FOOD CALORIES DETECTOR & COUNTER ANDROID APPLICATION USING DEEP LEARNING

PROJECT DOCUMENTATION

BY

ERICK IVUTO MUTUNGA
SCT212-0562/2018
BSc Computer Technology

Dated 15th January 2022

Project source code GitHub link
https://github.com/erickmoon/calorie-counter

CONFIRMATION OF DOCUMENTATION SUBMISSION

(Student)

Signature

[illegible]

Name

Date

(Supervisor)

Signature

Name

Date

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Isaiah Onando Mulang' who has given me this bright opportunity to engage in this project. His willingness to spare her time in assisting and providing guidance to me are truly appreciated. It is my first attempt in the field of deep learning and my first step to establish my career in the field.

ABSTRACT

Poor dietary choices and lack of physical activity are two main contributing factors for the increasing prevalence of overweight and obesity. Overweight and obese individuals are at risk for developing major life-threatening diseases.

Weight loss is an effective means for reversing these adverse health effects, and smartphone applications (apps) may be an effective means for supporting weight loss outside of formal clinical settings.

This project uses a framework to compute calorie from classified dessert in an Android application based on deep learning. Transfer learning is being applied in order to train a convolutional neural network model that is able to identify desserts including both western and local. The main benefits of deep learning are that it eliminates the need of feature engineering and proven to excel in performance when compared to traditional image classification techniques. Nonetheless, Tensor Flow Inference API made it possible for trained model to be used in local mobile device that usually has restrictionsin terms of memory, computing power and storage space.

There will be lots of future work, one of them could be implementation of object detection so that multiple dessert objects can be detected and sum of calories can be computed instead. Moreover, model performance can be further improved in terms of accuracy and dessert variety. This would help to extend the usability of the project. Nonetheless, the implementation of application can also be further optimized especially the frames per second during real-time mode in order to produce a smoother user experience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
BMI	Body Mass Index
CNN	Convolutional Neural Network
CPU	Central Processing Unit
GPU	Graphics Processing Unit
HTML5	Hypertext Markup Language revision 5
KNN	K-Nearest Neighbors
RAM	Random Access Memory
RGB	Red, Green and Blue
RNN	Recurrent Neural Network
SURF	Speeded Up Robust Feature
SVM	Support Vector Machine
VRAM	Video Random Access Memory

CHAPTER 1: INTRODUCTION

1.1. Problem statement

Problems of overweight and obesity in Kenya and across the world is worsening (Economist Intelligence Unit, 2017) and it is directly related to uncontrolled calorie intake. Desserts could be the one to be blamed. The psychological sugar-craving is the reason of desserts high popularity. While people are enjoying their desserts, they might notbe aware of the high calorie content of it. Desserts could be calorie bombs and if they are being over-consumed, it will have a negative impact on our body such as obesity and high blood pressure due to the high saturation of sugar. Moreover, some local or traditional desserts information are not very well documented and therefore their details including calorie content are hard to be identified.

1.2. Scope and limitations

The scope of this project is to cover only individual food items and the application also works on only android devices, that way, iOS mobile devices cannot use the service and also taking photos of a mixed food item may not produce desirable results.

1.3. Background and motivation

Body mass index (BMI) can be calculated by the formula *BMI = (Weight in kg / Height in m$^{2)}$*. According to World Health Organization (2016), BMI >= 25kg/m^2 is considered overweight and BMI >= 30kg/m^2 is considered obese for adults aged above 18.An example of BMI range and their categories is shown in Figure 1. As stated by Haslamand James (2014), obesity could be linked to serious diseases such as cardiovascular disease, diabetes, and cancer. These are serious problems that need be paid attention as notonly it has direct impact on life expectancy but also quality of life.

Body Mass Index
Men
Underweight
Healthy weight
Overweight
Obese
<18.5
18.5-24.9
25.0-29.9
>30.0
Body Mass Index
Women
Underweight
Healthy weight
Overweight
Obese
<18.5
18.5-24.9
25.0-29.9
>30.0

One of the factors contributing to overweight and obesity problems is directly relatedto excessive calorie intake. Not to miss out that insufficient calorie intake could also lead to underweight problems which is also undesired. The daily calorie intake for average male is 2,500 calories while 2,000 for average female. However,these standards might not be suitable for anyone as there are several factors need to be considered aside from gender. These factors are age, height, weight and also activity level which indicates how often a person exercises in a week. Therefore, daily calorie intake foreach person will be different and in order to maintain a good health, people must be aware of their daily calorie intake and manage it properly.

According to World Health Organization (2003), daily sugar intake should not exceed 10% of total energy which means it should be within 50 grams. Following by American Heart Association (2009) which states that daily added sugar intake should be within 100 calories for females and 150 calories for males. However, regardless to rising price of sugar, In Kenya and across the world sugar intake increased by 33% between early 1960 and 2005. In addition, certain western desserts are also calorie bombs. For example, according to (Johnson et al.,2009), one NLEA serving of a plain, commercially prepared cheesecake weights 125 grams and has 401 calories. Not just that, most of the western desserts are made up of ingredients such as butter, eggs etc. that contains heavy calorie.

Recent rapid progression in development of smartphone hardware had made performing heavy computation in the device locally possible. Besides, deep learning had been the state-of-the-art in the field of image classification as it excels in accuracy when compared to the others. At the same time, smartphones are getting more common and affordable for everyone. By applying deep learning, a mobile device could be a useful toolto solve the problems of overweight and obesity. It provides convenience for people of allage to maintain their daily calorie intake and also gain better understanding towards their favorite dessert. In a nut shell, deep learning can be implemented into amobile application that is capable of identifying food and estimating calories based on inputs from phone camera itself.

This project uses a mobile application framework to estimate food calorie in two modes, based on a photograph. It will use apply deep learning instead of classical or traditional machine learning methods due to recent advancement of deep learning in terms of accuracy and reliability in the field of image classification.

1.4. Project objective

The project’s main objective uses a framework to estimate food calorie for different kind of desserts by applying deep learning. The in depth objectives are further discussed in the functional requirements in the Analysis chapter.

1.5. Proposed approach / study

The project applied deep learning as a backbone of the framework to classify frames and used the predicted results to compute dessert calories. Each frame captured from phone camera will be used as input to the model and the highest output scores will be used to compute calories.

1.6. Report Organization

The details of this project are shown in the following chapters. In Chapter 2, some literature reviews and experiments are done. Then, project overview and flows are shown in Chapter 4. Furthermore, more information such as methodologies, tools etc. are listed in Chapter 5. Finally, Chapter 6 concludes the project and suggested some possible future works that can be done.

1.7. Limitations

The system mostly relies on server to function properly. This prevents the system from being interactive due to absence of internet connection or communication delays. Likewise, the server used for image processing and food classification is a single point of failure. This renders the system to be non-functional in case the server fails. Then, there is no image segmentation performed in the server. This might reduce the overall accuracy of food classification as the background of the food image is not removed.

CHAPTER 2: LITERATURE REVIEW

There is a total of six papers being reviewed. Among them are existing systems and solutions to tackle the problem of difficulty in acquiring food calorie. Indeed, these papers focus on estimating food calorie based on a food photograph. The first paper discussed about applying the idea of crowdsourcing as a mobile application for user to upload food photograph in order for nutritionist experts to estimate its calorie content. The second paperproposed a system that uses API to acquire food nutritional value after successful food classification. The third paper proposed the calorie estimation system as a web service. The fourth paper introduces real-time food calorie estimation. Finally, the fifth and sixth paper applies Convolutional Neural Network (CNN) for food and traffic sign classification respectively.

2.1. Crowdsourcing of experts in nutrition and non-experts in identifying calories of meals.

Moorhead et al. (2015) proposed a solution to solve the problem by developing a personalized messaging system for a mobile application that is based on crowdsourcing, where calorie content of meal is determined by a group of nutritionist experts based on a photograph given. The architecture of the proposed application is shown below as Figure 2.

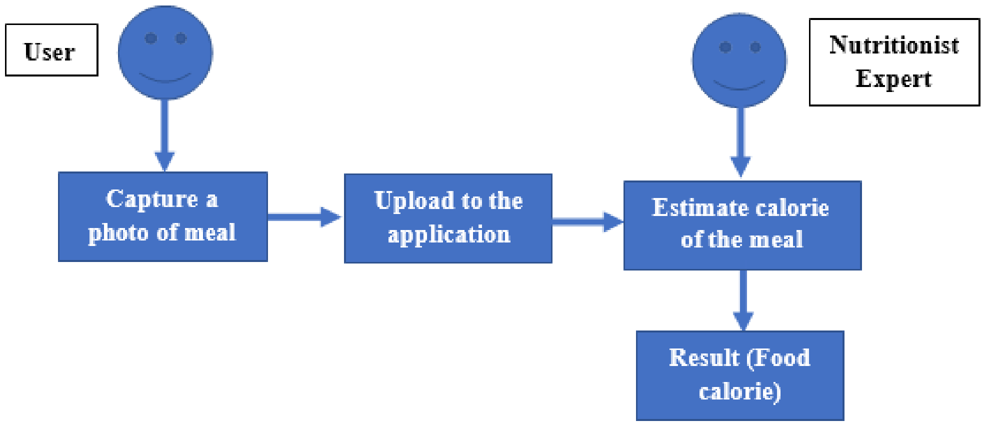


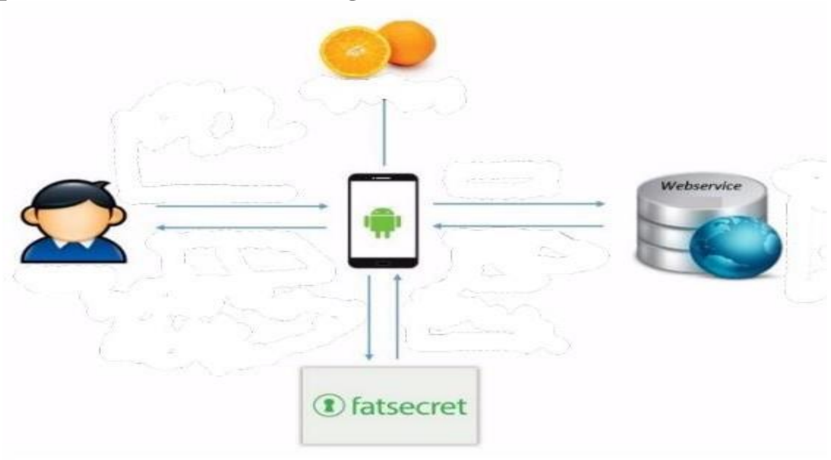
Figure 2: Framework of the SmartFood

This paper attempt to prove the benefit of crowdsourcing and how it could be used for an application that could help to prevent and manage obesity. To do that, an experiment is done on a group of 12 nutritionist experts and 12 non-experts by using online survey. Each group are required to estimate calories for 15 meals for twice, so there was a total of 720 estimates being made (15 meals * 24 persons * 2 instances).

In a nut shell, this paper proposed a solution that utilize crowdsourcing to achieve its goal to manage and prevent obesity. The solution stressed about estimating food calorie by human manually based on photograph taken. However, certain limitations exist in this solution as is over-reliance on human interaction with the system where it depends mostlyon manual input. To overcome this problem, an automated calorie estimation system will be a better way for users as it is much simpler and convenient to use.

2.2. Design and Implementation of Food Nutrition Information System using SURF and FatSecret API

The proposed application will take a food photograph as input and output its nutrition information to user. Among the returned nutrition information includes calorie, fat, carbohydrate and protein per serving. For the architecture used, user first take a photograph of their food. Then, the photograph is being sent to a server that will try to recognize the food and identify its name. Firstly, Speeded Up Robust Features (SURF) is used in the process of image recognition. Next, the classified name is used to get nutrition information via FatSecret API (FatSecret Platform API, 2017) and the result is then displayed to user. The framework for the proposed application is shown below Figure 3.

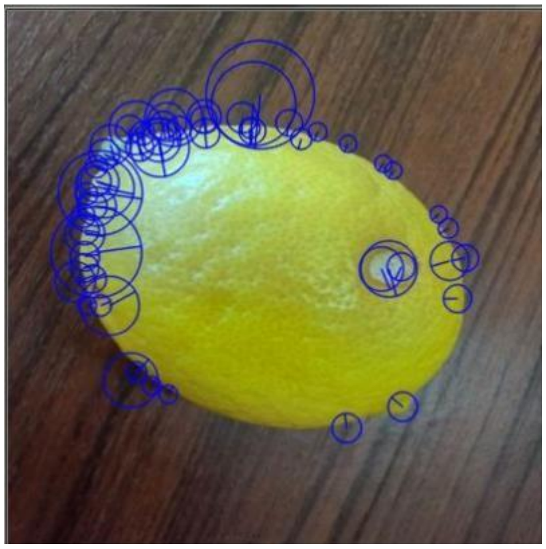


Using SURF, the keypoint features are extracted from image and compare with dataset by using Hessian matrix as shown in Equation 1 below. In addition, all of the symbol descriptions are shown in the table below.

$$H(x,\ \sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \qquad \textbf{(1)}$$

Symbol	Descriptio n
x	a point (x, y) in the image
σ	scale
$L_{xx}(x, \sigma)$	convolution of Gaussian second order derivative

Then, these keypoints will be used as the image descriptor. An example of an image oflemon with its keypoints is shown in Figure 2-3 below.



Finally, the similarity of key points is used to classify the objects compared with the datasetby calculating the Euclidean distance between two vectors.

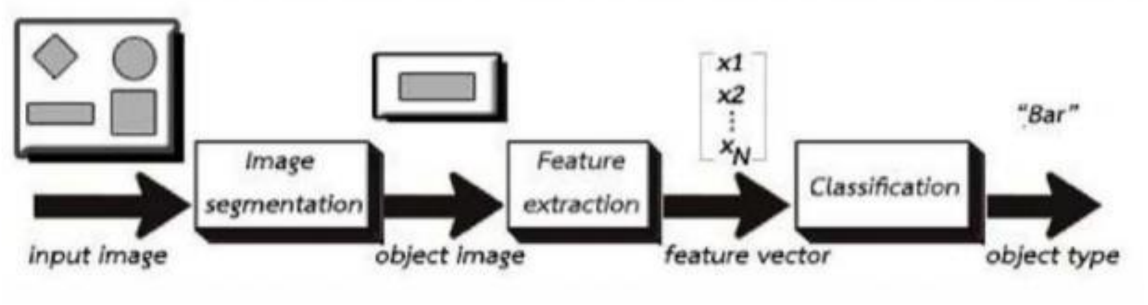
An experiment is conducted on correct recognition rate of 10 different objects suchas orange, banana etc. 20 unique photographs of each object are captured and tested with the application. In the end, the experiment achieved an average accuracy of 92% which issatisfiable.

One of the strengths of the proposed system is the usage of FatSecret API (FatSecret Platform API, 2017) which provide access to their food and nutrition database. It providescomplete access to their large database of food and its nutritional information. Also, the information retrieved are guaranteed to be accurate as all of the data are verified. Secondly, since that food classification are done in a server rather than the mobile device itself, this reduces the computational burden on the device. Thus, the phone specification requirements are significantly lowered as the application doesn't need to perform any heavy computation. Moreover, SURF is simple to implement, fast and requires smaller training datasets compared to Convolutional Neural Network (CNN) which is used in the project.

Nonetheless, FatSecret API is not an entirely free of charge service. There are several versions available and the basic version offers only up to 5000 API calls per day aswell as imposing throttling limit (FatSecret Platform API, 2017). That means that the application will not be able to retrieve any calorie information after maximum limit of APIcalls is reached.
To conclude that, the proposed system is tackled the problem of obesity by simplifying the process of acquiring food calorie that is to estimate food calorie from a photograph taken.

2.3. Calories Analysis of Food Intake Using Image Recognition

Tammachat and Pantuwong (2014) proposed a system that estimate calorie based on a food photograph taken. As discussed previously, several approaches to tackle the problem of obesity regarding to calorie intake is being proposed. Instead of applying the idea of crowdsourcing which is proposed by Moorhead et al. (2015) or acquiring information via an API as proposed by Hariadi et al. (2015), Tammachat and Pantuwong (2014) takes a different approach by applying the concept of calorie estimating as a service on the web.
The framework of the system is shown in Figure 5 below.



User will upload their food photograph onto the web server using browser. Then, the food portion is segmented

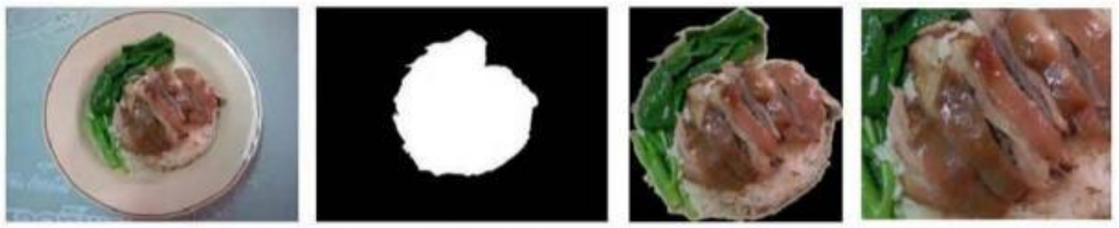


Figure 6: Example of image segmentation based on texture. Flow of the process is from left to right. (Tammachat and Pantuwong 2014)

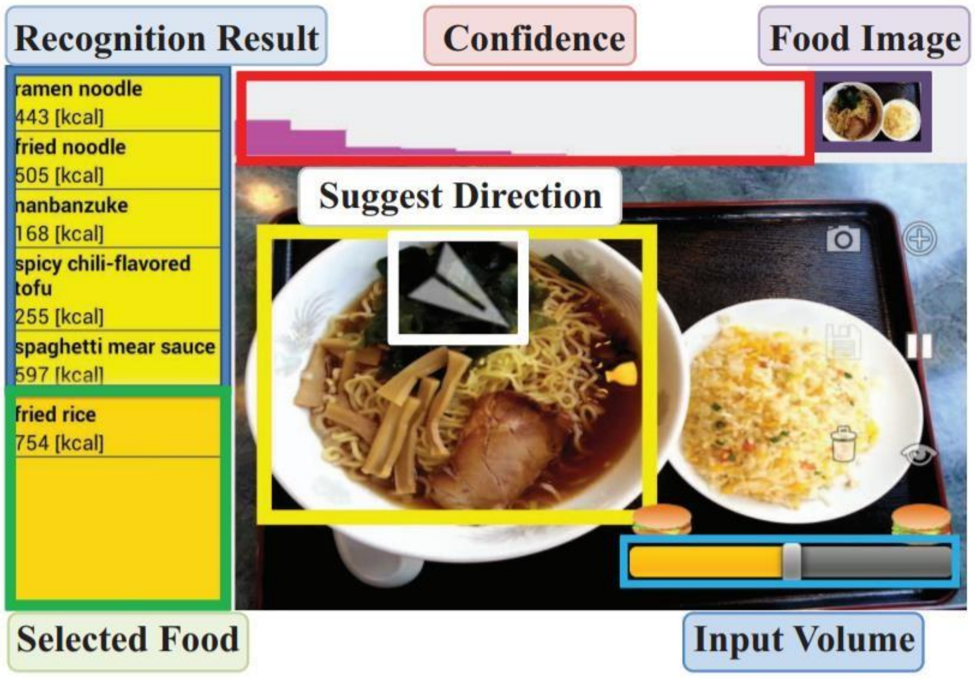
out from the whole image based on texture segmentation. An assumption that table and dish have no texture is being made. Then, feature vectors are extracted by Bag-of-Features (BOF), Segmentation based Fractal Texture Analysis (SFTA), and Colour Histogram. Finally, the feature vectors are used for food classification by Support Vector Machine (SVM) model with pre-trained datasets. Feature vectors with dimensions of 1x500, 1x24 and 1x30 are constructed respectively. Finally, a feature vector of 1x554 dimension (500 (BOF) + 24(SFTA) + 30 (Colour Histogram)) is created for each input image.

An experiment is performed on two kinds of SVM model. First model where constructed by food images grouped by food type achieved an average accuracy of 70%. The second model where constructed by food images grouped by range of calories achieved a slightly unsatisfying result with average accuracy of 46.4%. The paper concludes that first model is indeed better in terms of performance against second model.

One of the strength of the proposed system is being device independent. This is because it implemented as a service on website using HTML5 instead of a mobile application. Be it a desktop or smartphone, almost any devices can access the service via a browser. Furthermore, device requirements will not a problem since that all image processing operations and calculations are performed in a server. Nevertheless, the system will also record down user's calorie intake behavior and try to provide suggestions to use.

2.4. Real-time Mobile Food Recognition System

Kawano and Yanai (2013) proposed a mobile food recognition system. The systemis capable to estimating food calorie by simply pointing device camera towards it. Users are required to draw a bounding box around each food portion for the activate calorie estimation. Nonetheless, a slider is provided for user to indicate the size of the food portion.A screenshot of the proposed system is shown in Figure 7 below.



First of all, the proposed system harness use of GrabCut (Rother, Kolmogorov and Blake, 2004) to perform food portion segmentation. It segments out food portion from image background based on bounding boxed drawn by user. Then 50 categories of food are trained by Linear Support Vector Machine (SVM). Next, Colour Histogram (576 dimensions) and Bag-of-SURF (500 dimensions) are used to construct feature vectors from image. These feature vectors are then used to perform food classification using Linear SVM which is defined as inner product of two vectors. Equation 2 and Equation 3 are both equivalent Linear SVM formulas and the weighing factor w of input vector is calculated using Equation 4. All of the symbol descriptions are shown in the table below.

An experiment performed on the proposed system achieved a satisfying result of 81.55% correct classification rate with top five candidates of detected food

Table: Symbols with descriptions for Linear SVM formula

Symbol	Description
x	Input vector
x_i	Support vector
$f(x)$	Output SVM score
$yi \in \{+1, -1\}$	Class label
α_i	Weight of support vector
b	Bias vector
M	Number of support vector

One of the strengths of the proposed system is real time food classification, the system is capable of performing the action in real time, that is by simply point phone camera towards the food. Next, the system does not rely on any internet connection to function properly. This is because all image processing and recognition task are done in the device locally. As being said, users are able to acquire food calorie as quick as possible.To continue, the idea of using video instead of photograph allows user to locate correct food easily. Since the process of food recognition is repeated every second, user only need to continuously search for a new position to point their camera until the correct food is identified.

On the other hand, the system does impose a few weaknesses. For an instance, performance could be a problem when there are too many dishes present in a single frame. This is because GrabCut (Rother, Kolmogorov and Blake, 2004) is resource intensive, drawing too much bounding boxes at once will burdens the CPU of the device. Secondly, the system is not very autonomous. It displays top five candidates for the food detected and users are required to select the right food manually. Furthermore, users are required to draw bounding boxes around the food portion before the calculation of calorie activates. Of course, this is due to the system is not focus on only a single dish but more than one, so it requires assistance from user to locate the food. Likewise, the usage of GrabCut (Rother, Kolmogorov and Blake, 2004) is also another reason for it as the algorithm requires a bounding box to be drawn to determine background (table) and foreground (food portion).

Overall, the proposed system is creative and innovative in tackling the problem of obesity and calorie intake management. Not only it is able to detect more than one food at a time but also do it in real-time at the cost of performance. However, certain useful techniques proposed in the system such as GrabCut (Rother, Kolmogorov and Blake, 2004) could be applied to the proposed project and the performance can be improved by restricting detection to single food at a time.

2.5. Fruits and Vegetables Calorie Counter Using Convolutional Neural Networks

Akbari Fard et al. (2016) proposed a fruit and vegetables calorie counter system asmobile application. Convolutional Neural Network (CNN) is used as food classifier. The goal of the system is to detect and classify type of food based on a photograph and retrieve its relevant calorie content. The framework of the proposed system is shown in Figure 8.

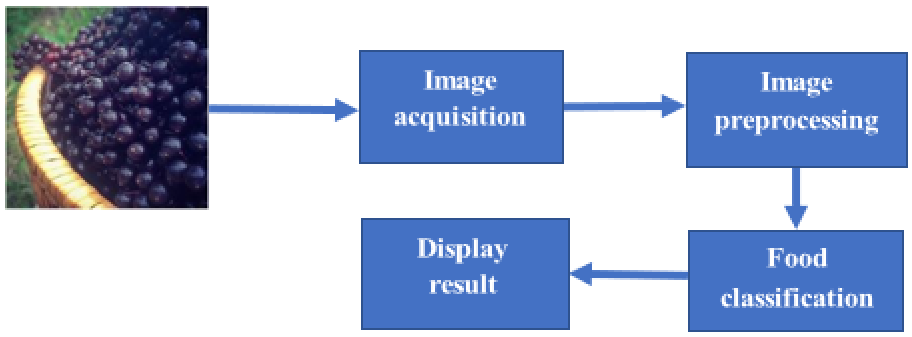


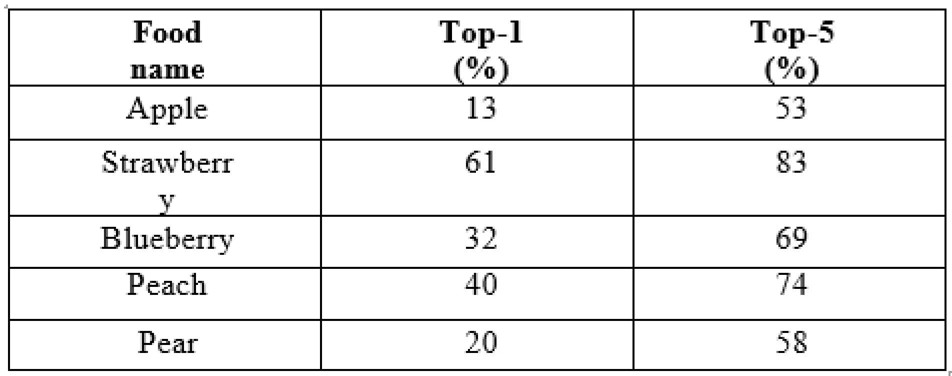
Figure 8: Framework for Fruits and Vegetables Calorie Counter

Firstly, test images were collected from ImageNet dataset where each of the imageswere pre-labeled. As a result, a dataset of 49,626 images spanning across 43 categories were handpicked. Among them, 80% of the images were used in training the CNN model while the rest were used for testing purposes. The features used for training are a combination of colour, texture, size and shape. The CNN consists of 3 convolution layers,3 pooling layers and 2 fully connected layers. The distribution of filters can be seen in the table below

Table: Number of filters and Filter sizes for each convolution layer

Convolution Layer	Number of filter	Filter size
First	48	11 x 11
Second	128	5 x 5
Third	128	3 x 3

Nonetheless, several preprocessing steps were performed on every image before they were used for CNN model training. These includes resizing each of them to dimensions of 256 x 256. Then, the resized images were cropped to dimensions of 227 x 227 with random offsets. These steps were performed to increase the accuracy of the foodclassification.
An experiment is conducted to identify the accuracy of the CNN model on fruit and vegetable classification. Two tests were conducted to determine the percentage of correct food classification in first prediction and percentage of correct food classification among top five predictions. The results are shown in the table below. Overall, the experiment achieved an average accuracy of 75% for Top-5 test and 45% for Top-1 test.



Food name	Top-1 (%)	Top-5 (%)
Apple	13	53
Strawberr y	61	83
Blueberry	32	69
Peach	40	74
Pear	20	58

The strength of the proposed system is the use of CNN as food classifier together with its huge image datasets which is 49,626 images used for training of the CNN model. This is because CNN

requires more training images to improve its accuracy compared to Support Vector Machine (SVM). The second strength is its framework simplicity which benefits the environment of a mobile device. Moreover, the system does not perform any communications with server which implies that it will work regardless of internet connection. This is because all operations including image classification are done locally in the device itself.
However, the system achieved a poor accuracy with only 45% of correct classification in Top-1 prediction. This might be caused by lack of image segmentation as the background of food are not being removed may have an impact on the results of classification. Furthermore, cropping the image with random offset might also introduces inaccuracy during classification. It should be standardised on every image.

2.6. Image classification applied to the recognition of traffic signs

Bruno and Osorio (2017) proposed an image classification system that is able to recognize traffic signals by applying deep learning. The main goal of the system is to increase road safety by usage of autonomous and semi-autonomous robotic vehicles. The paper adopted TensorFlow (Tensorflow, 2018), an open-source software library for machine intelligence to implement deep learning onto traffic signs classification.
Firstly, 21000 images which composed of 21 classes of German traffic sign was collected. Then, the image dataset was randomly split into 70% for as training set and remaining 30% for testing set for performance evaluation. The image dataset was converted from Portable Pixmap Format (PPM) to Joint Photographic Experts Group (JPEG). Image resolution were ranged between 15x15 and 256x256. Few examples of images are shown in figure below. The algorithm for detection is Slide Window where template is slide into input image to generate clippings. Then, the snippets are sent to Deep Convolutional Neural Network (DCCN) for classification. Figure 9 showcase some examples extracted from the image dataset.

Figure 9 Examples of traffic signs from image dataset (Bruno and Osorio, 2017)
An experiment is conducted by shuffling the dataset before splitting into training and testing set for 10 iterations. All training are performed only with CPU. The experiment achieved a minimum accuracy of 94% and averaged at 97.24% of correct classification for segmented and tagged

images. On the other hand, the system achieved an average of 75% accuracy for untagged images. Nonetheless, one of the main benefits of deep learning its ability to accept raw input images and perform feature extraction on its own to learn from it. Compared to classic machine learning, feature engineering needs to be done manually where learning is shallow. Besides, deep learning scales better with complex problems. As an example, a Support Vector Machine (SVM) usually require features in real-valued vectors while a Convolutional Neural Network (CNN) is end-to-end, which it has to ability to adapt to the problem it is solving. In general, CNN performed better with higher accuracy when compared to classic machine learning techniques.

On the other hand, deep learning techniques such as CNN do possess some disadvantages. Firstly, it requires a lot of training data in order to perform well. Then, it is associated with high computational costs where time taken to train a model is significantly long. Nonetheless, its benefits outweigh the cons.

In a nutshell, the rising popularity of deep learning had raised interest of people into it and once again raised other challenges in the field of image classification for others. Its usefulness can certainly be applied in many other ways.

2.7. Technology

TensorFlow, the open-source machine learning libraries is available for Python, Java and C, it is a tool designed to build deep neural network models (TensorFlow, 2018). Using TensorFlow, a built model can be deployed for usage on another platform including mobile platform, Android. This opened up wide opportunity for many new ideas to be implemented as it is impossible to perform machine learning in mobile platform before this without connecting to cloud (Greene, 2018). Besides, TensorFlow also supports deployment of computation across multiple CPU or GPU easily. This helps to save a lot of training time. Moreover, as compared to other machine learning library such as Theano (Deeplearning.net, 2018), TensorFlow compile times is much better. Nonetheless, there is also a tool named TensorBoard that allow visualization of graph and performance easily.

One of the other popular deep learning libraries is Keras (Keras.io, 2018). It’s much simpler and user friendlier compared to Tensorflow. The other benefits of Keras is its modularity. A complex neural network can be built in just a few lines of codes with Keras. However, TensorFlow provides much more flexibility than Keras as it offers more advanced operation. This means that

TensorFlow provides more control over the network. Furthermore, Tensorflow stands out as it supports for multi-platform provides more use case of neural network compared to others such as Theano and Keras.

2.8. Image classifying techniques

2.8.1. K-Nearest Neighbors (KNN)

KNN is a simple image classification technique that tries to find a predefined number of training samples, the k-nearest neighbours that have the closest distance to a new sample. (Scikit-learn.org, 2018). The first step of KNN is to memorize all the imagesdata and their labels. Then, a distance function is used to measure distance between two images. The common distance function are L1 and L2 distance which are also known as Manhattan and Euclidean distance respectively as shown in Equation 5 and 6 below (K Nearest Neighbors, 2018).

$$(l_1, l_2) = \sum_p |l_1^p - l_2^p| \qquad \textbf{(5)}$$

$$(l_1, l_2) = \sqrt{\sum_p (l_1^p - l_2^p)^2} \qquad \textbf{(6)}$$

The distance between each input samples are compared to every training sample and k number of training samples with closest distance to input samples is obtained and followedby their labels. A majority voting is performed on the predictions to get final predicted label for each input sample. In this case, the hyperparameter is value of k as it determines

how many predictions to be considered in the voting. An example of majority voting can be seen in the Table below. In this case, donut will be the final predicted label.

Table: Sample data to showcase majority voting

Labels	No. of occurrence in predictions
Donut	5
Egg tart	2
Mooncake	3

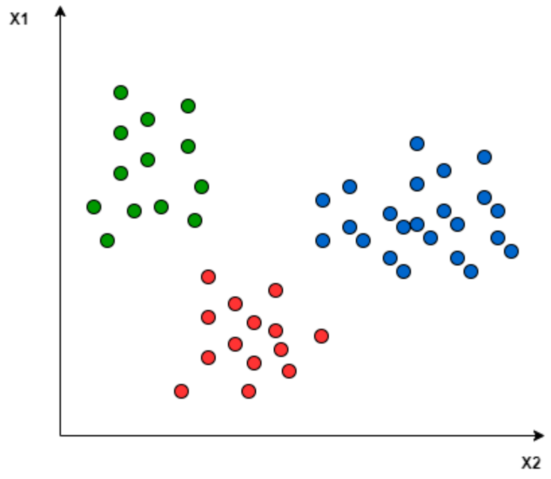
The strength of KNN is its simplicity to implement as the algorithm is pretty straightforward while the weakness of KNN is its high computational cost as the size of the dataset grows larger. This will cause fast training as its just saving all data and slow prediction as it needs to compute distances between every training sample against every input samples.

2.8.2. Multiclass Support Vector Machine (SVM) Linear Classifier

In its most simple type, SVM doesn't support multiclass classification natively. It supports binary classification and separating data points into two classes. For multiclass classification, the same principle is utilized after breaking down the multiclassification problem into multiple binary classification problems.

The idea is to map data points to high dimensional space to gain mutual linear separation between every two classes. This is called a One-to-One approach, which breaks down the multiclass problem into multiple binary classification problems. A binary classifier per each pair of classes. Another approach one can use is One-to-Rest. In that approach, the breakdown is set to a binary classifier per each class.A single SVM does binary classification and can differentiate between two classes. So that, according to the two breakdown approaches, to classify data points from m classes data set:

- In the One-to-Rest approach, the classifier can use m SVMs. Each SVM would predict membership in one of the m classes.
- In the One-to-One approach, the classifier can use $\frac{m(m-1)}{2}$ SVMs. Let's take an example of 3 classes classification problem; green, red, and blue, as the following image:



Applying the two approaches to this data set results in the followings:

In the One-to-One approach, we need a hyperplane to separate between every two classes, neglecting the points of the third class. This means the separation takes into account only the points of the two classes in the current split. For example, the red-blue line tries to maximize the separation only between blue and red points. It has nothing to do with green points:

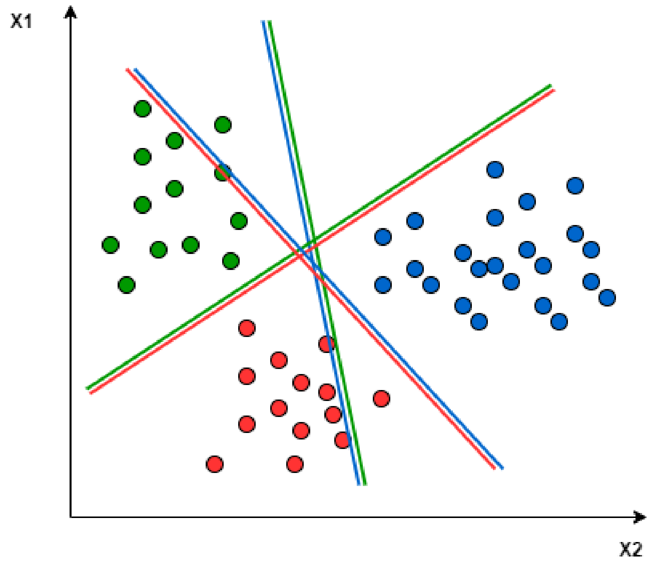
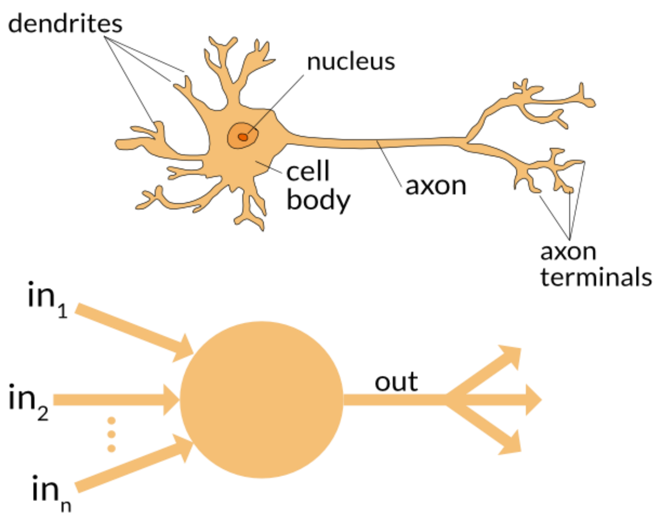


Figure 9: KNN classification graph

In the One-to-Rest approach, we need a hyperplane to separate between a class and all others at once. This means the separation takes all points into account, dividing them into two groups; a group for the class points and a group for all other points.

2.8.3. Deep Learning

Deep learning had gain huge success and applicable to wide area of applications such as voice recognition, image classification and games. It takes advantage of increasing computational power and availability of data nowadays. Deep learning actually inspired the way human brain works. Figure shows an artificial neuron and biological neuron which are very similar. Both of them work similarly

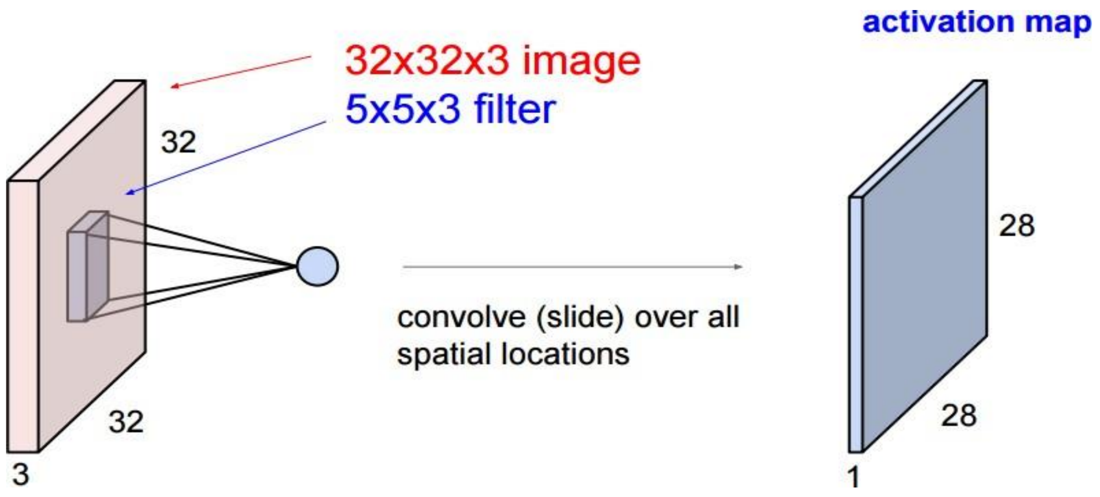


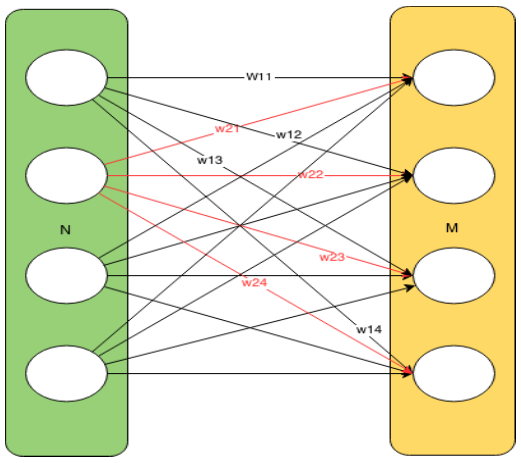
by taking in inputs, process them and send out the output. According to Applied Go (2016),neuron receives multiple inputs and process it into one output. Each input is multiplied by a weight and add to a bias. During the training phase, the neural network adjust the weights and biases accordingly through a process called backward propagation using gradient descent (Brilliant.org, 2018) where calculations of gradient proceeds backwards from last layer to the first. This indicates that the neural network is learning from its data through minimizing the loss function.

There are many types of deep learning, Convolutional Neural Network is one of the most popular one. CNN is built up by a sequence of layers that transform an input to output

with some differentiable function (Cs231n, 2018). The common layers in CNN are convolution layer, pooling layer and fully connected layer. First of all, Convolution layer is essentially a set of filters where each filter detects a feature. Its theory are similar to pooling where a sliding window scan across the inputs while each capture

different featuressuch as colour, contours, edges etc. Figure 11 shows an example of the convolution layer. Then, another common layer is Fully Connected layer. In this layer, every neurons from one layer connects to the other layer. An illustration of Fully Connected layer is shown in Figure 12 below.





Besides that, sometimes overfitting can happen, this is the case where the networkis too closely fit to the training set. It needs to be simpler and general in order to solve

problems outside the training data. Therefore, dropout layer can help in this situation. This improves generalization as the model drops some of the neurons causing them to be deactivated (Santos, 2018). An example of illustrating the neural network with dropout layer is shown in Figure 13 below.

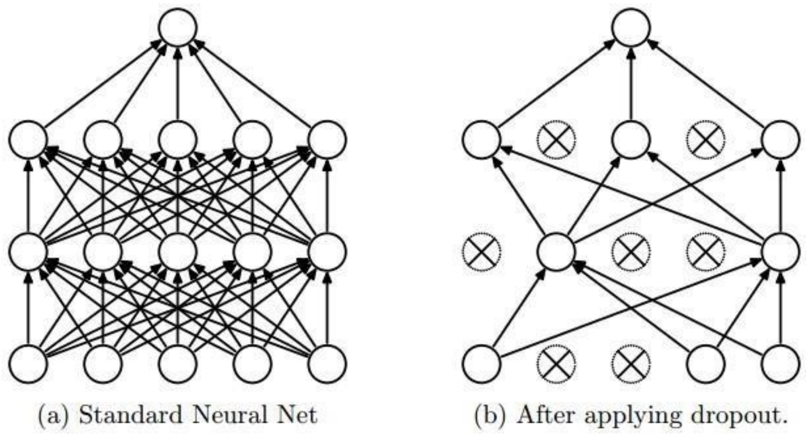


Figure 13: Neural network before and after dropout layer (Santos, 2018)

Aside from that, there is also another popular deep learning techniques known as Recurrent Neural Network (RNN). The idea behind RNN is the use sequential informationwhile the term 'recurrent' means that it perform the same task for every element in the sequence (Britz, 2015). The formula for Vanilla RNN can be seen in Equation 13 and 14 below (Karpathy, 2015). The symbols and descriptions for it are shown in the table below.

$$h_t = f_W(h_{t-1}, x_t) \tag{13}$$

$$y_t = W_{hy} h_t \tag{14}$$

Table: Symbols and descriptions for Vanilla RNN

Symbol	Description
ht, ht-1	New and old hidden state
f_w	Some function with parameters W
x_t	Input vector at some time step
y_t	Output vector

Since that RNN does not take in fixed size of input and produce fixed size of outputlike CNN did, it is ideal for sequential data such as text generation, speech recognition andmachine translation. On the other hand, CNN is more ideal for classification task such as image classification.

One of the strengths of multiclass SVM lies on its ability of generalization that creates simpler model. However, there are more hyperparameters to be considered and it is crucial to the performance of the classifier.

CHAPTER 3: SYSTEM ANALYSIS

3.1. Requirements specifications

3.1.1. Functional requirements

i. An android mobile app that is able to calculate BMI given Weight and Height
ii. Given a photo of a food item, the app should be able to detect nutritional information of the given food item.
iii. The returned nutritional information should be stored in a local SQLite Database
iv. User should be able to access the information above in (iii)
v. User should be able to set a daily calorie intake goal.
vi. The system should be able to check the limit/goal set by the user in (v) above and notify user when the goal is reached or exceeded.

3.1.2. Non-functional requirements

i. Speed

Speed determines how fast the application responds to commands. The application is expected to perform and respond within the shortest time possible.

ii. Security

To protect sensitive data, user's information is stored locally within SQLite database and is not stored or transmitted to any server or third party.

iii. Portability

For example, a user might purchase a new cell phone model and download a mobile application they had on their last device. The application runs as efficiently on the new phone as it did on the old phone.

iv. Compatibility

The application is compatible with android devices from 5 to 10 which can be considered a huge range of compatibility

v. Reliability

Technology that is highly reliable functions with the same or similar efficiency after extensive use.

3.2. Feasibility study

3.2.1. Technical Feasibility Needs and Resources

As for technical needs, it is critical to ensure that the proposed device would be accessible to a broad range of people. Therefore, it should be compatible with Android versions 5.0 and above with internet connection. In other words, technical feasibility should target motivation, social connectivity, and self-monitoring in order to assist customers in keeping an eye on their calorie intake.

3.2.2. Legal feasibility

The application's design, engineering and deployment process does not infringe the laws of any state, but irrespective, there was a great concern in how user's data is handles and thus for security and privacy purposes, user's data was stored locally in an SQLite database.

3.2.3. Operational feasibility

This is perhaps the most proven feasibility of my calorie counter mobile application. The application proposes a digital way of tracking calories from a food item by simply taking a picture using device camera or uploading an image from the end user's camera.

There is also high efficiency in the number of steps involved to obtain food nutrition information. The user simply logs in, or skips that process, and is immediately presented with the option of adding a food item from gallery or camera and under 10 seconds the nutritional value of the food item captured is displayed to the user.

3.2.4. Economic Feasibility

The app has no active monetization implementations because of the limited time of developing it but, should have Google Admob ads later to create passive income, that is, income which the user does not pay directly for.

Google Admob is an efficient and secure way of displaying controlled ads to users which pays for impressions and clicks and should generate enough revenue to sustain the app.

The app has hardly any running or setup expenditure and thus most of the income will be considered profit.

3.2.5. Schedule feasibility

The timelines provided for the development of this project are totally achievable and realistic and I came up with a well-structured breakdown of tasks against time.

CHAPTER 4: SYSTEM DESIGN

4.1. Project summary

4.1.3. Project flow and summary

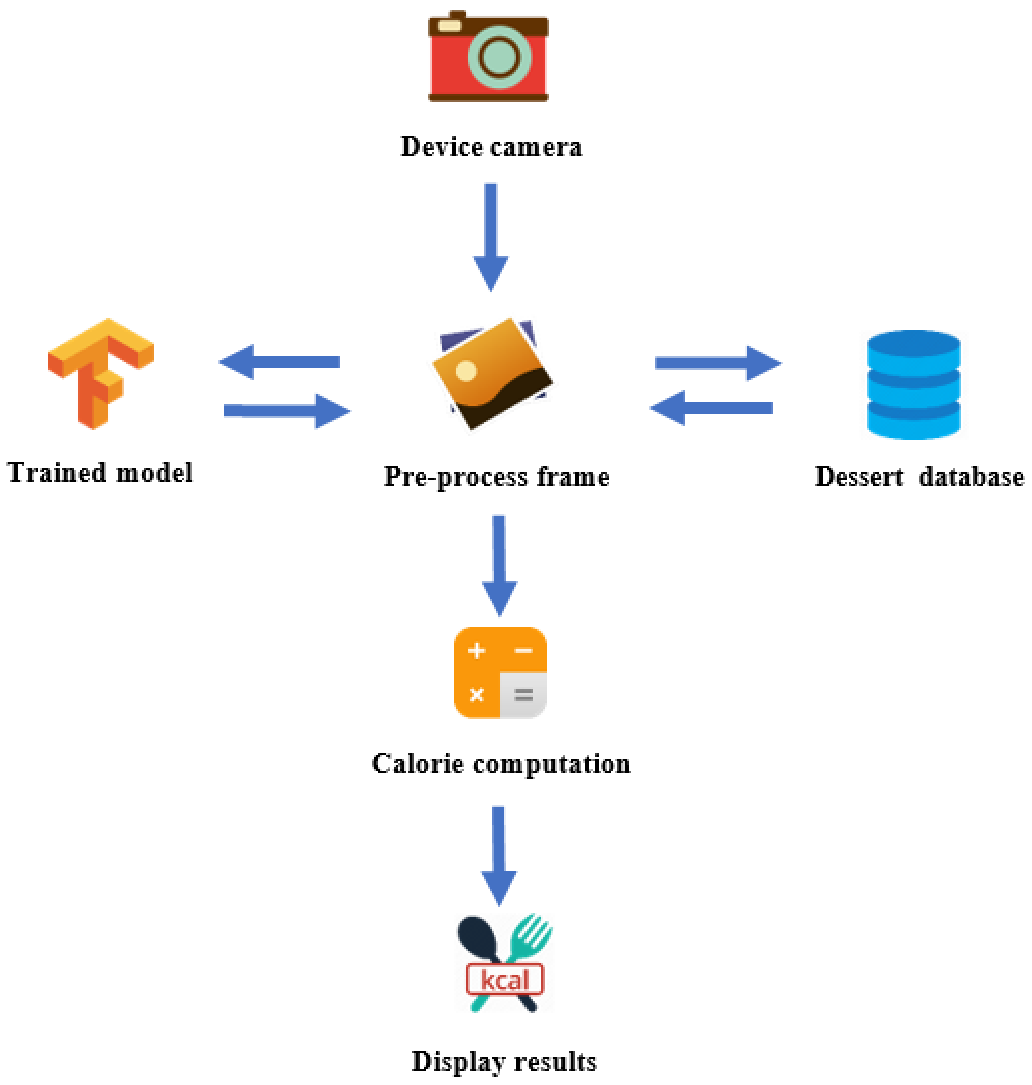


Figure 14: Overview of system design for the mobile application

4.1.2. Detailed flow of mobile application

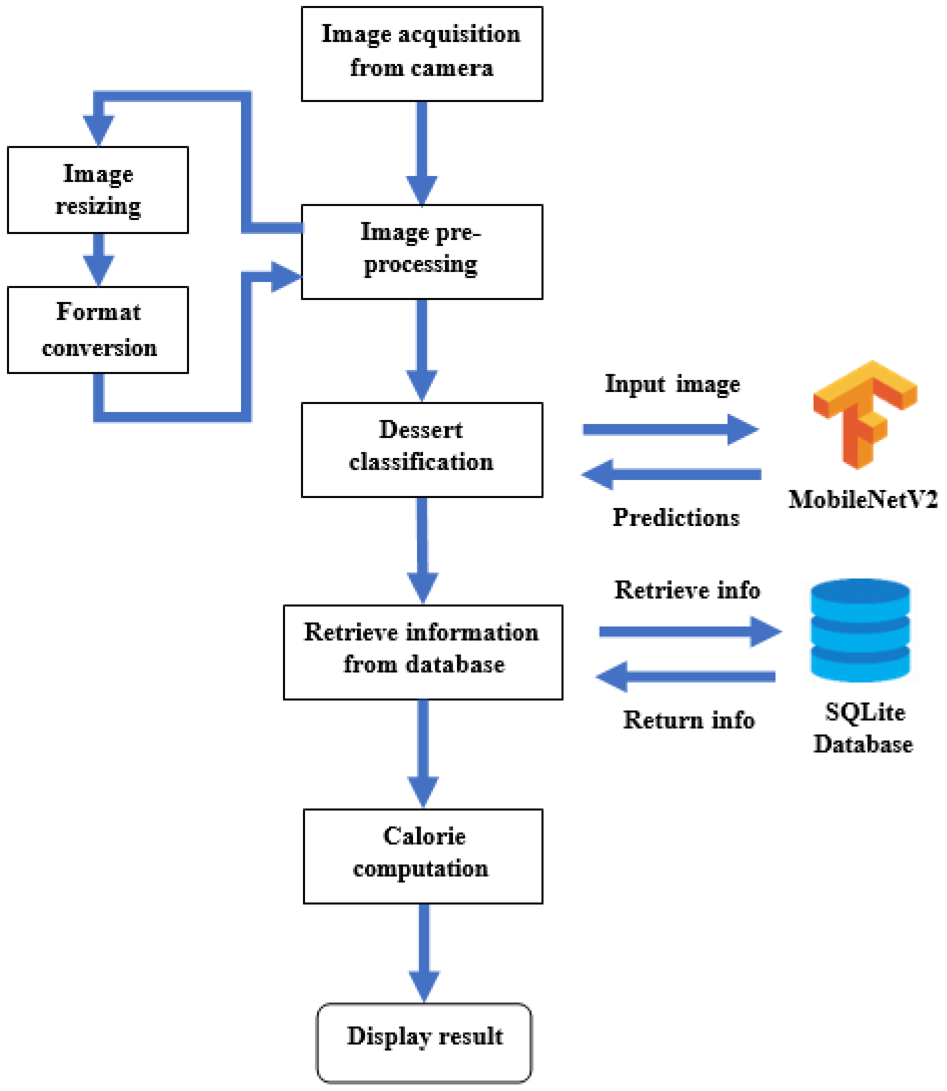


Figure 15: Detailed flow of the mobile dessert calorie counter application

Step 1: Image acquisition

- For the best result, user should try to locate and position the target food within the camera frame.
- The whole portion of the food should be present within the frame without any blockage.

Step 2: Image pre-processing

- Every time a new image frame is acquired, certain pre-process needs to be done before it is ready for inference.
- To start with that, each image frame is first resized into resolution of 224x224 according to the default set by MobileNetV2.
- Then, the array is arranged so that image channels are ordered last. The desired image format will be ordered by batch size, image width, image height and number of color channels.
- Finally, the array is flattened into a two-dimensional array.

Step 3: Dessert classification

- The trained TensorFlow model will be used to perform dessert classification.
- This is achieved through the use of TensorFlow Inference API which acts as abridge between TensorFlow model and the Android application.
- The pre-processed bitmap image is fed into the model for inference and the output predictions will be an array of scores.
- The scores are sorted and the dessert class who scored the highest will be the finaloutput

Step 4: Retrieve information from database

- Since the dessert classes inthe SQLite database is ordered the same as the model output,the index of classes who scored highest will be used to query for its relevant information

Step 5: Calorie computation

- The base dessert calorie of the predicted dessert is obtained.
- For several serving sizes, the calorie is computed accordingly and the final results are displayed to user

Splash screen

This is the welcoming screen which simply gives an idea of what the app is about.

Figure 16: Splash screen

User details entry screen

This is the point where user enters their personal details.

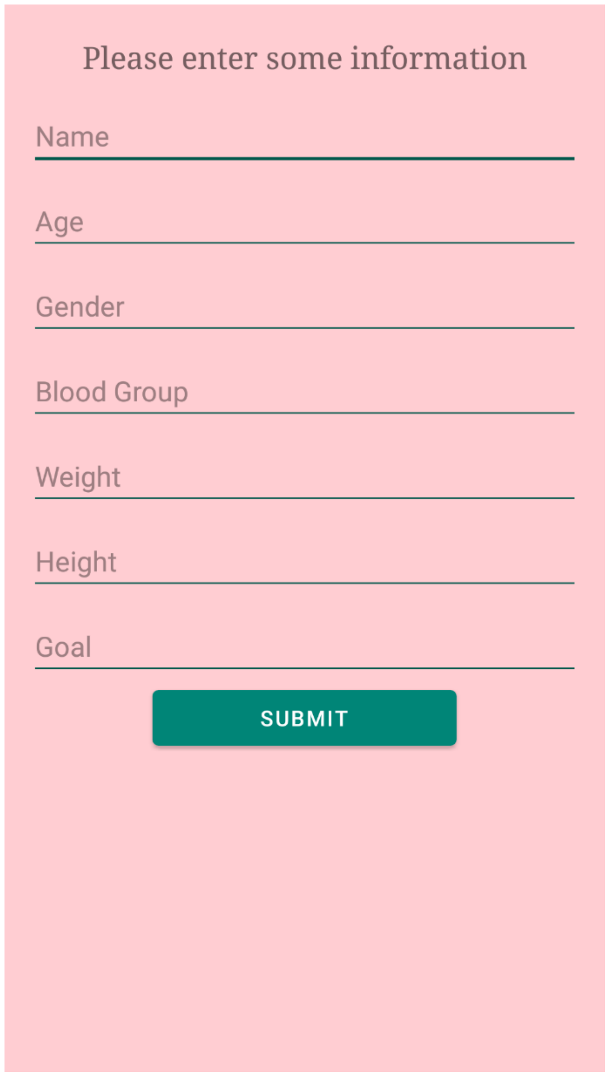


Figure 17: User details entry screen

BMI Calculator page

In this page, users can enter their height and weight and their BMI is calculated by the app.

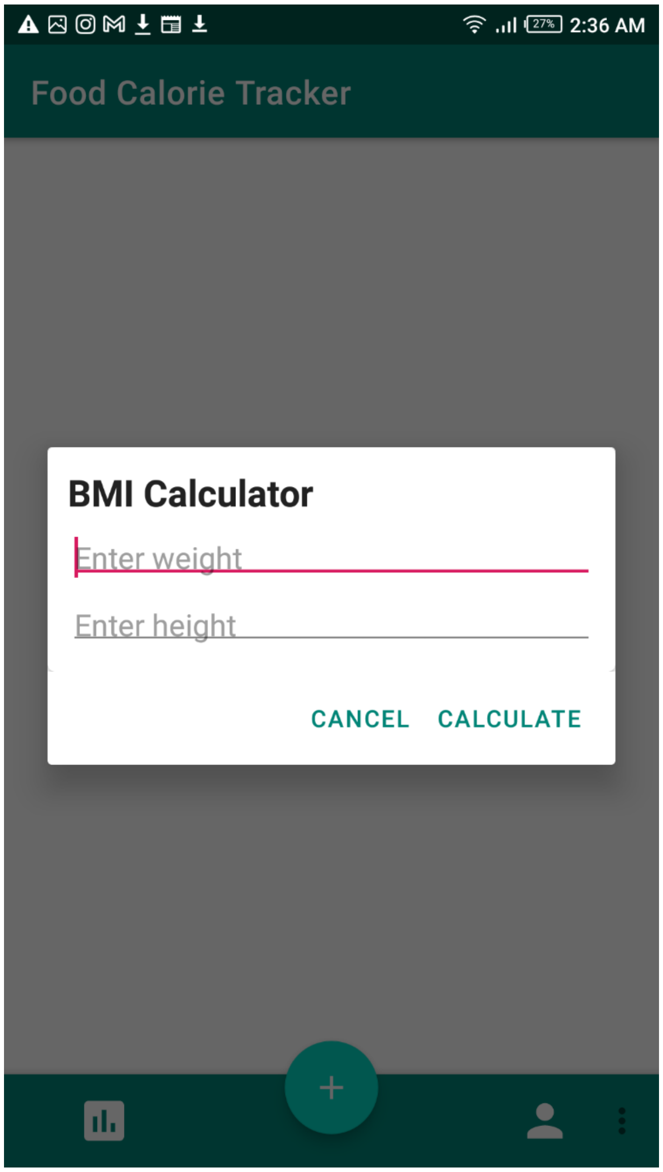
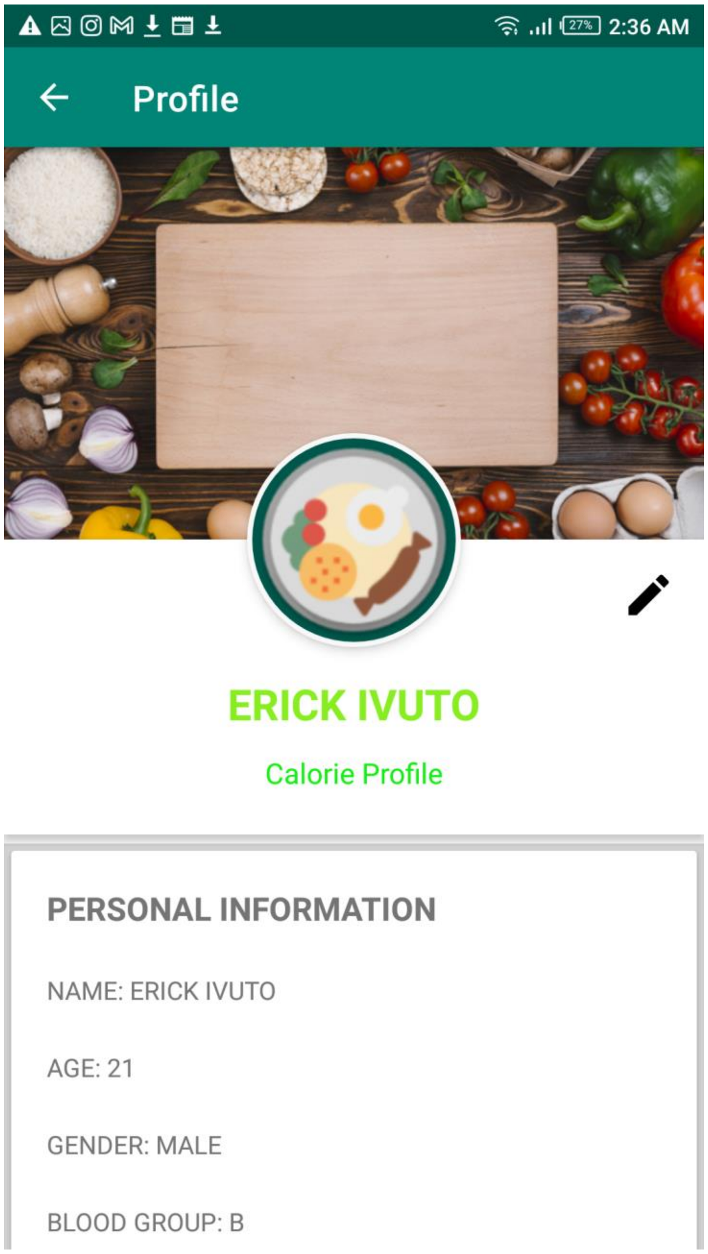


Figure 18: BMI Calculator page

Profile/Account page

This is the page that contains user's personal information which is also editable



PERSONAL INFORMATION

NAME: ERICK IVUTO

AGE: 21

GENDER: MALE

BLOOD GROUP: B

Figure 19: Profile/Account page

Homepage screen

This is the main screen that contains navigation buttons and most importantly the button to scan image and display calorie results.

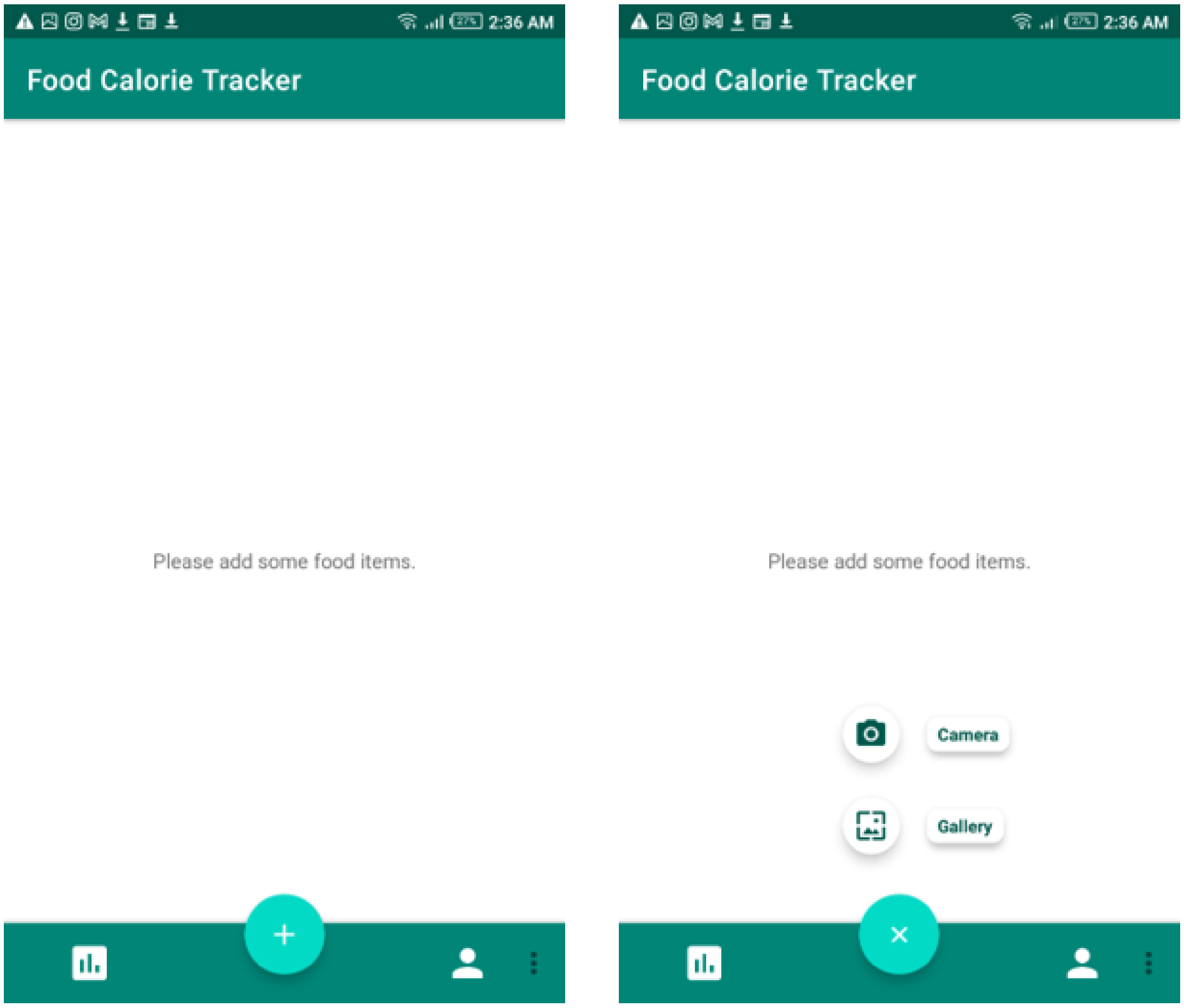
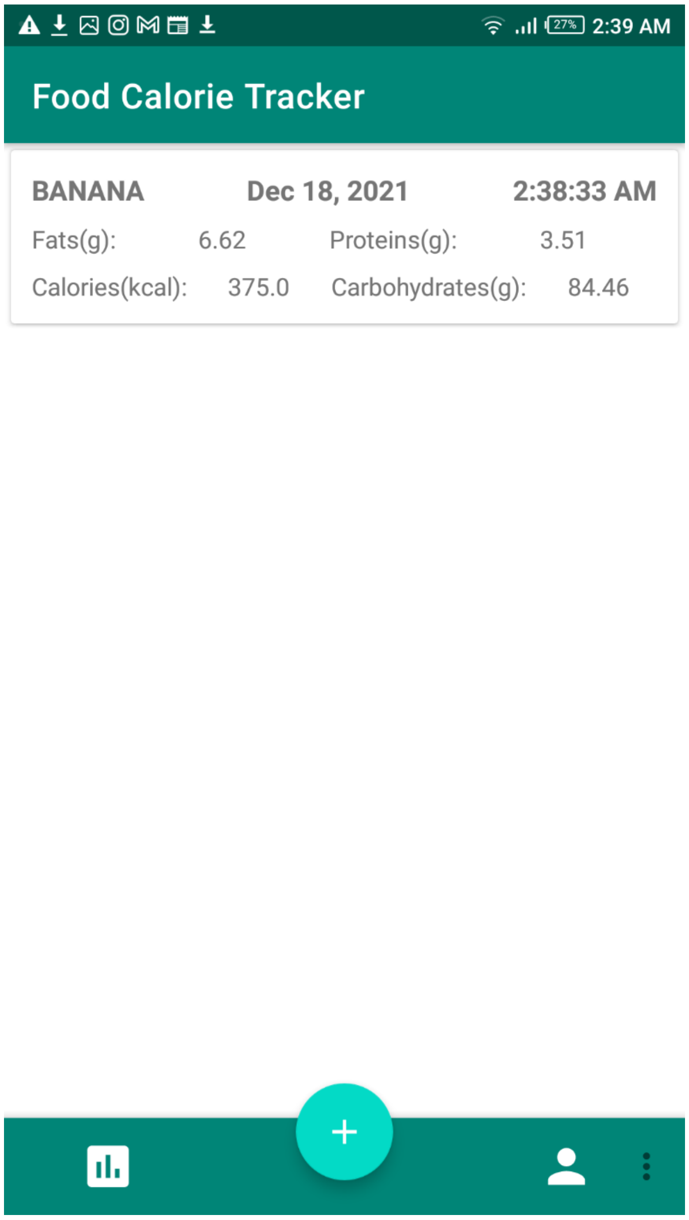


Figure 20: Homepage screen

Screen after capturing/selecting image

On pressing the gallery/camera button above an image is selected


2:39 AM

Food Calorie Tracker

BANANA	Dec 18, 2021		2:38:33 AM
Fats(g):	6.62	Proteins(g):	3.51
Calories(kcal):	375.0	Carbohydrates(g):	84.46

Figure 21: Screen after capturing/selecting image

Summary graph page

This page will contain a graph. The graph will be based on user's daily calorie intake.

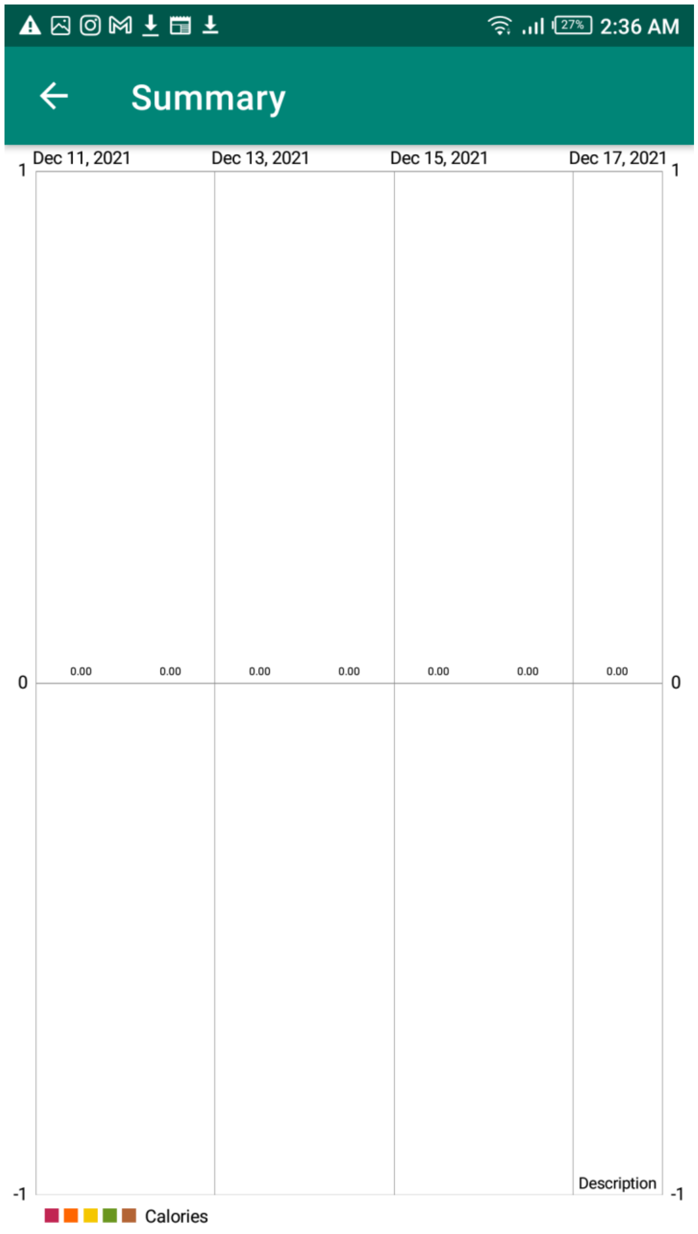


Figure 22: Summary graph page

4.2. Technical diagrams & models

4.2.1. Use case diagram

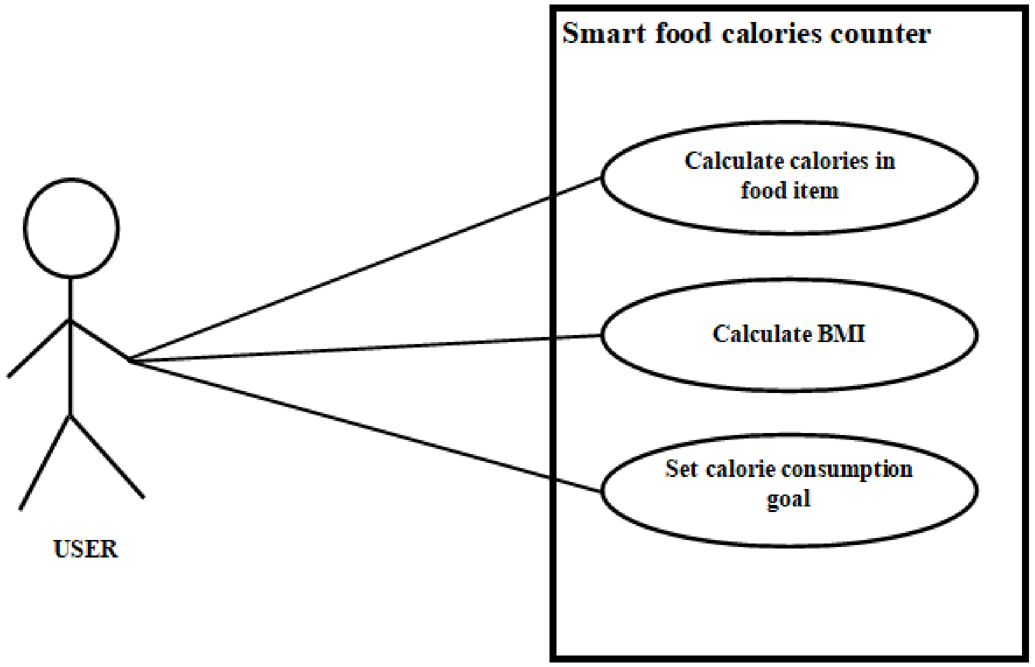

Smart food calories counter
Calculate calories in food item
Calculate BMI
Set calorie consumption goal
USER

Figure 23: Use case diagram

From the use case diagram, the (3) major functionalities of the app are brought out:

1. Using artificial intelligence to calculate the calorie estimate of a food product
2. Calculating user BMI
3. Setting a calorie consumption goal

4.2.2. State diagram

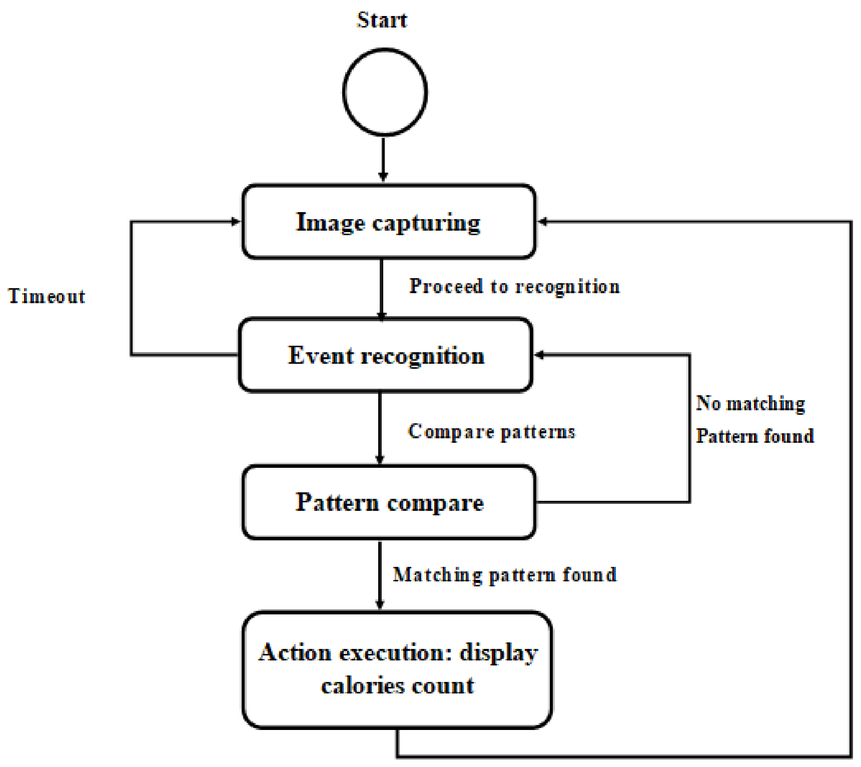


Figure 24: state diagram

The state diagram shows how the system works. The first step in the calorie detection process is of course capturing a photo of a food item where recognition is applied and pattern compared to give its calorie estimate.

4.2.3. Sequence diagram for calorie approximation action

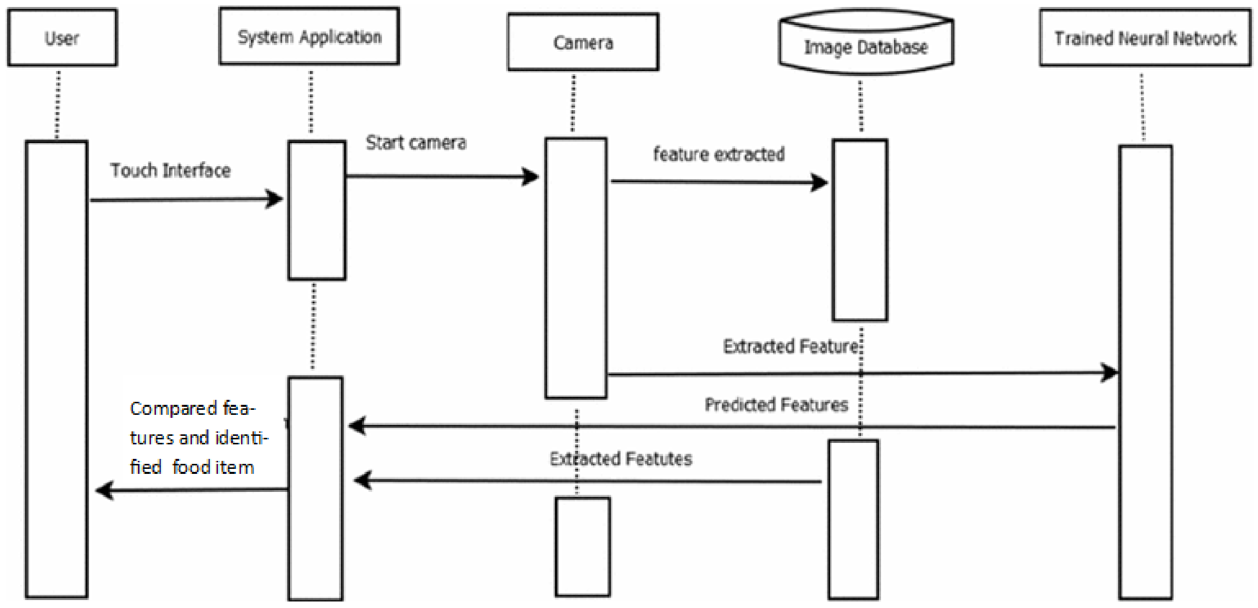


Figure 25: Sequence diagram for calorie approximation action

4.2.4. Class diagrams

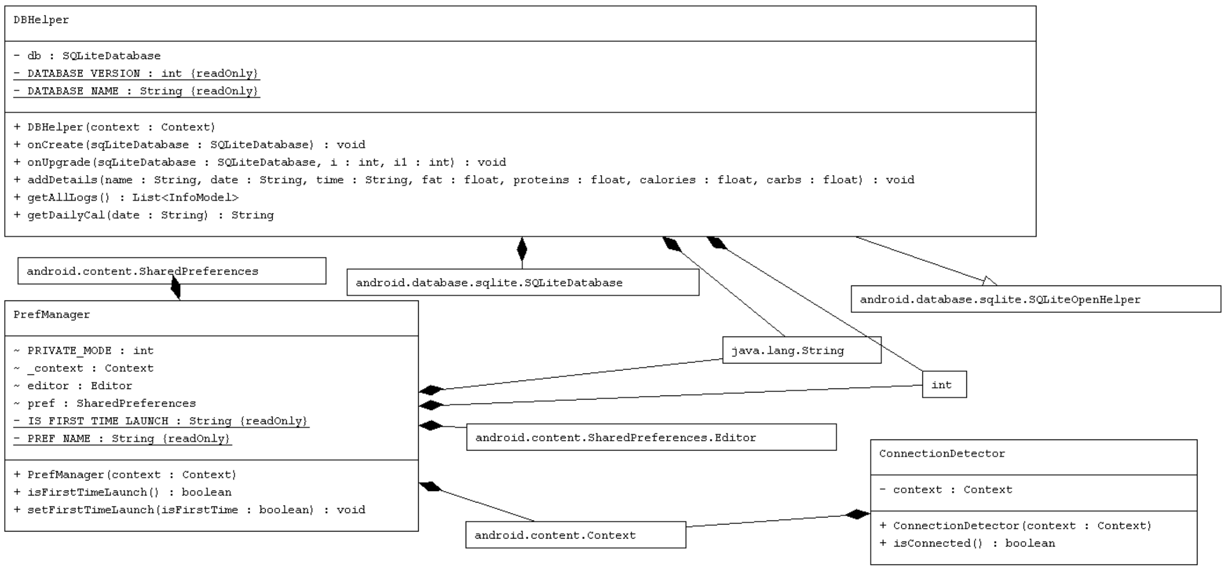

DBHelper
- db : SQLiteDatabase
- DATABASE_VERSION : int {readOnly}
- DATABASE_NAME : String {readOnly}
+ DBHelper(context : Context)
+ onCreate(sqLiteDatabase : SQLiteDatabase) : void
+ onUpgrade(sqLiteDatabase : SQLiteDatabase, i : int, i1 : int) : void
+ addDetails(name : String, date : String, time : String, fat : float, proteins : float, calories : float, carbs : float) : void
+ getAllLogs() : List<InfoModel>
+ getDailyCal(date : String) : String
android.content.SharedPreferences
android.database.sqlite.SQLiteDatabase
android.database.sqlite.SQLiteOpenHelper
PrefManager
~ PRIVATE_MODE : int
~ _context : Context
~ editor : Editor
~ pref : SharedPreferences
- IS_FIRST_TIME_LAUNCH : String {readOnly}
- PREF_NAME : String {readOnly}
+ PrefManager(context : Context)
+ isFirstTimeLaunch() : boolean
+ setFirstTimeLaunch(isFirstTime : boolean) : void
java.lang.String
int
android.content.SharedPreferences.Editor
ConnectionDetector
- context : Context
+ ConnectionDetector(context : Context)
+ isConnected() : boolean
android.content.Context

Figure 26: class diagram 1

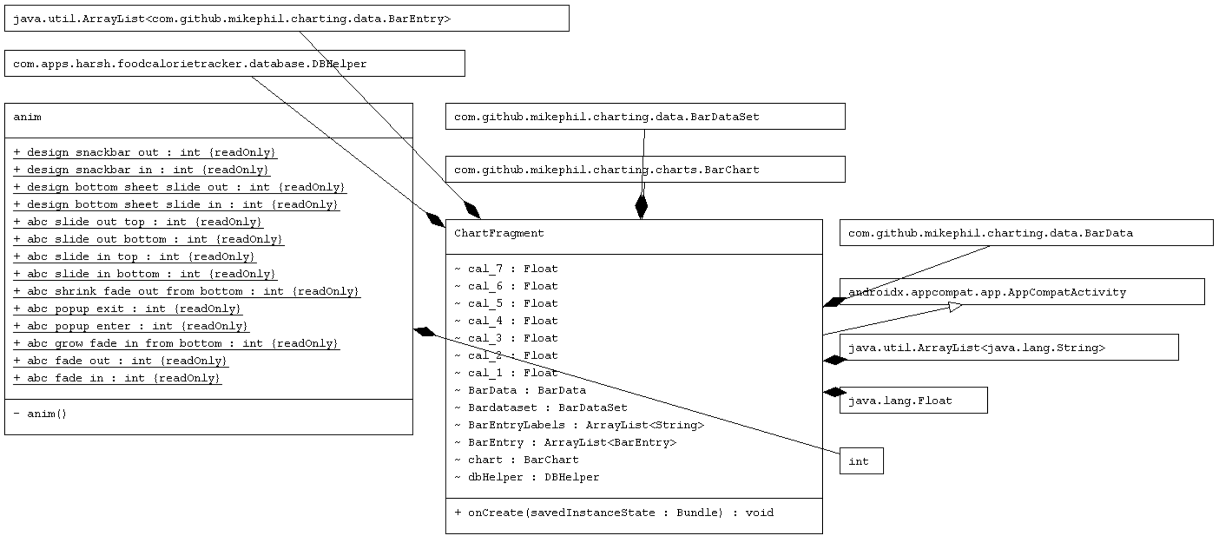

java.util.ArrayList<com.github.mikephil.charting.data.BarEntry>
com.apps.harsh.foodcalorietracker.database.DBHelper
anim
+ design_snackbar_out : int {readOnly}
+ design_snackbar_in : int {readOnly}
+ design_bottom_sheet_slide_out : int {readOnly}
+ design_bottom_sheet_slide_in : int {readOnly}
+ abc_slide_out_top : int {readOnly}
+ abc_slide_out_bottom : int {readOnly}
+ abc_slide_in_top : int {readOnly}
+ abc_slide_in_bottom : int {readOnly}
+ abc_shrink_fade_out_from_bottom : int {readOnly}
+ abc_popup_exit : int {readOnly}
+ abc_popup_enter : int {readOnly}
+ abc_grow_fade_in_from_bottom : int {readOnly}
+ abc_fade_out : int {readOnly}
+ abc_fade_in : int {readOnly}
- anim()
com.github.mikephil.charting.data.BarDataSet
com.github.mikephil.charting.charts.BarChart
ChartFragment
~ cal_7 : Float
~ cal_6 : Float
~ cal_5 : Float
~ cal_4 : Float
~ cal_3 : Float
~ cal_2 : Float
~ cal_1 : Float
~ BarData : BarData
~ Bardataset : BarDataSet
~ BarEntryLabels : ArrayList<String>
~ BarEntry : ArrayList<BarEntry>
~ chart : BarChart
~ dbHelper : DBHelper
+ onCreate(savedInstanceState : Bundle) : void
com.github.mikephil.charting.data.BarData
androidx.appcompat.app.AppCompatActivity
java.util.ArrayList<java.lang.String>
java.lang.Float
int

Figure 27: class diagram 2

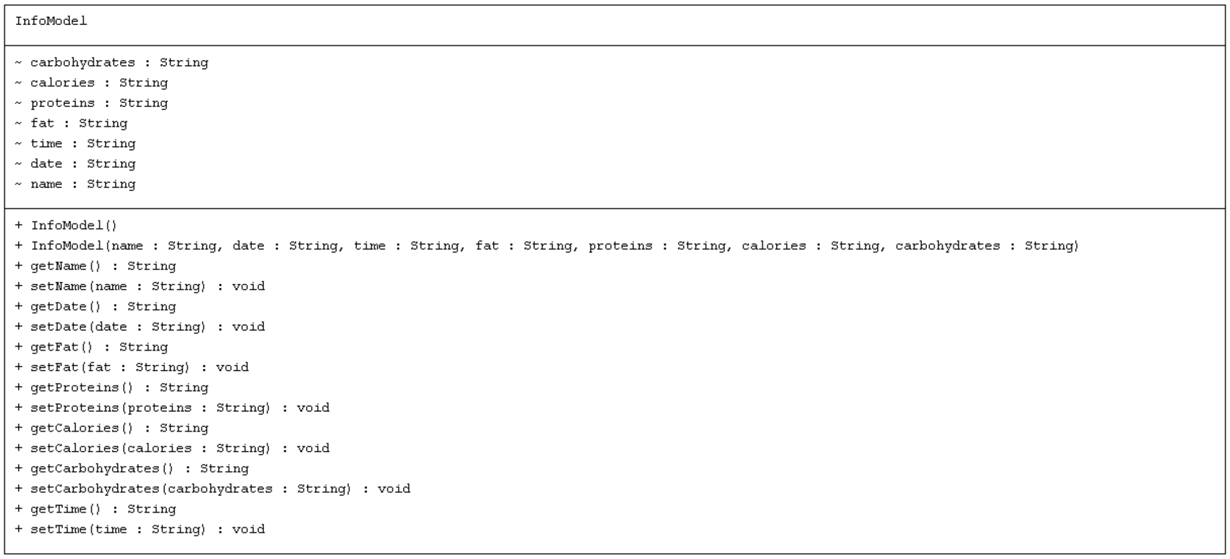

InfoModel
~ carbohydrates : String
~ calories : String
~ proteins : String
~ fat : String
~ time : String
~ date : String
~ name : String
+ InfoModel()
+ InfoModel(name : String, date : String, time : String, fat : String, proteins : String, calories : String, carbohydrates : String)
+ getName() : String
+ setName(name : String) : void
+ getDate() : String
+ setDate(date : String) : void
+ getFat() : String
+ setFat(fat : String) : void
+ getProteins() : String
+ setProteins(proteins : String) : void
+ getCalories() : String
+ setCalories(calories : String) : void
+ getCarbohydrates() : String
+ setCarbohydrates(carbohydrates : String) : void
+ getTime() : String
+ setTime(time : String) : void

Figure 28: class diagram 3

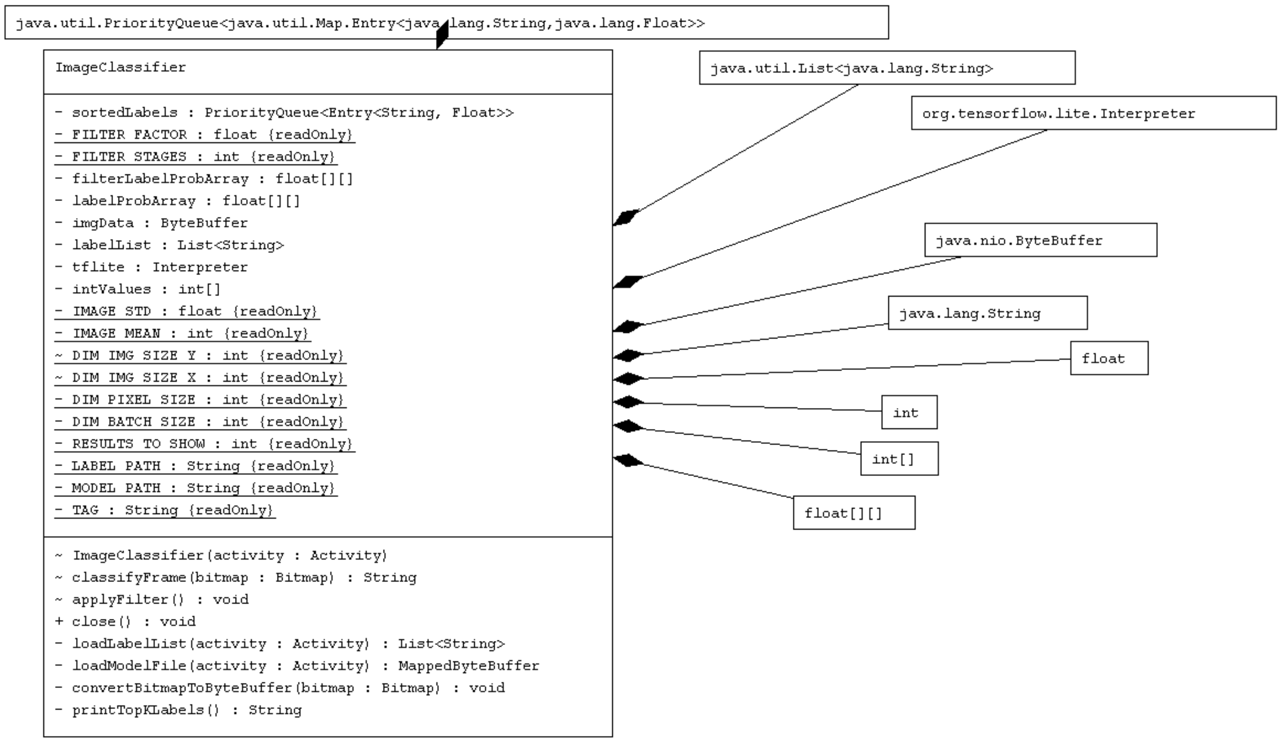

java.util.PriorityQueue<java.util.Map.Entry<java.lang.String,java.lang.Float>>
ImageClassifier
- sortedLabels : PriorityQueue<Entry<String, Float>>
- FILTER_FACTOR : float {readOnly}
- FILTER_STAGES : int {readOnly}
- filterLabelProbArray : float[][]
- labelProbArray : float[][]
- imgData : ByteBuffer
- labelList : List<String>
- tflite : Interpreter
- intValues : int[]
- IMAGE_STD : float {readOnly}
- IMAGE_MEAN : int {readOnly}
~ DIM_IMG_SIZE_Y : int {readOnly}
~ DIM_IMG_SIZE_X : int {readOnly}
- DIM_PIXEL_SIZE : int {readOnly}
- DIM_BATCH_SIZE : int {readOnly}
- RESULTS_TO_SHOW : int {readOnly}
- LABEL_PATH : String {readOnly}
- MODEL_PATH : String {readOnly}
- TAG : String {readOnly}
~ ImageClassifier(activity : Activity)
~ classifyFrame(bitmap : Bitmap) : String
~ applyFilter() : void
+ close() : void
- loadLabelList(activity : Activity) : List<String>
- loadModelFile(activity : Activity) : MappedByteBuffer
- convertBitmapToByteBuffer(bitmap : Bitmap) : void
- printTopKLabels() : String
java.util.List<java.lang.String>
org.tensorflow.lite.Interpreter
java.nio.ByteBuffer
java.lang.String
float
int
int[]
float[][]

Figure 29class diagram 4

CHAPTER 5: METHODOLOGY, TOOLS, IMPLEMENTATION & TESTING

5.1. Methodology

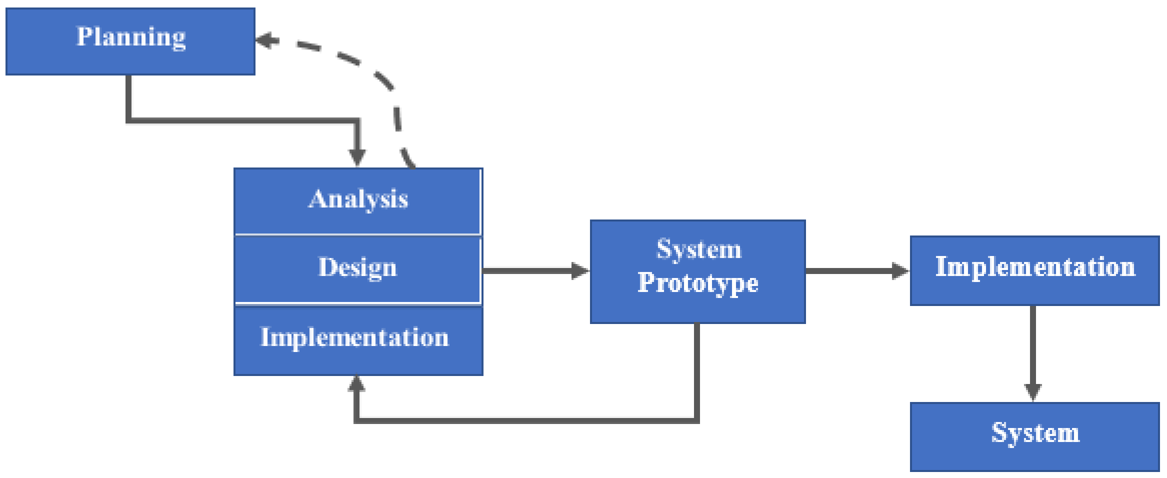


Figure 30: Prototyping methodology

Planning Phase - In the planning phase, the project is determined to take a duration of around 1 week to complete.

Analysis Phase -In analysis phase, the problem statement, project objectives and scopes are identified.

Design Phase - In design phase, the overall structure and flow of the system is determined.

Implementation Phase -In implementation phase, the actual coding is carried out according to the design chosen.

System Prototype - System prototypes are created to acquire feedback of its performance

5.2. Tools used

(i) Java

(ii) Android Studio - An office IDE designed by to build apps for every single Android device (Android Studio, 2020)

(iii) TensorFlow - Open-source machine learning framework (TensorFlow, 2020)

5.3. User requirements

• Android version higher than 5.0 or API level 21

According to minimum user requirement for use of camera2 API (Android 5.0 API, 2018)

• Minimum available storage of 150MB

According to compiled application package (.apk)

5.4. Implementation and testing

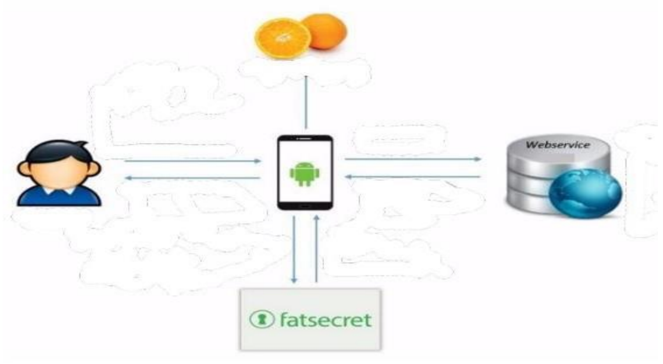
5.4.1. Model Selection

This project uses a pre-trained model on a RESTFul API provided by FatSecret as it is more accurate.

GitHub link **https://github.com/fatsecret/fatsecret4j**

5.4.1. Model Usage

Consider that the trained models are provided by the fat secret API. Thus, the app uses the models from Fat Secret API on TensorFlow for classification. Recall figure 3



To setup the API: I added the following configuration to my **`build.gradle`** file

```
repositories {
	mavenCentral()
}
android {
	defaultConfig {
		minSdkVersion 24
		jackOptions {
			enabled true
		}
	}
	compileOptions {
		sourceCompatibility JavaVersion.VERSION_1_8
		targetCompatibility JavaVersion.VERSION_1_8
	}
}
dependencies {
	compile 'com.fatsecret4j:fatsecret-platform:2.0'
	compile 'com.android.volley:volley:1.0.0'
}
```

Optionally, you may call the API from the URL `"https://platform.fatsecret.com/rest/server.api`

Supported methods are:

- food.get()
- foods.search()
- recipe.get()
- recipe.search()

Overall, the model performed well under situation between desserts that have similar appearance. However, there are circumstances where the model wrongly classified dessert when only partial portion of dessert was present.

5.5. Key functions

Image classifier function

```
/** Classifies a frame from the preview stream. */
String classifyFrame(Bitmap bitmap) {
    if (tflite == null) {
        Log.e(TAG, "Image classifier has not been initialized;
Skipped.");
        return "Uninitialized Classifier.";
    }
    convertBitmapToByteBuffer(bitmap);
    // Here's where the magic happens!!!
    long startTime = SystemClock.uptimeMillis();
    tflite.run(imgData, labelProbArray);
    long endTime = SystemClock.uptimeMillis();
    Log.d(TAG, "Timecost to run model inference: " +
Long.toString(endTime - startTime));

    // smooth the results
    applyFilter();

    // print the results
    String textToShow = printTopKLabels();
    //textToShow = Long.toString(endTime - startTime) + "ms" +
textToShow;
    return textToShow;
}
```

Fat secret get food() function

```
public JSONObject getFood(Long ab) {
    List<String> params = new
ArrayList<>(Arrays.asList(generateOauthParams()));
    String[] template = new String[1];
    params.add("method=food.get");
    params.add("food_id=" + ab);
    params.add("oauth_signature=" + sign(APP_METHOD, APP_URL,
params.toArray(template)));
    JSONObject food = null;
    try {
        URL url = new URL(APP_URL + "?" +
paramify(params.toArray(template)));
        URLConnection api = url.openConnection();
        String line;
        StringBuilder builder = new StringBuilder();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(api.getInputStream()));
        while ((line = reader.readLine()) != null)
            builder.append(line);
        JSONObject foodGet = new JSONObject(builder.toString());
        food = foodGet.getJSONObject("food");
    } catch (Exception e) {
        Log.w("Fit", e.toString());
        e.printStackTrace();
    }
    return food;
}
```

Search food function

```
public JSONObject searchFood(String searchFood, int page) {
    List<String> params = new
ArrayList<>(Arrays.asList(generateOauthParams(page)));
    String[] template = new String[1];
    params.add("method=foods.search");
    params.add("search_expression=" + Uri.encode(searchFood));
    params.add("oauth_signature=" + sign(APP_METHOD, APP_URL,
params.toArray(template)));

    JSONObject foods = null;
    try {
        URL url = new URL(APP_URL + "?" +
paramify(params.toArray(template)));
        URLConnection api = url.openConnection();
        String line;
        StringBuilder builder = new StringBuilder();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(api.getInputStream()));
        while ((line = reader.readLine()) != null)
builder.append(line);
```

```
        JSONObject food = new JSONObject(builder.toString());    //
{ first
        foods = food.getJSONObject("foods");                      //
{ second
    } catch (Exception exception) {
        Log.e("FatSecret Error", exception.toString());
        exception.printStackTrace();
    }
    return foods;
}
```

Calculate BMI function

```
private float calculateBMI (float weight, float height) {
    return (float) (weight / (height * height));
}
```

BMI Interpretation function

```
private String interpretBMI(float bmiValue) {

    if (bmiValue < 16) {
        return "Severely underweight";
    } else if (bmiValue < 18.5) {

        return "Underweight";
    } else if (bmiValue < 25) {

        return "Normal";
    } else if (bmiValue < 30) {

        return "Overweight";
    } else {
        return "Obese";
    }
}
```

5.6. Code algorithms

BMI Calculator

The body mass index (BMI), or Quetelet index, is a measure for human body shape based on an individual's mass and height. Devised between 1830 and 1850 by the Belgian polymath Adolphe Quetelet during the course of developing "social physics", it is defined as the individual's body mass divided by the square of their height - with the value universally being given in units of kg/m2 . Go-run is going to provide the function which could calculate your BMI which would show your body condition in an obvious way.

$$BMI = \frac{mass(kg)}{(height(m))^2} = \frac{mass(kg)}{(height(in))^2} * 703$$

Calorie Meter

The calories are calculated on the formula

$$Cal = 131.5 * distance + 23.5$$

The distance is the factor which is computed from the accelerometer and used for calorie meter. The computation can be found in the earlier section of the report mathematical modeling.

CHAPTER 6: CONCLUSION

6.1. Project review, discussions & conclusions

Uncontrolled and imbalanced calorie intake are both main sources of problem that are causing overweight and obesity in Kenya and across the world. Desserts played an important part of the causes of this due to its high sugar content that leads to high calorie. However, these problems are not being taken seriously enough from the people. Aside from lack of awareness of the problem, one of the causes could be difficulty in acquiring food calorie values as some local or traditional desserts are not very well documented. Traditional ways of food logging are inefficient and discouraging as it requires a lot of effort to perform.

Recent rise in popularity of deep learning had raised interest among many people in the field of machine learning. Not only it outperforms traditional machine learning techniques, it has a wide area of application in different fields. At the same time, rapid progression of smartphone development had made them much more affordable and increased capability to perform heavy computation locally.

This project proposed a mobile application that applied deep learning to acquire dessert calorie. Not only it provides an easier way to access information of food but could also serves as an educational tool for people to learn more about desserts. In a nutshell, the project aims to promote a healthy lifestyle by raising awareness among people towards the importance of balanced calorie intake. Hence, it can reduce the problem that the country is currently facing. Overall, the project has a potential to grow into a more usable tool as the classes of dessert that it is able to identify increases.

6.2. Novelties and contributions achieved

The project could be used in many cases. For instance, foreigners or tourist could use it to easily learn about local desserts without extensively digging up resources throughout the internet. While it could be served as an educational tool, it can also be used as a tool to manage calorie intake. As the country's alarming problem of obesity and overweight are arising, normal people could use it as a guidance to manage their calorie intake. Besides, this project could raise public awareness towards the impact of dessert towards our daily calorie intake. Moreover, this project could possibly eliminate traditional ways of food logging as there is no need to search for food information manually anymore. This project is unique as it also addressed local dessert where most of them are not very well documented and there is not any tool of its kind in the market yet.

6.3. Future work

There are many improvements that could be added to enhance the project in many aspects. One of them could be implementation of object detection so that multiple dessert objects can be detected and sum of calories can be computed instead. Moreover, model performance can be further improved in terms of accuracy and dessert variety. This would help to extend the usability of the project. Nonetheless, the implementation of application can also be further optimized especially the frames per second during real-time mode in order to produce a smoother user experience.

REFERENCES

- All About Web 2021, How Flexible is Python? Available from: <http://www.allaboutweb.biz/how-flexible-is-python/>
- Android 9.0 APIs 2021, Available from: <https://developer.android.com/about/versions/android-9.0.html
- Britz, D. 2015, Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs. Available from: <http://www.wildml.com/2015/09/recurrent-neural-networks- tutorial-part-1-introduction-to-rnns/>. [03 March 2021].
- Britz, D. 2015, Understanding Convolutional Neural Networks for NLP. Available from: <http://www.wildml.com/2015/11/understanding-convolutional-neural-networks.
- World Health Organization. (2011, October) Obesity Study. [Online]. http://www.who.int/mediacentre/factsheets /fs311/en/index.html
- World Health Organization. (2012) World Health Statistics 2012. [Online]. http://www.who.int/gho/publications/world_health_statistics/2012/en/index.html
- Fengqing Zhu, Anand Mariappan, Carol J Boushey, Deb Kerr, Kyle D Lutes, David S Ebert, Edward J Delp, "Technology-assisted dietary assessment", International Society for Optics and Photonics, p.p. 681411-681420, 2008.
- Bethany L Daugherty, TusaRebecca E Schap, Reynolette Ettienne-Gittens, Fengqing M Zhu, Marc Bosch, Edward J Delp, David S Ebert, Deborah A Kerr, Carol J Boushey, Novel "Technologies for Assessing Dietary Intake: Evaluating the Usability of a Mobile Telephone Food Record Among Adults and Adolescents", Published online 2012.
- P.Pouladzadeh, S.Shirmohammadi, and R.Almaghrabi, "Measuring Calorie and Nutrition from Food Image", IEEE Transactions on Instrumentation & Measurement, Vol.63, No.8, p.p. 1947 – 1956, August 2014.
- P. Pouladzadeh, S. Shirmohammadi, A. Bakirov, and A. Bulut, Abdulsalam Yassine "Cloud-Based SVM for Food Categorization", Multimedia Tools and Applications, Springer, Vol. 74, Issue 14, pp. 5243-5260.
- P.Pouladzadeh, S.Shirmohammadi, A.Yassine, "Using Graph Cut Segmentation for Food Calorie Measurement", IEEE International Symposium on Medical Measurements and applications, p.p.1-6, Lisbon, June 2014.
- Parisa Pouladzadeh, Pallavi Kuhad, Sri Vijay Bharat Peddi, Abdulsalam Yassine, Shervin

Shirmohammadi “Mobile Cloud Based Food Calorie Measurement” The 4th International IEEE Workshop on Multimedia Services and Technologies for E Health (MUST-EH), ICME, China, July 2014.

- Y. B. Yuri, G. F. Lea, “Graph Cuts and Efficient N-D Image Segmentation,”International Journal of Computer Vision, vol.70, no.2, pp.109-131, 2006.
- Y. Boykov, V. Kolmogorov, “An experimental comparison of mincut/max-flow algorrithms for energy minimization in vision,” IEEE transaction PAMI, vol.26, no.9. pp. 1124 1137, 2004.
- http://neuralnetworksanddeeplearning.com/chap2.html
- https://www.jetpac.com/
- M. Livingstone, P. Robson and a. J.Wallace, “Issues in dietary intake assessment of children and adolescents,” British Journal of Nutrition, vol. 92, p. 213–222, 2004.
- L. Bandini, A. Must, H. Cyr, S. Anderson, J. Spadano and W. Dietz, “Longitudinal changes in the accuracy of reported energy intake in girls 10-15 y of age,” The American Journal of Clinical Nutrition, vol. 78, p.p. 480–484, 2003.
- W. Luo, H. Morrison, M. d. Groh, C. Waters, M. DesMeules, E. Jones-McLean, A.-M. Ugnat, S. Desjardins and M. L. a. Y. Ma, “The burden of adult obesity in Canada,” Chronic Diseases in Canada, vol. 27, no. 4, p.p. 135-144, 2007.
- Chil, J.-H. Chen, H.-H. Chu and J.-L. Lo, “Enabling Calorie-Aware Cooking in a Smart Kitchen,” Springer-Verlag Berlin Heidelberg, vol. 5033, p.p. 116-127, 2008.
- M. S. Westerterp-Plantenga, “Eating behavior in humans, characterized by cumulative food intake curves-a review,” Neuroscience and Biobehavioral Reviews, vol. 24, p. 239–248, 2000.
- Y. Kato, T. Suzuki, K. Kobayashi, Y. Nakauchi, "A web application for an obesity prevention system based on individual lifestyle analysis," IEEE International Conference on Systems, Man, and Cybernetics (SMC), p.p. 1718 - 1723, Oct.2012.
- T. Miyazaki, G.C. De Silva, K. Aizawa, "Image-based Calorie Content Estimation for Dietary Assessment," IEEE International Symposium on Multimedia (ISM), pp.363-368, 5-7 Dec. 2011.

- H. C. n Chen, W. Jia, Z. Li, Y. Sun, M. Sun, "3D/2D model to-image registration for quantitative dietary assessment," 38th Annual Northeast Bioengineering Conference (NEBEC), p.p. 95-96, March 2012.
- M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. D. Fernstrom, M. H. Fernstrom, J. P. DeLany and R. J. Sclabassi, "Determination of Food Portion Size by Image Processing," International IEEE EMBS Conference, pp. 871 - 874, 2008.
- Zakaria Al-Battashi, John Bronlund, Gourab Sen Gupta, "Investigations Into Force Sensor Characteristics for Food Texture Measurements", IEEE International Conference on Instrumentation and Measurement Technology (I2MTC), p.p. 2089-2094, 2015.
- Y. Saeki and F. Takeda, "Proposal of Food Intake Measuring System in Medical Use and Its Discussion of Practical Capability," Springer-Verlag Berlin Heidelberg, vol. 3683, p.p. 1266–1273, 2005.
- Fengqing Zhu, Marc Bosch, Nitin Khanna, Carol J. Boushey, "Multiple Hypotheses Image Segmentation and Classification With Application to Dietary Assessment", IEEE Journal of Biomedical and Health Informatics, Vol. 19, NO. 1,pp. 377- 389, January 2015.
- Xi-Ai, C., Guang-Xin, Z., Ping-Jie, H., Di-Bo, H., Xu-Sheng, K., and Ze-Kui, Z.:Classification of the green tea varieties based on support vector machines using terahertz spectroscopy, in: Instrumentation and Measurement Technology Conf. (I2MTC), 2011 IEEE, 1–5, May, 2011.
- Brownlee, J. 2021, A Gentle Introduction to Transfer Learning for Deep Learning. Available from: <https://machinelearningmastery.com/transfer-learning-for-deep- learning/>
- Bruno, D. and Osorio, F. 2021, 'Image classification system based on deep learning applied to the recognition of traffic signs for intelligent robotic vehicle navigation purposes', 2021 Latin American Robotics Symposium (LARS) and 2021 Brazilian Symposium on Robotics (SBR), pp. 1-6.
- Caspi, G. 2021, What's the difference between deep learning and machine learning. Available from: < https://betanews.com/2016/12/12/deep-learning-vs-machine- learning/>
- Categorised Body Mass Index, 2021. Available from: <http://www.weightofthenation.org/wp-content/uploads/2016/02/bmi-overweight-obesity-men-and-women.png?3890d0>
- Deeplearning.net 2021, Welcome — Theano 1.0.0 documentation. Available from <http://deeplearning.net/software/theano/>. [03 March 2021].
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. 2021, 'MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications', pp. 1-9.

- Huang, G., Liu, Z., Maaten, L.V.D., Weinberger, K.Q. 2016, 'Densely Connected Convolutional Networks', pp. 1-9.
- Economist Intelligence Unit 2017, Tackling obesity in ASEAN Prevalence, impact, and guidance on interventions. Availablefrom:<https://www.eiu.com/public/topical_report.aspx?campaignid=ObesityInASEAN >.
- FatSecret Platform API, 2017. Available from: <https://platform.fatsecret.com/api/>. [12 August 2017].
- Floydhub 2018, FloydHub - Deep Learning Platform - Cloud GPU. Available from:<https://www.floydhub.com/>. [21 July 2018].
- Greene, T. 2018, Google brings on-device machine learning to mobile with TensorFlow Lite. Available from: <https://thenextweb.com/artificial intelligence/2017/11/15/google-brings-on-device-machine-learning-to-mobile- with-tensorflow-lite/>. [03 March 2018].
- Gunnars, K. 2017, How Many Calories Should You Eat Per Day to Lose Weight?. Available from: <http://www.healthline.com/nutrition/how-many-calories-per-day#section2>. [Accessed 14 August 2017].
- Hariadi, R. R., Khotimah, W. N. and Wiyono, E. A. 2015, 'Design and implementation of food nutrition information system using SURF and FatSecret API', 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), pp. 181-183.
- Haslam, D. and James, W. 2005, 'Obesity', The Lancet, vol. 366, no. 9492, pp.1197-1209.
- He, K., Zhang, X., Ren, S., Sun, J. 2016, 'Deep Residual Learning for Image Recognition',
- 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-777.