Calorie Estimation of Solid Food Using Machine Learning with YOLO Models

Minar Mahmud Rafi
Department of Computer Science & Engineering
Brac University
Dhaka,Bangladesh
minar.mahmud.rafi@g.bracu.ac.bd

William Ankan Sarkar
Department of Computer Science & Engineering
Brac University
Dhaka,Bangladesh
william.ankan.sarkar@g.bracu.ac.bd

Mohammad Zishan Tareque
Department of Computer Science & Engineering
Brac University
Dhaka,Bangladesh
mohd.zishan.tareque@g.bracu.ac.bd

Marium Khan Momo
Department of Computer Science & Engineering
Brac University
Dhaka,Bangladesh
marium.khan.momo@g.bracu.ac.bd

Farah Binta Haque
Department of Computer Science & Engineering
Brac University
Dhaka,Bangladesh
farah.binta.haque@g.bracu.ac.bd

Md. Sabbir Hossain
Department of Computer Science & Engineering
Brac University
Dhaka,Bangladesh
ext.sabbir.hossain@bracu.ac.bd

Annajiat Alim Rasel
Department of Computer Science & Engineering
Brac University
Dhaka,Bangladesh
annajiat@bracu.ac.bd

***Abstract*—In today's era of abundant data from diverse sources like social media, blogs, and newspapers, the quest for machines mirroring human capabilities is a reality. This study harnesses YOLO (You Only Look Once) models to detect and differentiate various solid food items encountered daily, aiming to estimate their calorie content. The unique focus of this research lies in utilizing existing technologies, specifically the YOLO model, to categorize manually countable solid food items and estimate their total calories. To achieve our objective, we curated a specialized dataset exclusively centered on solid food items, ensuring varied representations and ample data per category. This bespoke dataset formed the foundation for employing the YOLO model's object detection capabilities.Utilizing YOLO models, we rigorously evaluated their effectiveness in detecting and categorizing these food items, subsequently estimating their calorie content. Our emphasis was on leveraging existing technology rather than creating new models. Through meticulous experimentation with YOLO variants, encompassing different complexities and configurations, we gauged their accuracy in identifying diverse and intricate food images. Comparative analysis against existing works reaffirmed the efficacy of employing YOLO models for solid food detection and calorie estimation, showcasing the pragmatic application of established technology in this domain.**

***Index Terms*—Artificial Intelligence (AI); Computer Vision (CV); You Look Only Once (YOLOv5); Food Detection; Neural Network; Deep Learning**

I. Introduction

In recent years, the realm of image analysis has witnessed exponential growth, propelled by advancements in Computer Vision and AI technologies. Food, as a vital component of human sustenance, embodies cultural diversity through myriad unique recipes found across the globe. The global awareness of unfamiliar foods, facilitated by social media, parallels an escalating consciousness regarding health concerns tied to dietary habits [1]. This confluence emphasizes an increasing need for precise food recognition and understanding amid general consumption, dietary considerations, and fitness pursuits.

The proliferation of smartphones with high-resolution cameras has transformed food photography into a popular social media trend, offering an avenue for implementing food recognition technologies [2]. Beyond mere identification, these technologies hold promise for evolving into tools offering calorie information, divulging food origins, and sharing recipes. The data shared across social platforms presents an opportunity for corporations to track eating trends, predict food popularity, estimate sales, and tailor marketing strategies for new food-related products

In this paper, we delve into the implementation of modern Convolutional Neural Networks (CNNs) specifically aimed at detecting various solid food items and estimating their calorie content. Object detection systems using machine learning have persistently addressed real-world problems. However, detecting and characterizing solid foods pose unique challenges owing to their varied shapes, sizes, textures, and varying quantities within datasets [3]. These complexities mandate significant data, memory, and computational resources for precise detection, especially when multiple food objects coexist within a dataset.

Dataset	Class	Images	Year	Country
FoodNet	50	5000	2017	India
Custom	7	700	2020	Bangladesh
PFD	100	4928	2020	Pakistan
TBFI	7	2835	2021	Bangladesh
IndianFood20	20	17817	2022	India
CAFD	42	16449	2023	Asia
Ours	9	6574	2023	Bangladesh

TABLE I
DATASET SPECIFICATIONS

Existing large datasets predominantly cover Western or Chinese foods, lacking specificity and diversity concerning solid food items [4]. To address this gap, our research focuses on curating a comprehensive dataset exclusively comprising images of various solid food items. Our endeavor aims to create a sizable and distinctive dataset encompassing a diverse array of solid foods from different cultural backgrounds, avoiding specificity to any one region.

The deficiency in large, diverse datasets specific to South Asian cuisine is evident from existing repositories [4]. Notably, efforts like the TBFI dataset [5], comprising seven traditional Bangladeshi food classes, and the custom dataset [6] focused on a smaller scale with seven classes from Bangladesh have showcased accuracies of 86.0% and 95.2% respectively using transfer learning-based CNN models.

Other attempts, such as the CAFD dataset [7], encompassing 42 classes from Asia, and the PFD dataset [9], consisting of 100 classes from Pakistani culture, highlighted challenges in balancing dataset diversity and achieving accuracy, achieving 88.7% and 69.38% accuracy, respectively.

Various models like MobileNetV2[12], InceptionV3[13], DenseNet-201[14], and combinations of AlexNet[15], GoogleNet[16], and ResNet[17] have been employed on datasets like IndianFood10 and IndianFood20, showcasing accuracies ranging from 73.5% to 91.8% [11-17].

The adoption of the YOLOv5 algorithm stems from its enhanced accuracy and training efficiency over previous models. Leveraging bounding box coordinates and labeling during annotation, this algorithm facilitates real-time object detection, augmenting neural network referencing and Multi-Object Tracking (MOT) capabilities [19-20].

This paper meticulously documents the evolution of our dataset, models, and their comparative performance against existing research endeavors. In the larger framework of food recognition, we hope to further solid food detection and calorie estimation technologies through this investigation.

II. LITERATURE REVIEW

Rapid advancements in image recognition and classification technologies have ushered in a new era of refined applications, marked by notable improvements in accuracy. Despite these strides, the accurate identification of food items remains a persistent challenge within the realm of object recognition. The intricate nature of food recognition presents complexities that have hindered numerous proposed methods, resulting in suboptimal classification accuracies. However, the detection of food items holds immense significance across various industrial operations and boasts potential applications in health, fitness, and dietary domains.

Diving deeper into specific experiments, several studies have utilized YOLOv2 for food detection, primarily focusing on the categorization of Japanese cuisine [21]. These endeavors not only sought to identify food items but also gave rise to applications like Food Tracer. However, it's crucial to note that these applications were primarily designed for research purposes rather than practical real-world implementations. Subsequently, the development of YOLOv3 by J. Redmon & Farhadi [22] aimed to enhance accuracy by leveraging the Darknet 53 framework for improved feature extraction. Larger data batches caused performance problems, although significant progress was achieved in achieving optimal floating-point operations per second. When YOLOv3 was improved and compared to the previous version using the UECFOOD100 dataset for Asian food recognition, the improved version performed much better in terms of MAP (Mean Average Precision) scores.

Further studies focused on algorithmic comparisons, pitting various versions of YOLO against RCNN, emphasizing the importance of speed without compromising accuracy [24]. Researchers dedicated efforts to enhance the accuracy of identifying specific cuisines, such as Thai Food, by augmenting neural network layer counts [25]. Despite these efforts, the Inception network [26] stood out for its pivotal role in advancing CNN classifiers, achieving a noteworthy accuracy rate of 68.7% despite modifications using NU Inception modules. Additionally, studies underscored the efficacy of functions like Categorization and Segmentation when deployed in conjunction with Deep CNNs [27].

Efforts to gauge meal size detection deployed a revised U-Net rooted in CNN, segmenting specific food regions for accurate labeling [28]. However, challenges persisted due to dataset limitations, including image capture orientation and sensitivity. Introducing Support Vector Machine (SVM) techniques to resolve multi-class issues [29], coupled with CNN features, facilitated the successful classification of unique fast-food datasets into ten distinctive categories [30]. Further comparative analyses showcased the effectiveness of techniques such as combining classifiers K-NN and SVM and CNN4096, with the latter yielding optimal results through posterior probability techniques [31].

The evaluation of GoogleNet, Res-Net, and MobileNet on the Food101 dataset unveiled GoogleNet's superior accuracy of 87.2% [32]. Extending its application to identify diverse food classes from archive images, system tweaks pushed the accuracy to an impressive 95.97%. However, refashioned versions of Inceptions for food recognition encountered limitations in handling specific types of foods, particularly liquids [33]. Pre-trained InceptionV3 models, augmented using shear and flip techniques, achieved an accuracy of 91.5% for 20 food classes [34].

Moreover, studies incorporated methodologies such as the bag-of-features theory, Extreme Learning Committee strategy, and multi-scale multi-view fusion to bolster food recognition [35-37]. Advocacy for computer vision in automating food detection gained traction [38], proving its efficacy compared to alternative methods [39]. Instance segmentation and pixel segmentation techniques showcased improvements in prediction quality [40-42].

Successful execution of neural network models necessitates significant computational resources and parameter tuning to enhance factors like training time and precision. Despite efforts to curate a sizable dataset encompassing 9 unique food classes i, challenges persist due to the intricate and diverse nature of the cuisine.

III. Food Detection & Tracking

Object detection involves the identification of objects within an image and precisely defining their spatial boundaries using bounding boxes. On the other hand, image classification revolves around determining the presence of an object in an image based on calculated probabilities. Images possess distinctive features, such as edges, crucial for extraction within an object recognition model.

Automating this process is feasible through the integration of Convolutional Neural Networks (CNNs) in tandem with Auto Encoder algorithms and various other methodologies [44]. The optimal object detection technique strikes a balance between accurately identifying diverse objects with varying sizes and shapes while demonstrating robust computing capabilities for swift processing. Techniques like YOLO and SSD offer promising results, albeit with a trade-off between speed and accuracy. Therefore, the selection of the algorithm heavily relies on the specific application's requirements and constraints.

IV. Proposed Methodology

Our methodology, depicted in (Fig. 1), commenced with the collection of diverse images portraying a wide array of multiple sources.

Upon completion of the image collection phase, the gathered images underwent a meticulous curation process involving

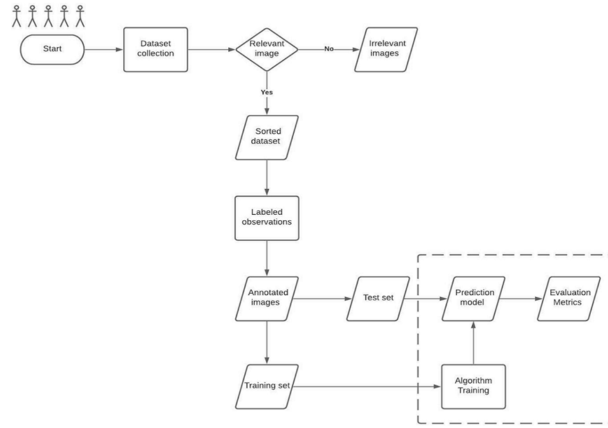


Fig. 1. Food Detection Methodology Flowchart

categorization into a comprehensive dataset. Irrelevant images were systematically eliminated from consideration. Each image was meticulously sorted and manually labeled to ensure its suitability for integration into the deep learning model.

Following this categorization, the dataset underwent a random split adhering to the conventional method. Eighty percent of the data constituted the training set, while the remaining 20% formed the validation set. This partitioning strategy aimed to facilitate accurate model evaluation while mitigating the risk of overfitting. The training set provided the learning algorithm, and the validation set allowed for the evaluation of the model's predictive performance.

Upon completion of the training phase, the resulting outcomes furnished a comprehensive array of metrics and statistics for each variation, facilitating a thorough evaluation and comparison of the models.

After the detection process, we initiated the calculation phase. In this stage, a CSV file containing food item names and quantities was processed using Node.js. Utilizing the 'fs' module, the script read and parsed the file, subsequently calculating the total calories by correlating items with predefined calorie values. The output was a comprehensive report outlining both individual and cumulative caloric content. This approach prioritized accuracy by harnessing JavaScript and file operations to estimate calorie counts using provided quantities and predefined values for each food item.

After completing all the training, we will initiate the calorie estimation process. Initially, the system will count the images. Subsequently, the counting results will be sent to the main system, which possesses the calorie estimation counting capacity. In this project, we will train our model with only nine classes, each representing a solid food item. Additionally, we will assign a specific calorie number to each solid food. Consequently, when the system tallies the total count, it will calculate the total calorie count and display the result."

A. Design & Implementation

We trained three different architectures from the YOLOv5 family – small, medium, and large – using our custom food dataset. The dataset, as detailed in Table 1 below (TABLE 1)These images underwent preprocessing and augmentation as part of the preparation process. Due to the random collection of images for the dataset, certain classes have a higher number of instances compared to others, resulting in a non-uniform distribution within the dataset. For instance, the 'shingara' class contains the highest number of instances at 10,680, while classes like 'jalebi' and 'samosa' have fewer instances. The figure below showcases a selection of images from our dataset."

It would appear something like this if there were 9 classes overall and 6574 photos overall:

We trained three different architectures from the YOLOv5 family – small, medium, and large – using our custom food dataset. This dataset comprised a total of 6,574 images, distributed among 9 distinct classes representing various types . These images underwent preprocessing. Due to the random collection of images for the dataset, some classes have a higher number of instances compared to others, resulting in a non-uniform distribution within the dataset. For instance, while the 'shingara' class contains the highest number of instances, others such as 'jalebi' and 'samosa' have fewer instances. The figure below illustrates a selection of images from our dataset.

B. Model Training

We trained three different YOLOv5 architectures (YOLOv5l, YOLOv5m, and YOLOv5s) using the dataset. An average of around 10 hours were needed to train each model for 100 epochs. The flowchart shown in Fig. 8 provides a visual representation of the model training procedure. The network is first trained on a predetermined set of training data and is trained to predict target values that correspond to the training set. A Train-Test split based on available data is necessary for the proper dataset, which is necessary for the training of a deep learning network. Validation losses are continuously monitored during the training phase, which causes values to fluctuate after few epochs. The model with the lowest validation score was chosen for further testing, and hyperparameters were changed to reduce the validation loss. When a model performs well after training on an enhanced dataset or exhibits high recall and precision rates with new datasets, it is considered effective.

Run the training command with the following parameters to start training:

- img: Indicates the input image's dimensions.
- batch: Establishes the size of the batch.
- epochs: Indicates how many training epochs there are.
- data: Indicates where our YAML file is located.
- weights: Specifies the route for determining the starting weights.
- cfg: Shows our model's configuration.
- name: Indicates the unique number for the outcomes.
- no save: Just the last checkpoint is saved.

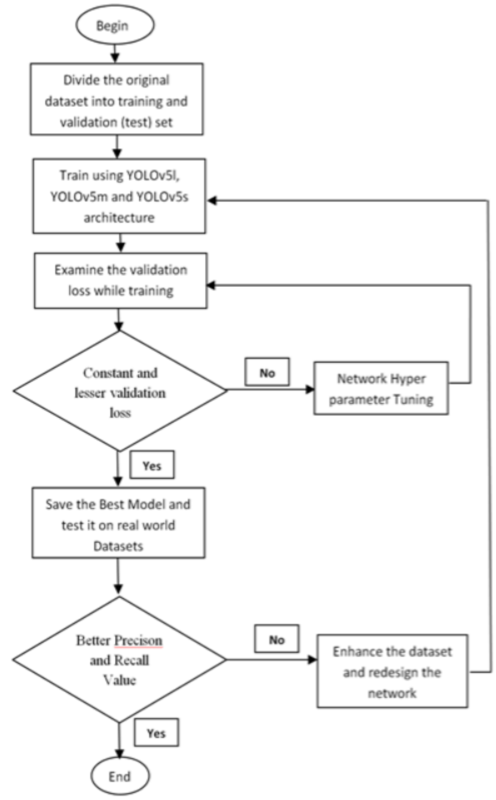


Fig. 2. Flow Diagram of Model Training

- cache: Holds onto photos to speed up learning."

C. Calorie Estimation Calculator

The caloric estimation process was initiated by importing a CSV file containing food item names and respective quantities. The data was processed using Node.js with the 'fs' module to read the file, parse its contents, and conduct the computations.

Data Reading and Parsing: The script used the 'fs' module to read the 'input.csv' file, which contained food item names and their corresponding quantities. The data was parsed by splitting it into rows and columns for further computation.

Caloric Mapping and Computation: A predefined mapping ('CALORIE_MAPPING') associating food item names with their respective calorie values was utilized. The script calculated the total caloric content by multiplying the quantity of each food item with its assigned calorie value and summing these values.

Result Generation: After computation, the script generated a report presenting the total calories alongside individual calorie counts for each food item quantity. This data was then written to an 'output.txt' file using the 'fs' module.

The methodology outlined a straightforward approach leveraging JavaScript and file system operations to process food quantity data, apply calorie mappings, and compute the overall caloric content of the listed food items from the CSV file. The approach focused on accuracy in estimating calorie content

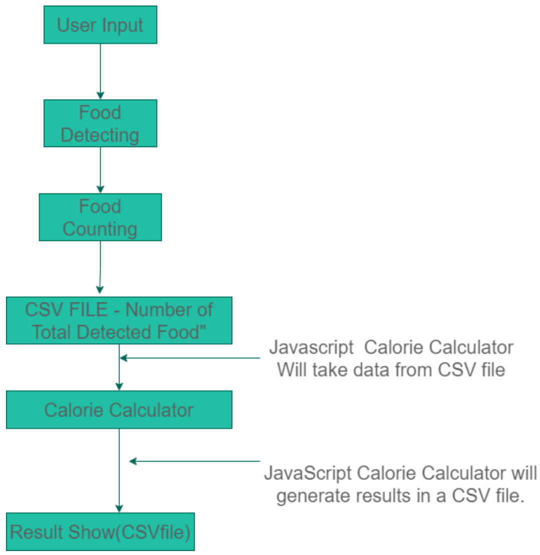


Fig. 3. Calorie Estimation Process

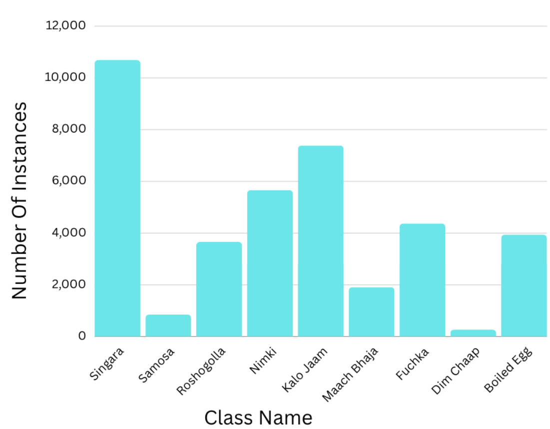
Dataset	Class	Images	Year	Country	Model	Accuracy
FoodNet	50	5000	2017	India	EnsembleNet	73.5
Custom	7	700	2020	Bangladesh	MobileNetV2	80
PFD	100	4928	2020	Pakisthan	DenseNet	69.4
TBF1	7	2835	2021	Bangladesh	CustomCNN	86
IndianFood20	20	17817	2022	India	YOLOv4	91.8
Ours	7	6574	2023	Bangladesh	YOLOv5	96

Fig. 4. Model Accuracy Comparison

based on the provided quantities and predefined calorie values for each food item.

V. Result & Analysis

Our efforts led us to gather a significant collection of data customized from more than 9 distinct categories, comprising roughly 6574 images. This extensive dataset was utilized to train our models and derive pertinent findings for analysis and comparison.



A. Performance Metrics

To comprehend the outcomes derived from training on the YOLOv5 model, we utilize various metrics to gauge its performance. These metrics, detailed below, assist in assessing its effectiveness:

Confusion Matrix: Provides a thorough analysis of the model's output, emphasizing its errors. With reference to Table 2, the collected data allows us to carry out various performance evaluations.

		Actual Values	
Predicted Values		Positive	Negative
	Positive	TP	FP
	Negative	FN	TN

TABLE II
Confusion Matrix

Attributes	Features
Image Type	RGB
Image Extension	JPG, PNG
Image Dimension	1920 * 1920

TABLE III
Dataset Specifications

True Positive (TP): When the expected value and the actual positive value line up. True Negative (TN): A situation in which the expected value and the real negative value coincide. False Positive (FP): A situation in which a positive value is mistakenly identified when a negative value actually exists. False Negative (FN): A situation in which a positive value is actually recorded but the predicted value is mistakenly classified as negative.

Accuracy: Since accuracy gives all errors the same weight, it is invalid as a stand-alone metric. It is determined by the following formula and is only appropriate for balanced datasets:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{1}$$

Precision: Precision measures the accuracy of the positive predictions made by a model and aids in assessing the reliability of each model. It is calculated using the formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall: Measures how well the model can reliably distinguish positive cases from all real positive instances. It is computed using the following formula, which quantifies the percentage of true positives the model successfully captures:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

The mean average precision (mAP), an essential statistic in object detection, is very important. It is used to assess object detection models such as Mask R-CNN, YOLO, and Fast R-CNN. This statistic addresses both false positives (FP) and false negatives (FN), accounting for the trade-off between precision and recall.

The mAP is calculated by averaging the Average Precision (AP) of each class over a number of classes.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{4}$$

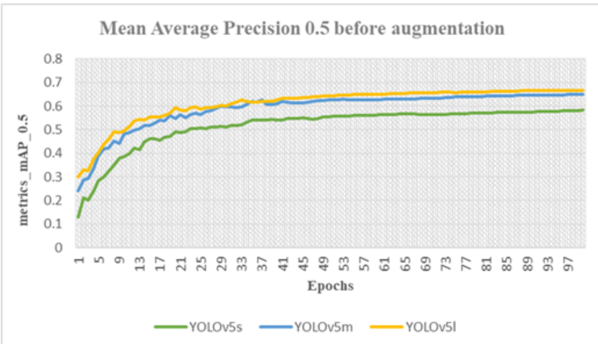


Fig. 11: mAP graph of models before augmentation

Fig. 5. mAP of models

The F1 Score aims to maximize the Precision and Recall values together into a single metric that will improve our model. But it can be difficult to understand the F1 score, which hides the precise metric that the model is trying to optimize. It is computed with the following formula: The box_loss, obj_loss, and cls_loss components make up the YOLO loss function.

Three different architectures—small, medium, and large—from the YOLOv5 family were used in this study. The dataset was preprocessed and augmented prior to training. These YOLOv5 architectures differ in terms of depth and complexity. The best model for food recognition was found by comparing the results of each model that was run on our dataset.

B. Dataset Epoch Results

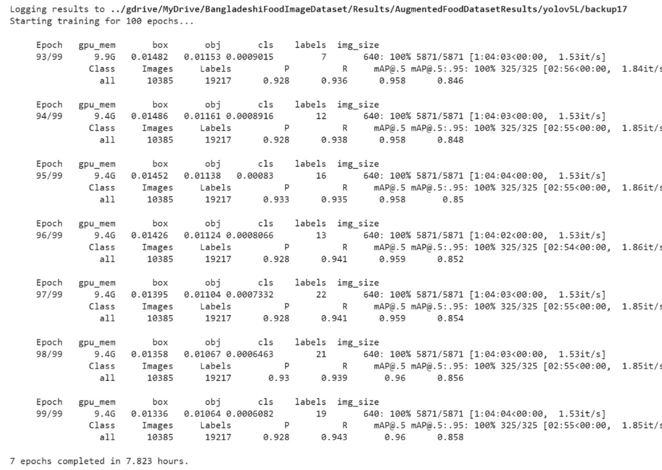
Subsequently, we trained our augmented dataset for another 100 epochs. The best of which was the large model whose precision was 0.943, recall was 0.943 and mAP was 0.96. The results for the YOLOv5l model can be seen in Fig. 13.

C. Overall Performance Analysis

We have documented the performance results before and after the application of augmented data in order to examine the performances of YOLOv5 based on its three distinct designs.

D. Conclusion

The exploration and assessment of various modules within YOLOv5 enabled a closer examination of their impact on food item detection. Our meticulously assembled dataset provided ample rich data for thorough testing, aiming for favorable outcomes across each model. To compare their effectiveness, models were chosen based on distinct differences in nodes,



```
Logging results to ../gdrive/MyDrive/BangladeshiFoodImageDataset/Results/AugmentedFoodDatasetResults/yolov5L/backup17
Starting training for 100 epochs...

     Epoch   gpu_mem       box       obj       cls    labels  img_size
     93/99      9.9G   0.01482   0.01153 0.0009015         7       640: 100% 5871/5871 [1:04:03<00:00,  1.53it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 325/325 [02:56<00:00,  1.84it/s]
                 all      10385      19217      0.928      0.936      0.958      0.846

     Epoch   gpu_mem       box       obj       cls    labels  img_size
     94/99      9.4G   0.01486   0.01161 0.0008916        12       640: 100% 5871/5871 [1:04:03<00:00,  1.53it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 325/325 [02:55<00:00,  1.85it/s]
                 all      10385      19217      0.928      0.938      0.958      0.848

     Epoch   gpu_mem       box       obj       cls    labels  img_size
     95/99      9.4G   0.01452   0.01138   0.00083        16       640: 100% 5871/5871 [1:04:04<00:00,  1.53it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 325/325 [02:55<00:00,  1.86it/s]
                 all      10385      19217      0.933      0.935      0.958       0.85

     Epoch   gpu_mem       box       obj       cls    labels  img_size
     96/99      9.4G   0.01426   0.01124 0.0008066        13       640: 100% 5871/5871 [1:04:02<00:00,  1.53it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 325/325 [02:54<00:00,  1.86it/s]
                 all      10385      19217      0.928      0.941      0.959      0.852

     Epoch   gpu_mem       box       obj       cls    labels  img_size
     97/99      9.4G   0.01395   0.01104 0.0007332        22       640: 100% 5871/5871 [1:04:03<00:00,  1.53it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 325/325 [02:55<00:00,  1.85it/s]
                 all      10385      19217      0.928      0.941      0.959      0.854

     Epoch   gpu_mem       box       obj       cls    labels  img_size
     98/99      9.4G   0.01358   0.01067 0.0006463        21       640: 100% 5871/5871 [1:04:03<00:00,  1.53it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 325/325 [02:55<00:00,  1.85it/s]
                 all      10385      19217       0.93      0.939       0.96      0.856

     Epoch   gpu_mem       box       obj       cls    labels  img_size
     99/99      9.4G   0.01336   0.01064 0.0006082        19       640: 100% 5871/5871 [1:04:04<00:00,  1.53it/s]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 325/325 [02:55<00:00,  1.85it/s]
                 all      10385      19217      0.928      0.943       0.96      0.858

7 epochs completed in 7.823 hours.
```

Fig. 6. Epoch Results

Attributes	Small	Medium	Large	Comment
mAP_0.5	0.583	0.650	0.667	Higher is better
mAP_0.5:0.95	0.425	0.482	0.501	Higher is better
train/box loss	0.027	0.024	0.023	Lower is better
train/class loss	0.009	0.004	0.004	Lower is better
val/box loss	0.022	0.022	0.021	Lower is better
val/class loss	0.009	0.009	0.008	Lower is better
F1 Score	0.604	0.600	0.651	Higher is better

TABLE IV
MODEL RESULTS BEFORE AUGMENTATION

complexity, and speed. Following intensive training sessions, it became evident that among the three, YOLOv5l stood out as superior. Due to its higher complexity—it had a lot more convolutional layers than the others—it was the most effective in terms of processing speed, but at the cost of longer processing times. YOLOv5l's exceptional performance and efficiency when compared to YOLOv5s and YOLOv5m have potential for practical uses in food detection and calorie estimate in a variety of commercial areas.

REFERENCES

[1] S. Banerjee and A.C. Mondal, "Nutrient food prediction through deep learning," 2021 Asian Conference on Innovation in Technology (ASIANCON). (2021)

[2] K. Aizawa et al., "Personalized Food Image Classifier Considering Time-Dependant and Item-Dependant Food Distribution," IEICE Transactions on Information and Systems, 102(11), 2120-2126. (2019)

[3] Xu, B., He, X., and Qu, Z. (2021) Asian Food Image Classification Based on Deep Learning. Journal of Computer and Communications, 9, 10-28.

[4] D. Sahoo, W. Hao, S. Ke, W. Xiongwei, H. Le, P. Achananuparp, E.-P. Lim, and S. C. H. Hoi, "FoodAI," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, Jul. 2019.

[5] Uddin, Asif Mahbub & Miraj, Abdullah Al & Sen Sarma, Moumita & Das, Avishek & Gani, Md. (2021). "Traditional Bengali Food Classification Using Convolutional Neural Network." 1-8.

[6] Karabay, A., Bolatov, A., Varol, H. A., Chan, M.-Y. (2023). "A Central Asian food dataset for personalized dietary interventions." Nutrients, 15(7), 1728.

[7] G. A. Tahir and C. K. Loo, "An Open-Ended Continual Learning for Food Recognition Using Class Incremental Extreme Learning Machines," in IEEE Access, vol. 8, pp. 82328-82346, 2020.

[8] Nazir, N., Singh, R. P., & Mehra, Dr. M. (2022). "A deep neural network approach on spleen cancer detection using densenet-201." International Journal for Research in Applied Science and Engineering Technology, 10(11), 1102–1116.

[9] D. Shubhangi, B. Gadgay, S. Jabeen and M. A. Waheed, "State wise Indian Food Recognition and Classification Using CNN And MobileNetV2," 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 669-672.

[10] He, Z., Xiong, L. (2021). "Image classification of zinc dross based on improved MobileNetV2." 2021 China Automation Congress (CAC).

[11] P. Pandey, A. Deepthi, B. Mandal, and N. B. Puhan, "Foodnet: recognizing foods using ensemble of deep networks," IEEE Signal Processing Letters, vol. 24, no. 12, pp. 1758–1762, 2017.

[12] Kayadibi, I., Güraksın, G. E., Ergün, U., Özmen Süzme, N. (2022). "An eye state recognition system using transfer learning: AlexNet-based deep convolutional neural network." International Journal of Computational Intelligence Systems, 15(1).

[13] Mohammed, A. H., Cevik, M. (2022). "GoogleNet CNN classifications for diabetic's retinopathy." 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA).

[14] Zahisham, Z., Lee, C. P., Lim, K. M. (2020). "Food recognition with ResNet-50." 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET).

[15] Pandey, D., Parmar, P., Toshniwal, G., Goel, M., Agrawal, V., Dhiman, S., Gupta, L., Bagler, G. (2022). "Object detection in Indian food platters using transfer learning with Yolov4." 2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW).

[16] Patil, S., Patil, S., Kale, V., Bonde, M. (2021). "Food item calorie estimation using Yolov4 and image processing." International Journal of Computer Trends and Technology, 69(5), 69–76.

[17] Jocher, G., Chaurasia, A., Stoken, A., Borovec, J. (2022). "Ultralytics/yolov5: V7.0 - YOLOv5 Sota Realtime Instance Segmentation." Zenodo.

[18] S. Tarashima, "Object hypotheses as points for efficient multi-object tracking," Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, (2021).

[19] J. Sun, K. Radecka and Z. Zilic, "FoodTracker: A Real-time Food Detection Mobile Application by Deep Convolutional Neural Networks," The 16th International Conference on Machine Vision Applications, 2019.

[20] J. Redmon, and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767 (2018).

[21] X. He, D. Wang and Z. Qu, "An Improved YOLOv3 Model for Asian Food Image Recognition and Detection," Open Journal of Applied Sciences 11, no. 12 (2021): 1287-1306.

[22] P. Adarsh and P. Rathi, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," Coimbatore, India: IEEE, 2020.

[23] C. Termritthikun, P. Muneesawang, and S. Kanprachar, "NU-InNet: Thai food image recognition using convolutional neural networks on smartphones," Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9, no. 2-6 (2017): 63-67.

[24] V. Burkapalli, and P. Patil, "An Efficient Food Image Classification By Inception-V3 Based Cnns," International Journal of Scientific & Technology Research Volume 9, Issue 03, (2020).

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," v6, 2015.

[26] T. Ege et al., "Image-based estimation of real food size for accurate food calorie estimation," In 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 274-279. IEEE, 2019.

[27] M. Kamble et al., "Calorie detection of food images based on SVM algorithm," International Journal of Research in Applied Science and Engineering Technology, 9(VI), 1836-1840, (2021).

[28] F. Akter et al., "Recognition and Classification of Fast Food Image," v1, 2018.

[29] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: a new dataset, experiments, and results," IEEE journal of biomedical and health informatics 21, no. 3 (2016): 588-598.

[30] M. Taskiran and N. Kahraman, "Comparison of CNN Tolerances to Intra Class Variety in Food Recognition," IEEE International Symposium on Innovations in Intelligent Systems and Applications, 3-5 July 2019, 1-5.

[31] Z. Shen et al., "Machine learning based approach on food recognition and nutrition estimation," Procedia Computer Science 174, (2020): 448-453.

[32] Chaitanya, A., Shetty, J., Chiplunkar, P. (2023). "Food image classification and data extraction using convolutional neural network and web crawlers." Procedia Computer Science, 218, 143–152.

[33] M. Anthimopoulos et al., "A food recognition system for diabetic patients based on an optimized bag-of-features model," IEEE journal of biomedical and health informatics 18, no. 4 (2014): 1261-1271.

[34] N. Martinel, C. Piciarelli, and C. Micheloni, "A supervised extreme learning committee for food recognition," Computer Vision and Image Understanding 148, (2016): 67-86.

[35] S. Jiang et al., "Multi-scale, multi-view deep feature aggregation for food recognition," IEEE Transactions on Image Processing 29, (2019): 265-276.

[36] M. Subhi, S. Ali, and M. Mohammed, "Vision-based approaches for automatic food recognition and dietary assessment: A survey," IEEE Access 7, (2019): 35370-35381.

[37] P. Pouladzadeh and S. Shirmohammadi, "Mobile multi-food recognition using deep learning," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 13, no. 3s (2017): 1-21.

[38] P. Poply and J. Arul Jothi, "Refined image segmentation for calorie estimation of multiple-dish food items," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS).

[39] Y. Sari et al., "Multiple Food or Non-Food Detection in Single Tray Box Image using Fraction of Pixel Segmentation for Developing Smart NutritionBox Prototype," International Journal of Innovative Technology and Exploring Engineering (IJITEE) 9, no. 3S (2020).

[40] V. Kumar, A. Namboodiri and C.V. Jawahar, "Region pooling with adaptive feature fusion for end-to-end person recognition," 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), (2020).

[41] Jaswanthi, R., Amruthatulasi, E., Bhavyasree, Ch., Satapathy, A. (2022). "A hybrid network based on Gan and CNN for food segmentation and calorie estimation." 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS).

[42] Li, J., Ji, C., Yan, G., You, L., Chen, J. (2021). "An ensemble net of convolutional auto-encoder and graph auto-encoder for auto-diagnosis." IEEE Transactions on Cognitive and Developmental Systems, 13(1), 189–199.

[43] I. Hrga and M. Ivasic-Kos, "Effect of data augmentation methods on face image classification results," Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods, (2022).

[44] E. Pulfer, "Different Approaches to Blurring Digital Images and Their Effect on Facial Detection," Computer Science and Computer Engineering Undergraduate Honors Theses, (2019).

[45] E. Rusak et al., "A simple way to make neural networks robust against diverse image corruptions," Computer Vision - ECCV 2020, 53-69, (2020)

[46] J. Azzeh et al., "Salt and pepper noise: Effects and removal," JOIV: International Journal on Informatics Visualization, 2(4), 252. (2018).