© June 2023 | IJIRT | Volume 10 Issue 1 | ISSN: 2349-6002

Deep Learning Instance Segmentation for Estimating the Nutritional Value of Food

Tilak Patil[1], Sohan Shrungare[2], Gaurav Lokhande[3], Govind Pole[4]
[1, 2, 3, 4] *Department of Computer Engineering, Modern Education Society's College of Engineering, Pune, Maharashtra, India*

***Abstract*— Poor dietary choices can have detrimental effects on health, impacting overall well-being. Consuming a diet containing excess of saturated fats and sugar, also overconsumption of processed foods can increase the risk of obesity and increase the susceptibility to heart disease, diabetes, high blood pressure, and certain cancers. Conversely, inadequate intake of essential nutrients such as vitamins, minerals, and fiber can lead to malnutrition, resulting in a weakened immune system, impaired growth and development, and heightened vulnerability to infections and diseases. Mental health and cognitive function can also be adversely affected by a suboptimal diet. It is crucial to maintain a balanced and diverse diet that provides nutrients required for optimal health. To address this issue, one potential solution is the daily consumption of quantified food. Recent advancements in computer vision and deep learning have facilitated the estimation of food calories from images, leading to the development of mobile applications that employ food image recognition to not only identify food items but also estimate their calorie content. This paper provides research on image-based food estimation for accurate calorie counts, evaluating aspects such as scalability, feasibility, and potential avenues for future work. This research paper introduces an image-based approach for estimating the nutrient content of food items from their images. The study explores various methodologies, including Fast R-CNN and Mask R-CNN, to achieve accurate nutrient estimation. The paper delves into the details of these methodologies and evaluates their effectiveness in this context.**

***Keywords* – Body Mass Index (BMI), Word2Vec, VGG16, Calorie Estimation, Convolution Neural Network, Recurrent Neural Network.**

I. INTRODUCTION

According to the National Institute of Health, obesity is the one of the leading causes of preventable death. Obesity is a medical condition characterized by excessive accumulation of body fat, which increases the risk of various health problems, including diabetes ,heart disease, high blood pressure, and certain kinds of cancer. It is usually defined by body mass index (BMI), which is a measure of body fat based on a person's weight in relation to their height. A BMI of 30 kg/m2 or higher is considered obese. Obesity is caused by a combination of genetic, environmental, and behavioral factors, such as poor diet, lack of physical activity, and certain medical conditions. It is a major public health concern worldwide, and its prevalence has been increasing in many countries over the past few decades.

Every year, obesity claims about a million lives. In order to tackle obesity, one must switch to eating healthy food and try to reduce the daily consumption of calories, for which one needs calculate and keep a track of calories from every day foods. But in today's busy and hectic world it is not that easy for an individual to track the amount of calories consumed by him/her. The amount of calories consumed plays a very crucial role in an individual's healthy lifestyle. In earlier time people used to track their calorie consumption with the help of charts. These traditional methods of nutrient estimation, such as manual entry or food composition tables, are time-consuming and prone to errors. These conventional methods are very tiring to follow and may lead to an unquantified meal diet. Consumption of a quantified meal helps in reducing excess fat and maintain fitness.

However, with the recent advancements in deep learning, accurate and efficient nutrient estimation from food images has become a possibility. A food identification system with the functions of detecting food and assessing nutrients might be suggested as a help to them. Food recognition systems is a system that could identify the type of food in an image that is captured with a camera. This is an idea to help the users to keep track of their calorie intake. The user can automatically record their food and calorie intake with just a snap of a photo. Numerous mobile applications have been developed to aid individuals in tracking their daily meals. Many of these applications utilize food image recognition technology, which can identify the types of food as well as estimate their calorie content. But, in the majority of cases, the amount of calories estimated are only linked to general food categories or the relative size of the food in comparison to standard serving sizes, which are typically inputted manually by the user. These applications often do not provide calorie estimates based on the actual quantity of food consumed.

In this paper, we will explore the latest developments in the use of deep learning for food nutrient estimation. We will discuss the various deep learning models and techniques that have been used for this task, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention mechanisms and implement deep learning models and improve models accuracy by improving the annotations required for model training and fine tuning the models by changing various hyperparameters.

Overall, this paper aims to provide an approach for food nutrient estimation using deep learning. It will serve as a valuable resource for researchers, nutritionists, and healthcare professionals interested in the application of deep learning in the field of nutrition and health.

II. LITERATURE SURVEY

Yanchao Lianga and Jianhua Li (2017), [1] proposed a method where they used the ECUSTFD dataset which is publicly available. They used the volume estimation method to determine the calories in a particular food. They also took into consideration the shooting conditions for the image. Once classification is performed, a Mean Average Precision obtained for R-CNN is 93%, whereas for ESVM is 75.9%. It can be stated that the methods used for the calculation of volume may affect the model accuracy.

Koichi Okamoto and Keiji Yanai [2] proposed a methodology that combines edge-based dish localization along with k-means-based dish bounding box estimation, and Grabcut to estimate the quantity of food in images. They employed pre-defined reference objects to aid in this estimation process. The results of their study were promising, with an absolute error of 50kcal. The relative error of their study was 20% which was also a good accomplishment. The time taken by the model to recognize food was approximately around 0.2 seconds. The average absolute error was found to be 52.231kcal, alongside a relative average error of 21.3%. However, there can be challenges in handling meal photos with non-uniform backgrounds.

In the study presented by, Priya Gupta and Shikha Gupta (2018) [3] mentions a super pixel-based LDC (Low-Level Descriptor) methodology that is suitable for processing constrained data. Their model employed various parameters such as Major Axis, Eccentricity, Euler Number, Solidity, Equivalent Diameter, Extent, Orientation, Convex Area, Perimeter, Area, Filled Area, etc. to predict the food item. The experimental results demonstrated that the model achieved the highest classification precision of 72.26% on the UEC-FOOD100 dataset. Additionally, the framework attained a classification accuracy of 78% for the order of food items. A significant enhancement desired in this research is the incorporation of automatic calorie estimation based on the food type.

Takumi EGE and Keiji Yanai (2018) [4] presented a method utilizing Multi-task CNNs for food ingredient recognition. The approach involves converting food ingredient information into real-number vectors using Word2Vec for the purpose and employing these vectors for CNN training. The architecture of the multi-task CNN is based on VGG16. It was demonstrated through the experimental results that the independent single-task CNNs along with image-search-based calorie estimation method were outperformed by multi-task CNNs as proposed by Miyazaki et al. As a potential future direction, the framework could incorporate segmentation or volume estimation based on multiple views of the food as techniques to further enhance accuracy. Additionally, the utilization of multiple view-based volume estimation has the potential to improve the overall model performance.

Pouladzadeh, Parisa and Pallavi Kuhad, among others, introduced a novel approach in their study conducted in 2016 [5].Their method integrates image processing techniques to identify individual and mixed food objects, utilizing deep learning and support vector machine (SVM) algorithms. Calorie estimation is performed through finger-based measurement and distance calculation methods. Additionally, they propose a block resize method that combines the measured distances and recognized food object names to enhance the accuracy of calorie estimation. Their approach achieves an impressive accuracy rate of 97%.

Shaikh Mohd. Wasif and Swapnil Thakery, along with their colleagues, proposed an innovative method in their 2019 study [6].Their approach involves taking an input image from the user, which will be then processed using a faster R-CNN model. After image detection is performed, they applied the grab cut algorithm for image segmentation. This segmentation step separated the food item from the background. To estimate the calorie content, they calculated the volume of the identified food item and based on the volume of food they determined the corresponding calorie count. The researchers successfully integrated all the aforementioned components into a cohesive software system specifically developed for accurately estimating calorie content based on food images. The experimental results demonstrated an impressive accuracy rate of 90%.

Several studies have proposed deep learning-based systems for food recognition. In order to achieve high accuracy in classifying various types of foods, Pouladzadeh and Shirmohammadi (2017) [7] built a mobile multi-food recognition system utilising a pre-trained VGG16 model that was refined using their food dataset. Similarly, Singla et al. (2016) [8] introduced a system for classifying images into food and non-food categories, as well as categorizing different types of food using a pre-trained GoogleNet model. Their approach demonstrated excellent accuracy in both tasks. Kagaya et al. (2014) [9] proposed a two-stage approach for food detection and recognition using a CNN, achieving high accuracy on a dataset of 32 types of foods. These studies demonstrate the effectiveness of deep learning for food recognition tasks and highlight the importance of fine-tuning pre-trained models for specific food datasets.

Ege et al. (2019) [10] suggested an image-based technique for determining the actual size of food items to increase the precision of food calorie calculation. Harshitha et al. (2020) [11] and Suma and Bharathi (2020) [12] also proposed food recognition and calorie estimation systems using computer vision and image processing techniques. They used deep learning models to recognize different types of foods and estimate their calorie content. The authors evaluated their systems on different datasets of food items and achieved high accuracy. Vishnu et al. (2020) [13] focused on fruit recognition and calorie management and proposed a deep learning-based system that can recognize different types of fruits and estimate their calorie content. These results show that deep learning and image-based algorithms are successful for calorie calculation and food recognition, which has important ramifications for food monitoring and dietary control.

III. DATASET

For training we have used FIDS30 dataset [14]. FIDS30 is a fruit dataset is a publicly available dataset of images of 30 different types of fruits commonly found in the market, captured using a smartphone camera. The FIDS30 dataset contains 30 classes of fruits, with each class containing 100 images. The fruits included in the dataset are: cherries, grapefruits, pears, papayas, blueberries, apricots, kiwis, figs, peaches, limes, coconuts, pomegranates, lemons, blackberries, watermelons, plums, avocados, nectarines, pineapples, tangerines, raspberries, strawberries, tomatoes, oranges, bananas, mangos, apples, cantaloupes, and white grapefruits. The images in the dataset were captured under different lighting conditions and with different backgrounds, making it a challenging dataset for image classification tasks.

For the testing phase, we created a custom dataset by gathering random images from the internet for each fruit category listed above. The dataset comprises 9 images for each fruit category, providing a diverse set of samples to evaluate the performance of our model. By using a custom dataset, we can ensure that our model can accurately detect and classify different fruits, even when presented with new and previously unseen images.

IV. PROPOSED METHODOLOGY

1. DATA ANNOTATION

GROUNDING DINO and SAM are two popular tools used in computer vision to aid in object detection and annotation. GROUNDING DINO is a tool that helps to standardize annotation by providing a graphical user interface (GUI) for labelling images GROUNDING DINO and SAM are two popular tools used in computer vision to aid in object detection and annotation. GROUNDING DINO is a tool that helps to standardize annotation by providing a graphical user interface (GUI) for labelling images in the Pascal VOC format. It allows users to annotate objects in images by drawing bounding boxes around them and specifying the object category. GROUNDING DINO also provides tools for editing annotations and exporting them in various formats. SAM (Semantic Annotation Management) is another annotation tool that is commonly used in computer vision research. It provides a web-based interface for annotating images and supports multiple annotation formats, including Pascal VOC and COCO. After the annotations have been made using either GROUNDING DINO or SAM, the next step is to convert the annotations and bounding box into the COCO format. COCO (Common Objects in Context) is a widely used format for object detection tasks that provides a more comprehensive annotation scheme than Pascal VOC. COCO annotations include object segmentation masks, which provide a more accurate representation of object boundaries. The conversion process involves mapping the annotation categories and bounding boxes to the corresponding COCO format. This is done using tools manually by writing custom scripts. Once the annotations have been converted, they can be easily integrated into popular object detection frameworks like Detectron.

After converting the annotations from Pascal VOC format to COCO format, the next step is to save the datasets into the Detectron dataset catalogue. The Detectron dataset catalogue is a standard way of organizing datasets for object detection tasks using Detectron2, a popular open-source object detection library. The dataset is divided into two distinct sets: a training set and a testing set. The training of object detection model is performed over the training set, whereas the testing set is utilized to assess and evaluate its performance.

2. MODEL BUILDING

2.1. DATA AUGMENTATION

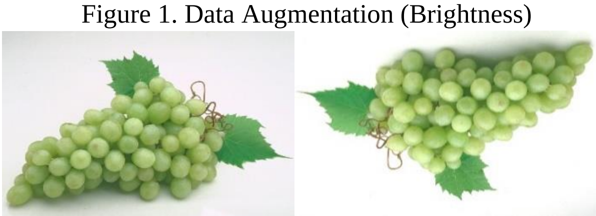
Data augmentation is a deep learning strategy that entails generating new training examples that are variants of the original pictures in order to artificially increase the size of the training dataset. This makes the model more generalizable and prevents overfitting. The following are the details of the data augmentation process used in this system:

- Resize((800,600))
- RandomBrightness(0.8, 1.8)
- RandomContrast(0.6, 1.3)
- RandomRotation(angle=[90, 90])
- RandomFlip(prob=0.5,horizontal=False, vertical=True)

These data augmentation techniques are applied to each image in the training set before it is fed into the model for training. By using these techniques, the model is better able to generalise to new photos and learn more robust Characteristics

(a) (b)

Figure 1. Data Augmentation (Brightness)



(a) (b)

Figure 2. Data Augmentation (Rotation)

2.2. MASK R-CNN

Mask R-CNN is a deep learning model that extends the popular Faster R-CNN model, which is widely used for object detection and instance segmentation tasks Mask R-CNN expands upon the Faster R-CNN framework by incorporating an additional branch dedicated to predicting object masks alongside the existing branch responsible for bounding box detection. This means that in addition to detecting the object and its location, Mask R-CNN is also capable of generating a binary mask that outlines the object in the image. During training, the model learns to predict both bounding boxes and masks simultaneously using a multi-task loss function.

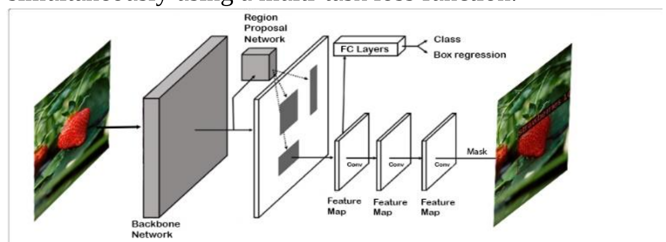


Figure 3. Network Architecture of MASK-R-CNN.

Mask R-CNN has achieved state-of-the-art performance on a number of benchmark datasets for instance segmentation, including COCO (Common Objects in Context) and Cityscapes. The model architecture used for training is the MASKRCNN R50 FPN 3x.yaml, which is a state-of-the-art object detection algorithm. The model is pre-trained on a large dataset, and its weights are fine-tuned on the FIDS30 dataset. In order to diversify the training data and strengthen the model's robustness, data augmentation techniques are used during training. The hyper parameters are set to a learning rate of 0.0025.

2.3. NUTRITIONAL VALUE ESTIMATION

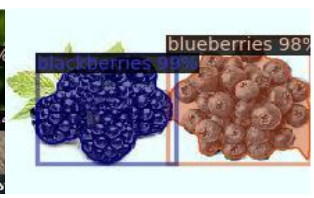
After the fruit is detected, its predicted label is matched with the corresponding nutritional values from a pre-prepared dataset. The dataset contains nutritional values for each fruit category, including calorie content, macronutrients such as Calories, Carbohydrates, Protein, Fat, Fiber, and micronutrients such as Vitamin C, Vitamin A, Potassium, Calcium, Iron, Magnesium, Phosphorus, Zinc, Vitamin E, Folate, Vitamin K, Niacin, Riboflavin, Vitamin B12, Vitamin B6, Vitamin D, Thiamin, Sodium (mg). The system outputs the nutritional values of the detected fruit by calculating the new nutritional values based detected area of the fruit, providing users with information about the fruit's nutritional content. This information can be used to make informed decisions about their diet and improve their overall health.

V. TRAINING

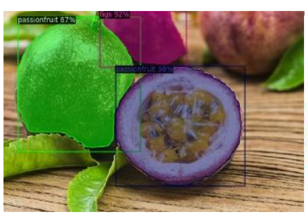
In this project, the training is performed using the COCO evaluator, which is a standard tool for evaluating object detection and instance segmentation models. Object detection, segmentation, and captioning on a broad scale are all part of the COCO (Common Objects in Context) dataset that is commonly used for training and evaluating computer vision models. The COCO evaluator computes the average precision (AP) for both instance segmentation and bounding box detection, which are two common evaluation metrics used in object detection tasks. The AP is a measure of the accuracy of the model in detecting objects in an image, taking into account both precision and recall.

The AP is calculated separately for each category of fruit in the dataset, and the average of all the AP values is reported as the overall performance of the system. A higher AP indicates better performance, with a maximum value of 1.0 indicating perfect detection and segmentation.

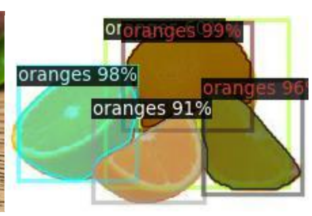
(a)



(b)



(c)



(d)

Figure 4. Training Results

The graphical representations illustrate the training accuracy and loss progression for various models, such as Fast R-CNN and Mask R-CNN, in the context of bounding box detection. Upon analysing the graphs, it becomes apparent that there is a notable convergence point where both accuracy and loss reach their optimized values. This convergence point occurs around the 13,000-iteration mark for all the models. This significant observation highlights an intriguing relationship between accuracy and loss during the training process. As the iterations progress, the models undergo refinement and learning, resulting in an improvement in accuracy, which refers to the models' ability to correctly identify and localize objects within bounding boxes. Simultaneously, the loss, which represents the discrepancy between predicted and ground truth bounding boxes, steadily decreases.

The convergence point at approximately 13,000 iterations signifies the stage at which the models have achieved a desirable level of accuracy while effectively minimizing the loss. This suggests that the models have undergone sufficient training and optimization, leading to a balanced trade-off between accurate predictions and minimized errors.

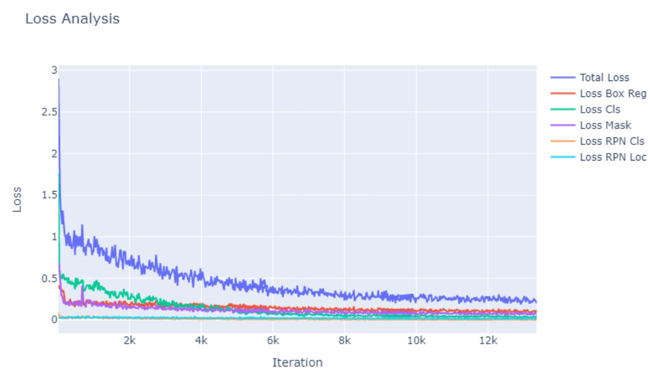


Figure 5. Loss Analysis

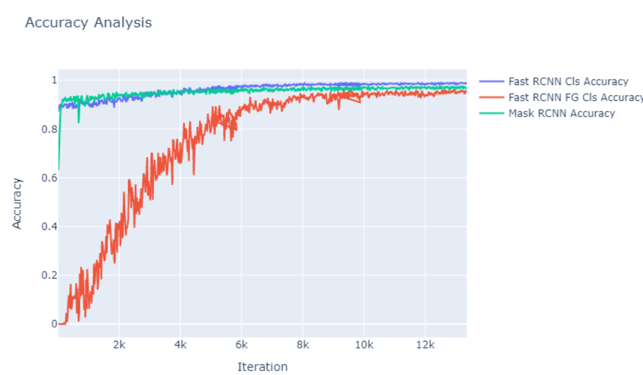


Figure 6. Accuracy Analysis

Examining the graphs, it is evident that the training accuracy reaches an impressive level of 80%, while the loss is minimized to a value of 0.19. These metrics serve as crucial indicators of the models' performance during the training process. The accuracy of 80% demonstrates the models' ability to correctly detect and localize bounding boxes for objects with a significant degree of precision. Additionally, the low loss value of 0.19 signifies the models' success in minimizing the discrepancy between predicted and ground truth bounding boxes, reflecting their effective learning and optimization.

VI. TESTING AND EVALUATION

The testing phase involved evaluating the system's performance on a manual dataset consisting of 9 samples for each category. The results showcased promising accuracy rates for both masking or segmentation (82-83%) and bounding box detection (87%) tasks, indicating the system's ability to accurately identify and localize objects within images. The achieved classification precision level of around 67-68% demonstrates the system's proficiency in assigning the appropriate category to the objects. These findings validate the system's robustness and reliability in processing and analysing the manual dataset, highlighting its potential for practical applications involving object identification, localization, and categorization.

In addition to evaluating the system's performance on the manual dataset, GUI testing was conducted to ensure the proper display and functionality of all frontend components. The purpose of this testing was to verify that the graphical user interface (GUI) accurately presented the system's features and functions. During the GUI testing phase, all components were thoroughly examined, and it was determined that they were functioning correctly. The frontend successfully displayed all the necessary elements, such as buttons, menus, input fields, and result displays, without any issues or discrepancies.

Furthermore, the GUI was tested with different user inputs to assess its ability to handle various scenarios. The system appropriately processed and responded to the different inputs, demonstrating its robustness and adaptability. Based on these outcomes, the GUI testing was declared successful.

VII. RESULTS

The user interface (UI) of the food nutrient value estimation system utilizing Deep Learning provides valuable outputs in the form of nutrient values for the analysed food items, accompanied by the Body Mass Index (BMI) of the user. This combination of information serves as a comprehensive tool for assessing the nutritional content of consumed foods and understanding its potential impact on an individual's overall health. These nutrient values encompass crucial macronutrients such as fats, carbohydrates and proteins, along with other micronutrients that include minerals and vitamins. This detailed breakdown allows users to gain insights into the specific nutritional composition.

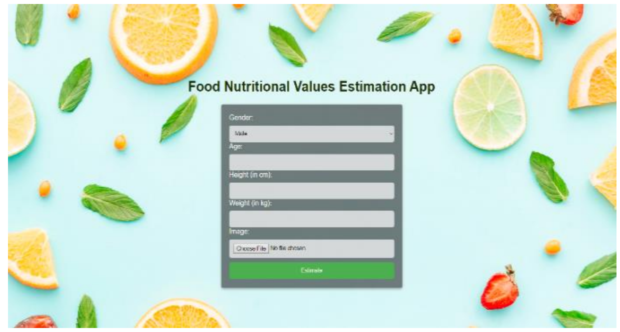


Figure 7. Image Upload

The BMI is derived from the user's provided information, such as weight, height, age, and gender. The BMI serves as an indicator of the user's body composition and can provide insights into their overall health status and potential risk factors associated with weight and body fat levels.

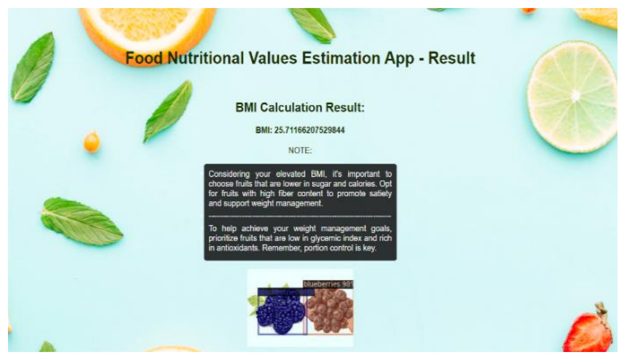


Figure 8. BMI suggestion

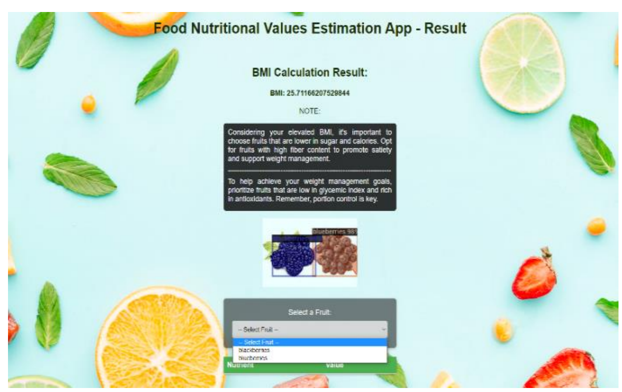


Figure 9. Dropdown for multiple food-items

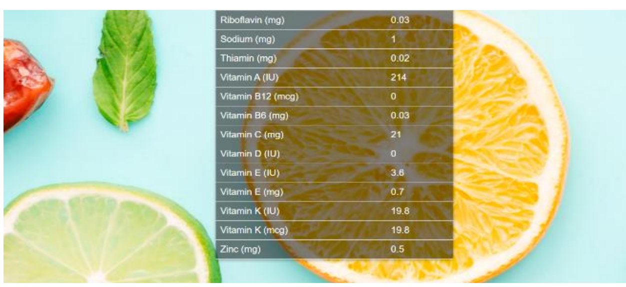
The UI effectively presents these outputs to users in a clear and user-friendly manner. The nutrient values and

BMI are prominently displayed, allowing users to quickly grasp the nutritional content of their food choices and understand the potential implications for their overall health and well-being. The GUI provides the user with tips related to his BMI, also it provides the user with a drop-down box to select from, if multiple food items are detected.

The system effectively displays comprehensive nutritional information to users in a clear and understandable manner. All macronutrients and other relevant components are included, and the nutrient values are presented in a well-structured table format. This organized display allows for easy reference and informed decision-making regarding dietary choices. Users can readily access and comprehend the breakdown of carbohydrates, proteins, fats, vitamins, minerals, and other essential nutrients.

Nutrient	Value
Calcium (mg)	29
Calories (kcal)	43
Carbohydrates (g)	9.6
Fat (g)	0.4
Fiber (g)	5.3
Folate (mcg)	35
Iron (mg)	0.6
Magnesium (mg)	20
Niacin (mg)	0.6
Phosphorus (mg)	22
Potassium (mg)	162
Protein (g)	1.4

(a)



Riboflavin (mg)	0.03
Sodium (mg)	1
Thiamin (mg)	0.02
Vitamin A (IU)	214
Vitamin B12 (mcg)	0
Vitamin B6 (mg)	0.03
Vitamin C (mg)	21
Vitamin D (IU)	0
Vitamin E (IU)	3.6
Vitamin E (mg)	0.7
Vitamin K (IU)	19.8
Vitamin K (mcg)	19.8
Zinc (mg)	0.5

(b)

Figure 10. Displaying nutrient values.

VIII. CHALLENGES AND ISSUES

We close by considering several challenges and issues to the current system: Variation in food appearance is one of the major challenges. The same type of food can look different depending on factors such as lighting, the angle of the camera, and the cooking method. These variations can make it challenging for Deep Learning models to accurately estimate calorie intake from food images. Also, every image may differ concerning its background, surroundings, distance from the camera, etc. Hence, estimating the calorie content of a food item requires an accurate measurement of its portion size. However, estimating portion sizes from images can be challenging, especially when food items are overlapping or presented in a crowded setting.

There are several Deep Learning models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants, that can be used for food calorie estimation from images. However, selecting the most suitable model for a specific application can be challenging and requires careful consideration of the trade-offs between accuracy, computational efficiency, and interpretability. Privacy also pops up as a major concern due to the fact that food images are personal data, and there may be privacy concerns related to their collection and use for calorie estimation. Careful consideration of ethical and legal issues related to privacy is necessary. The method shows non-uniformity as food recipes vary widely across regions. Also, it is a very challenging task to identify the ingredients present in any food item solely on the basis of an image containing the food item. We need to know all the ingredients and the quantity of the same in the food. Also, similar-looking food ingredients create ambiguity for the model, making it difficult to correctly identify the food items, which in turn may lead to imprecise results.

IX. CONCLUSION

In conclusion, this research paper has highlighted the significant advancements made in the field of food calorie estimation using deep learning techniques. From the studies and research discussed, it is evident that deep learning models, trained with approximately 13,000 iterations, can accurately estimate food calories from images, achieving a training accuracy of 80% and minimizing the loss to a value of 0.19. This convergence point signifies the stage where the models have achieved optimal performance, striking a balance between accuracy and loss. These findings demonstrate the effectiveness of deep learning models in accurately estimating food calories with high precision and efficiency.

The integration of deep learning with computer vision techniques has facilitated the development of various applications, including calorie-tracking mobile applications, which greatly assist individuals in maintaining a healthy lifestyle. However, it is important to acknowledge the challenges that still need to be addressed. These challenges include obtaining images with clear and uniform backgrounds, addressing portion size estimation, handling recipe variations, and improving ingredient recognition. These factors significantly impact the accuracy of food calorie estimation and require further investigation and innovation.

Furthermore, future research should focus on enhancing the generalization of deep learning models to different cuisines and food types. Ensuring that the models can accurately estimate calories across a diverse range of foods is essential for their practical implementation and widespread adoption. By addressing these challenges and advancing the field, deep learning-based food calorie estimation from images holds tremendous potential to revolutionize the way we track and manage our diets, ultimately promoting healthier lifestyles.

REFERENCES

[1] Y. Liang and L. Jianhua, "Computer Vision-Based Food Calorie Estimation: Dataset, Method, And Experiment," 2018.

[2] K. Okamoto and K. Yanai, "An Automatic Calorie Estimation System Of Food Images On a Smartphone".

[3] P. Gupta and S. Gupta, ""Hybrid Algorithm For Food Recognition, Calorie Estimation & Dietary Enforcement"," 2018.

[4] T. Ege and K. Yanai, ""Image-Based Food Calorie Estimation Using Recipe Information"," 2018.

[5] P. Pouladzadeh, P. Kuhad, B. Peddi, A. Yassine and S. Shirmohammadi, "“Food calorie measurement using deep learning neural network," pp. 1-6, 2016.

[6] S. M. Wasif, S. Thakery, A. Nagauri and S. I. Pereira, ""Food Calorie Estimation Using Machine Learning And Image Processing"," 2019.

[7] P. Poladzadeh and S. Shirmohammadi, ""Mobile Multi-Food Recognition Using Deep Learning"," 2017.

[8] A. Singla, L. Yuan and T. Ebrahimi, ""Food/Non-Food Image Classification And Food Categorization Using Pre-Trained GoogleNet Model"," 2015.

[9] H. Kagaya, K. Aizawa and M. Ogawa, ""Food Detection And Recognition Using Conventional Neural Network"".

[10] T. Ege, Y. Ando, R. Tanno, W. Shimoda and K. Yanai, ""Image-Based Food Calorie Estimation Of Real Food Size For Accurate Food Calorie Estimation"," 2019.

[11] D. S, H. S. V, M. D. Varma and M. P, ""Food Classification And Calorie Estimation Using Computer Vision Techniques"," 2018.

[12] S. Suma, M. Bharathi, G. Harikumamr, B. Deepak and G. Har, ""Food Recognition And Calorie Estimation Using Image Processing"," 2020.

[13] V. H. S, S. B, P. A, A. K and P. G, ""Fruit Recognition System For Calorie Management"," 2020.

[14] S. Marko, "Automatic fruit recognition using computer vision," 2013.

`