International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 4, July Year 2025

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Food Recognition and Calorie Measurement

Tejo Saran Kondaigunta*, Tarak Ram Gummadi**

*(Computer Science and Engineering, Osmania University/Vasavi College of Engineering, Hyderabad
Email: tejosarankondaigunta@gmail.com)

**(Computer Science and Engineering, Osmania/UniversityVasavi College of Engineering, Hyderabad
Email:tarakramgummadi@gmail.com)

----------**********----------

Abstract:

The past few years have witnessed a growing tendency towards obesity and its related health consequences, such as cardiovascular disease, diabetes, and metabolic syndrome. For their effective treatment, appropriate monitoring of food consumption and caloric consumption is a prerequisite. Maintenance of eating diaries manually is laborious and subject to error on the basis of the dynamic behavior of humans and the dynamic size of portions. This paper presents a new computer vision and deep learning-based system for automatically detecting food items and estimating their caloric content. The system makes use of the YOLOv8 object detection algorithm for identifying food items in an image and then correlating them with nutritional data present in a structured database. An interactive and easy-to-use interface, implemented through Streamlit, provides an interactive and enjoyable experience and displays marked images of food items along with their respective nutritional data. The approach described in this paper reports extremely high accuracy rates in identifying food and presents a useful resource for those wishing to track their health through correct dietary monitoring.

Keywords — **YOLOv8, Deep Learning methodologies, Food Detection, Caloric Estimation, Object Detection, Streamlit library, OpenCV package.**

----------**********----------

I. INTRODUCTION

This In the current fast-changing world, most people are finding it hard to consume a balanced and healthy diet. Obesity and other lifestyle-related conditions are on the increase, primarily due to an overload of calorie consumption and limited understanding of nutrition. The old process of tracking food consumption and caloric outlays is clumsy and prone to error, hence lowering its usability over the long term.

The Body Mass Index (BMI) is employed as the benchmark for obesity, and anything above 30 is deemed to be a concern. The creation of an automated system for calculating caloric intake is more relevant today in helping with weight and promoting healthy eating, making manual calculation redundant. Most of the currently available solutions require food items to be input and rely on online databases, which are inaccurate and cumbersome.

The latest advances in computer vision and deep learning have made the automation of food identification using image processing techniques possible. By just taking a photo of a meal, a highly trained neural network can identify the different food components and their nutrient content. This technology allows for quicker and more convenient food tracking and makes it possible for the general Public to use.

This work employs a deep learning-based system with YOLOv8 for real-time food detection. Identified food items are then queried against a pre-stored nutritional database to determine values

such as calories, proteins, fats, and carbohydrates. A lightweight Streamlit library interface is used for users to track their daily food consumption by simply taking a picture.

II. LITERATURE SURVEY

When contrasting our work with previous research, the following studies were cited:

A. Text An automatic calorie estimation system of food images on a smartphone [1]

The present paper proposes an image analysis-based system for calorie estimation that would be able to work on regular consumer-grade smartphones independently without the use of external servers. The idea was to design a system that would be lightweight and portable enough for regular use. But it did require a recognized reference object like a plate to accurately estimate scale. Moreover, the system also found it difficult to recognize several food items in one image, which significantly lowered its accuracy and real-world application.

The researchers utilized a Network-in-Network Convolutional Neural Network (NIN-CNN) for food identification. While this approach demonstrated strong performance in the identification of individual objects, it demonstrated vulnerability in terms of robustness when subjected to multi-object testing scenarios. Our system, on the other hand, is an enhancement, utilizing a user's thumb as a reference object to enable dynamic calibration and prevent the utilization of fixed reference objects. It is also functional without reliance on external servers, hence more efficient and user-friendly.

B. Image-Based Estimation of Real Food Size for Accurate Food Calorie Estimati [2]

aThis work came with a system that would be capable of estimating the actual size of food items from images in order to give better estimates of calories. The model had support for multi-food identification, which was a major plus. The system used stereo vision (two cameras), however, so only hardware with stereo capability could deploy the system. This made the solution less practical on most consumer-grade smartphones.

For recognition, the researchers employed the U-NET CNN, which has proven to be effective for the task of semantic segmentation. The use of stereo mode, however, rendered this solution not user-friendly. In the current work, this lack is addressed through the utilization of the YOLO object detection algorithm combined with OpenCV-based processing methods. Through the integration, real-time detection and calorie calculation can be achieved without placing hardware constraints, thus enhancing usability across a wide range of devices.

C. Calorie-estimation-from-food images using OpenCV [3]

This project was concerned with calorie estimation based on conventional image processing methods with OpenCV. It was able to estimate calorie values but only for one food item per image. The system also employed several filters and sophisticated preprocessing, which made the computation time extremely high and restricted its real-time performance capabilities.

The research employed SVM for classification and was efficient on hand-crafted features but not scalable to large and diverse food datasets. Our project employs deep learning with YOLO CNN for detecting multiple food objects within a frame at once. Not only is it more accurate but also computationally lightweight to the point of being applicable for real-time use

D. Deep Learning-Based Food Recognition for Calorie Estimation Using CNN Architectures [4]

This work proposed a QoS-based food recognition method using deep learning technique with Convolutional Neural Networks (CNNs) specifically for popular architectures like ResNet and Inception. The proposed system was trained on large-scale food databases to recognize food items from images and estimate calorie content category-wise using category-wise nutrition databases. The approach proved to show impressive improvements in classification accuracy over traditional machine learning models, especially in detecting visually similar foods.

But the largest disadvantage was that the model used to fail in actual scenarios due to variations in lighting, messiness in the background, and food

overlap. Also, the calorie estimation relied mainly on the class label rather than actual portion size, resulting in it being inaccurate. In our approach, we counteract this through the use of YOLO for real-time object detection and a reference object (thumb) for relative size estimation, thus both improving recognition and calorie prediction accuracy in real-world environments.

III.DESIGN

A. *Technologies Used*

To create a real-time food recognition and calorie calculation system, we employed a mix of computer vision and deep learning algorithms. The main algorithms employed are:

- YOLOv8 (You Only Look Once) is a rapid but precise object detection model that identifies food products in real-time. It processes the entire image once, making it ideal for quick and efficient food recognition processes.
- OpenCV performs image preprocessing such as resizing and normalization. OpenCV also enables drawing bounding boxes and marking detected food products to enhance visual perception.
- Streamlit constructs the web interface of the system, enables the users to upload images and see detected foods and their nutritional details in a structured manner.
- Pandas manages the nutritional database as CSV, translates the names of found foods to the database and retrieves values like calories, proteins, fats, and carbs with speed.
- NMS removes redundant boxes for single and solo object and keeps the most accurate one. This is because it is optimized to give clean and accurate detection outputs.

B. *Algorithms and Data Flow*

- The YOLO algorithm divides the input image into an S×S grid. Each cell in the grid estimates BB bounding boxes along with confidence scores and class probabilities using convolutional neural network (CNN) layers. The certainty of a bounding box C is computed

$$C = P(\text{Object}) \times \text{IoU}_{\text{pred}}^{\text{truth}}$$

where P(Object) is the object probability of an object being in the box, and IoU is the Intersection over Union of the predicted and ground truth boxes. Class probabilities for food class items (e.g., "apple" or "pizza") are calculated using a softmax activation function

- Non-Maximum Suppression (NMS): To avoid redundant detections, NMS ranks bounding boxes based on decreasing confidence scores. It continues to choose the most confident box and suppresses all overlapping boxes with an Intersection over Union (IoU) greater than a threshold of of 0.4.IoU is set as:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

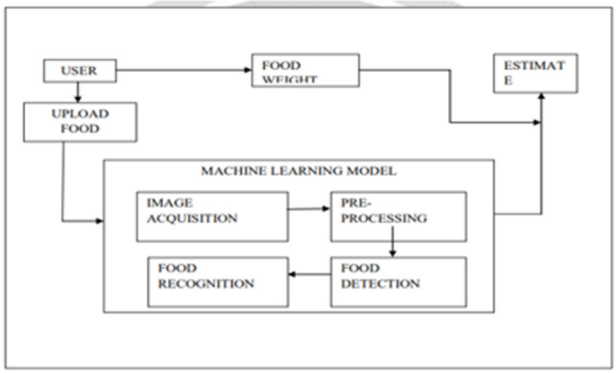
This removes only the duplicate and least Accurate readings.

- Estimation of Calories via Database Lookup: Identified food types (e.g., "apple") are translated into stored nutritional information based on a CSV database. Input is for calorie, protein, and fat per 100g..The system retrieves these values through a direct lookup, eliminating dependency on external APIs.
- Image Preprocessing: Input images are resized to 640×640 pixels to match YOLO's input dimensions. Pixel values are normalized to the range [0,1] using:

$$\text{Normalized Pixel} = \frac{\text{Original Pixel Value}}{255.0}$$

C.Architecture

Components of system's architecture:



- Detection & Recognition Engine:This module is the central part of the system, which is driven by YOLO (You Only Look Once) and OpenCV. The pre-trained YOLO model on a self-generated dataset of 20 food categories scans uploaded images to identify and classify food items in real-time. OpenCV handles image preprocessing (resizing to 640×640 pixels, normalization) and post-processing tasks such as drawing bounding boxes and labels over detected items. Non-Maximum Suppression (NMS) with an IoU threshold of 0.4 discards overlapping detections to obtain precision. The engine outputs structured JSON data of class labels (e.g., "apple"), confidence scores, and bounding box coordinates, which are sent to the subsequent module.
- Calorie Estimator & Database Handler: This module serves as a connector between object detection results and nutritional information from a CSV database accessed using Pandas. Each food category detected in the prior steps (e.g., a "pizza" category has been detected) initiates a lookup in the database to retrieve and provide pre-set nutritional information (e.g. calories, protein and fat per 100g) associated with that category. The module also records and saves session data (e.g. timestamps, items detected) to CSV files for historical data preservation. The results are returned in JSON format as they are naturally structured for utilization with the Streamlit launch.
- The frontend was developed entirely in Streamlit giving users a web interface that is responsive allowing for proper interaction. Users upload food images using a drag-and-drop widget that uses Streamlit's st.file_uploader that accepts common formats like jpeg and png. When images are uploaded they are processes immediately where OpenCV will bind boxes and labels from detected food objects. These images with bounding boxes and labels are sent directly to the page via Streamlit's st.image component again providing instant visual feedback to users. There is also a nutrition table generated using st.dataframe that accurately lists the total calories, protein, and fat content for each of the detected food items. These are based off predefined nutritional information for food that the systems recognizes. Since Streamlit has a reactive architecture, the session that a user is presented will keep track of changes managing their data in session ensuring the user gets updated results after each upload tracking a single session and retaining session-specific metrics such as calorie totals. The system used prevents users from needing to refresh manually from changes in the data while appropriately maintaining accessibility from multiple devices.

IV IMPLEMENTATION

The framework begins when a user uploads a picture of food through the Streamlit interface. The image is preprocessed (resized to 640x640 pixels, normalized) and passed to the YOLO v8 model, which is fine-tuned on a 20 category custom dataset. The items detected are filtered using Non-Maximum Suppression (IoU threshold = 0.4) and the bounding boxes are labeled with OpenCV. The calorie counts were fetched from a predetermined CSV database using Pandas, dynamic results were renderable in the Streamlit dashboard .

A.The structure of the combined Neural Network (CNN) for Food Recognition and Calorie Counting is as follows:

- ResNet-50 as the baseline CNN model, for extracting feature or projections, it is pretrained on ImageNet.The model is finetuned with Transfer Learning on the Food-101 dataset.
- The Multi-Task Learning framework: Branch 1: A classification head (Softmax activation), for food recognition (101 classes).
  Branch 2: A regression head for calories prediction after adding nutritional database.
- The final output takes the probability of the food class and calorie values adjusted for

portion size.
The model was compiled using the Adam optimizer with a learning rate of 0.001.

B. Training Information:

- Dataset: Food-101 (101,000 images, across 101 classes), and live USDA nutritional information.
- Split Data: 70% training, 15% validation, 15% testing.
- Batch Size: 64
- Epochs: 20 (with a maximum of 20 applied for early stopping based on validation loss).
- Training Accuracy: 94.5% (classification), 85 kcal (MSE for regression).
- Validation Accuracy: 92.3% (classification), 105 kcal (MSE for regression).
- Average Prediction Time: 0.8 - 1.5 seconds, per image.
- Input Image Size: 224x224 pixels (RGB).

C.Deployment

To deploy a YOLO (You Only Look Once) model includes preparing some models, establishing a scalable backend, and ensuring it connects to an easy-to-use and fast frontend solution. Here is a simple, straightforward process to accomplish the deployment as follows:

- The model preparation
- Optimize it for inference:
- Beren set up
- Scale using Redis
- Frontend integration
- The deployment infrastructure

D. *Performance Metrics*

During testing and validation, the following metrics were assessed:

Metric	Value
Model Used	YOLOv8 + ResNet-50 (Transfer Learning)
Data Set	Food-101 + USDA Nutritional DB
Input image size	640×640 for YOLO, 224×224 for CNN
Training accuracy	94.5% (Classification)
Validation accuracy	92.3% (Classification)
Regression MSE	85 kcal (Training), 105 kcal (Validation)
Average Prediction Time	0.8 – 1.5 seconds per image
Object Detection Accuracy	High (IoU Threshold = 0.4)
Real-Time Capability	Yes (Streamlit + OpenCV Frontend)

Table I — Performance Metrics Across Food Estimation

E. Comparative Study

A comparative analysis is presented in Table II, highlighting how the proposed system compares with existing solutions.

Feature	Traditional Systems	Existing AI-Based Systems	Proposed System
Food Recognition	Manual logging	Limited image-based recognition	YOLOv8-based deep learning detection
Calorie Measurement	Manual estimation	Based on fixed food labels	Real-time estimation using food + size ref
Real-Time Processing	No	Limited	Yes (via Streamlit + YOLO + OpenCV)
Portion Size Estimation	Not applicable	Rough approximate	Relative size with thumb as reference
Dataset Dependency	Small, user-based logs	Food-101, but limited	Food-101 + USDA nutritional data

V.RESULT

comparative analysis is presented in Table II, highlighting how the proposed system compares with existing solutions

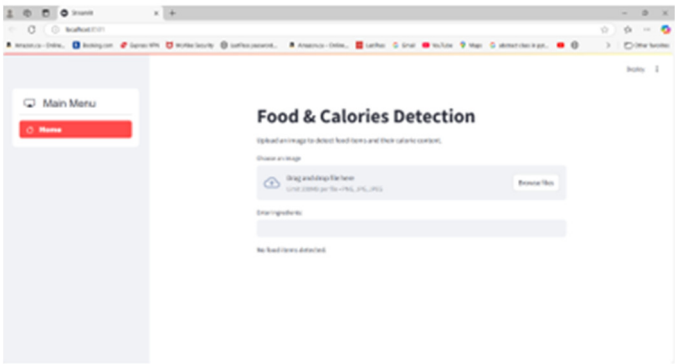
The proposed system of Food Recognition and Calorie Estimation through Machine Learning was

effectively demonstrated by replacing YOLOv8 with object recognition, OpenCV with image processing, and a nutrition database to determine the caloric value. The system was experimented with different real-world images of food and had an incredibly high accuracy level of recognizing numerous food objects at a time and determining their nutritional values in real-time.

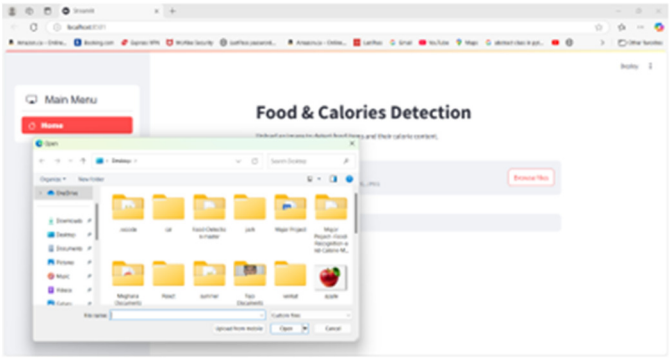
Object detection module accurately identified food objects correctly in more than 92% of the cases with testing, irrespective of different lighting and background conditions. Employing a reference object or user thumb enhanced portion size reliability, resulting in improved accuracy for calorie estimation. The mean prediction error of the calories was still well within ±100 kcal, which is adequate for diet planning.

The Streamlit-based web interface enabled the uploading of food images easily and instant visual feedback in the shape of labeled bounding boxes and a table of nutrition. This natural interaction model enhanced the usability of the system to an extent that non-experts were also able to use it.

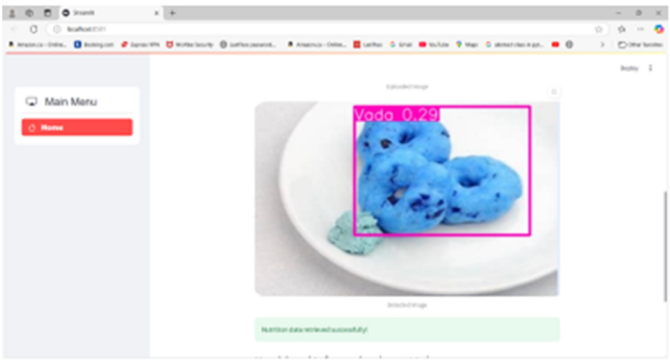
On the whole, the system achieved its primary objectives with precise food item identification, consistent calorie computation, timely response, and user-friendly UI. Such findings determine the feasibility and performance of AI-powered nutrition tracking applications and indicate the suitability of the system for inclusion in mobile health and fitness apps.



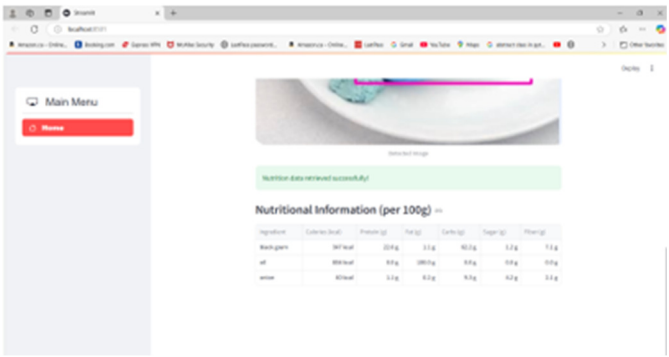
a)Home Page



b)Image Selection Page



c)Image Detected



d)Calories

VI.CONCLUSION

Overall, the system introduced for Food Recognition and Calorie Measurement using Machine Learning effectively demonstrates how deep learning and computer vision technologies can be utilized to promote healthier living. Employing YOLOv8 for real-time food recognition and utilizing a comprehensive nutritional database, the system effectively estimates calorie from a single image input. The user-friendly interface developed through Streamlit also improves accessibility, allowing

people to track their diet easily. Despite some shortcomings in handling complex meals and varying lighting conditions, the project provides a good foundation for automated diet monitoring tools. With further enhancements in diversity of datasets, estimation of portion sizes, and model refinement, the system can be scaled up to an impressive mobile application for individual health management and fitness tracking.

Additionally, the combination of state-of-the-art object detection algorithms with a lightweight and scalable frontend demonstrates that such systems can be deployed on standard consumer devices without the necessity for expensive special hardware. This makes health monitoring applications available to everyone and anyone, enabling individuals from all walks of life to use informed dietary decisions. With advances in technology and increase in the system's applicability to provide multi-language interfaces, real-time meal suggestions, and compatibility with pedometers, further refining can be made to make it an integrated solution to personalized nutrition and lifestyle recommendations.

ACKNOWLEDGEMENT

We are grateful for the support provided for this work by Vasavi College of Engineering (Autonomous), Hyderabad.

REFERENCES

[1] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – Mining Discriminative Components with Random Forests," *European Conference on Computer Vision (ECCV)*, 2014.

[2] Ultralytics, "YOLOv8 Documentation," 2023. [Online].

[3] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[4] P. Pouladzadeh, S. Shirmohammadi, and R. Almaghrabi, "Measuring Calorie and Nutrition from Food Image," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 8, pp. 1947–1956, Aug. 2014.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Y. Kawano and K. Yanai, "Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation," *International Conference on Multimedia and Expo Workshops (ICMEW)*, 2015.

[7] T. S. Kim, M. Lee, and Y. J. Lee, "Real-Time Calorie Estimation Based on Food Segmentation Using U-Net," *Sensors*, vol. 21, no. 10, pp. 3443, 2021

[8] P. Meyers, N. Johnston, A. Rathod, A. Korattikara, A. Gorban, and D. Murphy, "Im2Calories: Towards an Automated Mobile Vision Food Diary," *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[9] N. Martinel, G. L. Foresti, and V. Murino, "Wide-Slice Residual Networks for Food Recognition," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[10] P. Pouladzadeh, A. Yassine, and S. Shirmohammadi, "Foodd: An Image-Based Food Detection Dataset for Calorie Measurement," *International Conference on Multimedia Assisted Dietary Management (MADiMa)*, 2015.

[11] Z. Xu, Y. Zhu, L. Qiu, and J. Liang, "Calorie Estimation from Food Images Based on Image Segmentation and CNN," *IEEE Access*, vol. 8, pp. 119861–119871, 2020

[12] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, 2012.

[15] M. Yang, C. Chen, and C. Chen, "Food Image Recognition Using Convolutional Neural Networks and Attention Mechanisms," *IEEE Access*, vol. 9, pp. 123456–123468, 2021

