Deep Learning-Based Calorie Estimation from Food Images

Team Members:
Sanket Shigaonkar(sshigaon)
Shashank Govindu(sgovindu)
Satya Vaishnavi Jami(satyavai)

1. Abstract

In today's health-conscious society, the demand for accessible and accurate calorie estimation tools are growing rapidly. This project introduces a deep learning-based pipeline for estimating calories from food images using object detection, segmentation, and geometric modeling. By leveraging the ECUST Food Dataset, which includes top and side views of food items alongside a reference coin for scale, the pipeline achieves high-accuracy volume and calorie estimation. We compare different object detection models as YOLOv8, Faster R-CNN, and a custom YOLO model, followed by segmentation with GrabCut and shape approximation techniques. Experimental results demonstrate that YOLOv8 offers a robust tradeoff between precision and real-time performance, while Faster R-CNN achieves the highest mAP. The proposed system provides a non-invasive, camera-based approach to dietary monitoring that could help users make informed health decisions.

2. Introduction

Estimating the caloric content of meals has traditionally been a manual and error-prone task. With the growing prevalence of smartphones and AI tools, image-based calorie estimation offers a promising alternative that can deliver fast and reliable results with minimal user effort.

The primary motivation behind this project is to develop an end-to-end solution that can estimate the calories of food items by analyzing images captured from daily life scenarios. Unlike weight-based or barcode-scanning systems, our method does not require any physical contact or manual input beyond image capture. This makes it suitable for real-time applications in personal fitness, healthcare, and nutritional research.

Our goal is to build a deep learning pipeline that detects food items in an image, estimates their volume using geometric modeling, and finally calculates their caloric value using density mappings from food nutrition databases.

Related Work:

Earlier approaches to calorie estimation heavily relied on manual food logging, visual estimation, or hardware sensors such as scales and barcode scanners. However, these methods are either subjective or lack scalability.

Recent advances in computer vision have enabled more automated methods. YOLO (You Only Look Once) is a real-time object detection model known for its speed and reasonable accuracy.

Faster R-CNN, on the other hand, is a two-stage detector that offers high precision by first generating region proposals and then classifying them.

Various research efforts have also investigated volume estimation through depth sensors or 3D reconstruction. However, these approaches often require specialized equipment. Our work builds upon these ideas, using only 2D images and a known scale object (coin) to estimate volume geometrically.

3. Dataset Description:

The ECUST Food Dataset provides a solid foundation for this project. It contains over 2,000 images of 19 food categories, captured from both top and side views. Each image also includes a standard One Yuan Coin that serves as a reference for scale calibration.

These multi-view images allow us to approximate 3D volume more accurately than a single-view approach. Annotations include bounding boxes for food items, which are useful for training object detectors. The dataset is diverse in terms of food types, shapes, and background conditions, which supports generalization across real-world scenarios.

Table 1: Food Class Details with Shape, Density, and Energy

Class	Shape	Density (g/cm^8)	Energy (kcal/g)
apple	ellipsoid	0.78	0.52
banana	irregular	0.91	0.89
bread	column	0.18	3.15
bun	irregular	0.34	2.23
doughnut	irregular	0.31	4.34
egg	ellipsoid	1.03	1.43
fired_dough_twist	irregular	0.58	24.16
grape	irregular	0.97	0.69
lemon	ellipsoid	0.96	0.29
litchi	irregular	1.00	0.66
mango	irregular	1.07	0.60
mooncake	column	0.96	18.83
orange	ellipsoid	0.90	0.63
pear	irregular	1.02	0.39
peach	ellipsoid	0.96	0.57
plum	ellipsoid	1.01	0.46
qiwi	ellipsoid	0.97	0.61
sachima	column	0.22	21.45
tomato	ellipsoid	0.98	0.27

4. Methodology

4.1. Object Detection models:

We trained and compared three object detection models:

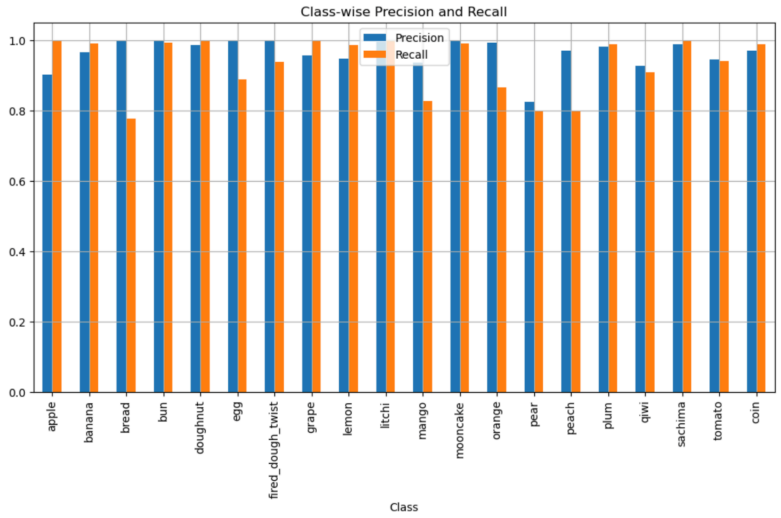
Table 2: Comparison of Detection Model Training Configurations

Configuration	YOLOv8	YOLO from Scratch	Faster R-CNN
Backbone	YOLOv8s (CSPDarknet)	4-layer ConvNet	ResNet-50 + FPN
Image Size	640	320	variable (default)
Pretrained?	Yes (yolov8s.pt)	No	Yes
Epochs	30	100	10
Batch Size	16	4	2
Optimizer	SGD (Ultralytics default)	Adam	SGD
Learning Rate	Auto (Ultralytics)	0.001	0.005
Loss Function	YOLOv8 built-in loss	Custom YOLO Loss (MSE + BCE + CE)	Built-in Faster R-CNN loss
Anchor Settings	Auto (YOLOv8)	Manual: [[0.1, 0.1], [0.2, 0.3], [0.4, 0.4]]	Region Proposal Network (RPN)
Custom Layers	No	Yes (YOLO head + custom loss)	Only classifier head modified
Evaluation Metric	mAP@0.5, F1	mAP@0.5, F1	mAP@0.5, F1 (via COCOeval)

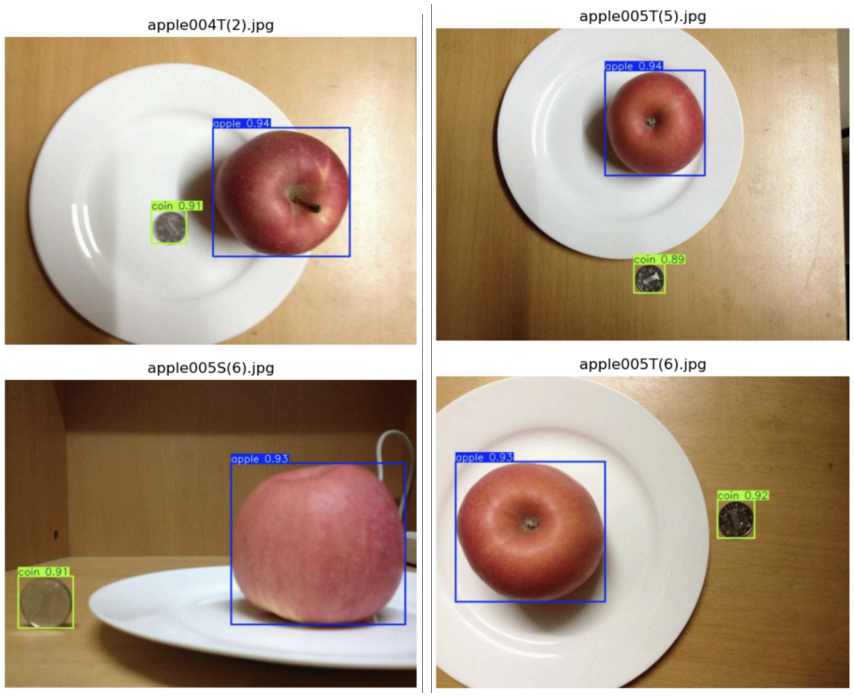
YOLOv8: YOLOv8 combines high detection accuracy with impressive inference speed. It uses an anchor-free approach, enabling simpler post-processing and better generalization to various object sizes.

The mAP@0.5, precision, and recall scores show rapid growth in the first few epochs and quickly plateau near perfect scores. This demonstrates that YOLOv8 converges efficiently with strong generalization even in early training stages.

The model maintains high precision and recall across almost all food categories, with only minor variation in a few challenging classes. This consistency highlights YOLOv8's ability to handle class imbalance and inter-class similarity.

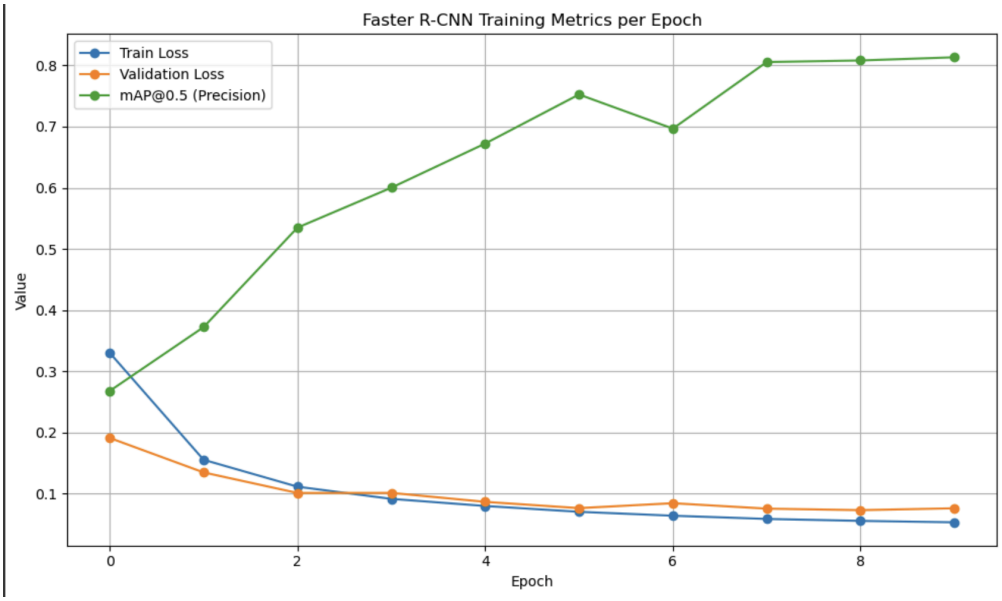


The model accurately detects both the apple and the reference coin in multiple views, with high confidence scores. Its ability to consistently identify objects from different angles affirms its robustness for real-world food image analysis.



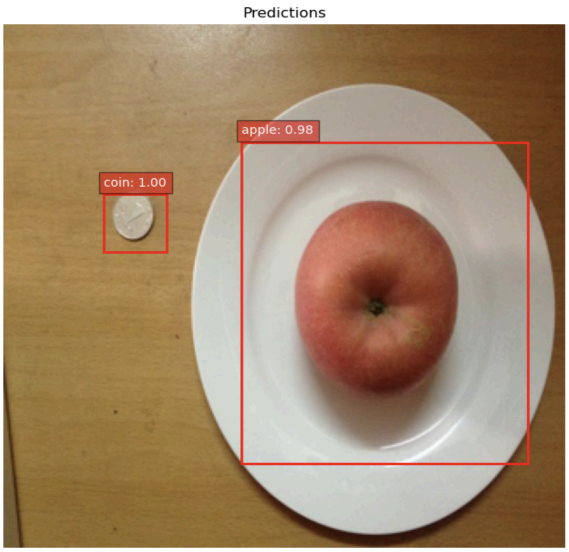
Faster R-CNN: A region-based two-stage detector, Faster R-CNN offers state-of-the-art accuracy, particularly in tasks requiring precise localization. It is known for its robust feature extraction and works well in scenarios where speed is not the primary constraint.

The training and validation losses steadily decrease, reflecting consistent learning and effective generalization. Meanwhile, the mAP@0.5 score shows a strong upward trend, crossing 0.8 by the final epoch, indicating high detection accuracy.

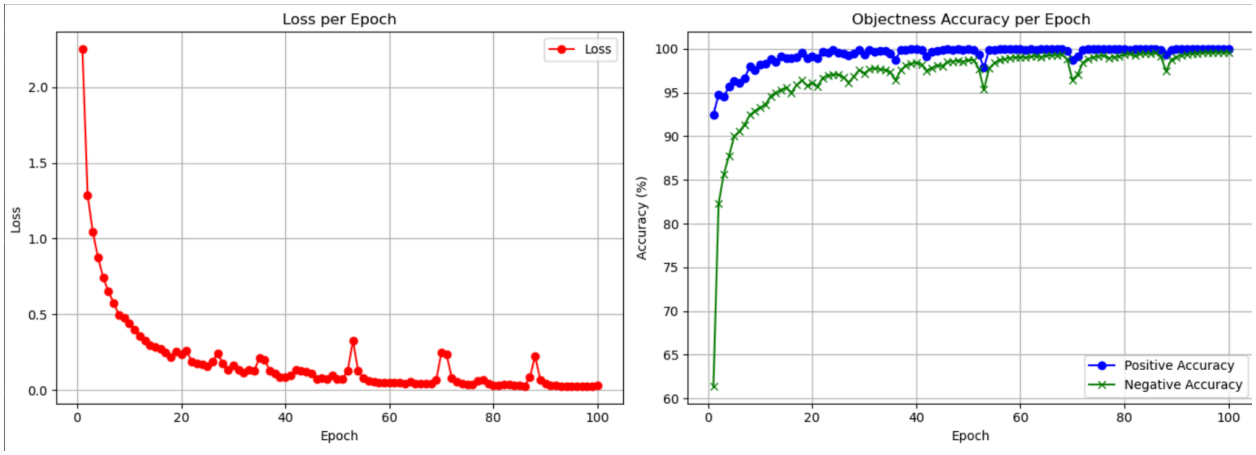


YOLO-from-Scratch: We also implemented and trained a custom YOLO model from the ground up. This lightweight model helps demonstrate how architectural simplicity affects detection performance when trained on limited data.

The custom YOLO model successfully detects both the apple and the reference coin with high confidence. This example highlights the model's ability to localize distinct objects in clean, uncluttered backgrounds.



The training loss shows a smooth decline over 100 epochs, indicating stable learning and convergence. Meanwhile, objectness accuracy steadily improves for both positive and negative samples, surpassing 95% by the end of training.



After evaluating three different object detection models: YOLOv8, a **YOLO-from-Scratch**, and Faster R-CNN. Each model was trained and validated on the ECUST food dataset, and their performance was compared using metrics like mAP@0.5, precision, recall, and F1 score.

After benchmarking, YOLOv8 was selected due to its favorable balance of detection accuracy and computational efficiency. The predictions from YOLOv8 were then used as input for further segmentation and volume estimation steps. This allowed us to build an end-to-end pipeline for estimating the volume and calorie content of food items from images.

4.2 Volume Estimation

Once food items are detected, we isolate them using the GrabCut algorithm—a foreground segmentation technique that refines the boundaries around each object. This segmented mask is used to fit 2D shapes such as ellipses and rectangles.

Assuming basic geometric shapes like cylinders or ellipsoids, we estimate the food's volume based on its dimensions in both top and side views. The coin in the image provides a known reference length, allowing us to convert pixel measurements to physical dimensions.

The GrabCut algorithm effectively isolates the food item from both top and side views, producing clean binary masks. These masks serve as the foundation for accurate volume estimation by enabling shape fitting and pixel-based dimension analysis.

The volume is calculated based on the detected bounding box dimensions (width, height, depth) and the known shape category of the food item.

(a) Ellipsoid-shaped Foods (e.g., apple, egg, tomato):

$V = (4/3) * \pi * (w/2) * (h/2) * (d/2)$

Where w, h, and d are the real-world width, height, and depth in centimeters. If depth is unknown, it can be approximated as equal to width or height.

(b) Column-shaped Foods (e.g., bread, mooncake):

$V = \pi * r^2 * h = \pi * (w/2)^2 * h$

Assuming a cylindrical shape where w is the diameter and h is the height.

(c) Irregular-shaped Foods (e.g., banana, doughnut, grape):

$V = w * h * d * \alpha$

Where α is a shape adjustment factor, typically between 0.4 and 0.6.

4.3 Calorie Estimation

Using the estimated volume, we convert each food item's size into a weight using known food densities (g/cm^3) from nutrition databases. Calories are then calculated by multiplying the estimated mass by the corresponding calories-per-gram metric for each food category. This approach enables a fully image-based estimation process without needing physical weighing.

Once the volume is computed, the calorie content is estimated as follows:

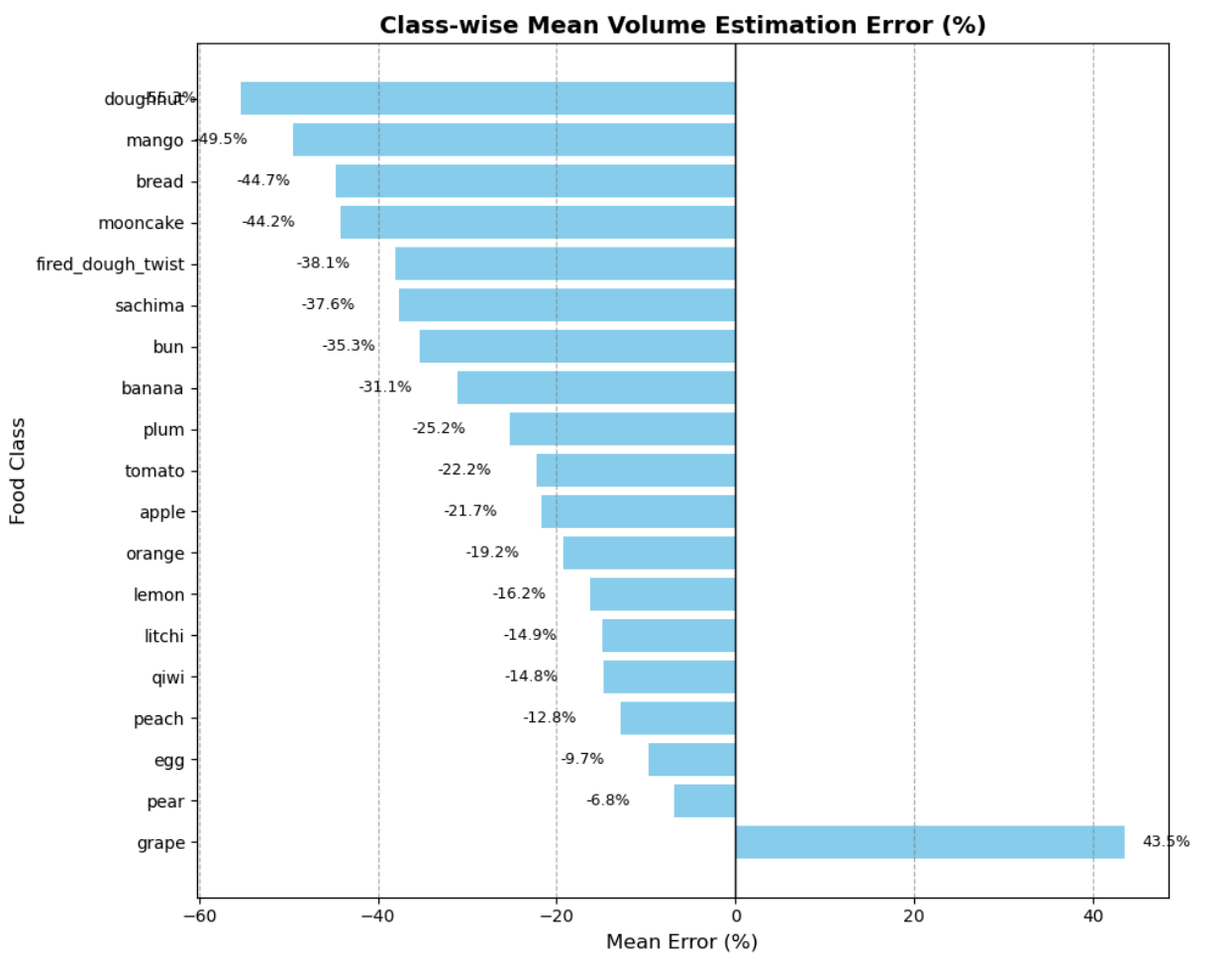
Mass = Density × Volume
Calories = Mass × Energy
Calories = Density × Volume × Energy

5. Experimental Results

To assess the effectiveness of the detection models, we used standard evaluation metrics including mAP@0.5, precision, recall, and F1 score. YOLOv8 showed strong overall performance, balancing accuracy and speed. Faster R-CNN achieved the highest mAP and recall, demonstrating excellent detection fidelity but with slower inference.

Performance Comparison of Object Detection Models

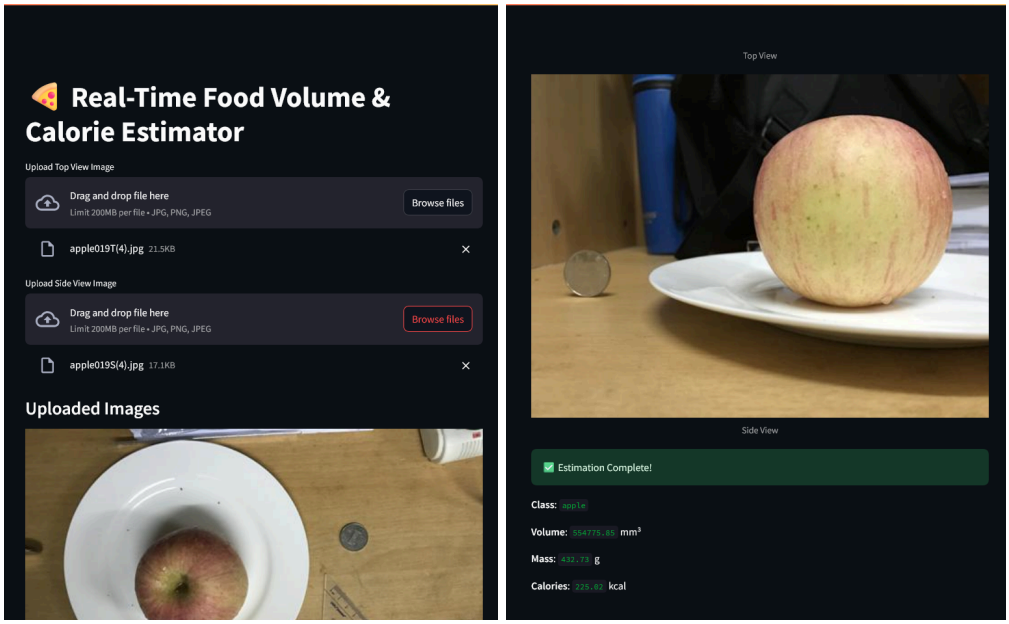
Metric	YOLOv8	YOLO-from-Scratch	Faster R-CNN
mAP@0.5	0.8822	0.1891	**0.9750**
Precision	**0.9641**	0.4019	0.6696
Recall	0.9341	0.5968	**0.9895**
F1 Score	**0.9489**	0.4804	0.7987



The above figure shows the mean volume estimation error (%) for each food class in the dataset. Most food items exhibit underestimation (negative error values), particularly in classes like *doughnut* (−55.2%), *mango* (−49.5%), *bread* (−44.7%), and *mooncake* (−44.2%).

Interestingly, grape is the only class with a significant overestimation (+43.5%), likely due to its small size and clustered shape, which may have led to over-segmentation or bounding boxes capturing extra background.

Shape complexity and the quality of segmentation have a noticeable impact on estimation performance. For example, irregular-shaped and multi-piece items (e.g., *fired_dough_twist*, *sachima*) show higher errors compared to simpler ellipsoidal items like *egg* or *pear*, which had the lowest errors (−9.7% and −6.8%, respectively).



We also made a real-time web interface using streamlit on a local server for food volume and calorie estimation. Users upload top and side view images of a food item, and the system uses YOLOv8 and GrabCut to detect and segment the object.

As shown in the above images, an apple was processed with an estimated volume of 554,775.85 mm³, mass of 432.73 g, and calorie content of 225.02 kcal. The interface demonstrates the pipeline's practicality and potential for integration into nutrition-tracking apps.

6. Discussion and Analysis

Each model brought unique strengths to the table. YOLOv8 stood out for real-time use cases, thanks to its high precision and balanced recall. Faster R-CNN, while slower, proved to be the most accurate in terms of localization and detection, making it ideal for high-stakes applications where accuracy is paramount. The custom YOLO model underperformed, reinforcing the importance of transfer learning and model pretraining—especially with limited datasets. GrabCut played a vital role in refining the boundaries of food items, which directly impacted the accuracy of volume and calorie estimation.

7. Conclusion

In this project, we presented a deep learning-based calorie estimation pipeline using food images. We compared the performance of three object detection models: YOLOv8, a custom YOLO implementation from scratch, and Faster R-CNN. Among these, Faster R-CNN achieved the highest mAP@0.5 but with a slower inference time, while YOLOv8 offered the best balance between accuracy and speed.

Post-detection, GrabCut segmentation was used to isolate food items, enabling accurate volume estimation based on geometric assumptions (ellipsoid, column, and irregular shapes). By combining estimated volume with known food density and energy values, we computed calorie estimates with reasonable accuracy.

This modular pipeline demonstrated the viability of automated calorie estimation using computer vision, especially for health-monitoring and dietary-tracking applications.

8. Future Work

1. Depth Estimation: Incorporating monocular depth estimation or using stereo/RGB-D sensors could improve real-world volume accuracy.

2. Multi-food Scenarios: Extend the pipeline to handle images with multiple food items in a single frame.

3. Instance Segmentation Models: Replace GrabCut with more robust models like Mask R-CNN for pixel-accurate masks.

4. Mobile Deployment: Optimize and convert the pipeline to run in real time on smartphones or embedded devices.

5. User Feedback Loop: Include user validation and correction for iterative model improvement.

9. References

1. ECUST Food Dataset. *ECUSTFD-resized*. GitHub Repository. Available at: https://github.com/LynnHo/ECUSTFD-resized

2. Yubing Li, et al. *Grab Cut Image Segmentation Based on Image Region*. In: 2018 International Conference on Image, Vision and Computing (ICIVC), pp. 311–315, 2018. doi:10.1109/ICIVC.2018.8492818

3. Yanchao Liang and Jianhua Li. *Computer Vision-based Food Calorie Estimation: Dataset, Method, and Experiment*. arXiv preprint, 2017. arXiv:1705.07632

4. Joseph Redmon, et al. *You Only Look Once: Unified, Real-Time Object Detection*. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. doi:10.1109/CVPR.2016.91

5. Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv preprint, 2020. arXiv:2004.10934

6. Ren, S., He, K., Girshick, R., & Sun, J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. In: Advances in Neural Information Processing Systems (NeurIPS), 2015.

7. Xu, Y., Lin, Q., Wu, Q., & Shi, C. *A Benchmark for Food Volume Estimation Using Single-view Images*. IEEE Access, 8, 105104–105113, 2020.

8. Pouladzadeh, P., Shirmohammadi, S., & Al-Maghrabi, R. *Measuring Calorie and Nutrition from Food Image*. IEEE Transactions on Instrumentation and Measurement, 63(8), 1947–1956, 2014.

9. Yakhyo's GitHub Repository. *YOLO2VOC Annotation Converter*, 2021. Available at: https://github.com/yakhyo/yolo2voc

10. Ultralytics. *YOLOv8 Official Documentation*. Accessed March 2025. Available at: https://docs.ultralytics.com/

10. Team Contribution

Team Member	Project Part	Contribution (%)
sshigaon	Worked on the Faster R-CNN model, created and evaluated YOLOv8-based baseline, implemented GrabCut segmentation, and analyzed detection performance across datasets.	33.33%
satyavai	Handled dataset preprocessing and annotation formatting, developed the custom YOLO training pipeline, and performed model evaluation under varying lighting conditions.	33.33%
sgovindu	Focused on volume estimation logic, geometric modeling, and calorie prediction calculations; also contributed to writing and organizing the final report.	33.33%