Design of a Smart Meal Choice Assistant for University Students using Artificial Intelligence

Anamta Ansari
Department of Computer Science
California State University, Fullerton
Fullerton, California 92831, USA
anamtaansari@csu.fullerton.edu

Tabashir Nobari
Department of Public Health
California State University, Fullerton
Fullerton, California 92831, USA
tnobari@fullerton.edu

Anand Panangadan
Department of Computer Science
California State University, Fullerton
Fullerton, California 92831, USA
apanangadan@fullerton.edu

Abstract—**Campus dining significantly influences student health, academic performance, and well-being. Limited access to nutritional information at the point of purchase complicates informed dietary choices. This study describes three artificial intelligence (AI) approaches to automatically classify and provide nutritional data for campus meals. The approaches use deep learning neural networks for object recognition that are trained using a custom-built dataset. Specifically, three approaches were implemented and evaluated: (1) object detection with Roboflow/YOLO v8, (2) image classification using MobileNetV2, and (3) Optical Character Recognition (OCR)-based menu text analysis. The methods were quantitatively evaluated using information available from public nutrition information sources for university campus restaurants. Our evaluation shows that these systems are able to categorize meals into healthy, moderately healthy, and unhealthy, and can provide comprehensive nutrition metrics. Preliminary results demonstrate that the Roboflow/YOLOv8-based object detection models outperform the other models. These AI-driven solutions can enable students to make healthier dietary choices at the point of purchase, potentially improving health outcomes among university populations.**

Index Terms—**Computer Vision, Nutrition Assessment, Campus Dining, OCR, Tesseract, AI-assisted health**

I. Introduction

The nutritional quality of meals consumed by university students plays a critical role in their overall health, academic performance, and well-being. However, campus dining environments often present challenges for making informed dietary choices due to limited availability of clear nutritional information at the point of purchase. Although many campus restaurants provide nutritional information online or through printed materials, this information is rarely accessible when students make real-time meal decisions.

Recent studies have highlighted concerning trends in college students' dietary patterns, showing a frequent consumption of meals high in calories, saturated fats, and added sugars, yet low in essential nutrients. Such dietary habits have been linked to multiple health issues, including weight gain, decreased energy levels, and poorer academic outcomes.

The rapid development of artificial intelligence (AI) technologies, particularly computer vision and machine learning, presents new opportunities to overcome these challenges. AI-driven systems can potentially recognize meals from images and instantly deliver accurate nutritional information, helping students make healthier dietary decisions. However, effectively developing such systems involves addressing multiple technical challenges, including: (1) creating comprehensive datasets that accurately reflect campus dining options, (2) developing robust classification models capable of identifying various dishes across diverse presentation styles, and (3) integrating nutritional data into visual recognition systems.

This research aims to develop and evaluate three distinct AI approaches —object detection, image classification, and text analysis of menu — to estimate health quality of meals available on university campus restaurants. By comparing these approaches, we seek to identify the most effective methods for automated nutritional assessment.

The contributions of this work include (1) Creating a comprehensive dataset of campus meals paired with accurate nutritional data, (2) Developing three AI-based methods for real-time meal identification and nutritional assessment, and (3) Evaluating the real-world performance of these approaches in university campus dining contexts.

II. Related Work

We synthesize key findings from recent studies exploring AI-driven solutions for dietary monitoring, food quality assessment, and meal analysis.

A. Food Recognition and Classification Systems

Vision-based approaches for automatic food recognition have shown promising results. Subhi et al. [1] highlighted challenges in segmentation and mixed-food recognition, underscoring the superiority of deep learning models in dietary assessment. Transfer learning techniques have been used for nutrition assessment from images. Mezgec and Seljak [2] modified the GoogLeNet architecture for food recognition and dietary assessment, achieving 86.5% accuracy on a custom image dataset. Rohini et al. [3] integrated the CNN and VGG16 models, and combined the architecture with OCR for allergen detection, achieving 97.37% accuracy on images of fruits and vegetables.

B. Nutritional Content Estimation and Dietary Assessment

Advanced dietary monitoring applications have leveraged OCR and CNN methodologies. FoodScan [4] utilizes OCR for simplifying food logging with high usability. goFOOD [5] integrates CNNs and Structure from Motion for real-time nutritional analysis. Experimental results indicate that these approaches are effective in tracking dietary habits.

Ju et al. [6] developed MenuAI, a menu analysis system that uses a transformer-based model for dynamic cafeteria environments. Abdullah et al. [7] describe Food Code Breaker (FCB) that uses OCR technology and image libraries for multilingual dietary inclusivity and ingredient analysis.

C. OCR Applications in Food and Nutritional Analysis

OCR has increasingly facilitated extracted of nutrition data from textual sources. Najam et al. [8] developed a CNN-LSTM-CTC model for OCR of handwritten text. Their model also emphasizes the importance of error correction for nutrition assessments. Myers et al. [9] and Ege and Yanai [10] have integrated OCR technology with food recognition systems to estimate nutritional content from images.

III. Approach

A. Dataset Creation

Nutritionix.com provided comprehensive nutritional data for the major campus restaurants—Togo's, Baja Fresh, Pieology, Panda Express, and Carl's Jr.—available primarily in PDF format. This required significant pre-processing to ensure consistency and convert the data into a structured Excel dataset. The dataset included detailed nutritional profiles such as calories, calories from fat, total fat (g), saturated fat (g), trans fat (g), cholesterol (mg), sodium (mg), total carbohydrates (g), dietary fibre (g), sugars (g), and protein content (g).

Each dish was categorized as "healthy," "moderately healthy," or "unhealthy" based on standardized nutritional guidelines. The classification criteria considered multiple nutritional factors rather than calories alone, providing a more holistic health assessment. Healthy meals are characterized by lower calorie content, reduced saturated fat and sodium levels, and a high presence of fiber, lean protein, and essential nutrients. Moderately Healthy meals have a balanced composition, containing moderate amounts of calories, fat, and sodium while still offering some beneficial nutrients. In contrast, Unhealthy meals are identified by their high levels of saturated fat, trans fat, sodium, and added sugars, with minimal fiber content, making them less suitable for a nutritious diet.

Image data was collected from multiple sources including dish images collected at California State University, Fullerton, restaurant review images from publicly available sources to represent real-world meal presentations and promotional photographs directly from the restaurants.

We implemented three distinct approaches each leveraging different aspects of computer vision and machine learning.:

1) Object Detection (Roboflow/YOLO v8): Our first approach utilized object detection to identify specific dishes within images. We employed Roboflow 3.0's object detection framework, which is derived from YOLOv8 (You Only Look Once), a state-of-the-art real-time object detection system. YOLOv8 utilizes a CSPDarknet-based backbone with Cross Stage Partial (CSP) connections, which enhance gradient flow while reducing computational redundancy, ensuring efficient feature extraction. Its Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) enable multi-scale feature fusion, allowing for accurate detection of objects at various sizes. Additionally, YOLOv8 replaces the traditional ReLU activation function with SiLU (Swish), which provides smoother gradients and improves learning in regions with small gradient updates.

a) Roboflow Architecture: Roboflow's architecture is designed to streamline and enhance object detection and image classification tasks by leveraging deep learning models trained on large-scale datasets such as Microsoft COCO and ImageNet.

Microsoft COCO (Common Objects in Context) is a widely used dataset containing over 330,000 images with 80 object categories, annotated with bounding boxes and segmentation masks. It serves as a benchmark for object detection models, providing diverse real-world images that enhance model generalization. ImageNet, in contrast, is a large-scale image classification dataset with over 14 million images spanning 1,000 object classes, primarily used for training convolutional neural networks (CNNs) for feature extraction and classification.

Roboflow integrates pretrained models from both datasets, allowing for transfer learning to enhance detection accuracy and efficiency. Models trained on COCO are optimized for object detection, enabling them to recognize multiple objects within an image using architectures like YOLO (You Only Look Once) and Faster R-CNN. These models benefit from COCO's fine-grained annotations, making them ideal for real-world applications such as food recognition, medical imaging, and autonomous systems. On the other hand, ImageNet-trained models, such as ResNet, MobileNet, and EfficientNet, provide robust feature representations that are useful for image classification tasks. Roboflow enables developers to fine-tune these models on custom datasets, leveraging ImageNet's hierarchical classification structure to improve recognition performance in domain-specific applications.

b) Model Implementation: The implementation followed these steps:

i. Dataset Preparation: Images were annotated with bounding boxes around individual dishes. The annotations included dish name, calorie count, and health category. The COCO format was used as the checkpoint for initial model training, with pretrained weights initialized from the COCO dataset, which includes 80 classes and 330,000 images.

ii. Preprocessing Operations: Images were normalized with pixel values scaled to [0,1] range and standardized to μ=0, σ=1 using ImageNet statistics. Systematic adjustments to hue (±10°), saturation (0.5-1.5×), and brightness (0.5-1.5×) were performed to correct photometric distortion. We used a progressive Training Strategy. We initially trained using 93 annotated images. In the second phase, we expanded the training dataset to 154 images. In the final phase, we added all 305 images.

iii. Model Configuration: The base architecture used was YOLO v8n (nano version for efficiency) with a batch size of 16 and learning rate of 0.01 with cosine decay. Augmentations performed included random rotation, zoom, brightness adjustment, contrast, color, horizontal flips, crops, etc.

2) Image Classification with MobileNetV2: Our second approach treated dish identification as an image classification problem using transfer learning with MobileNetV2 on actual food images, where each dish category corresponds to a class label. It connects food images to nutritional data loaded from an Excel file, trains a model on real images, and saves predictions, along with dish details, into a CSV file. MobileNetV2 pre-trained on ImageNet (a huge general-purpose image dataset) was chosen due to its efficiency for embedded applications and to leverage its existing knowledge about general image features. The base layers were freezed to prevent altering pretrained weights during initial training.

i. Dataset Preparation and Preprocessing:
Images were organized into class directories based on dish names. Preprocessing included resizing all images to 224×224 pixels and rescaling pixel values to [0,1]. We implemented data augmentation for the training set where rotation range was set to 20 degrees, width and height shifts to 0.2, shear range to 0.2, zoom range to 0.2, horizontal flip was set to True, and fill mode was set to 'nearest'.

ii. Model Architecture
We employed transfer learning using MobileNetV2, a lightweight CNN architecture designed for mobile and embedded vision applications.

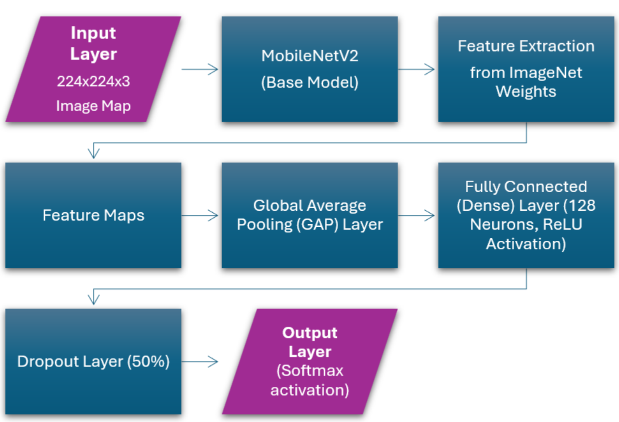

Fig. 1. Architecture Block Diagram for MobileNetV2

The model architecture is built on MobileNetV2, a pre-trained convolutional neural network (CNN) with ImageNet weights, serving as the base model for feature extraction. It processes input images of size (224, 224, 3) while keeping its base layers frozen during the initial training phase to retain pre-learned features. The extracted features are passed through a custom classification head, which includes a Global Average Pooling layer to reduce spatial dimensions, followed by a fully connected (Dense) layer with 128 neurons and ReLU activation for feature transformation. To prevent overfitting, a Dropout layer (0.5) is applied before the final output layer, which uses softmax activation with neurons corresponding to the number of dish classes for multi-class classification. The model is compiled using the Adam optimizer, with categorical cross-entropy as the loss function and accuracy as the evaluation metric.

iii. Training and Evaluation
The model was trained with epochs set to 10, batch size to 32 and the train-validation split was 80%-20%. During evaluation, predictions with confidence below 0.5 were labeled as "Uncertain Prediction" to prevent misclassification.
iv. Metadata Retrieval
After classification, the system retrieved nutritional information by mapping the predicted class to the corresponding dish name, then looking up the dish in the Excel dataset to obtain calories and health category and returning "Unknown" for metadata when no exact match was found.

3) Menu Text Analysis with OCR: Our third approach focused on extracting and analyzing text from menu images to identify dishes and their nutritional attributes. We developed an OCR-based system using Tesseract, embedded within a Flask web application. Images underwent preprocessing (grayscale conversion and thresholding), and text was cleaned and matched against nutritional data. Extracted menu items were classified into health categories with caloric data. Recognizing the challenges posed by visual similarity, lighting variations, and potential image quality issues, the OCR system also allows manual input of dish names.
i. Tesseract OCR Implementation
We implemented a Flask-based web application using Tesseract OCR for text extraction. Image Preprocessing included conversion to grayscale and binary thresholding (threshold value: 150). Text was extracted using Tesseract OCR engine (version 4.1.1). Finally text cleaning and processing included removal of non-alphanumeric characters, elimination of multiple spaces, removal of dietary indicators (VG, Vegan, GF, etc.) and case-insensitive matching

ii. Dish Classification Methodology
The extracted text was processed to identify dishes. Clean text was compared against entries in the dish_categories.xlsx file. For each match, the system retrieved the Dish name, Health category (healthy, moderately healthy, unhealthy) and Calorie count. The results were returned as a JSON response with a list of identified dishes with their categories and calories, status indicator (success/error) and raw extracted text for verification purposes iii. Web Application Integration The OCR system was wrapped in a Flask web application allowing users to upload their menu images, receive immediate analysis results, and view both the uploaded image and the classified dishes. Results were color-coded by respective health category (red for unhealthy, yellow for moderately healthy and green for healthy)

IV. Results and Discussion

A. *Roboflow 3.0 Object Detection Model*

Out of the three successive models trained on the Panda Express dataset, the final version had Mean Average Precision (mAP) of 55.3%, Precision of 72.4%, and Recall of 49.2% showing significant improvement compared to the previous two models. Figure 2 shows the precision scores for the 4 models.

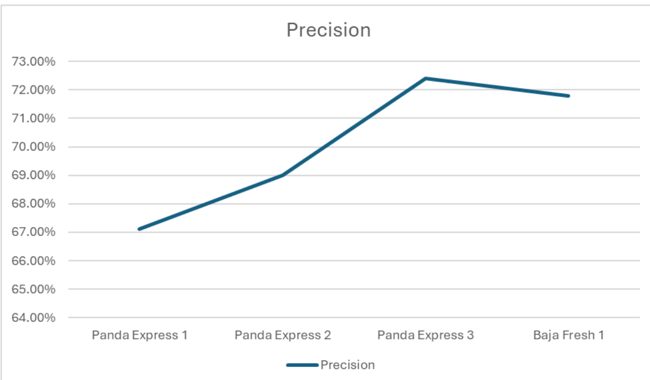

Fig. 2. Comparing successive Roboflow models by their Precision Scores.

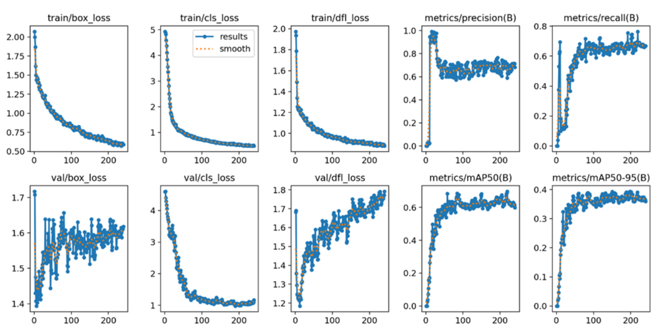

Fig. 3. Loss curves and Performance Metrics (Baja Fresh).

Results shown in Figure 3 show that Train Box, Class, and DFL Losses steadily decrease, indicating the model learns effectively from training data. Validation Box Loss initially decreases as shown in Figure 3, but later becomes noisy and slightly increases, signaling potential overfitting or difficulty in predicting boxes accurately on new data. Validation Class Loss consistently decreases sharply and remains stable, indicating strong classification performance. Validation DFL Loss initially decreases but then rises significantly over time, suggesting bounding box accuracy deteriorates, again hinting at slight overfitting.

Precision and Recall rise and stabilize at moderate to high levels, indicating decent prediction performance. The mean Average Precision at 0.5 Intersection over Union (mAP@50) measures the accuracy of predicted bounding boxes when they overlap with ground truth boxes by at least 50%, commonly used in object detection tasks. mAP50 rapidly increases and stabilizes around 0.6 (60%), showing reasonable object detection accuracy. mAP50-95 stabilizes around 0.3-0.4 (30-40%), indicating that while the model performs well at IoU of 50%, it struggles at stricter thresholds (common for most models).

Overall, the model demonstrates effective training behavior, successfully learning robust classification skills. Validation metrics indicate reasonable detection accuracy; however, there are signs of slight overfitting, as evidenced by gradually increasing box and Distribution Focal Losses (DFL) on the validation set. Additionally, the model shows strong performance at standard detection thresholds (mAP50) but exhibits weaker accuracy at stricter thresholds (mAP50-95), which is typical in practical, real-world object detection scenarios.

Overall performance:

- Validation mAP@50: 79%
- Test mAP@50: 79%

The model performs consistently and strongly overall, with identical validation and test set performance of 79% (see Figure 4 and Table II .

The model's performance was evaluated based on mean Average Precision (mAP@50) for both validation and test datasets across different food categories. Black Beans (150 cal) achieved a perfect detection score of 100% on both validation and test sets, indicating excellent and highly reliable recognition. Lettuce (5 cal) showed strong performance, with a validation score of 79% and a perfect 100% on the test set, reflecting notable improvement during testing. Similarly, Lettuce Cabbage Mix (5 cal) improved from 69% in validation to 87% in testing, demonstrating moderate enhancement. Chicken (280 cal) maintained consistently high accuracy, scoring 92% in validation and 100% in testing, confirming its robustness across datasets. Mexican Rice (130 cal) was only evaluated in the test set, achieving 83%, which suggests good reliability despite the absence of a validation score. Overall, these results highlight the model's strong generalization capability and strong potential for accurate practical implementation.

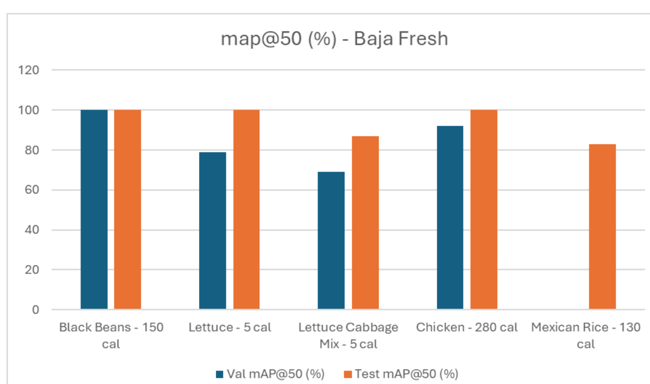

Fig. 4. Validation and test performance on dishes in Baja Fresh.

Table I shows that the Roboflow 3.0 object detection models trained for Panda Express demonstrated progressive improvement in performance through successive training

TABLE I
PERFORMANCE COMPARISON OF ROBOFLOW OBJECT DETECTION MODELS

Model Name	mAP	Precision	Recall	Total Images	Model Type	Checkpoint
Panda Express 1	22.8%	67.1%	17.2%	93	Roboflow 3.0 Object Detection	COCOn
Panda Express 2	20.3%	69.0%	15.3%	154	Roboflow 3.0 Object Detection	Panda Express 1
Panda Express 3	55.3%	72.4%	49.2%	302	Roboflow 3.0 Object Detection	Panda Express 1
Baja Fresh 1	68.3%	71.8%	64.4%	220	Roboflow 3.0 Object Detection	COCOn

TABLE II
CLASS-WISE PERFORMANCE ANALYSIS (MAP@50)

Class (Category)	Validation mAP@50 (%)	Test mAP@50 (%)
Black Beans - 150 cal	100	100
Lettuce - 5 cal	79	100
Lettuce Cabbage Mix - 5 cal	69	87
Chicken - 280 cal	92	100
Mexican Rice - 130 cal	-	83

TABLE III
IMAGE DETECTION PERFORMANCE BY TESSERACT

#	Image Name (Characteristics)	Total Dishes	Correctly Detected	Accuracy (%)
1	PE M5 (tilted image, clear text)	8	7	87.5
2	PE M3 (tilted image)	5	5	100
3	Entrees (white text)	3	0	0
4	Entrees4 (yellow tint, blurred)	8	7	87.5
5	PE Broccoli Beef	1	1	100
6	PE MM3	6	4	66.7
7	PE M2 (clear)	3	3	100
8	M1 (clear text)	20	19	95
9	M5 (glare, blurred text)	4	3	75
10	M3 (glare, unclear text)	6	6	100

iterations and incremental dataset expansion. The initial Panda Express model (Panda Express 1), trained on 93 images using a COCO checkpoint, achieved a mean Average Precision (mAP) of 22.8%, precision of 67.1%, and recall of 17.2%. The subsequent iteration (Panda Express 2), despite being trained on a larger dataset of 154 images initialized from Panda Express 1, showed a slight drop in performance (20.3% mAP, 69% precision, 15.3% recall), likely reflecting increased dataset complexity or challenging new examples. However, this incremental training approach ultimately proved beneficial, as the third model iteration (Panda Express 3), trained on an expanded dataset of 302 images, significantly improved performance to 55.3% mAP, 72.4% precision, and 49.2% recall, indicating successful adaptation and generalization. Additionally, the Baja Fresh model, trained independently with 220 images and initialized from the COCO checkpoint, achieved strong results (68.3% mAP, 71.8% precision, and 64.4% recall), further validating the effectiveness of data quality and adequate representation of object classes in improving object detection performance.

B. Tesseract OCR

1) Dish-level Accuracy Analysis: The overall accuracy of the Tesseract OCR model at the dish-level was calculated as:

Overall Accuracy = (Total Correctly Detected / Total Dishes) × 100

With a total of 55 correctly detected dishes out of 64 total dishes (see Table III), the computed accuracy is:

Overall Accuracy = (55/64) × 100 = 85.94%

The model is precise with a perfect Average Precision score of 1.0, which means that it did not generate any incorrect dishes, but it missed some actual dishes which lowered the recall (0.812) and F1-score (0.850).

Figure 5 shows the total number of dishes versus the number of correctly detected dishes using the Tesseract OCR model across 10 distinct images. This visualization clearly highlights strong performance on most images, with several images (Img 2, Img 5, Img 7, Img 10) achieving perfect accuracy (100%). Other images, despite challenges such as glare or blurring, also displayed good performance, while a few (notably Img 3) demonstrated poor performance due to significant readability issues. Figure 6 illustrates the

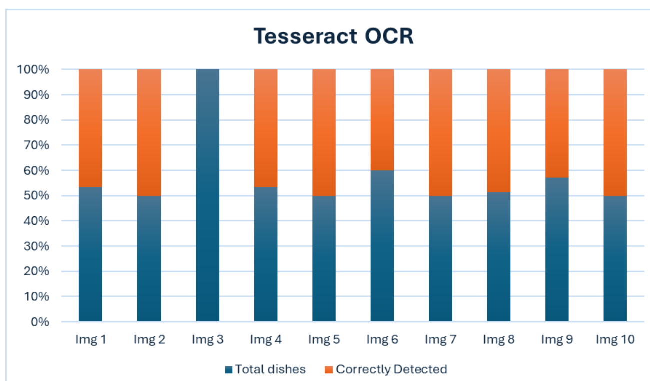

Fig. 5. Total and correctly detected dishes for 10 sample images.

relationship between the total number of dishes present in each image and the OCR model's accuracy percentage. The trend depicted shows generally high accuracy, typically ranging from 75% to 100%, with one prominent dip at Img 3 (0%), indicating that severe image-quality issues significantly impacted model accuracy. Notably, the number of dishes in an image does not directly correlate to performance, as both small and large numbers of dishes produced high accuracy in clear conditions.

The Tesseract OCR model demonstrated strong overall robustness, achieving an accuracy of approximately 86%. The results highlight that clear and adequately captured images consistently resulted in near-perfect accuracy (95–100%), whereas declines in accuracy were predominantly due to poor image quality and readability.

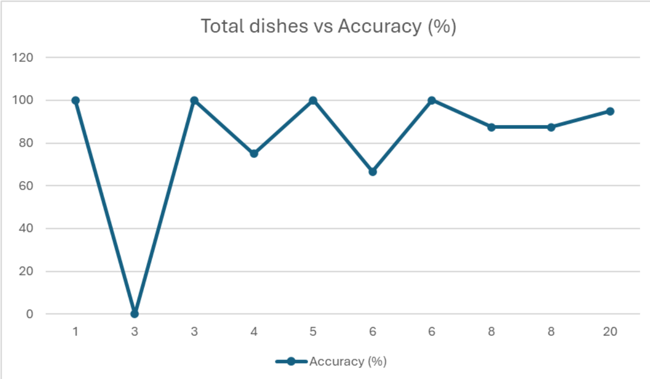

Fig. 6. Total number of dishes per image for 10 sample images vs Accuracy.

C. MobileNetV2 model

During initial training epochs, the MobileNetV2-based model achieved approximately 92% training accuracy and around 89% validation accuracy. However, upon deeper evaluation using precision, recall, and F1-score metrics, performance significantly declined. Specifically, due to the limited size of the validation dataset. The model showed poor class-level predictive capabilities, reflected by precision, recall, and F1-score metrics nearing zero in 2 out of 5 classes. Only one class, 'Super Greens,' achieved moderate accuracy (precision, recall, and F1-score = 0.50). This discrepancy indicates the validation accuracy initially reported was inflated by the small sample size and insufficient to represent true model performance or generalizability.
Due to initial limitations observed in dataset size and class imbalance, additional images per class were collected, and model fine-tuning was implemented (97.5% accuracy during second training epochs but validation accuracy dropping to 56.14%). These adjustments resulted in slightly improved classification accuracy clearly indicating the effectiveness of transfer learning combined with fine-tuning and proper dataset management. However, the overall accuracy of approximately 30% was still significantly low compared to the alternative approaches.

V. Conclusions

This study demonstrates the feasibility of employing AI-driven methods for campus meal recognition and nutritional assessment, highlighting the significant potential to improve dietary choices among university students. Among the three approaches tested—YOLOv8-based object detection, Tesseract OCR for menu text analysis, and MobileNetV2-based classification—object detection exhibited moderate to high mean Average Precision (mAP upto 79%), successfully identifying specific dishes across varying presentation styles. The Tesseract OCR approach provided robust text extraction and dish identification whenever image clarity and lighting conditions were adequate, proving particularly useful for menu-based analysis. Although the MobileNetV2 model showed high training accuracy (more than 90%), it suffered from overfitting and class imbalance within a smaller dataset, underscoring the importance of additional data collection and more rigorous fine-tuning of the architecture.

A. Limitations

Although promising, our approaches have certain limitations. The current dataset is restricted to a few campus dining vendors, which may limit generalizability. The MobileNetV2 classifier suffered from class imbalance and small dataset size, and OCR results were highly sensitive to image clarity. Additionally, object detection methods like YOLOv8 require significant computational resources compared to the other two approaches. Finally, ethical considerations, such as potential biases in AI-based dietary guidance, protecting user-uploaded images, and meal data for data privacy, can be explored in future work.

Acknowledgment

This work is supported by the Agriculture and Food Research Initiative, project award no. 2023-69015-40684, from the U.S. Department of Agriculture's National Institute of Food and Agriculture.

References

[1] M. A. Subhi, S. H. Ali, and M. A. Mohammed, "Vision-based approaches for automatic food recognition and dietary assessment: A survey," *IEEE Access*, vol. 7, pp. 35 370–35 381, 2019.
[2] S. Mezgec and B. K. Seljak, "Nutrinet: A deep learning food and drink image recognition system for dietary assessment," *Nutrients*, vol. 9, no. 7, p. 657, 2017.
[3] B. Rohini, D. M. Pavuluri, L. Naresh Kumar, V. Soorya, and J. Aravinth, "A framework to identify allergen and nutrient content in fruits and packaged food using deep learning and ocr," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2021, pp. 72–77.
[4] B. Sainz-De-Abajo, J. M. García-Alonso, J. J. Berrocal-Olmeda, S. Laso-Mangas, and I. De La Torre-Díez, "Foodscan: Food monitoring app by scanning the groceries receipts," *IEEE Access*, vol. 8, pp. 227 915–227 924, 2020.
[5] Y. Lu, T. Stathopoulou, M. F. Vasiloglou, L. F. Pinault, C. Kiley, E. K. Spanakis, and S. Mougiakakou, "gofoodtm: An artificial intelligence system for dietary assessment," *Sensors*, vol. 20, no. 15, 2020.
[6] X. Ju, F. Lo, J. Qiu, P. Shi, J. Peng, and B. Lo, "Menuai: Restaurant food recommendation system via a transformer-based deep learning model," 10 2022.
[7] S. Abdullah, M. Elshaikh, M. B. Mahmood, N. Nordin, S. Z. Syed Idrus, R. Mahadi, and U. N. Saraih, "Food code breaker (fcb) – developing the multi-lingual food code translation android application using multi– options code reader system," *Journal of Physics: Conference Series*, vol. 1019, no. 1, p. 012013, jun 2018.
[8] R. Najam and S. Faizullah, "Analysis of recent deep learning techniques for arabic handwritten-text ocr and post-ocr correction," *Applied Sciences*, vol. 13, no. 13, 2023.
[9] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2calories: Towards an automated mobile vision food diary," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1233–1241.
[10] T. Ege and K. Yanai, "Image-based food calorie estimation using knowledge on food categories, ingredients and cooking directions," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, ser. Thematic Workshops '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 367–375.