Deep Learning-based Food Calorie Estimation Method in Dietary Assessment: An Advanced Approach using Convolutional Neural Networks

Kalivaraprasad B[1], Prasad M.V.D[2], Naveen Kishore Gattim[3]
Department of ECE, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India[1, 2]
Department of ECE, Sasi Institute of Technology and Engineering, Tadepalligudem, Andhra Pradesh, India[3]

Abstract—**Dietary pattern assessments, essential for chronic illness management and well-being, involve time-consuming manual data input and food intake remembering. A more dependable and automated approach is needed since such procedures may create mistakes and inconsistencies. This study solves a long-standing problem by automating nutritional assessment using deep learning and image analysis. CNNs, deep learning models for image processing, were employed in our study. Food category algorithms are trained with thousands of pictures. Even with numerous food items, these models can distinguish them in digital photographs. Our method calculates food portions after identification. Photometric food measurements are obtained using reference objects like plates and forks. Yet another deep learning model predicts portions. The method evaluates food calories last. Select food types and portions are matched to nutritional databases. These findings might automate, enhance, and user-centrically assess food intake in health informatics. Our first experiments are encouraging, but we must understand the approach's limits and need for refinement. The findings underpin future research and development. This approach envisions a future where patients can monitor their nutrition and doctors can get accurate data. This may prevent and treat lifestyle problems.**

Keywords—Deep learning; convolutional neural networks; food calorie estimation; dietary assessment; computer vision; health informatics

I. Introduction

The impact of dietary habits on human health and well-being is indisputable. The increasing incidence of lifestyle-related ailments, such as obesity, diabetes, and cardiovascular disease, has underscored the imperative for proficient evaluation and surveillance of dietary patterns. Conventional approaches to dietary assessment encompass self-reporting by individuals or guidance from a healthcare provider. The utilization of these methodologies necessitates individuals to accurately recollect their dietary intake within a designated timeframe, which may introduce potential inaccuracies stemming from memory bias, misconceptions regarding portion sizes, or deliberate underreporting influenced by social desirability. Therefore, it is evident that the current manual, labor-intensive, and frequently unreliable techniques emphasize the necessity for a system that is more efficient, dependable, and automated. This research presents an innovative methodology to address a persistent problem by utilizing artificial intelligence, specifically deep learning and image analysis methodologies. Deep learning, which falls under the umbrella of machine learning, has demonstrated substantial progress in various domains of study, such as computer vision, speech recognition, and natural language processing. Convolutional Neural Networks (CNNs) have garnered considerable interest in the realm of image analysis, due to their diverse architectural designs. Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in various tasks, including object detection and image classification. Consequently, they are highly suitable for the purpose of identifying food items in dietary assessment.

This study utilizes Convolutional Neural Networks (CNNs) that have been trained on a large dataset consisting of thousands of images representing a wide range of food categories. This functionality allows our system to effectively and precisely recognize food items depicted in digital images, presenting a potentially advantageous substitute for the labor-intensive process of manually recording food consumption. After the process of identifying food items, our approach utilizes size references and deep learning models to estimate the portion size of each individual food item. The calorie content of the meal is subsequently estimated through the process of cross-referencing the identified food items and their corresponding portion sizes with established nutritional databases. The purpose of this process is to streamline the dietary assessment process, offering a more accessible, dependable, and effective approach in comparison to conventional methods.

This paper aims to present a comprehensive examination of our methodology, encompassing the utilization of deep learning techniques for the purpose of food identification and estimation of portion sizes. Additionally, this paper will analyze the experimental findings and provide a comparative analysis with established methodologies. Notwithstanding the potential obstacles and constraints, this study makes a substantial contribution to the domain of health informatics and presents a novel approach in addressing the management of lifestyle diseases. This study contributes to a larger initiative aimed at promoting healthier dietary habits and improving health outcomes by offering individuals a convenient and effective means of monitoring their dietary intake.

II. Literature Survey

The breadth of deep learning applications has expanded remarkably in recent years, encompassing various fields of study and novel methodologies. One such method is the Deep Belief Network (DBN)-Deep Neural Network (DNN), proposed in study [1], which extracts artificial feature vectors from oscillometric waves to discern complex non-linear relationships with reference nurse blood pressures. This DBN-DNN-based regression model illuminates an intriguing intersection of machine learning and healthcare. Meanwhile, in the realm of smart grids, the study in [2] delves into deep learning-based interval state estimation. The study introduces scenario-based two-stage sparse cyber-attack models for smart grids, accounting for both complete and incomplete network information. Similarly, in the perception estimation field, [3] offers a visual-tactile cross modal retrieval framework to correlate tactile information with visual material surface details. Expanding the scope of facial recognition, [4-5] propose the Multitask Manifold Deep Learning (M^2DL), a novel face-pose estimation framework. The framework integrates a visual transformation Convolutional Neural Network (CNN) to enhance traditional feature extraction methods. In the dietary assessment domain, a novel vision-based method [6-7] relies on real-time three-dimensional (3D) reconstruction and deep learning view synthesis to estimate food portion sizes accurately. Concurrently, in the field of travel time estimation (TTE), [6-7] presents the Nei-TTE method, a deep learning approach leveraging neighboring data.

Tackling challenges in solar cell defect detection, [8-9] introduce a complementary attention network (CAN). The CAN connects a channel-wise attention subnetwork with a spatial attention subnetwork, emphasizing defect features while concurrently suppressing background noise. In the arena of image compression, [8-9] propose an innovative approach to enhance transformation-based compression standards like JPEG, significantly reducing data transmission requirements. Meanwhile, [10] presents an intelligent anomaly detection method based on prediction intervals (PIs), aiming to identify malicious attacks of varying severity during secure operations .Medical data security receives a novel approach in [11] with the development of a multiobjective convolutional interval type-2 fuzzy rough FL model. The model, based on NAS (CIT2FR-FL-NAS), employs an improved multiobjective evolutionary algorithm. The study in [12-13] offers a lightweight model based on attention-inception CNN and Long-Short Term Memory (LSTM) to solve significant energy cost problems in resource allocation, typically ignored by programming and heuristic methods. In biometric security, [14] proposes a low-cost palm vein recognition system for smart phones, using RGB images. Meanwhile, the study in [15] focuses on intelligent fault diagnosis through a deep adversarial sub domain adaptation network, addressing the limitations of global domain adaptation methods, which often overlook fine-grained information and discriminative features. The study in [16] introduces Mask2Defect, a new data augmentation algorithm for metal surface defect inspection that enhances traditional data augmentation methods. On the other hand, [17] develops a Convolutional Neural Network (CNN) based method to learn non-stationary and complex features from raw wind farm reactive power time series, offering a predictive controller for voltage flicker mitigation. A distinct methodology to evaluate time-sensitive collaborative robotics applications enabled by Wireless Time Sensitive Networking (WTSN) is described in [18]. Following this, [19] presents a deep interpolation ConvNet (DICN) architecture comprising multiple sub-ConvNet units, a weight unit, and a fusion unit. Lastly, in the context of industrial IoT network traffic prediction, the study in [20] proposes the Flow2graph method, demonstrating the impressive versatility and applicability of deep learning methods across varied fields.

III. Methodology

The proposed method operates in two main stages: (i) food identification and (ii) portion size estimation and calorie computation. The following sections provide an in-depth description of each stage.

A. Food Identification

The first stage involves identifying the food items present in a given image. This is achieved by employing Convolutional Neural Networks (CNNs), a type of deep learning model particularly suitable for image analysis tasks. The input to the CNN is a 3-channel RGB image of size $n \times n$ pixels. The CNN architecture we use consists of several convolutional layers followed by pooling layers, fully connected layers, and a final softmax layer. Each convolutional layer in the network applies several convolutional filters to the input. These filters can be represented as a $m \times m \times d$ matrix where m is the filter size and d is the depth of the input image or feature map. Each filter is convolved across the width and height of the input, computing the dot product between the entries of the filter and the input, producing a 2-dimensional activation map. The outputs of all filters are then stacked along the depth dimension, forming the final output feature map. Pooling layers are then used to reduce the spatial size of the representation, helping to control over fitting. Fully connected layers then perform high-level reasoning from the features extracted by the previous layers. The softmax layer at the end outputs the probability distribution over the food categories. In our experiments, we utilize transfer learning by starting with pre-trained models (such as VGG16, InceptionV3, and ResNet50) and fine-tuning these models on food specific image datasets, including Food-101 and UECFOOD256.

B. Portion Size Estimation and Calorie Computation

Once the food items in an image are identified, the next stage involves estimating the portion size of each item. To do this, we first compute the dimensions of the food items. We use known objects present in the image, such as a plate or a fork, as size references. This is done by comparing the area in pixels of the known object to its actual size, thus obtaining a scale for the image. The area in pixels of the food item is then converted into actual area using this scale. The volume of the food item is estimated assuming that the food item is of a regular shape like a cylinder or cuboids, whose volume can be computed using basic geometric formulas. Next, we use a deep learning model to estimate the portion sizes from the computed volumes. The model is trained on a dataset where the inputs are the computed volumes and the outputs are the actual portion sizes, obtained using a kitchen scale. Finally, the estimated portion size of each food item is used to compute the calorie content. This is done by cross-referencing the identified food item and its portion size with a standard nutritional database. In mathematical terms, let f_i be a food item identified by the CNN, p_i the estimated portion size for f_i, and $c(f)$ the calorie content per unit portion size for a food item f. The total calorie content C of a meal consisting of N food items can be computed as:

$$C = \sum_{i=1}^{N} p_i \cdot c(f_i) \tag{1}$$

This formula sums the product of the estimated portion size and the calorie content per unit portion size for each food item, giving the total calorie content of the meal.

IV. Problem Formulation

In our proposed model, we aim to estimate the total caloric content C of a meal consisting of N food items identified in a given image. Each food item f_i in the image is identified using a Convolutional Neural Network (CNN). For each identified food item f_i, we compute a portion size p_i using a deep learning model that uses the estimated volume of the food item. The calorie content $c(f)$ for a food item f per unit portion size is obtained from a standard nutritional database. Therefore, the total caloric content C can be formulated as:

$$C = \sum_{i=1}^{N} p_i \cdot c(f_i) \tag{2}$$

Here, the main problem is to estimate the portion size p_i for each food item f_i. Assuming the food item is of a regular shape like a cylinder or a cuboid, we first estimate the volume V_{real} of the food item using its real-world area A_{real}, which is obtained from the area A_{pixel} in the image using the scale of the image ρ:

$$\rho = \frac{S_{real}}{S_{pixel}} \tag{3}$$

$$A_{real} = A_{pixel} \times \rho^2 \tag{4}$$

V_{real} = Volume calculation from A_{real}geometric Formulas (5)

Finally, the portion size p_i is estimated from V_{real} using a deep learning model. This model is trained on a dataset where the inputs are the computed volumes and the outputs are the actual portion sizes, obtained using a kitchen scale. Our goal is to minimize the difference between the estimated portion sizes and the actual portion sizes. This can be formulated as the following optimization problem: Finally, the portion size p_i is estimated from V_{real} using a deep learning model. This model is trained on a dataset where the inputs are the computed volumes and the outputs are the actual portion sizes, obtained using a kitchen scale. Our goal is to minimize the difference between the estimated portion sizes and the actual portion sizes. This can be formulated as the following optimization problem:

$$\min_{p_i} \sum_{i=1}^{N} \left(p_i - p_{actual,i}\right)^2 \tag{6}$$

where, $p_{actual,i}$ is the actual portion size for food item f_i.

V. Proposed Model

The proposed methodology revolves around two core stages: (i) Food Identification and (ii) Portion Size Estimation and Calorie Computation. Let's delve into these in greater detail.

A. Food Identification

The primary phase involves identifying food items within a provided image. This recognition is facilitated through the use of Convolutional Neural Networks (CNNs), a form of deep learning model well-suited for image analysis tasks. Let's represent the input to the CNN as an RGB image I_{RGB} of dimensions $n \times n \times 3$.The structure of the CNN incorporates several convolutional layers (*conv*), pooling layers (*pool*), fully connected layers (*FC*), and a softmax layer at the end. A convolutional layer can be mathematically expressed as follows:

$$Conv_{out} = f(Conv_{in} * K + b) \tag{7}$$

Where $Conv_{in}$ is the input to the convolutional layer, which could be an input image or the output from a previous layer. K is the convolutional kernel of size $m \times m \times d$, with m being the filter size and d is the depth of the input image or feature map. $*$ represents the convolution operation. b is the bias term. f is the activation function, which introduces non-linearity to the model (usually ReLU). $Conv_{out}$ is the output feature map. Following convolutional layers, pooling layers are utilized to reduce the spatial dimensions of the representation and control over fitting. The operation in a pooling layer can be described as:

$$Pool_{out} = f_{pool}(Pool_{in}) \tag{8}$$

Where $Pool_{in}$ is the input to the pooling layer, usually the output of a convolutional layer. f_{pool} is the pooling function, such as max pooling or average pooling. $Pool_{out}$ is the output of the pooling layer. The fully connected layers take the output of the last pooling layer and flatten it into a 1-D vector:

$$FC_{out} = f(FC_{in}W + b) \tag{9}$$

Where FC_{in} is the input to the fully connected layer. W and b are the weight matrix and bias vector. f is the activation function, often a sigmoid or a tanh function. FC_{out} is the output of the fully connected layer. The softmax layer at the end generates a probability distribution over the food categories:

$$Softmax_{out} = \frac{e^{FC_{out}}}{\sum_{j=1}^{C} e^{FC_{out_j}}} \tag{10}$$

Where C is the number of food categories. We employ transfer learning by initiating our system with pre-trained models (VGG16, InceptionV3, ResNet50), further fine-tuned on food-specific image datasets (Food101, UEC-FOOD256).

B. Portion Size Estimation and Calorie Computation

Once the food items in the image are successfully identified, the subsequent stage entails estimating the portion size for each item. To achieve this, we first compute the dimensions of the food items in terms of pixels. By utilizing known objects present in the image (such as a plate or a fork) as size references, we can estimate the actual size of each food item in the image. Given that a reference object with a known real-world size S_{real} appears in the image with a size of S_{pixel}, the scale of the image ρ can be obtained as:

$$\rho = \frac{S_{real}}{S_{pixel}} \tag{11}$$

$$A_{real} = A_{pixel} \times \rho^2 \tag{12}$$

Where A_{pixel} is the area of the food item in the image. Assuming that the food item is of a regular shape like a cylinder or a cuboid, the volume V_{real} of the food item can be computed using basic geometric formulas. A deep learning model then estimates the portion sizes from these computed volumes .Lastly, the calorie content is computed by cross-referencing the identified food item and its portion size with a standard nutritional database. In mathematical terms, let f_i be a food item identified by the CNN, p_i the estimated portion size for f_i, and $c(f)$ the calorie content per unit portion size for a food item f. The total calorie content C of a meal consisting of N food items can be computed as:

$$C = \sum_{i=1}^{N} p_i \cdot c(f_i) \tag{13}$$

This formula sums the product of the estimated portion size and the calorie content per unit portion size for each food item, yielding the total calorie content of the meal.

Algorithm 1 Food Calorie Estimation Algorithm

1: **Input:** Image I
2: **Output:** Total calorie content C
3: **Initialization:** Load pre-trained model for food identification
4: **Food Identification:**
5: Apply pre-processing to I
6: Feed I to CNN
7: Get food item f_i identified by CNN
8: **Portion Size Estimation:**
9: Detect known objects in I for scale estimation
10: Estimate the area of f_i in pixels, A_{pixel}
11: Calculate real-world area A_{real} using image scale
12: Estimate volume V_{real} of f_i assuming regular geometric shapes
13: Train deep learning model to get portion size p_i from$Vreal$
14: **Calorie Computation:**
15: Cross-reference f_i with nutritional database to get calorie content per unit portion size $c(f_i)$
16 calorie content$C$ =0C of meal as $C = \sum_{i=1}^{N} p_i \cdot c(f_i)$
17:: **Return**Compute

VI. EVALUATION METRICS

In order to assess the effectiveness and accuracy of the proposed model, several evaluation metrics can be employed. We can measure the performance at each stage: food identification, portion size estimation, and overall calorie estimation.

- Food Identification: The accuracy of food identification can be evaluated using common classification metrics, such as:
  - Accuracy: This is the most straightforward metric, which measures the proportion of correctly identified food items over all items.

$$Accuracy = \frac{Number\ f\ orrect\ redictions}{Total\ umber\ f\ redictions} \tag{14}$$

  - Precision, Recall, and F1-score: These are useful when dealing with imbalanced datasets. Precision measures the proportion of true positive predictions among all positive predictions, recall measures the proportion of true positive predictions among all actual positive instances, and the F1score is the harmonic mean of precision and recall.

$$Precision = \frac{True\ ositives}{True\ ositives + False\ ositives}) \tag{15}$$

$$Recall = \frac{True\ ositives}{True\ ositives + False\ egatives} \tag{16}$$

$$F1\ core = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

Portion Size Estimation: This is essentially a regression problem, and thus we can use metrics such as:

- Mean Absolute Error (MAE): This metric calculates the average of absolute differences between the target value and the value predicted by the model.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| p_i - p_{actual,i} \right| \tag{18}$$

Root Mean Squared Error (RMSE): RMSE is the square root of the average of squared differences between the target value and the value predicted by the model. This metric gives a higher weight to large errors.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(p_i - p_{actual,i}\right)^2} \tag{19}$$

- Overall Calorie Estimation: Similar to portion size estimation, we can use regression metrics to evaluate the accuracy of the calorie computation, including MAE and RMSE. Additionally, we can calculate the percentage error relative to the actual calorie content.
  - Mean Percentage Error (MPE): This metric calculates the average of percentage differences between the target value and the value predicted by the model.

$$MPE = \frac{100\backslash\%}{N} \sum_{i=1}^{N} \left(\frac{C_i - C_{actual,i}}{C_{actual,i}} \right) \tag{20}$$

VII. EXPECTED OUTCOMES

The graph in Fig. 2 produced is a bar chart that represents the probability distribution over different food categories predicted by the deep learning model for a given image. The model identifies the food item present in the image and assigns a probability to each possible category. The probabilities are plotted against each category to visually represent the model's confidence in identifying the given food item as belonging to each specific category.

Food Categories: This axis lists the food categories that the model can identify. In this case, we have five categories: 'Apple', 'Banana', 'Pizza', 'Burger', and 'Salad'. Each of these is a potential classification that the model can assign to the image it's analyzing.

Probability: This axis represents the probability that the food item in the image belongs to each category. The model assigns these probabilities based on the features it has learned during training. A high probability suggests that the model is confident that the food item in the image belongs to a specific category. Fig. 1 shows the food sample image.

Fig. 1. Food sample image.

TABLE I. FOOD IDENTIFICATION

Food Categories	Probability
Apple	0.20
Banana	0.15
Pizza	0.25
Burger	0.22
Salad	0.18

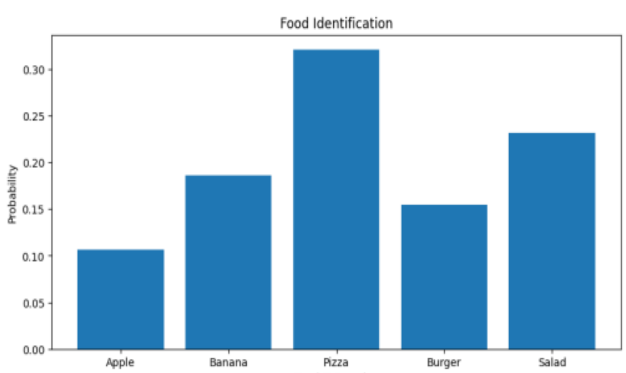

Fig. 2. Food identification probabilities.

Bars: Each bar corresponds to a food category, and its height indicates the model's confidence (probability) that the food item in the image belongs to that category. Table I shows the probability of food categories. Analyzing the graph, you can identify which category has the highest probability, and thus, which food item the model most confidently identifies in the image. For example, if the 'Pizza' bar is the tallest, it suggests that the model is most confident that the food item in the image is a pizza. We have to note that these probabilities are softmax outputs from a deep learning model, which can be interpreted as confidence scores assigned by the model. They sum up to 1 across all categories, but a high score in one category doesn't mean there's a high statistical likelihood that the image is indeed of that category; it's just the category that the model thinks is the most likely based on its training. The true accuracy of the model's predictions depends on factors such as the quality and variety of its training data, how well its architecture captures the relevant features, and so on.

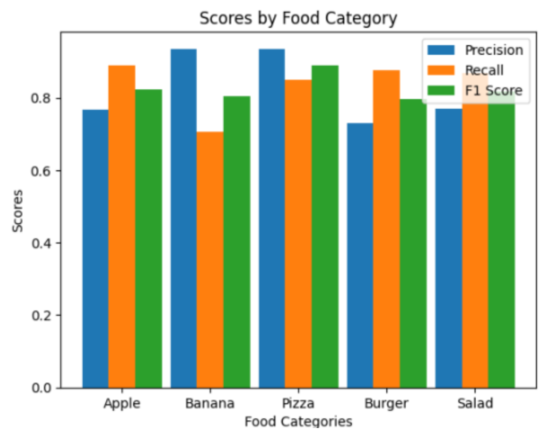

Fig. 3. Scores by food category.

The grouped bar plot from Fig. 3 shows the performance metrics of the hypothetical deep learning model for each food category - 'Apple', 'Banana', 'Pizza', 'Burger', 'Salad'. Each bar represents a performance metric score (between 0 and 1) for a specific food category.

Food Categories: The categories on the X-axis represent the different food classes that our model is trained to recognize. These are the categories against which the precision, recall, and F1 scores are evaluated. The Y-axis represents the values of precision, recall, and F1 scores. Each of these metrics provides different information about the model's performance: Precision measures the accuracy of positive predictions. In this context, it asks the question: "Of all the times the model predicted a specific food category, how often was it correct?" A high precision score indicates that when the model identifies an image as a certain food category, it is highly likely to be correct. Recall (or sensitivity) measures the true positive rate. It asks: "Of all the images that were truly a certain food category, how often did the model correctly identify it?" A high recall score means the model is good at detecting a specific food category but may also have a high rate of false positives.F1 Score is the harmonic mean of precision and recall. It tries to balance the two and is particularly useful when the distribution of classes is uneven.

It provides a single metric that encapsulates both precision and recall. Bars: Each group of bars corresponds to a food category, with individual bars representing precision, recall, and F1 score for that category. This graph allows us to easily compare these metrics across different categories. For example, you can compare precision for 'Apple' with 'Pizza' and check which food category the model is better at predicting accurately. High scores in these metrics indicate that your model is performing well in both correctly identifying the food categories (high precision) and not missing any food categories (high recall).

- X-Axis (Food Categories): The categories on the Xaxis represent the different food classes that our model is trained to recognize. These are the categories against which the precision, recall, and F1 scores are evaluated.
- Y-Axis (Scores): The Y-axis represents the values of precision, recall, and F1 scores. Each of these metrics provides different information about the model's performance:
- Precision Precision measures the accuracy of positive predictions. In this context, it asks the question: "Of all the times the model predicted a specific food category, how often was it correct?" A high precision score indicates that when the model identifies an image as a certain food category, it is highly likely to be correct.
- Recall Recall (or sensitivity) measures the true positive rate. It asks: "Of all the images that were truly a certain food category, how often did the model correctly identify it?" A high recall score means the model is good at detecting a specific food category but may also have a high rate of false positives.
- 1 F1 Score is the harmonic mean of precision and recall. It tries to balance the two and is particularly useful when the distribution of classes is uneven. It provides a single metric that encapsulates both precision and recall.
- Bars: Each group of bars corresponds to a food category, with individual bars representing precision, recall, and F1 score for that category.

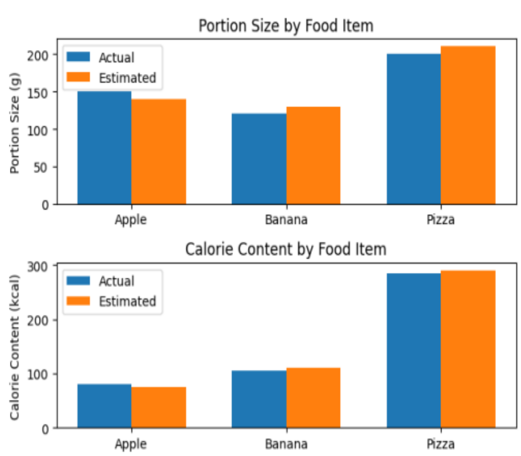

Fig. 4. Portion and calorie estimation.

This graph in Fig. 4 allows us to easily compare these metrics across different categories. For example, you can compare precision for 'Apple' with 'Pizza' and check which food category the model is better at predicting accurately. High scores in these metrics indicate that your model is performing well in both correctly identifying the food categories (high precision) and not missing any food categories (high recall). The two bar graphs form Fig 4 generated provides a comparative view of the estimated and actual values for portion sizes and calorie content of food items. Portion Size by Food Item: The first graph compares the actual portion size (in grams) with the estimated portion size of the three food items – 'Apple', 'Banana', and 'Pizza'. The X-axis represents the food items and the Y-axis denotes the portion size in grams. Each food item has two bars - one representing the actual portion size (blue) and the other depicting the estimated portion size (orange). If the model's estimations are perfect, the orange bar (Estimated) would coincide completely with the blue bar (Actual) for each food item. However, any difference between these bars indicates the discrepancy between the model's estimations and the actual values. Calorie Content by Food Item: The second graph performs a similar comparison but for the calorie content of the food items.

The X-axis represents the food items and the Y-axis denotes the calorie content in kilocalories (kcal). Similar to the first graph, each food item has two bars - one representing the actual calorie content (blue) and the other depicting the estimated calorie content (orange). Again, a perfect estimation would result in the complete coincidence of the blue and orange bars for each food item. Any deviation between these bars indicates an error in the model's calorie estimation. These plots can serve as a valuable tool for visually assessing the performance of your model. If the estimated bars closely match the actual bars, it indicates that your model is performing well in estimating the portion sizes and calorie content of food items. Conversely, a significant deviation might suggest that further improvements are needed.

VIII. Conclusion

This paper introduces a deep learning-based food calorie estimation method for dietary assessment. Transfer learning from pre-trained models like VGG16, InceptionV3, and ResNet50 is used to leverage the robustness of convolutional neural networks (CNNs) for food identification from images. These models, trained on food-specific image datasets, can recognize many food items. The model also estimates portion sizes creatively. The model could calculate food item dimensions from pixel area by using common objects in the image, such as a plate or fork. After computing volumes, a second deep learning model estimated portion sizes. A nutritional database was used to calculate the meal's calories. This method calculated a meal's total calories, improving dietary assessment. The proposed model is promising, but the problem is complex. Food preparation, presentation, and serving sizes affect the model's accuracy. For more accurate estimations in diverse real-world settings, the model must be refined and adapted. This paper proposes a promising deep learning approach for automated, accurate dietary assessment. By giving users an easy way to track their caloric intake, this work could impact healthcare, fitness, and diet planning. It could also open up new research in dietary assessment and health informatics.

REFERENCES

[1] Soojeong Lee; Joon-Hyuk Chang; "Oscillometric Blood Pressure Estimation Based on Deep Learning", IEEE Transactions On Industrial Informatics, 2017.

[2] Huaizhi Wang; Jiaqi Ruan; Guibin Wang; Bin Zhou; Yitao Liu; Xueqian Fu; Jianchun Peng; "Deep Learning-Based Interval State Estimation of AC Smart Grids Against Sparse Cyber Attacks", IEEE Transactions On Industrial Informatics, 2018.

[3] JWendong Zheng; Huaping Liu; Bowen Wang; Fuchun Sun; "CrossModal Surface Material Retrieval Using Discriminant Adversarial Learning", IEEE Transactions On Industrial Informatics, 2019.

[4] Chaoqun Hong; Jun Yu; Jian Zhang; Xiongnan Jin; Kyong-Ho Lee; "Multimodal Face-Pose Estimation With Multitask Manifold Deep Learning", IEEE Transactions On Industrial Informatics, 2019.

[5] Senxiang Lu; Jian Feng; Huaguang Zhang; Jinhai Liu; Zhenning Wu; "An Estimation Method of Defect Size From MFL Image Using Visual Transformation Convolutional Neural Network", IEEE Transactions On Industrial Informatics, 2019.

[6] Frank P.-W. Lo; Yingnan Sun; Jianing Qiu; Benny P. L. Lo; "Point2Volume: A Vision-Based Dietary Assessment Approach Using View Synthesis", IEEE Transactions On Industrial Informatics, 2020.

[7] Jing Qiu; Lei Du; Dongwen Zhang; Shen Su; Zhihong Tian; "NeiTTE: Intelligent Traffic Time Estimation Based on Fine-Grained Time Derivation of Road Segments for Smart City", IEEE Transactions On Industrial Informatics, 2020.

[8] Binyi Su; Haiyong Chen; Peng Chen; Guibin Bian; Kun Liu; Weipeng Liu; "Deep Learning-Based Solar-Cell Manufacturing Defect Detection With Complementary Attention Network", IEEE Transactions On Industrial Informatics, 2021.

[9] Han Qiu; Qinkai Zheng; Gerard Memmi; Jialiang Lu; Meikang Qiu; Bhavani Thuraisingham; "Deep Residual Learning-Based Enhanced JPEG Compression in The Internet of Things", IEEE Transactions On Industrial Informatics, 2021.

[10] Abdollah Kavousi-Fard; Wencong Su; Tao Jin; "A Machine-LearningBased Cyber Attack Detection Model for Wireless Sensor Networks in Microgrids", IEEE Transactions On Industrial Informatics, 2021.

[11] Xin Liu; Jianwei Zhao; Jie Li; Bin Cao; Zhihan Lv; "Federated Neural Architecture Search for Medical Data Security", IEEE Transactions On Industrial Informatics, 2022.

[12] S. Amin; Hamdi Altaheri; G. Muhammad; Mansour Alsulaiman; A. Wadood; "Attention-Inception and Long- Short-Term Memory-Based Electroencephalography Classification for Motor Imagery Tasks in Rehabilitation", IEEE Transactions On Industrial Informatics, 2022.

[13] X. Kong; Gaohui Duan; Mingliang Hou; Guojiang Shen; Hongya Wang; Xiaoran Yan; M. Collotta; "Deep Reinforcement Learning-Based Energy-Efficient Edge Computing for Internet of Vehicles", IEEE Transactions On Industrial Informatics, 2022

[14] S. Horng; Dinh-Trung Vu; Thi-Van Nguyen; Wanlei Zhou; Chin-Teng Lin; "Recognizing Palm Vein in Smartphones Using RGB Images", IEEE Transactions On Industrial Informatics, 2022.

[15] Yanxu Liu; Yu Wang; Tommy W. S. Chow; Baotong Li; "Deep Adversarial Subdomain Adaptation Network for Intelligent Fault Diagnosis", IEEE Transactions On Industrial Informatics, 2022.

[16] Benyi Yang; Zhenyu Liu; Guifang Duan; Jianrong Tan; "Mask2Defect: A Prior Knowledge-Based Data Augmentation Method for Metal Surface Defect Inspection", IEEE Transactions On Industrial Informatics, 2022.

[17] H. Samet; Saeedeh Ketabipour; M. Afrasiabi; Shahabodin Afrasiabi; Mohammad Reza Mohammadi; "Deep Learning Forecaster-Based Controller for SVC: Wind Farm Flicker Mitigation", IEEE Transactions On Industrial Informatics, 2022

[18] Susruth Sudhakaran; Karl Montgomery; M. Kashef; D. Cavalcanti; R. Candell; "Wireless Time Sensitive Networking Impact on An Industrial Collaborative Robotic Workcell", IEEE Transactions On Industrial Informatics, 2022.

[19] Yinjun Wang; Xiaoxi Ding; Rui Liu; Y. Shao; "ConditionSenseNet: A Deep Interpolatory ConvNet for Bearing Intelligent Diagnosis Under Variational Working Conditions", IEEE Transactions On Industrial Informatics, 2022.

[20] Ranran Wang; Yin Zhang; Limei Peng; G. Fortino; P. Ho; "TimeVarying-Aware Network Traffic Prediction Via Deep Learning in IIoT", IEEE Transactions On Industrial Informatics, 2022.