and Resnet-50V2 are the techniques that are used for this paper | FoodyDudy dataset is taken from Kaggle that contains 11,530 Thai food images with 48 distinct classes or menus.<br>The dataset was split into 80% for training and 20% for testing. | Accuracy<br>MobileNetV2: 75%<br>EfficientNetB0: 97%<br>ResNet-50V2: 93% |

### 2.3 Justification selection of EfficientNetB0

EfficientNet is a family of convolutional neural networks designed to achieve high predictive performance while using computational resources efficiently. A key contribution of the EfficientNet architecture is its compound scaling strategy, which scales network depth, width, and input resolution in a balanced manner instead of increasing only one dimension of the model. EfficientNetB0 is the baseline variant of this family and serves as the foundation for the larger EfficientNet models (GeeksforGeeks, 2024).

EfficientNetB0 was selected in this study because it offers a strong balance between classification accuracy and computational efficiency, which is important for mobile deployment. Compared with heavier deep learning architectures, EfficientNetB0 is better suited to applications requiring real-time inference, lower memory usage, and practical implementation on resource-constrained devices. This makes it particularly appropriate for a mobile food image classification application, where responsiveness and lightweight execution are as important as predictive accuracy.

Another advantage of EfficientNetB0 is that it is commonly used with ImageNet pre-trained weights, enabling transfer learning for domain-specific image classification tasks (Yudeepanao et al., 2024). This allows the model to learn generalized visual features from large-scale image data before being fine-tuned for a more specific task, thereby improving learning efficiency when the target dataset is relatively limited. In food image classification, recent studies have shown that EfficientNetB0 performs competitively while maintaining a lightweight architecture suitable for deployment-focused applications.

Therefore, EfficientNetB0 is a suitable model for this study because it supports accurate food image classification. It remains efficient enough for integration into a mobile application. Its balance of performance, compact design, and transfer learning capability aligns well with the objective of developing a practical and user-friendly system for safer food selection.

https://doi.org/10.24191/jcrinn.v11i2.612

## 3. METHOLODOGY

This section describes the methodology used to develop and evaluate the proposed mobile food image classification system with allergen information. The process consists of dataset construction, image pre-processing, model training, comparative evaluation, mobile application integration, and system testing. The methodology was designed to assess both the model's classification performance and the practical functionality of the deployed mobile application.

### 3.1 Dataset construction

The dataset used in this study was constructed from multiple online sources, including Kaggle and recipe-based websites, to obtain food images relevant to the selected food categories. The collected images were curated to support the development of a food image classification model for allergen-related guidance in a mobile application. A total of 7,556 food images were collected and organized into eight food categories representing the targeted food classes in the study.

To improve dataset quality, we manually reviewed the collected images to ensure that each image matched its assigned food category and was visually suitable for classification. We excluded images with poor quality, unclear food objects, or irrelevant content from the dataset. In addition, efforts were made to reduce duplication and overlap among images collected from different sources to minimize the risk of data leakage during model training and testing. After dataset collection and cleaning, the images were grouped by food class and prepared for subsequent pre-processing and model development. Table 2 presents the class-wise distribution of the dataset, including the number of images and data sources for each food category.

Table 2. Overview of collected food category dataset and preprocessing details

| Food Class | Main Allergen | Total Images | Kaggle | Recipe Websites | Notes |
|---|---|---|---|---|---|
| Class 0 (baklava) | Tree Nuts / Wheat / Milk | 982 | 1000 | https://www.delish.com/cooking/recipe-ideas/a36943000/baklava-recipe/ | Cleaned (incorrect images removed) |
| Class 1 (laksa) | Shellfish | 959 | 1000 | https://rasamalaysia.com/assam-laksa-nyonya-noodles-with-fish-broth/ | Cleaned (incorrect and low quality images removed) |
| Class 2 (nasi_lemak) | Peanut / Egg / Shellfish | 918 | 1000 | https://rasamalaysia.com/nasi-lemak-recipe/ | Cleaned (unreadable, irrelevant, incorrect, duplicate and low quality images removed), Selected images cropped |
| Class 3 (pad_thai) | Soy / Shellfish / Peanut | 988 | 1243 | https://rasamalaysia.com/pad-thai-recipe/ | Cleaned (incorrect and low quality images removed) |
| Class 4 (roti_canai) | Wheat / Egg / Milk | 913 | 1000 | https://rasamalaysia.com/roti-canai-roti-paratha-recipe/ | Cleaned, Selected images cropped |
| Class 5 (samosa) | Milk / Wheat / Egg | 920 | 1244 | https://rasamalaysia.com/easy-samosa/ | Cleaned (incorrect and low quality images removed) |
| Class 6 (satay) | Peanut / Soy | 974 | 1000 | https://rasamalaysia.com/recipe-chicken-satay/ https://rasamalaysia.com/peanut-sauce/ | Cleaned (incorrect and low qquality images removed) |

https://doi.org/10.24191/jcrinn.v11i2.612

| | | | | | |
|---|---|---|---|---|---|
| Class 7 (tiramisu) | Egg, Milk, Wheat | 902 | 1000 | https://www.delish.com/cooking/recipe-ideas/a45487852/tiramisu-recipe/ | Cleaned (incorrect, duplicate and low quality images removed) |
| **Total** | | 7556 | 8487 | | |

## 3.2 Data pre-processing

After dataset construction, the food images were pre-processed before model training and evaluation. The pre-processing stage consisted of three main steps: data cleaning, data splitting, and data augmentation.

(i) Data cleaning

Data cleaning was performed as the initial step to ensure the dataset used in this project was accurate, consistent, and suitable for training. We manually inspected the images to remove unreadable files, incorrect samples, low-quality images, and duplicate or irrelevant content. In addition, certain images were cropped to emphasize the food item of interest and to align the visual content with the corresponding class label. For instance, as in Fig. 1, incorrect images found in the nasi lemak folder were removed. Some images in that category were cropped (from Fig. 2) to highlight only the nasi lemak portion. This process improved dataset quality by reducing noise and ensuring that the final set of images was more representative of the selected food classes (Fig. 3).


Fig 1. Incorrect food in Nasi Lemak folder

Fig 2. Full image before cropping

Fig 3. Image after cropping

(ii) Data splitting

Following data cleaning, the dataset was partitioned into training, validation, and testing subsets using an 80:10:10 ratio. Specifically, 80% of the images were used for training, 10% for validation, and 10% for testing, as shown in Fig. 4. This division allowed the model to learn from a large portion of the dataset while keeping separate data for tuning and final performance evaluation. To maintain a fair experimental setup, each image was assigned to only one subset, reducing the risk of overlap and data leakage. In addition, the figure illustrates the process of transferring 7,556 food images from the FoodClassification folder in Google Drive to the FoodClassification folder in Google Colab. Fig. 5 shows the dataset directory after splitting. The final dataset contained 7,556 images, consisting of 6,042 training images, 752 validation images, and 762 testing images.

https://doi.org/10.24191/jcrinn.v11i2.612

```
import splitfolders

split_folder = splitfolders.ratio(
    input = '/content/drive/MyDrive/FoodClassification',
    output = './FoodClassification',
    ratio = (0.8, 0.1, 0.1)
)

Copying files: 7556 files [03:41, 34.05 files/s]
```

Fig 4. Code snippet for split folder

```
train_data_dir = './FoodClassification/train'
val_data_dir = './FoodClassification/val'
test_data_dir = './FoodClassification/test'
```

Fig 5. Train, validation, and test directory path

(iii) Data augmentation

As shown in Fig. 6, data augmentation was performed on the training set using ImageDataGenerator to increase dataset diversity and improve model generalization. Because food images may appear in different orientations, lighting conditions, and visual contexts, augmentation was used to expose the model to a wider range of training examples. The augmentation techniques applied in this project included rotation, width shift, height shift, channel shift, shearing, zooming, horizontal flipping, brightness adjustment, and fill mode. These techniques were applied only to the training data, while the validation and testing sets remained unmodified and were only pre-processed using the preprocess_input function to ensure unbiased evaluation. This approach helped reduce overfitting and improved the model's ability to recognize food images more robustly.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.efficientnet import preprocess_input

train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    channel_shift_range=30.0,
    shear_range = 0.2,
    zoom_range =0.3,
    horizontal_flip=True,
    vertical_flip=False,
    brightness_range=[0.8, 1.2],
    fill_mode='nearest'
)

valid_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input
)

test_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input
)
```

Fig 6. Code of data augmentation

https://doi.org/10.24191/jcrinn.v11i2.612

### 3.3 EfficientNetB0 implementation

EfficientNetB0th is used as the base model in this project, as shown in Fig. 7. The model was pretrained on the ImageNet dataset, which means it had already learned useful general visual features such as edges, shapes, and textures from a large number of images. This pretrained knowledge is beneficial because it allows the model to start from a strong foundation instead of learning everything from scratch. In this project, the include_top=False setting was applied to remove the original classification layer of EfficientNetB0 so that the model could be customized for the eight food classes in the dataset. The input size was set to 224 x 224 x 3 pixels to match the model's required input format. To adapt the model to the specific food image dataset, the last 20 layers of EfficientNetB0 were set as trainable, while the remaining layers were frozen. This approach allows the earlier layers to retain the general features learned from ImageNet, while the later layers are fine-tuned to learn food-specific patterns. As a result, the model is able to combine pretrained knowledge with task-specific learning, which improves classification performance and training efficiency.

```
from tensorflow.keras.applications.efficientnet import EfficientNetB0

model_efficientNet = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(224,224,3))

for layer in model_efficientNet.layers[-20:]:
  layer.trainable = True

Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_notop.h5
16705208/16705208 ━━━━━━━━━━━━━━━━━━━━ 2s 0us/step
```

Fig 7. Code for EfficientNetB0 model

Fig. 8. shows the custom classification head built on top of EfficientNetB0. This head consists of a global average pooling layer, followed by a dense layer with 256 units and ReLU activation, batch normalization, dropout with a rate of 0.5, and a final dense layer with eight output units and softmax activation for classification. The global average pooling layer reduces the spatial dimensions of the extracted feature maps, while the dense layer helps the model learn higher-level patterns from the food images. Batch normalization and dropout were included as regularization techniques to improve generalization and reduce overfitting. By combining the EfficientNetB0 feature extractor with this custom classification head, the model can learn discriminative features more effectively and classify the food categories more accurately.

```
from tensorflow.keras.layers import GlobalAveragePooling2D, BatchNormalization, Dense, Dropout
from tensorflow.keras.models import Model
from keras import regularizers
x = GlobalAveragePooling2D()(model_efficientNet.output)
x = Dense(256, activation='relu', kernel_regularizer=regularizers.l2(0.001))(x)
x = BatchNormalization()(x)
x = Dropout(0.5)(x)
predictions = Dense(8, activation='softmax')(x)

model = Model(inputs=model_efficientNet.input, outputs=predictions)
```

Fig 8. Snippet code for custom classification head

Finally, the optimizer and loss function for model training were configured during the model compilation stage, as shown in Fig. 9. The Adam optimizer is used with a learning rate of 1e-5 to enable stable fine-tuning of the pretrained EfficientNetB0 model. A small learning rate has been used to improve training stability, especially for food image classification. Additionally, categorical cross-entropy is applied as the loss function to optimize the result of multi-class image classification.

https://doi.org/10.24191/jcrinn.v11i2.612

```
from tensorflow.keras.optimizers import Adam

model.compile(
    optimizer=Adam(learning_rate=1e-5),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Fig 9. Snippet Code for model compilation

### 3.4 Performance evaluation

The performance of the proposed model was evaluated using the testing dataset. Several evaluation metrics were used to assess the classification results, including accuracy, precision, recall, and F1-score. Accuracy measures the overall proportion of correctly classified images, while precision indicates the proportion of correct predictions among the predicted classes. Recall measures the ability of the model to identify all relevant images in each class, and the F1-score provides a balanced measure between precision and recall. In addition, a confusion matrix was used to visualize the classification performance for each food allergen class and to identify correct and incorrect predictions. These evaluation metrics were selected to provide a more comprehensive assessment of the model's performance. The confusion matrix is shown in Fig. 10.

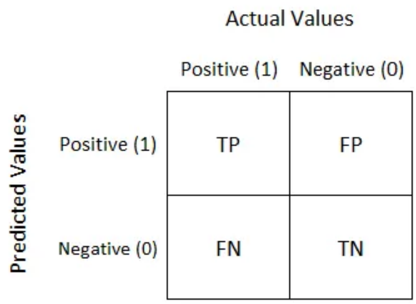


Fig 10. Confusion matrix

TP stands for "True positive," which means the number of cases that were predicted to be true and were correct. TN for "True negative," which means the number of cases that were predicted to be false and were correct. FP for "False positive," which means the number of cases that were predicted to be true but were wrong, and FN for "False negative," which means the number of cases that were predicted to be false but were wrong. Using the values from the confusion matrix, it was easy to figure out the performance rating metrics using the formula in Eq. (1) - (4). The way to measure accuracy is in Eq. (1), and the way to measure precision is in Eq. (2). Eq. (3) shows the recall, and Eq. (4) shows the F1 score test.

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

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

https://doi.org/10.24191/jcrinn.v11i2.612

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

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

### 3.5 System architecture

The system architecture diagram in Fig. 11 shows the interactions among the main components of the application. The user interface, local database, and image classification model are included in the system architecture diagram. This diagram shows how data flows through the application and how each component contributes to its functionality.

The interaction between the user and the application. When the user inputs the image of food, the application predicts the food name and displays the result. Furthermore, the application has three layers: presentation layer, application layer, and data layer. The presentation layer displays the mobile application interface between the user and the backend logic. It was developed using the XML layouts in Android Studio. The food dataset was processed and split into training, validation and testing sets using Google Colab. The EfficientNetB0 architecture was used to develop and train the model for food image classification. Then, the trained model was converted to a TensorFlow Lite (.tflite) file and integrated into the mobile application.

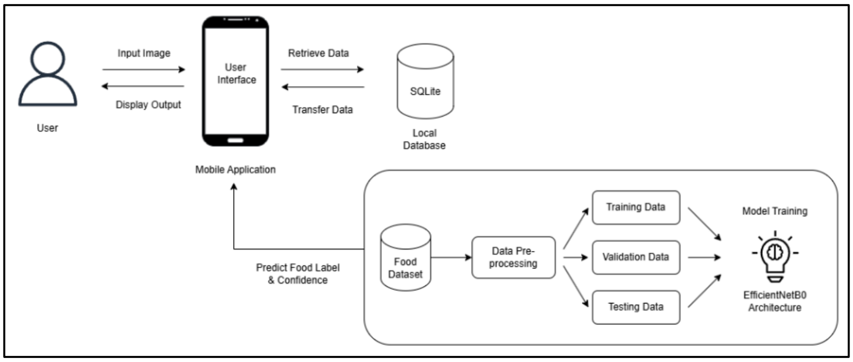


Fig 11. System architecture diagram

In the application layer, the model predicts the category and confidence score of the food in real time based on user input. The application retrieves and displays additional information about the food based on the predicted label in the CSV file. Additionally, the application layer implements some application functionality in Java, including create, read, update, and delete (CRUD) operations and search functions for related information. The database used for this project is SQLite and will manage data retrieval and transfer between the applications. It also stores user information, such as the user's account, historical classifications, and allergen profile.

## 4. RESULT AND DISCUSSION

This section covers the evaluation of the EfficientNetB0 model, including a confusion matrix and graphical analysis of training and validation accuracy over epochs. The mobile application's user interfaces are presented in the last part of the section.

https://doi.org/10.24191/jcrinn.v11i2.612

Table 3 shows the inference time and total latency results for the mobile application. This application has been tested on a Samsung Galaxy A52 with a Qualcomm Snapdragon 720G processor and 8GB of RAM. The deep learning model compressed in the application assets folder is 16.5MB. The performance evaluation was conducted by running multiple inference runs using the model.process() method, measuring the inference time from input image processing to the generation of the classification output. Based on 10 inference runs on different images, the model achieved an average inference time of approximately 99ms. In contrast, the average end-to-end device latency, including image pre-processing and result display, is 130ms. This result shows that the architecture can classify food in real time without noticeable delay.

Table 3. Results of inference time and total latency

| Image | Inference Time (ms) | Total Latency (ms) |
|---|---|---|
| 1 | 93.705937 | 247.270781 |
| 2 | 100.738646 | 145.860365 |
| 3 | 91.228386 | 103.505156 |
| 4 | 98.460260 | 114.884427 |
| 5 | 99.714218 | 114.539635 |
| 6 | 97.416198 | 112.542708 |
| 7 | 101.552396 | 115.624583 |
| 8 | 100.101511 | 114.291771 |
| 9 | 114.502917 | 128.759843 |
| 10 | 94.248802 | 106.599792 |
| **Average** | 99.1669271 | 130.3879061 |

## 4.1 Evaluation of EfficientNetB0

The accuracy testing was performed in Google Colab. It was conducted using a testing dataset (10%) in eight classes (762 images) during model evaluation. Fig. 13 shows the classification report for EfficientNetB0, displaying precision, recall, F1-score, and support for each class. Among all classes, the pad thai class achieved the highest precision of 0.99, recall of 0.98, and F1-score of 0.98 on 100 test images. In particular, as shown in Fig. 12, the pad thai class was evaluated on a larger test set of 100 images, which represents a more robust sample size than the other classes. Other than that, the unique visual elements of pad thai, such as the flat texture of the rice noodles, the crushed peanuts, and the characteristic warm, brown color, facilitated highly accurate classification. This is because these consistent visual features allowed the model to easily differentiate pad thai from other classes, thereby reducing inter-class confusion. This shows that the pad thai classes have recorded the highest and best performance among other classes. In addition, other classes such as baklava, laksa, nasi lemak, roti canai, samosa, satay, and tiramisu show precision, recall, and F1 scores between 0.91 and 0.98, indicating balanced and consistent model performance.

The model achieved an overall accuracy of 0.96 (96%) on the 762 testing images that had not been seen before. This shows that the model is unbiased and performs well in image classification. Most of the classes achieved high precision, recall, and F1 scores, indicating that the model is strong, highly accurate, and reliable in classification performance. These results indicate that the model is suitable for integration into the mobile application to capture and upload food images.

https://doi.org/10.24191/jcrinn.v11i2.612

Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| baklava | 0.94 | 0.91 | 0.92 | 99 |
| laksa | 0.94 | 0.97 | 0.95 | 97 |
| nasi_lemak | 0.96 | 0.99 | 0.97 | 93 |
| pad_thai | 0.99 | 0.98 | 0.98 | 100 |
| roti_canai | 0.96 | 0.93 | 0.95 | 92 |
| samosa | 0.98 | 0.96 | 0.97 | 92 |
| satay | 0.96 | 0.98 | 0.97 | 98 |
| tiramisu | 0.97 | 0.97 | 0.97 | 91 |
| accuracy | | | 0.96 | 762 |
| macro avg | 0.96 | 0.96 | 0.96 | 762 |
| weighted avg | 0.96 | 0.96 | 0.96 | 762 |

Fig 12. Classification report

To evaluate the class-level performance of the food classification model, a confusion matrix was generated across eight food categories, as shown in Fig. 13. The diagonal elements represent the true positive counts and indicate high classification accuracy for all classes. Specifically, the model achieved the highest number of correctly classified samples on pad thai (Class 3) with 98 correct classifications, followed by satay (Class 6) with 96 correct classifications, and then laksa (Class 1) with 94 correct classifications. The lowest individual class performance was recorded for roti canai (Class 4), with 86 correct instances and two misclassifications in each of laksa and satay. Moreover, a slight confusion was observed between the samosa (Class 5) and baklava (Class 0) classes, with three samosa samples predicted as baklava and two baklava samples predicted as samosa. Overall, the model minimized confusion between classes and achieved higher precision across food categories.

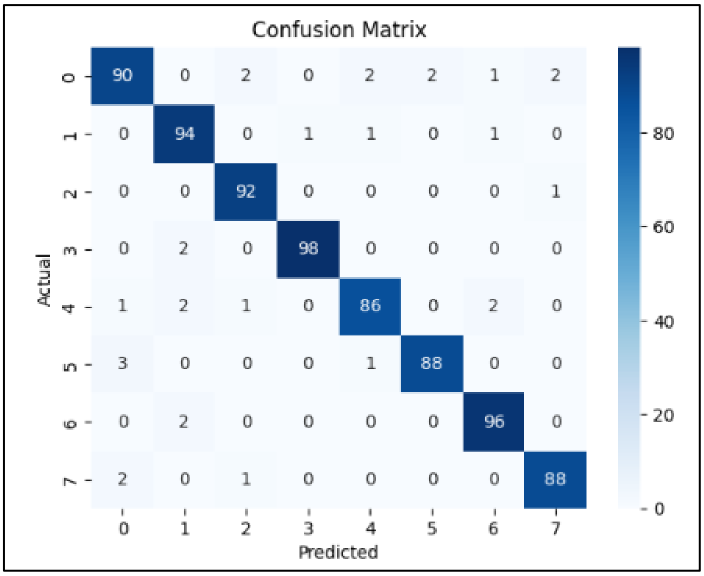


Fig 13. Confusion matrix

https://doi.org/10.24191/jcrinn.v11i2.612

### 4.2 Analysis of results and graphical representation

Fig. 14 shows the training and validation of the accuracy graph. The accuracy graph represents the accuracy value of the model on the y-axis and the number of training iterations (epochs) on the x-axis. The blue line shows the training accuracy, while the orange line shows the validation accuracy. The training accuracy shows a steady and smooth increase over 50 epochs. Similarly, validation accuracy increases rapidly in the early epochs and closely tracks the training accuracy, with both stabilising at approximately 96%. This shows that the model is not only memorising the training data but also learning the data patterns. The close convergence indicates excellent generalization to unseen data without overfitting.

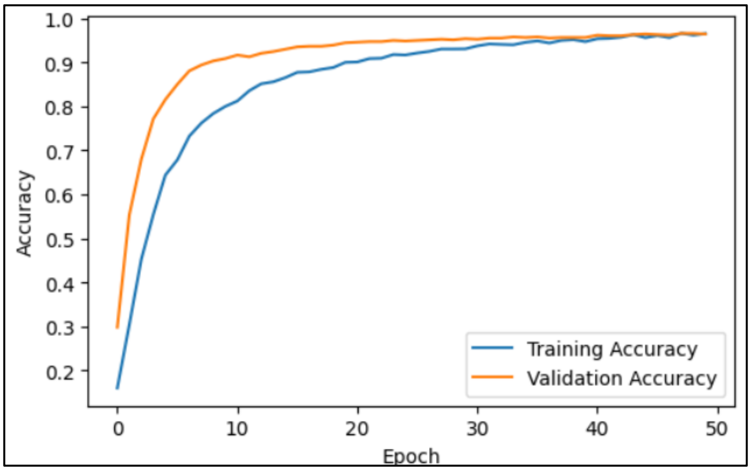


Fig 14. Graph of training and validation accuracy

The graph of training and validation loss is shown in Fig. 15. The y-axis shows the loss value, and the x-axis shows the epoch number, i.e., the number of iterations. Both losses decreased in the early epoch, indicating that the model is effectively learning. The accuracy graph increases rapidly, while the loss values decrease gradually, converging at epoch 48. This pattern indicates proper convergence and no signs of overfitting.

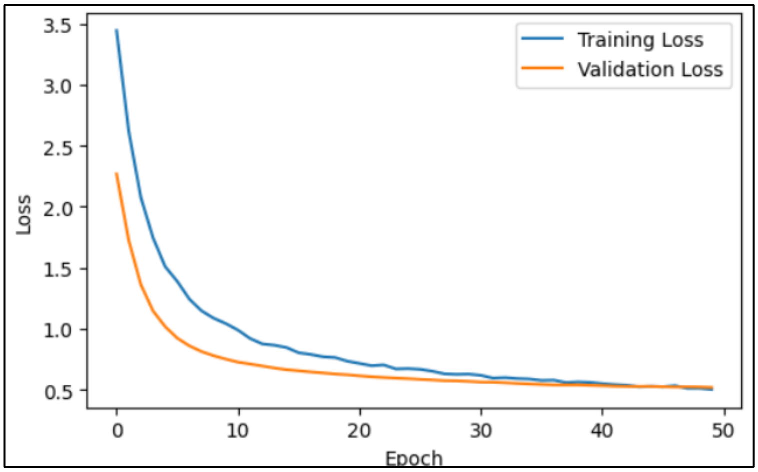


Fig 15. Graph of training and validation loss

https://doi.org/10.24191/jcrinn.v11i2.612

### 4.3 Functionality of the mobile application

The functional testing results for the mobile application are shown in Table 4. These test results include a description of the test case, the expected output, and the status of the result. Each test case will be tested to determine whether the application functions as intended. This functionality was tested by an external user from the registered page to the log out page in the mobile application. The mobile application's functionality was tested through eight test cases. All test cases were passed and produced the expected output. The test includes key features such as data input validation, user authentication, navigation, user interface elements, and the core functionality for capturing and uploading food images. Overall, these tests showed that the application meets its functional requirements and performs as intended.

Table 4. Functionality testing

| Test Case ID | Test Case Description | Expected Output | Status |
|---|---|---|---|
| TC001 | Verify successful registration with valid credentials. | The user account is created, and the application will go directly to the Login Page. | Pass |
| TC002 | Verify successful login with valid credentials. | User should be able login to the application home page. | Pass |
| TC003 | Ensure that the user can capture an image using camera or upload image from the device. | Capture: Capturing the image and displaying the image. | Pass |
| | | Upload: Displaying the food image after users upload it. | Pass |
| TC004 | Verify the results based on the input image, where the app predicts the food name with confidence and displays other relevant information from CSV file. | Users should view the results after capturing or uploading the input image, including the food name, ingredients, possible allergens in those ingredients, symptoms, allergens, and notes. | Pass |
| TC005 | Ensure that the user can manage history successfully.<br>i. Test the swipe-to-delete feature for deleting records.<br>ii. Test the search bar.<br>iii. Test the records view to ensure the selected records are displayed correctly. | i. Successfully delete the selected records.<br>ii. Display the correct result based on food name.<br>iii. Display the correct historical record based on the selected records. | Pass |
| TC006 | Ensure that the user can successfully edit their profile details, such as email address, profile picture, password, and allergens. | The email address, profile picture, password, and allergens should update successfully and display a success message. | Pass |
| TC007 | Verify successful logout. | The user should be able to log out of the application and be redirected to the login page. | Pass |
| TC008 | Verify that the AI system generates results based on the image of food. | AI system analyses the input image and displays the correct prediction results. | Pass |

### 4.4 User interface of the mobile application

This project has developed a mobile application that includes seven main user interfaces. These interfaces include a login page, a register page, an allergen page, a home page, a view results page, a manage history page, and a profile page. Each page has its own functionality that interacts with the user, including text, buttons, icons, images, and text fields.

The first main user interface displayed when the user opens the mobile application is the Login Page. Fig. 16 shows the login page of this project. On this page, the user must enter their username and password to verify their identity and access the home page. If the username or password does not match, the application displays an error message, as shown in Fig. 17. The user needs to click the 'LOG IN' button to access the application and the 'REGISTER' text link if they do not have an account.

https://doi.org/10.24191/jcrinn.v11i2.612

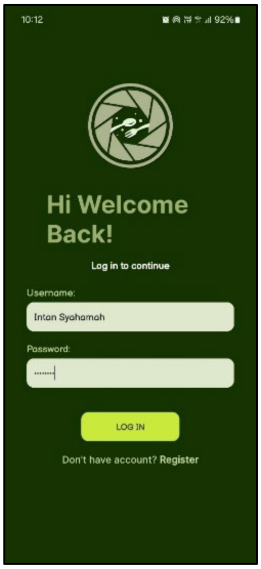


Fig 16. Login page

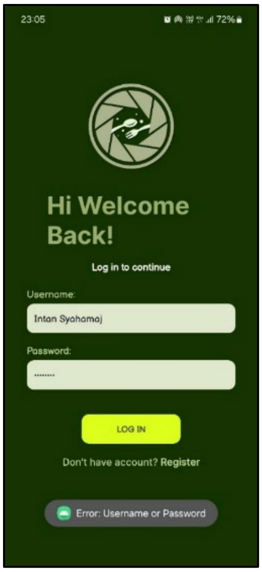


Fig 17. Toast message displayed in the Login Page for Incorrect Username or Password

Fig. 18. shows the Register Page, which stores new user data such as username, email address, and password. To create an account, the user needs to fill in all the text fields and click the "REGISTER" button to access the application. If the same username or email is already stored in the database, the application displays a message prompting the user to change it. Apart from that, the user needs to enter the same password in both the password and confirmation password text fields. If the user enters an incorrect password match in the password confirmation field, the application will prompt the user to re-enter the password. The message will display as shown in Fig. 19.

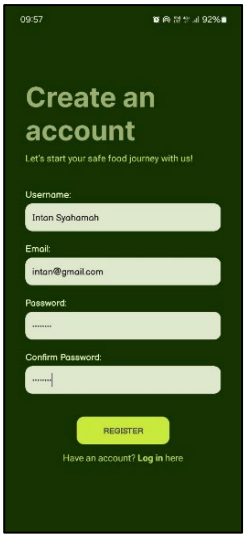


Fig 18. Register page

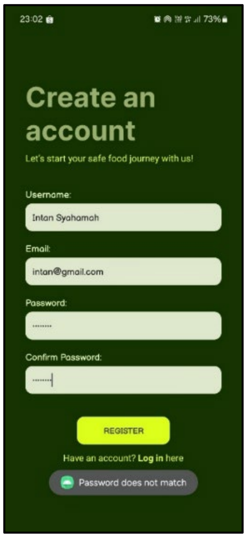


Fig 19. Toast message displayed in register for a password mismatch

https://doi.org/10.24191/jcrinn.v11i2.612

After logging in, new users are directed to the User Allergen Profile Page (Fig. 20). Here, interactive cards allow users to select their specific sensitivities to common allergens (e.g., peanut, tree nuts, shellfish, egg, milk, soy, and wheat) to match against future food predictions. Users can either “Save” to store these preferences in the database or “Skip” to proceed without saving; both actions redirect to the home page.

The Home Page Fig. 21 serves as the main interface, displaying the username and the user’s saved allergens (or a “No allergen” message if skipped). From here, users can initiate food classification by capturing a new photo via the camera or uploading an existing image. The top app bar features the page title, a profile icon, and a hamburger menu.

Accessed via the hamburger icon, the Navigation Drawer Fig. 22 features the app’s logo and name in its header. It provides straightforward routing to main destinations such as Home, Detection History, and Profile and includes a “Log Out” button at the bottom to terminate the session and return the user to the login screen.

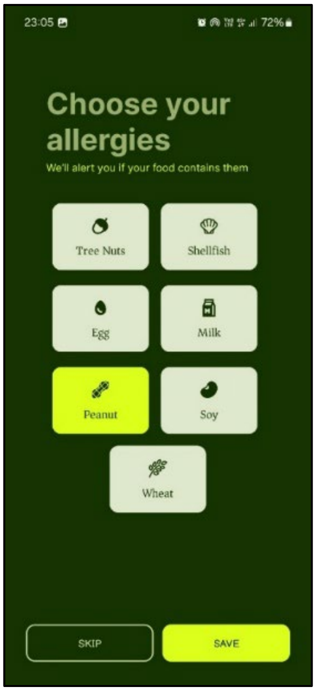


Fig 20. User allergen interface

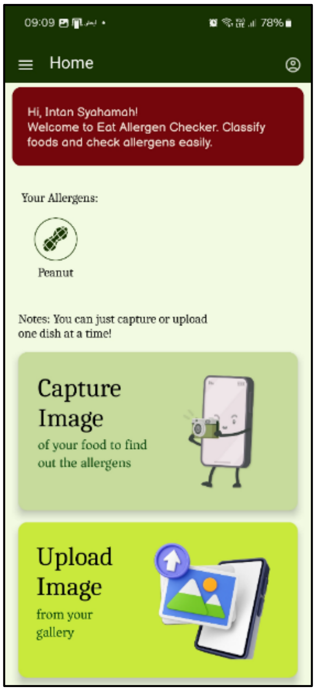


Fig 21. Home interface


Fig 22. Navigation drawer interface

The result page was displayed after the user had uploaded or captured the image. This page will predict food labels and their confidence values using the trained model. In addition, the page also displays other information, such as “*may contain*” allergens, additional information, and notes. This information was displayed based on the user’s predicted input image. Fig. 23 and Fig. 24 show the results in a scrollable view, including the predicted food label and other information based on the user’s input image. In Fig. 25, the results of the predicted unknown food were displayed, which contain messages such as “*Sorry, this image may not be in our database*!”. This message will display if the confidence value is below 70%.

https://doi.org/10.24191/jcrinn.v11i2.612


Fig 23. Result page showing scrolling up to view result


Fig 24. Result page showing scrolling down to view the result


Fig 25. Result page showing prediction on unknown food

## 5. CONCLUSION

The EfficientNetB0 Integrated Mobile Application for identifying food allergens has successfully achieved its objectives by employing an advanced deep learning approach to improve allergen identification in food products. The experiment demonstrated the effectiveness of EfficientNetB0 in reliably identifying food categories and providing associated allergen information. The identification of allergens corresponding to the user's profile emerged as the most critical factor in assessing the safety of food items for consumption, thereby mitigating the risk of allergic reactions. The feature offered a straightforward, efficient approach that enabled users to promptly identify food items based on their distinct allergen profiles.

The importance of this research lies in its alignment with Sustainable Development Goal 3 (Good Health and Well-being), established by the United Nations, which emphasizes the promotion of good health and well-being. It notifies users about food allergens that may cause various allergic reactions, which can be fatal. The provision of these applications facilitates the prediction of accuracy and real-time classification of food items. Consequently, it serves to prevent unintentional exposure and safeguard individuals with allergies.

The project had several identified limitations. The image classification accuracy depends on image quality and the limited number of food categories. The image quality, such as poor lighting conditions, blurred images, low resolution, and improper camera angles, causes misclassifications of food items if the image does not clearly represent their appearance. Apart from that, to classify the confidence value between the trained classes and unknown food, the threshold value was set to 0.7 to ensure high precision, but this can also lead to false negatives. For instance, if the image is correct but the image quality is low, the model's

https://doi.org/10.24191/jcrinn.v11i2.612

prediction confidence may fall below the 0.7 threshold. In this case, the application will display a message such as “Sorry, this image may not be in our database!” even if the food is technically trained in the model. Additionally, the application was trained in a limited number of food categories. Thus, the image classification models can accurately identify only a few food categories among the supported classes. When users capture or upload images outside the trained categories, or images that are similar to food categories, the application may produce incorrect classification results. These limitations affect the application’s performance in diverse real-life situations. In addition, the result display based on image classification relies on predefined information stored in CSV files, including ingredients, symptoms, and allergens associated with ingredients. As a consequence, the application can only provide information based on the existing data. This limitation requires manual updates to ensure the information for the food categories is accurate, functional, and up to date, especially when adding new food categories or modifying information. In summary, these limitations are useful in enhancing the reliability of the application and future development.

There are a few recommendations that could be made to the application that can be improved and enhanced in the future. One of the future recommendations is to increase the food dataset. The existing dataset used for training is limited to selected food classes, making it hard to generalize to other food categories. The image classification model can be trained to identify more food categories and reduce misclassification by increasing the number of food images and the variety of food categories. Improving the accuracy of the classification model can be achieved in the future through additional training data, advanced training techniques, and an optimised model architecture. In addition, future work may focus on extending the application from food classification to food detection, enabling the identification of multiple food items in a single image. Apart from that, the application can be enhanced by replacing CSV files with a centralized database, enabling more efficient and effective updates. This can save time spent manually updating the CSV file when adding new food categories or modifying information.

The primary contribution of this research project is the implementation of food allergen image recognition using EfficientNetB0, specifically designed for mobile platforms. This feature enables users to perform image classification in their everyday activities, free from constraints on time and location, achieving 96% accuracy on 762 test images across 8 allergen-related food categories.

## 6. ACKNOWLEDGEMENT/FUNDING

The authors would like to acknowledge the support of UiTM Kampus Shah Alam for the continuous support given to the research initiatives in the university.

## 7. CONFLICT OF INTEREST

The authors agree that this research was conducted in the absence of any self-benefits, commercial or financial conflicts and declare no conflict of interest associated with this publication.

## 8. AUTHORS’ CONTRIBUTION

**Intan Nur Syahamah Tajul Ariffin, Hana Fakhira Almarzuki** Development and analysis, writing original draft; **Khyrina Airin Fariza Abu Samah**: Supervision and editing draft; **Tajul Rosli Razak, Hafizatul Hanin Hamzah**: Review manuscript; **Nur Aina Khadijah Adnan**: Formatting and final review.

https://doi.org/10.24191/jcrinn.v11i2.612

## 9. REFERENCES


AdminBarakah. (2023, March 1). *Food allergen*. Al Barakah Training Solution. https://al-barakah.com.my/food-allergen/.

Anvari, S., Miller, J., Yeh, C. Y., & Davis, C. M. (2019). IgE-mediated food allergy. Clinical reviews in allergy and immunology, 57(2), 244–260. https://doi.org/10.1007/s12016-018-8710-3.

Chaitanya, A., Shetty, J., & Chiplunkar, P. (2022). Food image classification and data extraction using convolutional neural network and web crawlers. *Procedia Computer Science*, *218*, 143–152. https://doi.org/10.1016/j.procs.2022.12.410.

GeeksforGeeks. (2024, June 3). *EfficientNet architecture*. https://www.geeksforgeeks.org/computer-vision/efficientnet-architecture/.

Iweala, O. I., Choudhary, S. K., & Commins, S. P. (2018). Food allergy. In *Current Gastroenterology Reports* (Vol. 20, Number 5). Current Medicine Group LLC 1. https://doi.org/10.1007/s11894-018-0624-y.

Jiang, S., Min, W., Liu, L., & Luo, Z. (2020). Multi-scale multi-view deep feature aggregation for food recognition. *IEEE Transactions on Image Processing*, *29*, 265–276. https://doi.org/10.1109/TIP.2019.2929447.

Khor, P. L., Jee, W., & Raymond, K. (2022, December 28). Food allergen detection in Malaysian food using convolutional neural networks. *AIJR Proceedings*, 103-108. https://doi.org/10.21467/proceedings.141.

Kumar, S., Verma, A. K., Das, M., & Dwivedi, P. D. (2012). Molecular mechanisms of IgE mediated food allergy. *International Immunopharmacology*, *13*(4), 432–439. https://doi.org/10.1016/j.intimp.2012.05.018.

Lam, H. K., & Perera, K. M. P. (2024). Detecting Korean food using image using hierarchical model. http://arxiv.org/abs/2409.02448.

Leba, E., Faigao, J. A., Magante, M. V., Joseph, V., Tamba, Y., Nelson, I., & Bingcang, E. (2024). Allertify: Android-based mobile application for food allergens identification using barcode scanning and image recognition. In *11th International Scholars Conference* (Vol. 11, No. 5, pp. 1542-1557).

Masilamani, M., Commins, S., & Shreffler, W. (2012). Determinants of food allergy. *Immunology and Allergy Clinics of North America*, *32*(1), 11–33. https://doi.org/10.1016/j.iac.2011.12.003.

Tan, A. L., Trupp, A., & Tan, J. X. (2023). Dining out safely with food allergies: A comparative perspective from restaurant managers and servers in Malaysia. *International Food Research Journal*, *30*(3), 750–763. https://doi.org/10.47836/ifrj.30.3.17.

Tran, Q. L., Lam, G. H., Le, Q. N., Tran, T. H., & Do, T. H. (2021). A comparison of several approaches for image recognition used in food recommendation system. In *10th IEEE International Conference on Communication, Networks and Satellite* (pp. 284–289). https://doi.org/10.1109/COMNETSAT53002.2021.9530793.

Vishal A. A., Aakash, G., Sanchay Kumar, S., & Preethiya, T. (2024). DeepAllergy: A deep learning approach for accurate and rapid food allergen detection. In *10th International Conference on Advanced Computing and Communication Systems, ICACCS 2024* (pp. 2225–2230). https://doi.org/10.1109/ICACCS60874.2024.10717042.


https://doi.org/10.24191/jcrinn.v11i2.612

West, H., Northrop, A., MPH, RD, & LMT. (2024, March 7). The 9 most common food allergies. *Healthline*. https://www.healthline.com/nutrition/common-food-allergies.

Yudeepanao, K., Sakkato, K., Tosaman, P., Songmuang, P., & Kongkachandra, R. (2024). Detecting food items with possible allergens using deep neural network. In *International Conference on Cybernetics and Innovations, ICCI 2024* (pp. 1–6). IEEE. https://doi.org/10.1109/ICCI60780.2024.10532449.