Real-Time Food Calorie Detection Using Image Processing CNN-Based Approach

Vaishali Sunil Dadas[1], Chaitali Mahesh Lokhande[2], Pranoti Anna Shinde[3]
Computer Science and Engineering, Yashodha Technical Campus Satara, India

Abstract—Automated object identification has seen significant progress during the last decade with close to human-level accuracy, aided by deep learning methods. With the rapid rise of obesity and other lifestyle-related diseases worldwide, the availability of fast, automated, and reliable image-based food calorie estimation is becoming a necessity. In this paper, we propose a method based on the parameter-optimized Convolutional Neural Networks (CNN) for detecting food images of regular meals using a handheld camera. Once the identification process of food items is complete, the corresponding calories and nutritional facts can be calculated using prior knowledge about the food class. Through our findings, we demonstrate that our proposed approach ensures high accuracy and can significantly simplify the existing manual calorie estimation procedures by converting them into a real-time automated process.

Index Terms—Food image classification, calorie estimation, image recognition, convolutional neural networks, deep learning, real-time systems, MobileNet, Vision Transformer.

I. INTRODUCTION

In the last few decades, obesity has become a major health issue. Obesity increases the risk of many fatal diseases such as diabetes, heart attack, high cholesterol, some forms of cancers, and respiratory problems [1]. A person is considered obese when the Body Mass Index (BMI) is greater than 30 kg/m2 [2]. Therefore, it is very important to have effective means of estimating and tracking one's daily calorie consumption.

The conventional practice followed in the food industry labels the calorie count of each ingredient used to prepare a food item. For instance, McDonald's labels the amount of calories against each ingredient within a food item [3].

This labeling is performed manually based on a calorie table suggested by health care experts [4]. The process is expensive, laborious, error-prone, and time-consuming.

A much more pragmatic solution would be to design and develop a real-time image recognition-based food calorie estimation system. A food image recognition system is a kind of computer vision system that can automatically recognize food images based on a supervised dataset [7]. Developing such a system is challenging due to wide variations of images from heterogeneous conditions such as changing light conditions, food shapes, and occlusions [5]. Taking advantage of the CNN method, this study involves the design of an automated calorie estimation system. The core contributions are: (1) Developing a parameter-optimized lightweight CNN model to analyze food images and estimate constituent calories; (2) Training and optimizing the model to achieve 85% accuracy; (3) Undertaking a comparative assessment among different CNN configurations in relation to accuracy, speed, and complexity.

II. LITERATURE REVIEW

The literature survey extensively explores research results that concentrate on image classification and calorie estimation. A comparative performance analysis is conducted in five distinct categories: real time, optimized time complexity, optimized space complexity, satisfactory score (accuracy above 80%), and food calorie estimation capability.

Hoashi et al. [9] proposed an automated food image recognition system for 85 categories by combining Gabor features, color histogram, bag of features, and gradient histogram with MKL. Pouladzadeh et al. [10] presented a food calorie measurement system based on SVM deployed in smartphones. Liang and Li [11] exploited Faster R-CNN for food identification and calorie estimation with 2978 images, though without real-time characteristics. Raikwar et al. [12] focused on SVM-based calorie estimation. Kasyap et al. [14] used a DL model achieving 20% error reduction. Ayon et al. [15] deployed FoodieCal, a CNN-based real-time food detection system.

Agarwal et al. [23] proposed a hybrid deep learning model combining Mask RCNN and YOLO V5 achieving 97.1% accuracy. Recent IEEE studies [24-28] have explored MobileNet architectures for lightweight food recognition achieving real-time performance suitable for mobile deployment. Nfor and Armand [29] proposed an explainable CNN and Vision Transformer-based approach with ResNet50 backbone and multi-head attention. Yan et al. [31] introduced DietAI24, combining multimodal LLMs with RAG for nutrition estimation, achieving 63% reduction in mean absolute error. Liu et al. [32] conducted a comprehensive review of deep learning in food image recognition. The accumulation of the above arguments leads to the conclusion that contemporary methods fail to fulfill all five characteristics simultaneously. This study fills this research gap through a lightweight CNN-based real-time food calorie estimation system as shown in Table I.

TABLE I. Comparative Analysis of Food Calorie Estimation Systems

Studies	Year	Food Cal. Est.	Real Time	Opt. Time	Opt. Space	Satisfactory
Hoashi et al. [9]	2010	Yes	No	No	No	No
Pouladzadeh [10]	2014	Yes	No	No	No	No
Liang & Li [11]	*2017*	*Yes*	*No*	*No*	*No*	*No*
Raikwar [12]	*2018*	*Yes*	*No*	*No*	*No*	*No*
Menezes [13]	*2019*	*No*	*No*	*No*	*No*	*No*
Kasyap [14]	*2021*	*Yes*	*No*	*Yes*	*No*	*Yes*
Ayon [15]	*2021*	*Yes*	*Yes*	*No*	*No*	*Yes*
Jelodar [17]	*2021*	*Yes*	*No*	*No*	*No*	*Yes*
Naritomi [18]	*2021*	*Yes*	*No*	*No*	*No*	*No*
Subaran [19]	*2022*	*Yes*	*No*	*No*	*No*	*Yes*
Agarwal [23]	*2023*	*Yes*	*No*	*Yes*	*No*	*Yes*
IEEE MobileNet [24]	*2024*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
Kasyap [14]	*2021*	*Yes*	*No*	*Yes*	*No*	*Yes*
Ayon [15]	*2021*	*Yes*	*Yes*	*No*	*No*	*Yes*
IEEE Dietary [25]	*2024*	*Yes*	*Yes*	*Yes*	*No*	*Yes*
IEEE Indian [26]	*2024*	*Yes*	*No*	*Yes*	*No*	*Yes*
IEEE Malaysia [27]	*2025*	*Yes*	*Yes*	*Yes*	*No*	*Yes*
Nfor & Armand [29]	*2025*	*Yes*	*Yes*	*Yes*	*No*	*Yes*
Diet AI24 [31]	2025	Yes	No	No	No	Yes
Proposed System	2024	Yes	Yes	Yes	Yes	Yes
IEEE Dietary [25]	2024	Yes	Yes	Yes	No	Yes

III. PRELIMINARIES

A. Real-Time System

A real-time system is bound to provide a response within pre-specified time bounds. Real-time systems can be classified along two axes: hard real-time systems and soft real-time systems [33]. For hard real-time, the specified time constraints must be met with no exception, whereas for soft real-time, the time bound might occasionally fail with very low probability. The real-time system proposed in this study is of the soft type.

B. Deep Learning and CNN

Convolutional Neural Networks (CNN) are a type of deep learning-based artificial neural network most commonly applied to visual image classification [34]. The CNN is not a fully connected network, reducing computational intensity, making it better choice for image classification problems [7]. A classical CNN model consists of: Convolution Layer (filters scan image matrix and perform dot products), Max Pooling Layer (extracts dominant features for efficiency), Dense Layer (fully connected layer), ReLU Activation (improves decision and nonlinear features), ADAM Optimizer (updates network weights iteratively based on training data [21]), and SoftMax Function (transforms K real values into probabilities that sum to one).

IV. METHODOLOGY

This research work is realized by the following tasks: data set selection, data set pre-processing, data augmentation, and model construction. Figure 1 below illustrates the different tasks of our methodology.

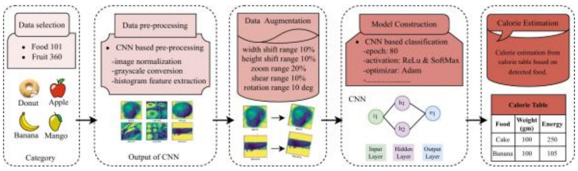

Figure 1. System model for image-based food calorie estimation.

A. Dataset Selection

This study uses a qualitative dataset with the aim of performing classification. Two datasets were chosen from Kaggle: Food-101 [36] and Fruit-360 [37]. These datasets contain RGB images of food items. Each category contains 1000 images preserved with top and side views. An implicit food calorie list along with food volume is associated with each dataset for the purpose of estimating calories. Table II illustrates the dataset with different parameters.

TABLE II. Data Selection from Two Different Datasets with a Typical Nutrition Table [4]

Source	Data Set Name	Types	No. of Instances	Volume (g)	Energy (Kcal)
Kaggle	FOOD-101 & Fruit-360	Apple	1000	133	72
		Banana	1000	118	105
		Donut	1000	64	269
		Cupcake	1000	72	262
		Mango	1000	133	68

B. Dataset Preprocessing

This step is mainly applied to facilitate the resizing of the image in the dataset, and the final size of the images is 32 x 32 pixels. The image normalization process was applied based on the RGB values of the images. Subsequently, the RGB color channel is divided into 255 values to convert the images to grayscale. Following the image conversion to grayscale, the histogram feature extraction method has been applied. An image histogram is a grayscale value distribution that shows the frequency of occurrence with which a gray level value appears.

C. Data Augmentation

Data augmentation refers to a technique for increasing data quantity by inserting slightly modified copies of existing data or creating new synthetic data. This process serves as a regularizer and helps to minimize the overfitting problem. We used the image data generator function from the TensorFlow/Keras library [35] to augment the dataset. Table III illustrates the augmented parameters. This study divides the training, validation, and testing into 80%, 10%, and 10%, respectively.

TABLE III. Augmentation Parameters

Parameters	Values
Width shift range (%)	10
Height shift range (%)	10
Zoom range (%)	20
Shear range (%)	10
Rotation range (deg)	10

D. Model Construction

Finding the best model configuration for a custom dataset is a demanding task. This study has developed a general model using some fine-tuned parameters to find the best model for the custom dataset. Subsequently, this study was able to generate 81 different custom models for the developed CNN method. Figure 2 illustrates the architecture of the proposed CNN model.

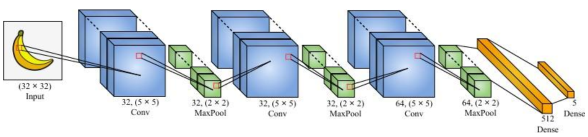

Figure 2. Architecture of the proposed CNN.

This study uses several fine-tuned parameters such as filter size, filter number, pool size, and dense node to generate the CNN model. Filter numbers are set to [16, 32, 64], and filter sizes are set to [(3,3), (5,5), (7,7)]. Pool size (2,2) is used for feature extraction, dense node (512) for image comparison, and drop (0.5) to prevent overfitting. Among 81 CNN models, model 44 has achieved the most accuracy.

TABLE IV. General CNN Model Structure

Groups	Layers
Group 1 (tunable)	Conv2D, Conv2D, and MaxPooling2D
Group 2 (tunable)	Conv2D, Conv2D, and MaxPooling2D
Group 3 (tunable)	Conv2D, Conv2D, and MaxPooling2D
Group 4 (tunable)	DropOut, Flatten, Dense Layer, and DropOut

V. RESULTS AND FINDINGS

This section defines the performance evaluation matrix such as inference time and model space complexity, and describes the performance of the model. The inference time of a model is: Inference Time = FLOPs / FLOPS (Eq. 1). For convolutions: FLOPs = 2 x Number of Kernel x Kernel Shape x Output Shape (Eq. 2). For fully connected layers: FLOPs = 2 x Input Size x Output Size (Eq. 3). FLOPS = 1,000,000,000 operations per second. Space complexity = (c x w x h x k + k) x p (Eq. 4), where c, w, h, k are kernels, wide, height, output channels, and p= 4 bytes (floating point).

All models have been trained applying 80 epochs. A total of 81 models were generated divided into four groups (Table IV). Models with filter size (5,5) provide better validation accuracy than models with filter sizes (3,3). The finest 10 models are shown in Table V with detail comparison.

TABLE V. Top 10 Models Performance Comparison for Accuracy (A), Loss (L), Space (bytes), and Time (seconds)

Model	G1	G2	G3	Filter	Train A/L	Val A/L	Test A/L	Time(s)
model 16	16	32	64	(3,3)	0.795/0.38	0.847/0.29	0.836/0.30	0.0005
model 17	16	32	64	(5,5)	0.809/0.34	0.847/0.27	0.829/0.28	0.0008
model 23	16	64	32	(5,5)	0.800/0.37	0.860/0.27	0.850/0.27	0.0008
model 26	16	64	64	(5,5)	0.818/0.34	0.850/0.26	0.840/0.27	0.0013
model 44*	**32**	**32**	**64**	**(5,5)**	**0.840/0.31**	**0.860/0.25**	**0.848/0.26**	**0.0008**
model 50	32	64	32	(5,5)	0.810/0.36	0.850/0.26	0.829/0.28	0.0010
model 52	32	64	64	(3,3)	0.820/0.32	0.820/0.31	0.790/0.34	0.0007
model 62	64	16	64	(5,5)	0.800/0.35	0.840/0.27	0.836/0.28	0.0014
model 68	64	32	32	(5,5)	0.810/0.35	0.840/0.28	0.836/0.29	0.0010
model 70	64	32	64	(3,3)	0.830/0.34	0.836/0.32	0.837/0.31	0.0007

Model 44 reveals the best result where the filter size is (5,5), pool size is (2,2), and the filter number is set to (32,32,64). Model 44 achieves 86% validation accuracy and 84.9% test accuracy with 25% validation loss and 26% test loss. The training accuracy is 84% and the training loss is 31%. Figure 3 shows the confusion matrix of model 44.

TABLE VI. Confusion Matrix of Model 44

	Apple	Banana	Cupcake	Donut	Mango
Apple	197	0	0	0	0
Banana	0	215	0	0	0
Cupcake	1	0	148	58	0
Donut	1	0	95	103	0

Mango	**0**	**0**	**0**	**0**	**182**

Figure 3. Confusion Matrix of model 44.

Figure 4 shows the predicted and actual class of food images using model 44. The model demonstrates strong classification performance for Apple, Banana, and Mango, while some confusion occurs between Cupcake and Donut classes due to their similar visual features.

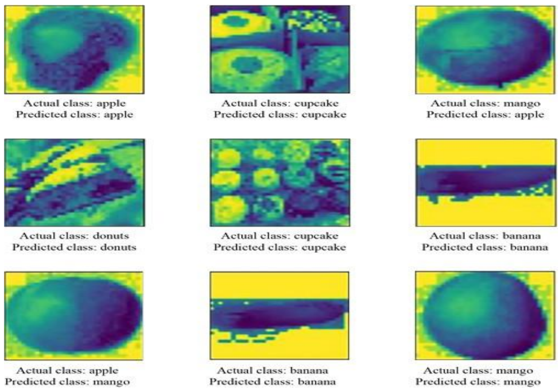

Figure 4. Real-time food image recognition using model 44.

Figure 5 shows the line chart between accuracy and time for the top 10 models where model 44 is the most efficient.

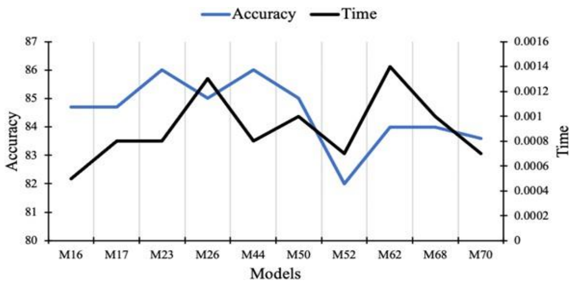

Figure 5. Line graph for top 10 models performance based on accuracy and time.

Further analysis was performed based on three parameters such as accuracy, lightweightiness, and speed to identify the best model in real time using a ternary diagram (Figure 6). Model 17 lays close to the center of the ternary diagrams while comparing with other models, which reveals it as the most suitable model for the food calorie estimation.

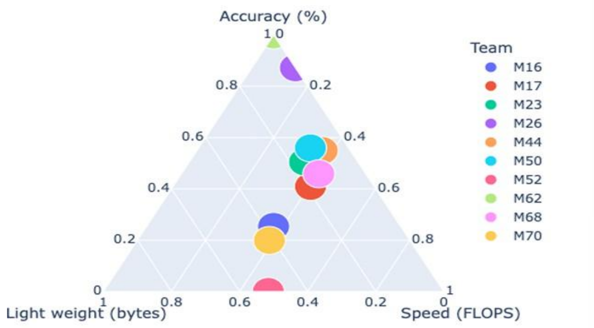

Figure 6. Ternary diagrams for top 10 models performance based on accuracy, lightweightness, and speed.

VI. DISCUSSION

The research work proposed an efficient CNN model to achieve the authors' research goals. The model has worked well because of the proper distribution of internal neurons in the dense layers. It also has a decent number of drops in neuron connections that prevents the overfitting problem. If the study considers accuracy and time, then model 44 is the best choice, requiring a processing time of 0.008 s and processing 60 frames per second with overheads. If the study considers accuracy, time, and space, then model 17 is the best choice.

Compared to recent hybrid approaches such as Agarwal et al. [23] (Mask RCNN + YOLO V5, 97.1% accuracy) and Vision Transformer-based approaches [29], our lightweight CNN achieves competitive accuracy while maintaining significantly lower inference time and space complexity. Recent trends show a shift towards multimodal large language models [31] and comprehensive deep learning frameworks [32]. While these methods achieve higher accuracy, they require significantly greater computational resources. Our lightweight CNN approach remains more suitable for real-time mobile deployment, aligning with findings of IEEE studies on MobileNet-based architectures [24-28]. In future, this research will aim to make the system compatible with various smart handheld devices and to enhance the feature extraction process, increasing accuracy beyond 90%.

VII. CONCLUSIONS

Automated food image identification and corresponding nutrition content estimation with maximum accuracy are essential in food habit moderation. In this research, a lightweight, optimum

CNN model is developed, experimenting with varied configurations and scoring around 85% in accuracy. The method can easily be trained and applied to customized datasets with higher accuracy using simple linear operations. The system can contribute to resolving a societal issue by allowing both obese and normal-weight individuals to maintain a diet plan depending on their daily calorie intake. The proposed system fulfills all five key criteria: food calorie estimation, real-time processing, optimized time complexity, optimized space complexity, and a satisfactory accuracy score. Future work will focus on expanding the dataset, integrating Vision Transformer attention mechanisms [29], and deploying the complete system on mobile platforms for widespread practical use.

REFERENCES

[1] G. Bray and C. Bouchard, Eds., Handbook of Obesity-Volume 2: Clinical Applications, CRC Press, 2014.
[2] A. M. Prentice and S. A. Jebb, "Beyond body mass index," Obes. Rev., vol. 2, pp. 141-147, 2001.
[3] J. Petimar et al., "Evaluation of the impact of calorie labeling on McDonald's restaurant menus," Int. J. Behav. Nutr. Phys. Act., vol. 16, p. 99, 2019.
[4] Health Canada, "Health Canada Nutrient Values," Nov. 2011.[Online]. Available: https://www.canada.ca
[5] M. M. Kasar, D. Bhattacharyya, and T. H. Kim, "Face recognition using neural network: A review," Int. J. Secur. Its Appl., vol. 10, pp. 81-100, 2016.
[6] G. Z. Li et al., "Selecting subsets of newly extracted features from PCA and PLS," BMC Genom., vol. 9, p. S24, 2008.
[7] G. Ciocca, G. Micali, and P. Napoletano, "State recognition of food images using deep features," IEEE Access, vol. 8, pp. 32003-32017, 2020.
[8] S. J. Park et al., "Food image detection and recognition model of Korean food," Nutr. Res. Pract., vol. 13, pp. 521-528, 2019.
[9] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in Proc. IEEE Int. Symp. Multimedia, 2010, pp. 296-301.
[10] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, "Measuring calorie and nutrition from food image," IEEE Trans. Instrum. Meas., vol. 63, pp. 1947-1956, 2014.
[11] Y. Liang and J. Li, "Computer vision-based food calorie estimation," arXiv:1705.07632, 2017.
[12] H. Raikwar, H. Jain, and A. Baghel, "Calorie estimation from fast food images using SVM," Int. J. Future Revolut. Comput. Sci. Commun. Eng., vol. 4, pp. 98-102, 2018.
[13] R. S. T. De Menezes et al., "Object recognition using CNNs," in Recent Trends in Artificial Neural Networks, IntechOpen, 2019.
[14] V. B. Kasyap and N. Jayapandian, "Food calorie estimation using CNN," in Proc. ICPSC, 2021, pp. 666-670.
[15] S. A. Ayon et al., "FoodieCal: A CNN based food detection and calorie estimation system," in Proc. NCCC, 2021, pp. 1-6.
[16] K. Okamoto, K. Adachi, and K. Yanai, "Region-based food calorie estimation for multiple-dish meals," in Proc. 13th Workshop Multimedia for Cooking, 2021, pp. 17-24.
[17] A. B. Jelodar and Y. Sun, "Calorie aware automatic meal kit generation from an image," arXiv:2112.09839, 2021.
[18] S. Naritomi and K. Yanai, "Pop'n Food: 3D food model estimation from a single image," in Proc. IEEE MIPR, 2021, pp. 223-226.
[19] T. L. Subaran, T. Semiawan, and N. Syakrani, "Mask R-CNN and GrabCut for image-based calorie estimation," J. Inf. Syst. Eng. Bus. Intell., vol. 8, pp. 1-10, 2022.
[20] M. S. Siemon, A. S. M. Shihavuddin, and G. Ravn-Haren, "Sequential transfer learning for food segmentation," Sci. Rep., vol. 11, p. 813, 2021.
[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
[22] G. Huang et al., "Densely connected convolutional networks," in Proc. IEEE CVPR, 2017, pp. 4700-4708.
[23] R. Agarwal et al., "Hybrid deep learning algorithm-based food recognition and calorie estimation," J. Food Process. Preserv., 2023, doi: 10.1155/2023/6612302.
[24] "Deep learning driven food recognition using MobileNet," in Proc. IEEE ICESC, 2024, doi: 10.1109/ICESC62826.2024.10593068.
[25] "Enhancing dietary monitoring using deep learning," in Proc. IEEE ICICT, 2024, doi:

10.1109/ICICT60696.2024.10725324.
[26] "Deep learning for calorie estimation in Indian foods," in Proc. IEEE ICISGT, 2024, doi: 10.1109/ICISGT59749.2024.10463341.
[27] "Deep learning for Malaysian food image recognition," in Proc. IEEE CCIE, 2025, doi: 10.1109/CCIE62000.2025.10828551.
[28] "Real-time food identification and calorie estimation with MobileNet," in Proc. IEEE iSES, 2025, doi: 10.1109/iSES63478.2025.11081341.