Article

Not peer-reviewed version

A Deep Learning–Driven Method for Bowl Tableware Reconstruction and the Prediction of Liquid Volume and Food Nutrient Content

Xu Ji , Kai Song , Lianzheng Sun , Haolin Lu , Hengyuan Zhang , Yiran Feng *

Posted Date: 1 January 2026

doi: 10.20944/preprints202512.2802.v1

Keywords: image segmentation; nutrient content prediction; model reconstruction; volume estimation

 doi:10.20944/preprints202512.2802.v1

Article

A Deep Learning–Driven Method for Bowl Tableware Reconstruction and the Prediction of Liquid Volume and Food Nutrient Content

Xu Ji [1], Kai Song [1], Lianzheng Sun [4], Haolin Lu [1], Hengyuan Zhang [1] and Yiran Feng [1,2,3,*]

[1] Department of School of Mechanical Engineering and Automation, Dalian Polytechnic University, Dalian, China
[2] SKL of Marine Food Processing & Safety Control, National Engineering Research Center of Seafood, Dalian Polytechnic University, Dalian 116034, China
[3] Department of Key Laboratory of Marine Food Processing Technology and Equipment of Liaoning Province, Dalian Polytechnic University, Dalian, China
[4] Qingdao Yeelink Information Technology Co., Ltd
* Correspondence: fengyr@dlpu.edu.cn; Tel.:+86-411-86323765

Abstract

To overcome the low accuracy of conventional methods for estimating liquid volume and food nutrient content in bowl-type tableware, as well as the tool dependence and time-consuming nature of manual measurements, this study proposes an integrated approach that combines geometric reconstruction with deep learning–based segmentation. After a one-time camera calibration, only a frontal and a top-down image of a bowl are required. The pipeline automatically extracts key geometric information, including rim diameter, base diameter, bowl height, and the inner-wall profile, to complete geometric modeling and capacity computation. The estimated parameters are stored in a reusable bowl database, enabling repeated predictions of liquid volume and food nutrient content at different fill heights. We further propose Bowl Thick Net to predict bowl wall thickness with millimeter-level accuracy. In addition, we developed a Geometry-aware Feature Pyramid Network (GFPN) module and integrated it into an improved Mask R-CNN framework to enable precise segmentation of bowl contours. By integrating the contour mask with the predicted bowl wall thickness, precise geometric parameters for capacity estimation can be obtained. Liquid volume is then predicted using the geometric relationship of the liquid or food surface, while food nutrient content is estimated by coupling predicted food weight with a nutritional composition database. Experiments demonstrate an arithmetic mean error of −3.03% for bowl capacity estimation, a mean liquid-volume prediction error of 9.24%, and a mean nutrient-content (by weight) prediction error of 11.49% across eight food categories.

Keywords: image segmentation; nutrient content prediction; model reconstruction; volume estimation

1. Introduction

With the growing global emphasis on healthy diets, precision nutrition, and quality-of-life management, dietary management and nutritional assessment systems have become active research topics in public health, intelligent healthcare, and data-driven science. Conventional dietary assessment approaches—such as manual weighing, handwritten logging, and subjective experience–based estimation—are labor-intensive and time-consuming, prone to error, and thus insufficient for the increasingly accurate and efficient dietary management demanded by modern society. In recent years, automated dietary assessment systems based on image analysis, computer vision, and deep learning have emerged, with particular interest in food nutrient prediction. However, most existing

 doi:10.20944/preprints202512.2802.v1

methods still rely on combining 2D imagery with depth cameras, which increases hardware cost and deployment complexity and makes robust use in everyday environments challenging[1-3].

With the rapid development of computer vision, deep learning, and model reconstruction techniques, image-based methods for predicting liquid volume and dish nutrient content have achieved substantial progress. Dehais *et al.* proposed a model reconstruction approach for bowl-type containers from 2D images, in which multi-view image reconstruction is employed to model the container geometry for liquid volume estimation[4]. In the deep learning domain, Lo *et al.* presented a food volume estimation method based on a single depth map, leveraging neural networks to extract spatial features from depth data for model reconstruction and volume computation[5]. In addition, researchers have attempted to improve the accuracy of liquid volume estimation through enhanced convolutional neural networks (CNNs) and advanced image processing. Jia *et al.* performed liquid surface segmentation and volume estimation using a Mask R-CNN–based model, demonstrating the potential of deep learning for image segmentation and object volume prediction[6].

To address the challenge of accurately predicting liquid volume and food nutrient content, this paper proposes a capacity prediction method that integrates deep learning–based segmentation with model reconstruction of bowl-type tableware. Using commonly used bowls as the study objects, we extract key geometric parameters—including rim diameter, base diameter, effective height, and the inner-wall contour—and construct a reconstructed model to enable subsequent estimation of liquid volume and prediction of food nutrient content.

The main contributions of this study are as follows:

(a) Compared with conventional manual measurement, which is time-consuming, and existing methods that often yield inaccurate reconstruction of bowl-type container models, this study uses only a frontal-view and a top-down-view image of the bowl, without requiring a depth camera. We adopt an improved deep learning–based mask segmentation model and leverage multi-view images to reconstruct a geometry-parameterized bowl model, enabling accurate capacity estimation. The resulting capacity model further supports subsequent prediction of the liquid volume and food nutrient content of items contained in the bowl.

(b) We propose Bowl Thick Net, a model that predicts bowl wall thickness by detecting and fitting circles to the inner and outer rims at the bowl mouth, thereby further improving the accuracy of bowl capacity estimation.

(c) We propose a Geometry-aware Feature Pyramid Network (GFPN) module that, when integrated into an improved Mask R-CNN framework, enables accurate segmentation of bowl contour masks. Combined with Bowl Thick Net and geometric reasoning, the proposed pipeline recovers the bowl's geometric parameters, constructs a reconstructed bowl model, and estimates its capacity, achieving an arithmetic mean error of −3.03% in bowl volume prediction.

(d) We propose a method for predicting the liquid volume contained in a bowl and the nutrient content of the served food, and we develop a visualization platform that integrates image processing, geometric modeling, and liquid-volume or nutrient-content prediction. After a one-time camera calibration, the system requires only a frontal view and a top-down view of the bowl to automatically extract the rim, base, height, and inner-wall contour dimensions, enabling non-contact, semi-automated modeling and capacity estimation. The resulting parameters are stored in a reusable bowl database to support repeated use. Given a top-down image of the liquid or food in the bowl, the platform can rapidly predict liquid volume and food nutrient content at different fill heights. Experimental results show a mean liquid-volume prediction error of 9.24% and a mean prediction error of 11.49% for nutrient content (by weight) across eight food categories.

2. Materials and Methods

2.1. Camera Calibration and Perspective Correction

The frontal-view and top-down images of bowl-type tableware in this study were captured using an industrial camera with a resolution of 3072 × 2048 and a maximum frame rate of 59.6 fps. Camera calibration is a critical step to ensure accurate estimation of tableware dimensions. Based on the pinhole camera model, we establish the mapping between pixel coordinates and world coordinates, thereby enabling dimensional measurement from 2D images to 3D models and

 doi:10.20944/preprints202512.2802.v1

compensating for errors caused by lens distortion. Variations in object distance induced by different camera models and settings may change the scale factor between image pixels and real-world physical units, thereby degrading measurement accuracy. To improve the precision of tableware dimension measurement, it is necessary to calibrate object-distance variations and compensate for height-related errors.

We adopt Zhang's calibration method using a 2D checkerboard calibration target with a square size of 25 mm[7]. From 15 images of the calibration target captured at different heights and viewing angles, the pixel coordinates of the four corner points of each square are extracted. The homography matrices under different viewpoints are then computed to estimate the camera's geometric and pose parameters. Lens distortion parameters are refined via nonlinear least-squares optimization, and all parameters are finally integrated using maximum likelihood estimation. Calibration is performed using the MATLAB Camera Calibrator tool, and Fig. 1 visualizes the calibration procedure.

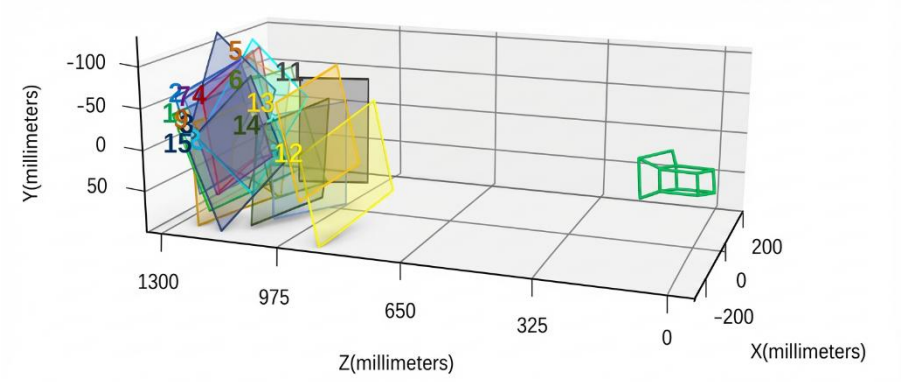


Figure 1. Visualization diagram of the camera calibration process.

In each calibration image, corner detection is performed to obtain the pixel coordinates of the checkerboard corners. We then solve for the mapping between the image pixel coordinates and the corresponding real-world geometric coordinates, thereby estimating the camera's geometric parameters—including focal length, principal point, and intrinsic matrix—as well as its pose parameters, i.e., the rotation matrix, translation vector, and extrinsic matrix, together with the lens distortion coefficients[8]. These parameters determine the camera pose in the physical world, and the accuracy of the calibration model is verified by statistics of the reprojection error. In our experiments, the mean reprojection error is 0.23 pixels, indicating a small calibration error and ensuring the reliability of subsequent measurements. Fig. 2 shows the reprojection error statistics of the camera calibration.

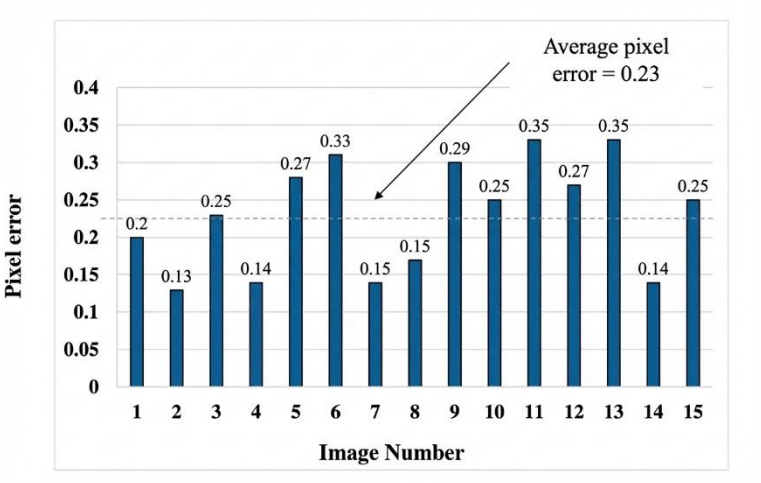


Figure 2. Statistical graph of re-projection error in camera calibration.

 doi:10.20944/preprints202512.2802.v1

During calibration, the checkerboard target was placed within the camera's depth of field, and its pose was corrected using a spirit level and the camera pose parameters. Based on the established mapping between pixel and world coordinates, we obtained a pixel-to-world scale of $P_x = 0.1719mm/pixel$ when the camera-to-target distance was 954.325 mm and the image resolution was 1280×1280 pixels. To facilitate subsequent image processing and bowl segmentation, the image resolution was resized to 640×640. Accordingly, the pixel-to-world mapping was adjusted using the scaling factor, yielding $P_x = 0.3438mm/pixel$ at a resolution of 640×640.

2.2. Image Preprocessing and Contour Extraction for Bowl-Type Tableware

Bowl contours are typically composed of straight segments and circular arcs. Because contour extraction and capacity estimation require fitting these geometric elements, it is essential to accurately obtain contour edge coordinates to ensure reliable dimensional and geometric characterization. In image processing, filtering is commonly used for denoising, feature enhancement, and contour smoothing; typical approaches include Gaussian, median, and mean filtering. Gaussian filtering performs a weighted average, effectively suppressing high-frequency random noise and smoothing fine details, but it may weaken edges[9]. Median filtering is a nonlinear technique that replaces each pixel with the median value in its neighborhood; it is particularly effective against salt-and-pepper noise and better preserves edges[10]. Mean filtering is computationally efficient and suitable for uniformly distributed noise, but it tends to blur edges and remove fine details[11]. For the frontal-view and top-down bowl images, we applied all three filters and compared their ability to preserve contour details. The results indicate that median filtering achieves effective denoising while introducing less blur and fewer artifacts, thereby retaining bowl edge information more completely. Accordingly, we adopt median filtering in the proposed pipeline to provide clearer and more accurate inputs for subsequent contour segmentation, feature extraction, and capacity prediction. Fig. 3 shows bowl images processed using the three filtering methods.

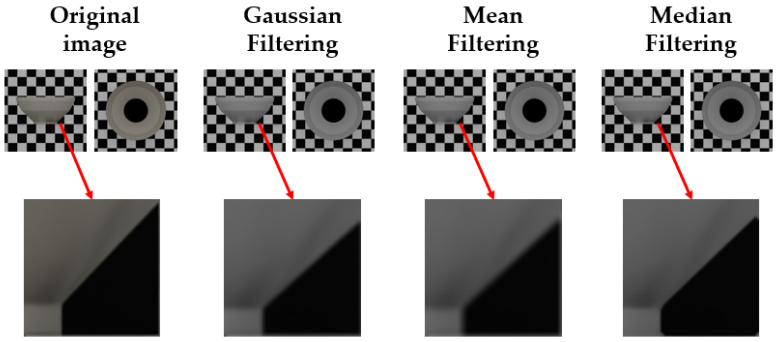


Figure 3. The bowl images processed by three filtering methods.

After filtering the frontal-view and top-down images, we extract the inner and outer rim contours from the top-down view to support subsequent bowl wall-thickness prediction. To this end, we design a two-stage pipeline consisting of (i) image processing and (ii) contour extraction. To evaluate the effectiveness of different method combinations, we tested 40 combinations (each comprising one image-processing method and one contour-extraction method). By visually inspecting and comparing the resulting images, we selected the optimal scheme for subsequent bowl wall-thickness prediction.

In stage (i), we apply seven common image-processing methods to enhance bowl contrast and improve contour separability. (a): CLAHE performs contrast-limited adaptive histogram equalization in local regions, enhancing low-contrast details and emphasizing bowl edges and specular highlights[12]. (b): Histogram Equalization adjusts the global intensity distribution to alleviate uneven background illumination[13]. (c): Inversion reverses pixel intensities to increase the visual distinction between the bowl and the background when their contrast is weak[14]. In morphological processing. (d): Top Hat subtracts the opened image from the original image to enhance bright details and suppress slowly varying background components, thereby highlighting

 doi:10.20944/preprints202512.2802.v1

contour structures[15]. (e): Black Hat subtracts the original image from the closed image to enhance dark-region details, which is beneficial when the bowl's inner surface appears darker[16]. (f): Closing Operation applies dilation followed by erosion to fill small gaps and holes, improving contour continuity. (g): Opening Operation applies erosion followed by dilation to remove small artifacts and isolated noise, producing cleaner images for subsequent contour extraction[17].

In stage (ii), we employ four edge-detection techniques for contour extraction. (a): High-Pass Filter preserves high-frequency components to accentuate intensity transitions, thereby sharpening the bowl-rim contours[18]. (b): Canny applies a multi-step procedure—including denoising, gradient computation, non-maximum suppression, and hysteresis thresholding—to stably extract the primary outer contours and edges while suppressing noise[19]. (c): The Laplace localizes edges using the second-order grayscale derivative; although it is sensitive to noise, it provides complementary capability for capturing inner-wall details[20]. (d): Morphological Gradient extracts contours by computing the difference between dilation and erosion, which is well suited to bowls with regular structures and distinct edges, highlighting the outer boundary while reducing interference from internal texture[21].

By combining the stage-one image processing methods with the stage-two contour extraction methods, we processed the top-down images of bowls and obtained image-processing results for 40 different method combinations. The experimental results show that, as illustrated in Fig. 4, the combination of CLAHE and Canny significantly enhances the difference between the inner and outer walls, outperforming the other methods as well as the original image. Under this setting, both high-pass filter and Canny can extract the inner- and outer-wall contours; however, high-pass filtering is prone to adhesion. In contrast, Canny produces fewer breakpoints and branches, yields smoother contours, and provides clearer separation, making the processed top-down bowl images more suitable for subsequent bowl thickness prediction.

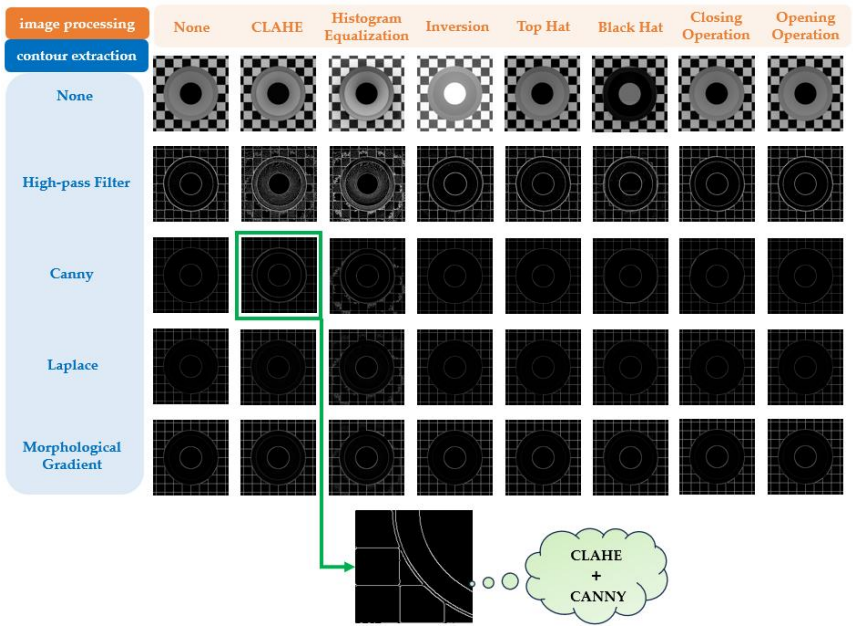


Figure 4. The combined experimental results of the top view of the bowl.

2.3. Bowl Wall Thickness Estimation Model – Bowl Thick Net

In bowl model reconstruction and capacity prediction, accurate estimation of geometric parameters is crucial, particularly because differences between the inner and outer wall structures affect the accuracy of capacity computation. In this study, we adopt Conv NeXt-Tiny as the backbone network and use Res Net-18 as a baseline for comparison. We incorporate a Transformer-based framework with the Hungarian matching algorithm, and investigate the effects of Spatial Positional Encoding and Circle NMS on the recognition accuracy of the inner- and outer-rim contours at the bowl mouth[22]. We design a Bowl Thick Net model that represents the inner and outer rim contours as two equivalent circles. The model detects the circular contours of the inner and outer rims in the

 doi:10.20944/preprints202512.2802.v1

top-down image and predicts bowl wall thickness from the size difference between the two fitted circles. This subsection details the feature extraction module, encoder–decoder architecture, post-processing, underlying principles, and experimental evaluation of the proposed model.

2.3.1. Feature Extraction for the Bowl Thick Net Model

In the feature extraction stage, the input consists of a batch of 640×640 RGB top-down image of a bowl. Features are extracted using a backbone network, where we compare Conv NeXt-Tiny with Res Net-18; this subsection focuses on Conv NeXt-Tiny[23]. Compared with the conventional Res Net-18, Conv NeXt employs convolutional operations such as depth wise convolution and pointwise convolution. Its hierarchical design from Stage 1 to Stage 4 progressively increases the channel dimension while reducing the feature-map resolution, thereby enhancing representational capacity. This design is particularly advantageous over Res Net-18 in capturing fine image details and higher-level features. The first layer is the stem, which maps the three-channel input to 96 channels, producing a feature map of size $B \times 96 \times 160 \times 160$. Stage 1 contains three Conv NeXt blocks and performs feature extraction via depth wise separable and pointwise convolutions, reducing the feature map to 80×80 with 192 channels. Stage 2 contains three Conv NeXt blocks, reducing the feature map to 40×40 with 384 channels. Stage 3 contains nine Conv NeXt blocks, further reducing the feature map to 20×20 with 768 channels. Finally, Stage 4 extracts higher-level features and outputs a feature map of size $B \times 256 \times 20 \times 20$.

2.3.2. Spatial Positional Encodings

To enable the network to capture spatial location information—particularly for detecting circular geometric parameters—we explicitly incorporate Positional Encoding (PE) into the feature maps. Because the Transformer architecture is inherently not position-aware, we generate PE using 2D sine and cosine functions to ensure that the network can model spatial relationships across different locations[24]. The formulations of the 2D sine and cosine positional encodings are given in Eqs. (1) and (2), respectively:

$$PE_{(i,j),2k} = \sin\left(\frac{i}{10000^{\frac{2k}{D}}}\right) \quad (1)$$

$$PE_{(i,j),2k+1} = \cos\left(\frac{i}{10000^{\frac{2k}{D}}}\right) \quad (2)$$

In this equation, i and j denote the row and column indices of a location in the feature map. Since the feature map size is 20×20, i and j range from 0 to 19, i.e., $i, j \in \{0,1,\ldots,19\}$. The variable k is the index of the positional-encoding dimension, and $D = 256$ is the positional-encoding dimensionality. The term $10000^{\frac{2k}{D}}$ is used to control the frequency of the sine or cosine functions at different dimensions, so that the encoding of each position has different scales across dimensions, ensuring encoding diversity and strong positional discriminability.

The PE generated by the above equations is added element-wise to the backbone output feature map F of size $B \times 256 \times 20 \times 20$, yielding a position-aware feature map $\tilde{F}$ with the same size $B \times 256 \times 20 \times 20$. The spatial resolution and channel dimension remain unchanged, while each spatial location now contains explicit positional information. The feature map is then flattened into a sequence Z_0 of size $B \times 400 \times 256$, where 400 is the sequence length after flattening (20×20). Specifically, each pixel in the 20×20 feature map is converted into a 256-dimensional token, preserving spatial correspondence and providing the input for subsequent processing.

2.3.3. Encoder and Decoder

The encoder consists of six layers, each comprising a multi-head self-attention module and a Feed-Forward Network (FFN). In each layer, the input is processed with Add & Norm to ensure stable gradient propagation. The encoder outputs a tensor of size $B \times 400 \times 256$, which is fed into the decoder and serves as the Key and Value in cross-attention. In the decoder, the inputs include the encoder output and 10 learnable object queries. Each query interacts with the encoded features through self-attention and cross-attention. Each decoder layer likewise contains multi-head self-attention and an FFN, with the central objective of generating the final predictions, including the

 doi:10.20944/preprints202512.2802.v1

circle center coordinates c_x, c_y and the circle diameter d. The regression is performed via two branches: the Circle Head, which predicts geometric attributes, and the Obj Head, which determines target existence[25]. The predicted circular parameters are then de-normalized by mapping the coordinates and diameter from $[0,1]$ back to pixel values, as shown in Eq. (3):

$$c_x = c_x^n \times 640, c_y = c_y^n \times 640, d = d^n \times 640 \quad (3)$$

In this equation, c_x, c_y, and d^n are the normalized outputs of the model, representing the normalized x-coordinate of the circle center, the normalized y-coordinate of the circle center, and the normalized circle diameter, respectively. All three values lie in the range $[0,1]$, and 640 denotes the input image resolution.

2.3.4. Post-processing and Matching

In the post-processing and matching stage, to accurately match each predicted circle to its corresponding ground-truth circle, we employ the Hungarian matching algorithm to compute the matching cost between predicted and ground-truth circles. The cost matrix consists of a geometric loss and an existence loss. The geometric loss is measured using the Smooth L1 loss[26], while the existence loss is computed using the BCE With Logits loss[27]. The existence loss is defined in Eq. (4):

$$L_{\text{cxist}} = -\frac{1}{N}\sum_i [y_i \log(\sigma(\hat{y}_i)) + (1 - y_i)\log(1 - \sigma(\hat{y}_i))] \quad (4)$$

In this equation, N denotes the number of samples, which here refers to the total number of detected circles. y_i is the ground-truth label for the i-th sample: $y_i = 1$ indicates that the circle corresponds to a target (i.e., the target exists), whereas $y_i = 0$ indicates that it is not a target (i.e., the target does not exist). $\hat{y}_i$ is the model's raw prediction score for the i-th sample. This score can take any real value and is used to determine whether the sample is a target. $\sigma(\hat{y}_i)$ is the output of the sigmoid activation function, representing the probability that the i-th sample is a target by converting the raw score into a probability. By combining the geometric loss and the existence loss, we obtain the overall matching cost, as defined in Eq. (5):

$$L_{\text{total}} = \lambda_{\text{gco}} \cdot L_{\text{gco}} + \lambda_{\text{exist}} \cdot L_{\text{exist}} \quad (5)$$

In this equation, L_{geo} denotes the geometric loss and L_{exist} denotes the existence loss. λ_{geo} and λ_{exist} are hyperparameters used to balance the respective contributions of the geometric and existence losses.

2.3.5. Circle NMS

Circle NMS is a critical step for handling multiple circle predictions, particularly when detecting the outer and inner rims of a bowl, where several predictions may be produced and some may be redundant or noisy. Circle NMS suppresses duplicate circles based on the distance between circle centers and the relative difference in diameters, ensuring that only the most accurate outer- and inner-rim predictions are retained. This procedure is essential for improving the stability and robustness of the model[28].

2.3.6. Principle of Bowl Thick Net for Wall-Thickness Prediction

The Bowl Thick Net model approximates the inner and outer rim contours at the bowl mouth as two circles. By detecting the two largest circular contours in the top-down image, it maps the predicted circle centers and diameters from the $[0,1]$ range back to pixel values. Using the pixel-to-world mapping and the camera calibration parameters, the diameters of the two circles are then computed in real-world units. The bowl wall thickness is obtained by halving the difference between the two diameters. The corresponding formula is given in Eq. (6):

$$T = \frac{d_{outer} - d_{inner}}{2} \quad (6)$$

In this equation, T denotes the bowl wall thickness, d_{outer} is the diameter of the largest circle (the outer-wall rim contour), and d_{inner} is the diameter of the second-largest circle (the inner-wall rim contour). Fig. 5 illustrates the architecture of Bowl Thick Net: Bowl Wall Thickness Estimation Network.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 January 2026 doi:10.20944/preprints202512.2802.v1

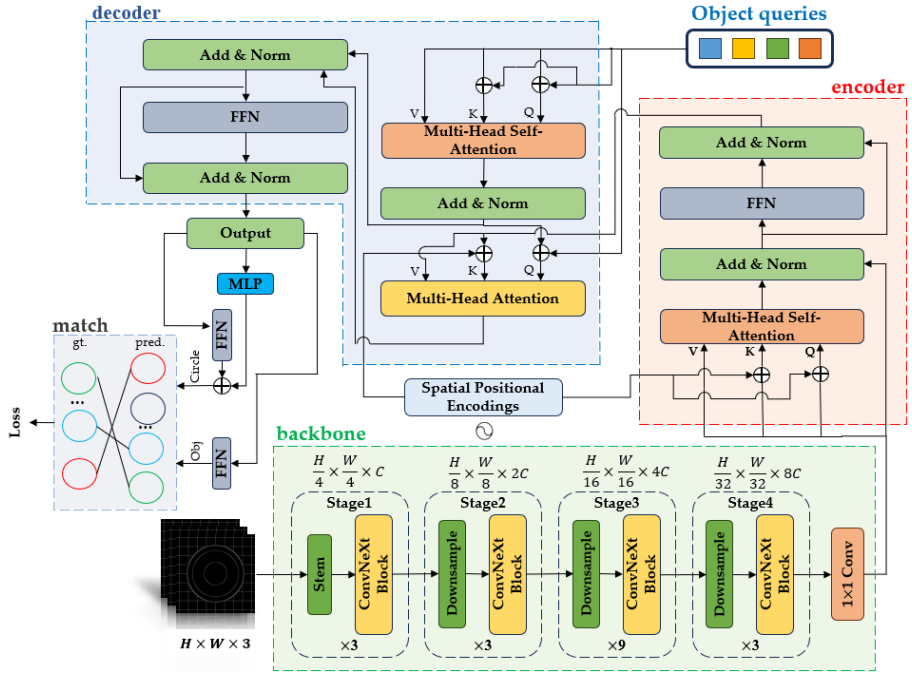


Figure 5. Bowl Thick Net: Model Architecture Diagram of Bowl Wall Thickness Estimation Network.

2.3.7. Experiments on the Bowl Thick Net Model

All images in the dataset were captured using an industrial camera under fixed height and illumination conditions, and were resized to 640×640 to ensure accuracy and consistency. Experiments were conducted on Windows 10 using PyTorch 1.9.2, Python 3.8, CUDA 11.2, and an NVIDIA RTX 4060 GPU. To enlarge the bowl image dataset, we applied data augmentation to expand the original 363 images to 1089 images. The augmentation pipeline included a series of operations such as rotation, cropping, and flipping, while ensuring that these transformations did not alter the mapping between image pixels and physical dimensions. To maintain geometric consistency with the original images, each augmented image was further processed with CLAHE and Canny edge extraction to enhance contrast and edge features, facilitating subsequent contour detection and analysis. For bowls of different sizes and materials, we manually annotated the standard circle parameters of the outer and inner rims, and measured the bowl wall thickness using vernier calipers to construct millimeter-level ground-truth labels. The dataset was split into training and test sets at a ratio of 8:2.

To systematically evaluate the impact of each module on the performance of Bowl Thick Net, we designed eight model variants. The identifiers and meanings of these variants are as follows. Bowl Thick Net uses Conv NeXt-Tiny as the backbone and includes all modules, representing the final model performance when all components operate jointly. A replaces Conv NeXt-Tiny with Res Net-18 to assess the performance of a conventional convolutional network for bowl wall-thickness prediction. B removes Spatial Positional Encoding, such that the model can no longer explicitly exploit positional encoding to capture spatial relationships. C removes Circle NMS, performing no geometric deduplication and directly using the circle parameters output by the decoder. D–F each remove two of the three variables and retain only one, enabling analysis of the effect of a single remaining component. G removes all three variables to examine the maximum performance degradation when they are all absent. Table 1 summarizes the modules used in each model variant.

Table 1. Usage of each model version module.

Model	Replace backbone*	Spatial Positional Encoding	Circle NMS
Bowl Thick Net	√	√	√

 doi:10.20944/preprints202512.2802.v1

A		√	√
B	√		√
C	√	√	
D			√
E		√	
F	√		
G			

* In the "Replace backbone" column, a check mark (√) indicates that the backbone is Conv NeXt-Tiny; otherwise, Res Net-18 is used as the backbone.

All input images were normalized to [0,1] and channel-wise standardized using ImageNet statistics. The backbone was initialized with ImageNet-pretrained weights. Both the encoder and decoder employed a 6-layer, 8-head self-attention architecture, and the number of object queries was set to 10. In the loss function, the weight ratio between the geometric term and the existence term was set to 2:1, and the model was trained for 100 epochs. Optimization was performed using the SGD optimizer, and the mini-batch size and initial learning rate were tuned on the validation set. To improve robustness, early stopping and cross-validation were adopted during training to mitigate overfitting and ensure that the model effectively learns the key geometric features in the images.

A prerequisite for accurate bowl wall-thickness prediction is the correct identification of the two circular contours corresponding to the outer and inner walls in the image. We therefore evaluated the classification accuracy of the largest detected circle C_1(outer-wall contour) and the second-largest detected circle C_2(inner-wall contour) produced by the eight models on 218 test images; experiments on dimensional accuracy are reported subsequently. The results are summarized in Table 2.

Table 2. Accuracy Statistics Table for Two Circle Recognition.

Model	**Number of C_1 images identified**	**Number of C_2 images identified**	**C_1 recognition accuracy (%)**	**C_2 recognition accuracy (%)**	**Both C_1 and C_2 have correct recognition rates (%)**
Bowl Thick Net	212	208	97.2	95.4	93.6%
A	203	196	93.1	89.9	87.4%
B	196	187	89.9	85.8	82.7%
C	194	184	89.0	84.4	81.1%
D	187	158	85.8	72.5	66.9%

 doi:10.20944/preprints202512.2802.v1

E	184	166	84.4	76.1	72.3%
F	185	162	84.9	74.3	67.5%
G	179	171	82.1	78.4	70.6%

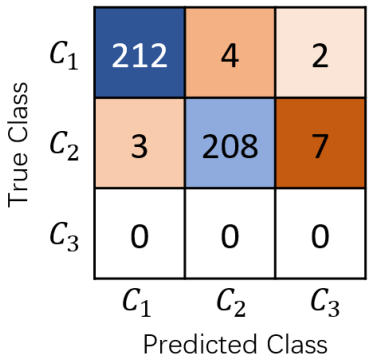


Figure 6. Two circles identify chaotic matrices (Bowl Thick Net).

The results on the 218 test images are summarized in Table 2. Bowl Thick Net achieves strong performance in classifying both the largest circle C_1 (outer-wall contour) and the second-largest circle C_2 (inner-wall contour). The recognition accuracy reaches 97.2% for C_1 and 95.4% for C_2, and the proportion of test images in which both C_1 and C_2 are correctly identified is 93.6%. The other model variants show slightly lower accuracies compared with Bowl Thick Net. In addition, the confusion matrix of Bowl Thick Net, which attains the highest recognition accuracy, is shown in Fig. 6, where C_3 denotes circles other than C_1 and C_2.

From the above model variants, we selected the output images in which both C_1 and C_2 were correctly identified to conduct experiments on the geometric size errors of the two circles. Since subsequent wall-thickness prediction requires only the circle diameters, we did not report the predicted circle-center coordinates. Instead, we de-normalized the predicted diameters to pixel values, computed the corresponding real-world diameters using the pixel-to-world mapping, and then calculated bowl wall thickness according to Eq. (6). We report the predicted diameters of C_1 and C_2 and the resulting wall-thickness estimates for each model variant, and compare them with manual measurements.

In this study, we use Mean Prediction Error (MPE) to quantify the discrepancy between the model-predicted circle diameter or bowl wall thickness and the corresponding manual measurements. MPE computes the absolute error between the predicted and measured values for each image and then averages these errors over all images; a smaller value indicates closer agreement with the ground truth. We also report Mean Wall Thickness Prediction Accuracy ($MWTPA$), defined as the ratio between the predicted wall thickness and the manually measured wall thickness for each image, averaged across all images. This metric effectively reflects the model's accuracy in wall-thickness prediction, with higher values indicating more accurate predictions[29].

Table 3. Predicted results of the diameter and bowl wall thickness of C_1 and C_2 in each model version.

 doi:10.20944/preprints202512.2802.v1

Model	MPE-C_1（mm）	MPE-C_2（mm）	MPE-Wall Thickness（mm）	$MWTPA$（%）
Bowl Thick Net	0.64	0.78	0.75	73.3
A	0.76	0.73	0.88	67.5
B	0.61	0.97	0.96	53.2
C	1.02	1.72	1.21	59.9
D	1.11	1.64	1.39	57.7
E	0.74	0.88	0.91	68.1
F	0.97	1.24	1.33	56.8
G	1.23	1.45	1.41	54.2

Table 3 presents the predicted diameters of $C_1$and $C_2$and the resulting bowl wall-thickness estimates for each model variant. Bowl Thick Net achieves the best overall performance among all versions: the MPE for the diameters of $C_1$and $C_2$are 0.64 mm and 0.78 mm, respectively; the MPE for wall-thickness prediction is 0.75 mm, and the $MWTPA$ is 73.3%. We further conducted capacity measurement experiments on multiple bowls. The results show that, using the wall-thickness predictions from this model, the percentage difference in capacity falls within 1.5%–4%, demonstrating the effectiveness and accuracy of the proposed model in practical applications.

2.4. Mask Segmentation and Volume Estimation of Bowls

In this subsection, we use the frontal-view images paired with the bowl top-down images as input and adopt Mask R-CNN as the base framework. Since each image contains only a single bowl, the primary goal is to obtain a high-quality mask with smooth boundaries and accurate details[30]. However, the standard Mask R-CNN, which is designed for multi-object and multi-class detection, introduces redundant computation and may provide insufficient contour delineation for our task. To address this issue, we build an improved Mask R-CNN framework and compare three backbone networks. We further propose a geometry-aware feature pyramid structure, GFPN (Geometry-aware Feature Pyramid Network), tailored to this study, and investigate its effect on bowl mask segmentation. Finally, we select the model with the highest segmentation accuracy and extract keypoints from the predicted bowl mask to infer the rim diameter, base diameter, bowl height, and the sidewall contour corrected by the predicted wall thickness, which are then used to estimate bowl capacity.

2.4.1. Improved Mask R-CNN framework

In this subsection, we propose an improved Mask R-CNN baseline that is optimized for the characteristics of the mask segmentation task. Conventional Mask R-CNN is primarily designed for multi-object, multi-class detection and must jointly support classification and detection, which leads to substantial computational overhead on classification and insufficient delineation of fine bowl contours. To address these issues, we retain the two-stage architecture and the mask branch, while improving the feature extraction and pyramid fusion components. For region proposal generation, we continue to use the Region Proposal Network (RPN). However, considering the single-bowl image setting and the requirement for high geometric precision, we simplify the RPN structure and its output configuration[31]. At each scale, we keep only a small set of scale and aspect-ratio

 doi:10.20944/preprints202512.2802.v1

combinations that better match bowl contours, thereby reducing the number of proposals and improving convergence efficiency. Standard Mask R-CNN produces hundreds of proposals and then filters them via NMS before further processing; in our task, this is redundant and may introduce uncertainty. Therefore, after NMS we retain only 50–100 high-confidence proposals and feed them into RoI Align and the segmentation head for subsequent processing[32]. In the RoI head, we perform single-class classification without distinguishing specific bowl types. This avoids allocating additional fully connected layers and classification losses for fine-grained categorization. Meanwhile, we keep the original loss functions, which helps the feature representations in the mask and geometric branches remain more focused, thereby improving contour segmentation accuracy.

2.4.2. Three Different Backbone Networks

For the backbone, we adopt three architectures—Res Net-50, Res NeXt-50 (32×4d), and Res Net-C4—to maintain compatibility with the standard Mask R-CNN and to facilitate controlled comparisons. The reasons for selecting these three backbones are as follows:

1. Res Net-50 serves as the baseline for comparison. By introducing residual blocks and skip connections, it effectively alleviates the vanishing-gradient problem in deep networks[33]
2. Res NeXt-50 (32×4d) can be regarded as augmenting the Res Net-50 bottleneck with 32 parallel 3×3 convolutional groups, each with 4 channels, implemented via grouped convolutions, thereby improving texture representation and contour modeling capability[34]
3. Res Net-C4 retains only C1–C4 as the shared backbone and does not further down sample to C5, trading some high-level semantic information for higher spatial resolution and finer local structure, which benefits precise localization of geometric boundaries such as the bowl rim and base[35]

The architectures of the three backbones are illustrated in the upper-left part of Fig. 8. The three boxes correspond to the three backbone variants. Beneath C2, using C2 as an example, the figure further compares the convolutional operation in Res Net-50 with standard convolutions against that in Res NeXt-50 (32×4d) with grouped convolutions. By comparing these three backbones, we can analyze how backbone architecture affects mask quality and geometric fitting accuracy.

2.4.3. Geometry-aware Feature Pyramid Network

The Feature Pyramid Network (FPN) is a multi-scale architecture widely used in object detection. It enhances the recognition of targets at different scales by fusing feature maps from multiple hierarchical levels in a bottom-up pyramid manner. In our task, FPN plays the following roles[35]:

1. Improved contour segmentation accuracy: Because bowl contours may exhibit subtle variations, FPN effectively combines fine-grained details from shallow layers with high-level semantic information from deeper layers, thereby improving the accuracy of bowl contour segmentation. In particular, for segmenting the bowl rim and base, FPN strengthens the model's sensitivity to fine details, ensuring accurate extraction of circular contours
2. Reduced computational redundancy: In conventional Mask R-CNN, processing multi-scale features can introduce redundant computation. FPN mitigates this issue through efficient feature fusion, improving computational efficiency. This advantage is especially important for bowl segmentation tasks that require real-time performance or high-throughput processing
3. Multi-scale feature extraction: FPN extracts features at different hierarchical levels and fuses them, enabling the model to recognize targets at various scales within the image

To further emphasize bowl contour features, we propose a Geometry-aware Feature Pyramid Network (GFPN) that includes only P3–P5, built on the C3–C5 outputs of Res Net-50 and Res NeXt-50 (32×4d). This pyramid follows the standard top-down design with 1×1 convolutions, up sampling, and 3×3 convolutions. On this basis, we introduce a Geometry-aware Prior Modulation (GPM) module and the CBAM (Convolutional Block Attention Module) attention mechanism. With multi-scale feature enhancement, GFPN strengthens boundary responses and thereby improves the accuracy of bowl contour segmentation.

GPM enhances the model's understanding of geometric structures by incorporating geometric prior knowledge. In mask segmentation, GPM modulates the feature maps according to the

 doi:10.20944/preprints202512.2802.v1

geometric shape information in the input image, thereby improving sensitivity to geometric boundaries such as bowl contours[36]. CBAM strengthens feature representation through two components: channel attention and spatial attention. Channel attention models the importance of each channel and selectively amplifies informative channel features, whereas spatial attention emphasizes different spatial locations to increase the model's responses to salient regions[37]. GFPN is derived from the original FPN with targeted modifications. The overall architecture of GFPN is shown in Fig. 7, and the procedure is described as follows:

The frontal-view bowl image is first converted to grayscale, and the resulting grayscale image is denoted as I_g. The corresponding morphological gradient image is defined in Eq. (7):

$$G = \text{dilate}(I_g) - \text{erode}(I_g) \quad (7)$$

In this formulation, $G \in \mathbb{R}^{H \times W}$ takes larger values near the bowl contour and is close to zero in background regions, where H and W denote the height and width of the original image. To inject scale-matched geometric priors into each FPN level, G is downsampled at multiple scales and encoded via convolution. Taking the third level with a stride of 8 as an example, average pooling is first applied to obtain a coarse-scale boundary map G_3, which is then encoded using a 3×3 convolution to produce the geometric prior feature E_3. The details are given in Eqs. (8) and (9):

$$G_3 = \text{AvgPool}_{k=8,s=8}(G) \in \mathbb{R}^{H_3 \times W_3} \quad (8)$$

In these equations, G_3 denotes the coarse-scale boundary map at the third level, $\text{AvgPool}(\cdot)$ represents the average pooling operation, and $H_3 = H/8$ and $W_3 = W/8$.

$$E_3 = \phi(G_3) = \text{Conv}_{3\times3}(G_3) \quad (9)$$

In these equations, E_3 denotes the geometric prior feature corresponding to a stride of 8. $\phi(\cdot)$ is a function symbol representing the convolution operation, which transforms the input map G_3 into a new feature map E_3. $\text{Conv}_{3\times3}(\cdot)$ denotes a 3×3 convolution.

Similarly, for the levels with strides of 16 and 32, the corresponding geometric prior features E_4 and E_5 can be obtained. Thus, for the three levels with strides of 8, 16, and 32, we derive the geometric prior features E_3, E_4, and E_5. These features are spatially aligned with the backbone outputs C3–C5 and serve as the geometric priors for their respective levels.

For feature fusion, we adopt a top-down pathway and retain only P3–P5, which match a single, medium-scale bowl target. Let the backbone outputs at C3, C4, and C5 be C_3, C_4, and C_5, respectively. The top-level pyramid output P_5 is first obtained by applying 1×1 and 3×3 convolutions to C_5 to produce the base fused feature $\hat{F}_5$, as defined in Eq. (10):

$$\tilde{C}_5 = \text{Conv}_{1\times1}(C_5), \hat{F}_5 = \text{Conv}_{3\times3}(\tilde{C}_5) \quad (10)$$

In these equations, $\tilde{C}_5$ denotes the channel-compressed top-level feature, and $\hat{F}_5$ denotes the base fused feature at the top level. $\text{Conv}_{1\times1}(\cdot)$ and $\text{Conv}_{3\times3}(\cdot)$ represent 1×1 and 3×3 convolutions, respectively. For the intermediate level, taking the fourth backbone output C_4 as an example, we first apply a 1×1 convolution to C_4. We then upsample the pyramid feature P_5 by a factor of two so that its spatial resolution matches that of C4. The two features are added element-wise and passed through a 3×3 convolution to obtain the base fused feature for this level. This process is defined in Eqs. (11) and (12):

$$\tilde{C}_4 = \text{Conv}_{1\times1}(C_4), U_5 = \text{Up}(P_5) \quad (11)$$

In these equations, $\tilde{C}_4$ denotes the channel-compressed feature at the fourth level. $\text{Up}(\cdot)$ denotes a two-fold up sampling operation; here, it upsamples the pyramid output from the upper level—P_5 in this example—by a factor of two, producing the upsampled feature U_5.

$$S_4 = \tilde{C}_4 + U_5, \hat{F}_4 = \text{Conv}_{3\times3}(S_4) \quad (12)$$

In these equations, S_4 denotes the fused feature obtained by element-wise addition at the fourth level, and $\hat{F}_4$ denotes the base fused feature at this level. Subsequently, the GPM module applies a 3×3 convolution followed by a sigmoid activation to each level's geometric prior feature E_l to produce a normalized geometric weight map. This weight map modulates the base feature $\hat{F}_l$ in a residual manner, yielding the geometry-modulated feature map F_l at level l. The process is defined in Eqs. (13) and (14):

$$M_l = \sigma(W_g * E_l), M_l \in [0,1]^{H_l \times W_l} \quad (13)$$

 doi:10.20944/preprints202512.2802.v1

In these equations, l denotes the pyramid level (here, $l = 3$ to 5). M_l is the normalized geometric weight map at level l, $\sigma(\cdot)$ denotes the sigmoid function, $*$ denotes the convolution operator, W_g is a 3×3 convolution kernel, and E_l is the geometric prior feature at level l.

$$F_l = \mathrm{GPM}(\hat{F}_l, E_l) = \hat{F}_l \odot (1 + M_l) \quad (14)$$

In these equations, l denotes the pyramid level (here, $l = 3$ to 5). F_l is the feature map at level l after geometric prior modulation. $\mathrm{GPM}(\cdot)$ denotes the geometric prior modulation module. $\hat{F}_l$ is the base feature at level l, and E_l is the geometric prior feature at the same level. $\odot$ denotes element-wise multiplication, and M_l is the normalized geometric weight map for level l.

Finally, we apply the CBAM module to F_l to adaptively reweight the features along both the channel and spatial dimensions, producing the geometry-aware pyramid output P_l. This process is given in Eq. (15):

$$P_l = \mathrm{CBAM}(F_l) \quad (15)$$

In this equation, l denotes the pyramid level (here, $l = 3$ to 5). P_l is the geometry-aware pyramid output at level l, and $\mathrm{CBAM}(\cdot)$ denotes the CBAM module.

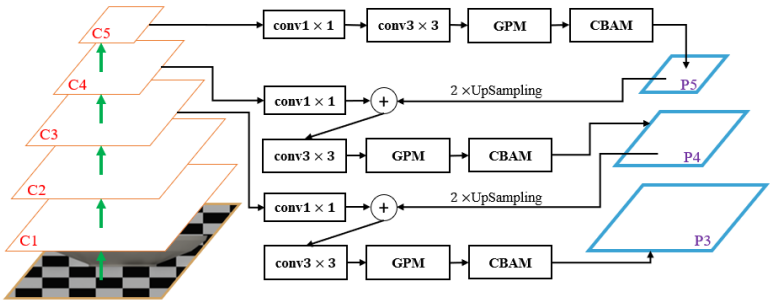


Figure 7. Structure diagram of GFPN module.

2.4.4. Bowl Mask Segmentation Model Process

This subsection summarizes the overall workflow of the proposed bowl mask segmentation model based on an improved Mask R-CNN. First, we adopt the improved Mask R-CNN as the baseline framework. To meet the requirements of single-bowl images and high geometric precision, we optimize region proposal generation and introduce the GFPN structure to strengthen the model's sensitivity to bowl contours. By incorporating the GPM module and the CBAM attention mechanism, the model can more accurately capture fine-grained features of the bowl rim, base, and sidewall. To further improve segmentation accuracy, GFPN enhances the bowl's geometric boundaries across multiple scales and performs feature fusion on the outputs of the C3–C5 layers. With these optimizations, the model effectively reduces background noise interference and improves the accuracy of bowl contour segmentation. Fig. 8 illustrates the architecture of the improved bowl mask segmentation model.

 doi:10.20944/preprints202512.2802.v1

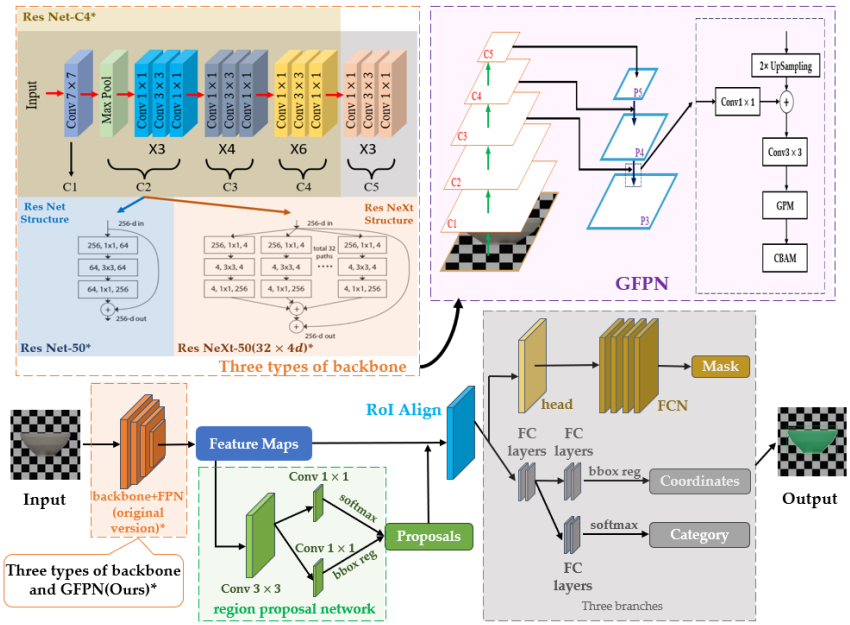


Figure 8. The structure diagram of the mask segmentation model for the improved bowl image. The lower part of the figure illustrates the overall pipeline of the bowl mask segmentation model, from image input to mask output. The three overlapping boxes in the upper-left corner represent the three backbone architectures: Res Net-50, Res NeXt-50 (32×4d), and Res Net-C4. Within the Res Net-50 and Res NeXt-50 (32×4d) backbone boxes, the figure highlights the standard convolutional structure of Res Net-50 and the 32 parallel 3×3 grouped convolutions of Res NeXt-50 (32×4d), with 4 channels per group, which enhances texture and contour modeling capability. The box in the upper-right corner shows the architecture of GFPN.

2.4.5. Experimental Results and Analysis

In this experiment, we use 363 frontal-view images of bowls that were captured together with the corresponding top-down images, and expand the dataset to 1089 images through data augmentation. The dataset includes bowls of various sizes. All images were manually annotated using the LabelMe tool and converted to the standard COCO annotation format. Unlike the original COCO setting, this task does not involve bowl-type classification; therefore, all instances are labeled as a single class, while still retaining bounding boxes and segmentation masks. The hardware and software settings are the same as those in Section 2.3. Each image has an original resolution of 640×640 pixels. The dataset was randomly split, with 80% of the data used for training and 20% used for testing.

In this study, we adopt a Mask R-CNN model pre-trained on the COCO dataset and fine-tune it by loading the corresponding pre-trained weights. To improve accuracy and efficiency, we adjust the RPN_ANCHOR_SCALES parameter to (16, 32, 64, 128, 256) to better accommodate feature extraction for smaller image sizes. The initial learning rate is set to 0.0005, with a momentum of 0.9, weight decay of 0.00005, a batch size of 8, and 100 training epochs to ensure stable optimization on the reduced image resolution. During training, we use a stepwise learning-rate decay schedule, reducing the learning rate by a factor of 0.1 every 15 epochs. To further improve training stability and convergence speed, we employ a staged training strategy. In the early stage, the first several layers of Res Net are frozen and only the subsequent layers are trained, allowing the network to focus on task-relevant features. As training progresses, more Res Net layers are gradually unfrozen, ultimately enabling end-to-end optimization of all layers.

For model evaluation, we adopt COCO-style segmentation metrics, including IOU (Intersection over Union), $mIOU$ (Mean Intersection over Union), AP (Average Precision), and mAP (Mean Average Precision)[38,39]. Specifically, in this experiment, IOU is defined as the ratio between the area of intersection and the area of union of the predicted mask and the ground-truth mask, and $mIOU$ is the average IOU over all images. Since our task involves segmentation of only a single class, mAP is equivalent to AP as used in multi-class settings. The calculation is given in Eq. (16):

 doi:10.20944/preprints202512.2802.v1

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (16)$$

To quantify the model's boundary prediction accuracy, we evaluate it using the $Boundary\ F1\ Score$ with a 3-pixel tolerance. This metric assesses boundary performance by computing $Precision$, defined as the proportion of the predicted boundary that overlaps the ground-truth boundary, and $Recall$, defined as the proportion of the ground-truth boundary that overlaps the predicted boundary[40]. The final $Boundary\ F1\ Score$ is the weighted harmonic mean of $Precision$ and $Recall$, as given in Eq. (17):

$$Boundary\ F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (17)$$

In this experiment, we report $mIOU$ and mAP at two IOU thresholds, 0.50 and 0.75, as well as the mean $mIOU$ and mAP averaged over these two thresholds. The corresponding metrics are $mIOU_{0.5}$, $mIOU_{0.75}$, $mAP_{0.5}$, $mAP_{0.75}$, $mIOU$, and mAP, together with the $Boundary\ F1\ Score$.

To systematically evaluate the effects of different backbones and the proposed GFPN on bowl mask segmentation performance, we conduct a comparative study based on the improved Mask R-CNN baseline with different configurations. Three representative residual-network backbones are considered: Res Net-50, Res NeXt-50 (32×4d), and Res Net-C4. Here, Res Net-C4 serves as a shallower backbone without FPN, whereas the other two backbones are equipped with either the standard FPN or the proposed geometry-aware feature pyramid GFPN. All models are trained and evaluated on the same grayscale-processed frontal-view bowl dataset using the metrics described above. Table 4 summarizes the comparative results of the bowl mask segmentation models.

Table 4. Comparison Experimental Results of Bowl Mask Segmentation Models.

Backbone	FPN Type*	$mIOU_{0.5}$	$mIOU_{0.75}$	$mAP_{0.5}$	$mAP_{0.75}$	$mIOU$	mAP	$Boundary\ F1\ Score$
Res Net-50	FPN	0.8662	0.8634	0.9521	0.9487	0.8658	0.9491	0.84
Res NeXt-50(32×4d)	FPN	0.8835	0.8812	0.9728	0.9676	0.8871	0.9694	0.89
Res Net-C4	/	0.8426	0.8453	0.9139	0.9021	0.8429	0.9083	0.76
Res Net-50	GFPN	0.9024	0.9057	0.9648	0.9532	0.9044	0.9576	0.87
Res NeXt-50(32×4d)	GFPN	0.9257	0.9341	0.9852	0.9817	0.9357	0.9824	0.95

*In the table, "/" indicates that Res Net-C4 is used without FPN or GFPN. The best result for each metric is highlighted in red, and the second-best is highlighted in blue.

Under the same experimental settings and hyperparameters, we select the best-performing configuration described above—using Res NeXt-50 (32×4d) as the backbone and GFPN as the FPN type—which we refer to as Ours, and compare it with other mask segmentation models, including U-Net, HTC, and B Mask R-CNN. All baseline models are evaluated using their original architectures without any module modifications. U-Net employs skip connections to fuse multi-scale features, balancing global semantics and local boundary details for pixel-level mask prediction[41]. HTC alternately optimizes the detection and mask branches and progressively refines predictions through a multi-stage cascade, improving localization and segmentation quality[42]. B Mask R-CNN is a boundary-enhanced variant of Mask R-CNN that introduces contour supervision and feature enhancement to strengthen edge representations and improve mask boundary accuracy and consistency[43]. Table 5 reports the comparative results of these four models. Figure 9 presents

 doi:10.20944/preprints202512.2802.v1

qualitative segmentation outputs for all models, along with the training and validation loss curves of our model and the $mIOU$ and mAP curves of the four models.

Table 5. Comparison of experimental results for four different mask models.

Network Model	$mIOU_{0.5}$	$mIOU_{0.75}$	$mAP_{0.5}$	$mAP_{0.75}$	$mIOU$	mAP	$Boundary\ F1\ Score$
Ours	0.9257	0.9341	0.9852	0.9817	0.9357	0.9824	0.95
U-Net	0.8169	0.8207	0.9094	0.9181	0.8173	0.9122	0.82
HTC	0.8416	0.8468	0.9167	0.9243	0.8455	0.9208	0.87
B Mask R-CNN	0.8825	0.8739	0.9273	0.9361	0.8786	0.9311	0.91

* The best value for each metric in the table is highlighted in red, and the second-best is highlighted in blue.

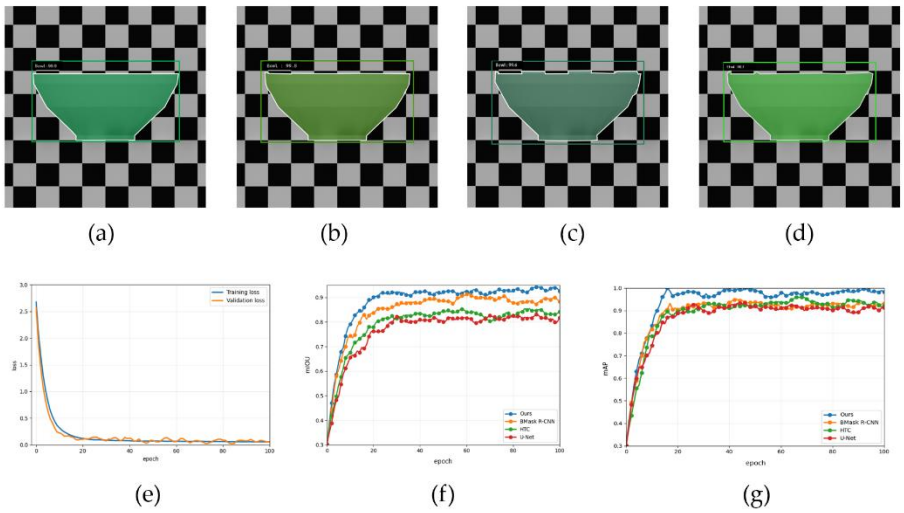


Figure 9. The mask segmentation results of the four models, along with the training and validation loss curves of our models and the $mIOU$ and mAP curves of the four models:(a) The mask segmentation results of our model.(b) B Mask R-CNN mask segmentation results.(c) U-Net mask segmentation results.(d) HTC's mask segmentation results.(e) Our model's training and validation loss curves.(f) $mIOU$ curves of four models.(g) mAP curves of four models.

2.5. Estimation of the Volume of a Bowl

After obtaining the bowl mask, we use the segmentation model from the previous subsection and define 10 geometric keypoints on the mask: the outer-contour extremal points $P_1 \sim P_6$ and four uniformly sampled points along the right-side arc length, $S_1 \sim S_4$. These keypoints are used to derive the rim diameter, base diameter, and bowl height, and to fit the inner-wall contour by incorporating the predicted wall thickness. Finally, the bowl volume is computed via axisymmetric integration[44]. The detailed procedure is as follows:

1. Foreground pixel set and contour set: After bowl mask segmentation, let the binary mask be $M(x,y) \in \{0,1\}$. We first construct the foreground pixel set and the contour set on the mask, which are used for subsequent definition of geometric keypoints and fitting of the outer-wall curve, as given in Eq. (18):

$$\Omega = \{(x,y) \mid M(x,y) = 1\} \quad (18)$$

In this equation, Ω denotes the set of foreground pixels in the mask image. $M(x,y)$is the mask value at pixel (x,y), where the foreground is 1 and the background is 0, and x and $y$are the horizontal and vertical pixel coordinates in the image coordinate system. $\partial\Omega$ denotes the contour set extracted using OpenCV's contour detection.

2. Six outer-contour extremal points $P_1 \sim P_6$: To stably obtain the geometric boundaries of the bowl rim and base from the mask contour, we define four extremal points on $\partial\Omega$: the left and right upper-rim points $P_1$and P_2, and the left and right lower-base points $P_3$and P_4. The upper-rim points

 doi:10.20944/preprints202512.2802.v1

characterize the rim width, whereas the lower-base points characterize the base width and help determine the base position. Their definitions are given in Eqs. (19) and (20):

$$P_1 = \arg\max_{(x,y)\in\partial\Omega, x=\min x} y \ , \ P_2 = \arg\max_{(x,y)\in\partial\Omega, x=\max x} y \quad (19)$$

In this equation, $\min x$ and $\max x$ denote the minimum and maximum horizontal coordinates on the contour set $\partial\Omega$, respectively. $\arg\max(\,)$ denotes the point within the specified domain that maximizes the objective function. $P_1 = (x_{p_1}, y_{p_1})$and $P_2 = (x_{p_2}, y_{p_2})$correspond to the left and right upper-rim points, respectively. Specifically, $P_1$and $P_2$are the highest points in the leftmost and rightmost contour columns, respectively.

$$P_3 = \arg\min_{(x,y)\in\partial\Omega, x=\min x} y \ , \ P_4 = \arg\min_{(x,y)\in\partial\Omega, x=\max x} y \quad (20)$$

In this equation, $\arg\min(\,)$ denotes the point within the specified domain that minimizes the objective function. $P_3 = (x_{p_3}, y_{p_3})$ and $P_4 = (x_{p_4}, y_{p_4})$ correspond to the left and right lower-base points, respectively. Specifically, P_3 and P_4 are the lowest points in the leftmost and rightmost contour columns, respectively.

Because the base primarily provides structural support and does not contribute to the effective holding volume, directly using $P_3$and P_4would lead to an overestimation of bowl height. Therefore, we analyze the variation of the mask's horizontal width along the vertical direction to locate the shape transition between the base and the bowl body. The left and right boundaries at this transition are defined as $P_5$and P_6, which are used for estimating the effective height and fitting the sidewall. The definition is given in Eq. (21):

$$w(y) = x_{max}(y) - x_{min}(y) \quad (21)$$

In this equation, $w(y)$denotes the foreground width of the mask at row y, and $x_{\max}(y)$and $x_{\min}(y)$are the rightmost and leftmost horizontal coordinates of the foreground region at row y, respectively.

In the base region, $w(y)$varies only slightly, whereas once entering the bowl body sidewall, $w(y)$increases markedly as $y$increases. We scan upward from the base and identify the first row at which the width change rate exceeds a threshold τ, denoted as $y = y_{\text{split}}$. In our implementation, $\tau$is set to 5 px, and the left and right boundary points at this split row are defined as $P_5 \sim P_6$, as given in Eq. (22):

$$P_5 = (x_{min}(y_{split}), y_{split}) \ , \quad P_6 = (x_{max}(y_{split}), y_{split}) \quad (22)$$

In this equation, y_{split} is the pixel y-coordinate of the height where the base transitions to the bowl body. $P_5$is the left boundary point at the start of the bowl body after removing the base, and $P_6$is the corresponding right boundary point.

3. Pixel-scale geometric quantities: rim diameter, base diameter, and effective height. After obtaining the keypoints $P_1 \sim P_6$, we can directly compute the bowl rim diameter, base diameter, and effective height in pixel units. The pixel-scale rim diameter is defined in Eq. (23):

$$D_{\text{rim}}^{(pix)} = |x_{P2} - x_{P1}| \quad (23)$$

In this equation, $D_{\text{rim}}^{(pix)}$denotes the rim diameter in pixels, and $x_{P_1}$and $x_{P_2}$are the horizontal coordinates of points $P_1$and P_2, respectively. The pixel-scale base diameter is defined in Eq. (24):

$$D_{\text{base}}^{(pix)} = |x_{P4} - x_{P3}| \quad (24)$$

In this equation, $D_{\text{base}}^{(pix)}$denotes the base diameter in pixels, and $x_{P_3}$and $x_{P_4}$are the horizontal coordinates of points $P_3$and P_4, respectively. The bowl's effective height uses the midpoint of the rim as the upper reference point and the base–body split line defined by $P_5$and P_6as the lower reference, thereby reducing the influence of the base height. The pixel-scale effective height is defined in Eq. (25):

$$H^{(pix)} = \left|\frac{y_{p_5}+y_{p_6}}{2} - \frac{y_{p_1}+y_{p_2}}{2}\right| \quad (25)$$

In this equation, $H^{(pix)}$denotes the effective bowl height in pixels, and y_{p_1}, y_{p_2}, y_{p_5}, and $y_{p_6}$are the vertical coordinates of P_1, P_2, P_5, and P_6, respectively.

4. Conversion from pixel units to physical scale: Using the pixel-to-world mapping obtained in Section 2.1, $P_x = 0.3438mm/pixel$, we convert the pixel-scale measurements to real-world dimensions, including the rim diameter D_{rim}, base diameter D_{base}, and bowl height H. The conversion is given in Eq. (26):

 doi:10.20944/preprints202512.2802.v1

$$D_{\text{rim}} = D_{\text{rim}}^{(pix)} \cdot P_x \ , \ D_{\text{base}} = D_{\text{base}}^{(pix)} \cdot P_x \ , \ H = H^{(pix)} \cdot P_x \quad (26)$$

In this equation, D_{rim}, D_{base}, and H denote the rim diameter, base diameter, and bowl height in real-world units, respectively, while $D_{\text{rim}}^{(pix)}$, $D_{\text{base}}^{(pix)}$, and $H^{(pix)}$ denote the corresponding measurements in pixel units.

5. Sampling the right-side outer contour: Since a bowl can be approximated as an axisymmetric solid generated by rotation about its central axis, we model the bowl using the right-side outer contour. Specifically, we select the contour arc segment from P_6 to $P_2$and denote the contour point sequence as $\Gamma_R = \{(x_i, y_i)\}_{i=0}^{N}$. Using arc-length parameterization, we uniformly sample four interior points $S_1 \sim S_4$ along this arc. Together with the endpoints P_6 and P_2, these points form a representative set that characterizes the outer-wall shape, as defined in Eq. (27):

$$\Delta s_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, i = 1, \dots, N \ , \ s_i = \sum_{j=1}^{i} \Delta s_j, s_0 = 0, s_{\text{tot}} = s_N \quad (27)$$

In this equation, Δs_i denotes the arc-length increment between adjacent contour points. $x_i$and y_i are the horizontal and vertical coordinates of the i-th point, and x_{i-1} and y_{i-1} are defined analogously. Here, i is the contour-point index ranging from 1 to N, where N is the total number of contour points. s_i represents the cumulative arc length from the starting point $P_6$to the i-th point. Δs_j denotes the arc-length increment between the j-th point and the $(j-1)$-th point, where j is the summation index ranging from 1 to i. s_0 is the arc length at the starting point and is set to 0. s_{tot} denotes the total arc length from the starting point $P_6$to the endpoint P_2, i.e., the full length of the selected contour segment. We then select four interior point locations by dividing the arc-length interval $[0, s_{\text{tot}}]$into five equal parts, as defined in Eq. (28):

$$s^{(k)} = \frac{k}{5} s_{\text{tot}}, k = 1,2,3,4 \quad (28)$$

In this equation, $s^{(k)}$denotes the arc-length position of the k-th sampled point, where k is the sampling index and takes values from 1 to 4. By performing linear interpolation on $\Gamma_R$at $s^{(k)}$, we obtain the sampled point $S_k = (x_{S_k}, y_{S_k})$, i.e., the coordinates of $S_1 \sim S_4$.

6. Outer-wall fitting and inner-wall correction: To convert the contour points into a continuous geometric representation, we determine the bowl's central axis using the midpoint of the rim and project the right-side sampled points onto the height–radius plane. In physical units, we fit the outer-wall radius function $r_{\text{out}}(z)$. We then combine this with the wall thickness $\tilde{t}$ predicted by Bowl Thick Net to obtain the inner-wall radius function $r_{\text{in}}(z)$, as defined in Eq. (29):

$$r_{out}(z) = a_3 z^3 + a_2 z^2 + a_1 z + a_0 \ , \ r_{in}(z) = r_{out}(z) - \tilde{t} \quad (29)$$

In this equation, $r_{\text{out}}(z)$ denotes the outer-wall radius as a function of height z, and $r_{\text{in}}(z)$denotes the inner-wall radius as a function of height z. The coefficients a_0, a_1, a_2, and $a_3$are obtained via least-squares fitting.

7. Bowl capacity via axisymmetric integration: After obtaining the inner-wall profile $r_{\text{in}}(z)$, we compute the bowl's inner-cavity volume using the volume-of-revolution formula. The integration limits correspond to the effective height range, starting from the height of P_6at the bottom and ending at the rim at P_2. The formulation is given in Eq. (30):

$$V_{\text{inner}} = \pi \int_{z_{\min}}^{z_{\max}} r_{\text{in}}^2(z) \mathrm{d}z \quad (30)$$

In this equation, V_{inner} denotes the bowl capacity, $z_{\min}$ and $z_{\max}$ are the lower and upper bounds of the effective height in physical coordinates, and $r_{\text{in}}(z)$ is the inner-wall radius function.

In summary, based on the frontal-view bowl images, we use the outputs of our improved mask segmentation model to extract key geometric parameters, including the rim diameter, base diameter, and bowl height. By further incorporating the wall thickness predicted by Bowl Thick Net, we construct axisymmetric contour curves for the inner and outer walls, derive an integral formulation for bowl capacity, and complete the bowl reconstruction model. Fig. 10 illustrates the bowl capacity estimation procedure and the corresponding results.

 doi:10.20944/preprints202512.2802.v1

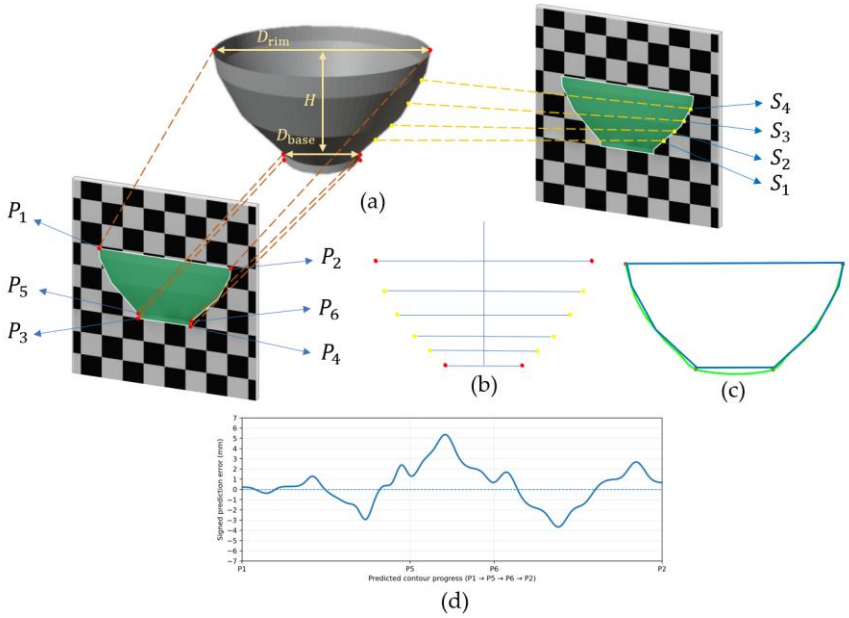


Figure 10. Bowl volume estimation process and results diagram：(a) Reconstruction model of the bowl.(b) Bowl key outline extraction.(c) The predicted inner contour curve and the actual inner contour curve of the bowl (green is the predicted curve, and blue is the actual curve).(d) Statistical chart of error values between the predicted inner contour curve of the bowl and the actual contour curve of the bowl.

Following the above pipeline, we conduct a capacity validation experiment on eight bowls of different sizes. For each bowl, we first acquire a top-down image and a frontal-view image. The rim contour in the top-down image is used to predict wall thickness, while the mask and contour point set from the frontal-view image are used to obtain the bowl's geometric parameters and contour curve. The outer-wall contour is then corrected to the inner-wall contour, yielding image-based geometric parameters, a reconstructed bowl model, and the predicted bowl capacity. As ground-truth references, we manually measure the rim diameter, base diameter, effective height, and actual holding volume of each bowl using tools such as vernier calipers and a graduated cylinder. Finally, we evaluate the accuracy of the proposed bowl model reconstruction and volume estimation method by comparing the predicted and measured geometric dimensions and capacities of the eight bowls on a per-bowl basis and conducting error analysis. Table 6 presents the measured and predicted values of rim diameter, base diameter, effective height, and bowl capacity for each bowl.

Table 6. A comparison table of measured and predicted values for the rim diameter, base diameter, effective height, and volume of each bowl.

	Rim diameter of Bowl			**Base diameter of Bowl**			**Effective height of Bowl**			**Volume of Bowl**		
	MV(mm)*	PV(mm)*	$Error$(%)*	MV(mm)	PV(mm)	$Error$(%)	MV(mm)	PV(mm)	$Error$(%)	MV(mL)	PV(mL)	$Error$(%)
*No.1**	159	163.5	2.83	74	72.3	-2.30	59	59.9	1.53	661	626.6	-5.20
No.2	183	189.2	3.39	85	86.8	2.12	79	77.9	-1.39	714	746.8	4.59
No.3	161	157.0	-2.48	77	78.4	1.82	58	58.6	1.03	608	570.9	-6.10
No.4	137	142.6	4.09	92	90.6	-1.52	74	75.5	2.03	375	387.8	3.41
No.5	151	154.3	2.19	68	68.7	1.03	80	81.3	1.62	458	424.6	-7.29
No.6	167	174.8	4.67	87	89.6	2.99	63	62.2	-1.27	350	335.6	-4.11
No.7	166	160.9	-3.07	63	64.0	1.59	84	82.1	-2.26	673	634.0	-5.79
No.8	160	155.4	-2.87	73	75.5	3.42	72	73.4	1.94	337	324.5	-3.71

* *No.1* to *No.8* denote the bowl IDs. MV is short for **Measured Value**, and PV is short for **Prediction Value**. The error is computed as $Error = \frac{PV-MV}{MV} \times 100\%$.

 doi:10.20944/preprints202512.2802.v1

As shown in Table 6, the prediction error for the rim diameter of the bowl ranges from 2.19% to 4.67%, with an arithmetic mean error of 1.09%. For the base diameter of the bowl, the error ranges from 1.03% to 3.42%, with an arithmetic mean error of 1.14%. For the effective height of the bowl, the error ranges from −2.26% to 1.03%, with an arithmetic mean error of 0.40%. For the bowl volume, the error ranges from −7.29% to 3.41%, with an arithmetic mean error of −3.03%.

2.6. Liquid Volume and Food Nutrient Composition Prediction

After obtaining the key geometric parameters of each bowl (rim diameter, base diameter, effective height, and capacity) and its reconstructed model, we store these parameters together with the corresponding bowl ID in a database, which serves as prior knowledge for subsequent prediction of liquid volume and food nutrient content. In practical use, the system first selects the bowl with the specified ID and captures a top-down image of the liquid or food in the bowl. To ensure consistent scale conversion and geometric mapping, the camera parameters, mounting height, and top-down viewing angle must be kept strictly identical to those used during the acquisition of the bowl top-down images described above. Next, the visible upper-surface region of the contents is segmented to obtain its top-down projection, and a circle is fitted to this region to estimate the diameter of the fitted surface circle. Combined with the inner-wall contour function of the selected bowl stored in the database and the corresponding cavity geometry constraints, this diameter can be mapped to the associated height, enabling inference of the liquid level and computation of the liquid volume via integration. Note that this volume inference assumes that the upper surface of the liquid or food is approximately flat; pronounced concavities, bulges, or stacked structures may introduce additional error. For food nutrient prediction, the system can estimate nutrients for a single food type. After estimating food volume, density is introduced to convert volume to weight, and the nutrient content (e.g., calories, carbohydrates, protein, and fat) is then obtained by linear scaling based on a per-unit-weight nutrient table for different dishes, yielding the estimated nutrient amounts for the food in the bowl[45,46].

2.6.1. Prediction of the Volume of Liquid in the Bowl

After completing bowl mask segmentation and obtaining the bowl geometric parameters as described in the previous subsection, we have the rim diameter D_{rim}, base diameter D_{base}, and effective height H in physical units, as well as the inner-wall radius function $r_{\mathrm{in}}(z)$, which is obtained by fitting the outer wall and then correcting it using the predicted wall thickness. Specifically, z denotes the height coordinate along the bowl's central axis, and $r_{\mathrm{in}}(z)$ is the radius from the inner wall to the central axis at height z. Under the axisymmetry assumption, the bowl cavity can be regarded as a solid of revolution generated by rotating $r_{\mathrm{in}}(z)$ about the central axis. Therefore, when the liquid surface is approximately level, the liquid volume can be computed by integrating the volume of revolution below the liquid level.

We perform mask segmentation on the surface region in the top-down liquid image and fit an equivalent circle to the liquid-surface contour to obtain the pixel-scale radius $R_{\mathrm{liq}}^{(\mathrm{pix})}$. Using the calibration factor P_x（mm/pixel） we convert it to the physical liquid-surface radius r_{liq}, as given in Eq. (31):

$$r_{\mathrm{liq}} = P_x \cdot R_{\mathrm{liq}}^{(\mathrm{pix})} \quad (31)$$

Since the inner-wall profile is represented as a continuous function $r_{\mathrm{in}}(z)$, the liquid level can be inferred from the liquid-surface radius. In implementation, $r_{\mathrm{in}}(z)$ is interpolated within the integration interval defined by the effective height, and the corresponding z_{liq} is solved using a bisection method. The liquid-surface radius r_{liq} and the liquid level z_{liq} satisfy Eq. (32):

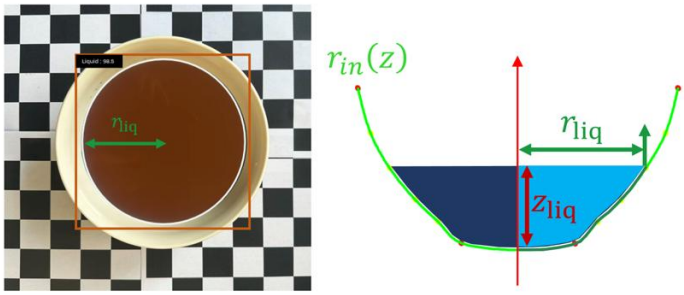
$$r_{\mathrm{in}}(z_{\mathrm{liq}}) = r_{\mathrm{liq}} \quad (32)$$

After determining the liquid level z_{liq}, the liquid volume is computed as the volume of revolution from the bottom z_{min}up to z_{liq}, as given in Eq. (33):

$$V_{\mathrm{liq}} = \pi \int_{z_{min}}^{z_{\mathrm{liq}}} r_{\mathrm{in}}^2(\mathrm{z})\mathrm{dz} \quad (33)$$

In this equation, V_{liq} denotes the predicted liquid volume in mL. Figure 11 illustrates the liquid mask segmentation and the principle of liquid volume prediction.

 doi:10.20944/preprints202512.2802.v1



(a) (b)

Figure 11. Schematic diagram of liquid mask segmentation and its volume prediction： (a) Liquid mask segmentation image. (b) Liquid volume prediction principle diagram.

2.6.2. Predicting the Nutritional Content of Food in a Bowl

After obtaining the bowl geometric parameters and the inner-wall contour function $r_{\text{in}}(z)$, this subsection further implements prediction of the nutrient content of food contained in the bowl. We consider eight common food categories: Rice, Mapo Tofu, Kung Pao Chicken, Fried Noodles, Stir-fried Vegetables, Braised Eggplant, Tomato Scrambled Eggs, and Stir-fried Shredded Potatoes. For prediction, the food is placed in a bowl that has already been registered in the database, and the food surface is kept as level as possible, without pronounced concavities or protrusions. We then capture images under the same top-down setup as in the previous experiments, keeping the camera parameters and viewing angle unchanged, and perform instance mask segmentation on the food surface region. Unlike the single-class frontal-view segmentation of the "bowl" in Section 2.4, this subsection requires simultaneous dish category recognition during segmentation. Therefore, the adopted mask segmentation network performs category classification alongside mask prediction. Table 7 lists the eight food categories and the number of samples within different portion-size ranges, which are used to train and evaluate the model.

Table 7. Eight food categories and the number of samples in different serving sizes.

		Number of images of food within each weight range				
Food Types	**Number of images**	$W * < 100g$	$100g \leq W < 200g$	$200g \leq W < 300g$	$300g \leq W < 400g$	$W \geq 400g$
Rice	194	40	70	36	42	6
Mapo Tofu	86	19	16	22	23	5
Kung Pao Chicken	71	16	1	37	15	1
Fried Noodles	58	10	19	13	16	0
Stir-fried Vegetables	64	17	16	16	9	6
Braised Eggplant	51	7	14	15	8	8
Tomato Scrambled Eggs	76	19	13	19	20	5

 doi:10.20944/preprints202512.2802.v1

Stir-fried Shredded Potatoes	43	17	8	14	1	2

* W is short for *Weight*.

In the geometric estimation stage, the procedure is consistent with the liquid volume prediction in Section 2.7.1. First, an equivalent circle radius (in pixel units) is obtained by fitting a circle to the segmented food surface in the top-down image, and the calibration factor is used to convert it to a physical-scale radius. Next, the corresponding height z_{food} is inferred from the inner-wall function $r_{\text{in}}(z)$, and the food volume in the bowl, V_{food}, is computed via axisymmetric integration. We then introduce the density parameter ρ_c for each food category and convert volume to weight as $m_{\text{food}} = \rho_c \cdot V_{\text{food}}$. Finally, based on the per-unit-weight contents of calories, carbohydrates, protein, and fat for each dish, we compute the total nutrient amounts for the serving in the bowl, thereby completing nutrient content prediction. Figure 12 presents the workflow for food nutrient content prediction.

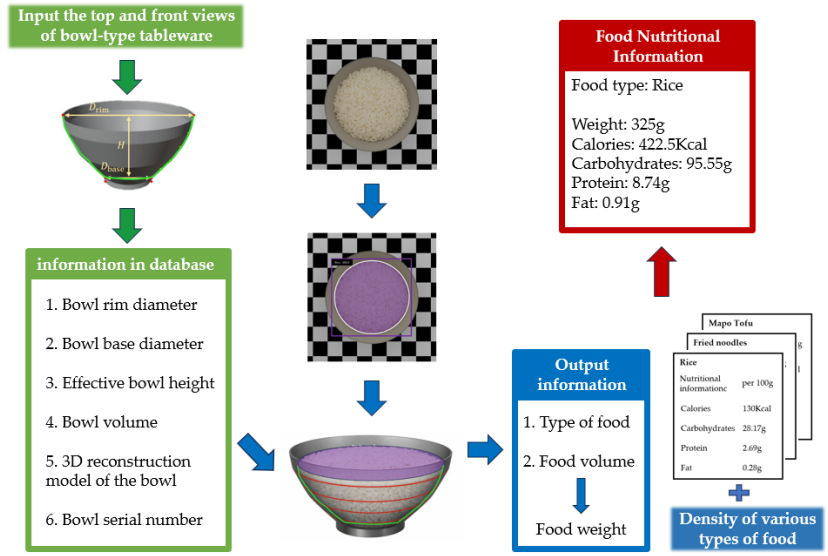


Figure 12. Flowchart for predicting the nutritional content of food.

2.6.3. Experimental Results and Analysis of Estimating the Relationship between Liquid Volume and Food Nutrient Content

In this subsection, under the same experimental equipment and imaging conditions as in Section 2.4.5 (with identical camera parameters, top-down viewing angle, resolution, and calibration factor), we evaluate the performance of different mask segmentation models on two tasks: liquid volume estimation in bowls and food volume estimation in bowls. In our dataset, there are 48 images of liquids in bowls, and the number of images for the eight food categories is as listed in Table 2. We assign an ID to each image and record the corresponding liquid volume and food weight in a spreadsheet. Data augmentation is applied to both the liquid and food images, resulting in 144 augmented liquid images and 1,929 augmented images in total for the eight food categories. We then train standard Mask R-CNN models for the two tasks separately. To ensure a fair comparison, both tasks use the same input preprocessing pipeline and the same training/test split strategy; the only difference lies in the category space. The liquid task is a single-class mask segmentation problem that distinguishes only liquid from background, whereas the food task is a multi-class instance segmentation problem that outputs both masks and category labels, with the category set consisting of the eight dishes listed in Table 7.

During inference, the model first outputs the mask of the liquid or the visible food surface in the bowl, and the food task additionally outputs the predicted category. The mask is then processed by geometric circularization. Specifically, boundary points are extracted from the mask, and the OpenCV toolbox is used for contour extraction and circle fitting. Based on the extracted boundary

 doi:10.20944/preprints202512.2802.v1

point set, an equivalent circle is estimated using a least-squares strategy, yielding the pixel-scale radius of the fitted surface circle. This radius is converted to a physical radius using the pixel-to-world mapping factor P_x. Because the bowl cavity is a solid of revolution formed by rotating the inner-wall profile $r_{\text{in}}(z)$ about the central axis, a given surface radius corresponds to a unique height. Therefore, the height of the liquid level or food surface above the bowl bottom can be obtained by solving $r_{\text{in}}(z) = r$. Finally, given the upper height limit, we integrate $r_{\text{in}}(z)$ using the axisymmetric volume formula to obtain the liquid volume or food volume in the bowl. For the food task, after volume estimation, the nutrient amounts are further quantified by combining the dish-specific density and the per-unit-mass nutrient table for each category.

We evaluate the mask segmentation quality of different models using COCO-style metrics, including the mean average precision for bounding-box detection ($bbox\_mAP$) and the mean average precision for mask segmentation ($mask\_mAP$).

In the bowl liquid-volume prediction experiment, we use the Mask R-CNN segmentation model to extract surface masks from the liquid images in the test set and predict liquid volume according to the aforementioned method. The predicted volumes are then compared with the recorded volumes in the spreadsheet to compute the prediction error. The liquid-surface mask segmentation performance and the liquid-volume prediction results are summarized in Table 8.

Table 8. Experimental Results of Mask Segmentation Performance and Volume Prediction of Liquid Surface Inside Bowl.

Test parameters	Result
$bbox\_mAP$	0.9453
$mask\_mAP$	0.8561
$Mean\ Liquid\ volume\ prediction\ error$*(%)	9.24

*$Mean\ Liquid\ volume\ prediction\ error = \frac{1}{N}\sum_{i=1}^{N}\frac{|Predict\ liquid\ volume - Measure\ liquid\ volume|}{Measure\ liquid\ volume} \times 100\%$, N denotes the number of test images.

As shown in Table 8, the mean prediction error for liquid volume estimation is 9.24%. We measured the density of each food category using a 200 mL measuring cup, following the same filling procedure for all foods. The nutrient composition data used in this study were primarily collected from public food nutrition databases such as FatSecret, from which we obtained nutrient values per 100 g for each dish[47]. Table 9 summarizes the food densities and the nutrient contents per unit weight for each category.

Table 9. Density and content of various nutrients per unit weight of various foods.

Food Types	$Density(g/mL)$	$Calories*(Kcal)$	$Carbohydrates(g)$	$Protein(g)$	$Fat(g)$
Rice	0.667	130	28.17	2.69	0.28
Mapo Tofu	0.943	119	5.96	5.91	8.38
Kung Pao Chicken	0.684	268	14.03	17.14	16.13

 doi:10.20944/preprints202512.2802.v1

Fried Noodles	0.932	88	11.23	5.88	2.27
Stir-fried Vegetables	0.537	106	4.96	2.29	8.35
Braised Eggplant	0.867	140	14.14	2.39	8.19
Scrambled Eggs with Tomatoes	0.916	179	5.07	13.25	11.58
Stir-fried Shredded Potatoes	0.527	71	15.73	1.67	2.48

* In the table, $Calories$, $Carbohydrates$, $Protein$, and Fat denote the amounts of the corresponding nutrients contained in 100 g of each dish.

In the food nutrient content prediction experiment, we likewise use a Mask R-CNN segmentation model to extract surface masks from the food images in the test set and estimate food volume according to the aforementioned procedure. The estimated volume is converted to food weight using the density of the corresponding food category, and the predicted weight is compared with the recorded weights in the spreadsheet to compute the prediction error. We then estimate the nutrient amounts by combining the predicted food weight with the per-100 g nutrient values in Table 9. Therefore, the accuracy of nutrient content prediction is closely tied to the accuracy of the predicted food weight. Under this approach, the prediction error of food nutrient content is equivalent to the prediction error of food weight. The mask segmentation performance for different food categories and the weight prediction results are reported in Table 10.

Table 10. Experimental Results of Food Mask Segmentation Performance and Weight Prediction on Food Surface Inside Bowl.

Food Types	***bbox_mAP***	***mask_mAP***	***Mean Food weight prediction error*** * (%)
Rice	0.9463	0.8537	8.55
Mapo Tofu	0.9316	0.8359	11.49
Kung Pao Chicken	0.9041	0.8280	13.61
Fried Noodles	0.9324	0.8538	10.37
Stir-fried Vegetables	0.9268	0.8103	14.91
Braised Eggplant	0.9574	0.8458	9.67
Scrambled Eggs with Tomatoes	0.9407	0.8174	11.28

 doi:10.20944/preprints202512.2802.v1

Stir-fried Shredded Potatoes	0.9352	0.8566	12.04

* $Mean\ Food\ weight\ prediction\ error = \frac{1}{N}\sum_{i=1}^{N}\frac{|Predict\ food\ weight - Measure\ food\ weight|}{Measure\ food\ weight} \times 100\%$, N denotes the number of test images.

As shown in Table 10, the mean weight prediction error for each food category ranges from 8.55% to 14.91%, and the overall mean weight prediction error across the eight food categories is 11.49%. Stir-fried Vegetables exhibits the largest weight prediction error. This is mainly because the density of a given dish is not a fixed constant and can vary substantially with cooking and serving conditions. Specifically, even within the same category, differences in oil usage, moisture content of sauces, and ingredient composition can directly change the solid content and porosity per unit volume. In addition, the stacking pattern and degree of compaction during filling affect the actual volume distribution and effective density of the food in the bowl, which in turn leads to larger deviations in weight prediction.

2.6.4. Visual User Platform

Based on the above analysis pipeline, we developed a visualization and user-facing platform, as shown in Fig. 13. After a one-time camera calibration, the user only needs to provide a frontal-view and a top-down image of a bowl. The system can then automatically identify geometric parameters such as the rim diameter, base diameter, effective height, and the inner-wall contour curve, and stores the recognized results and the reconstructed model as a bowl instance in the database. For liquid volume and food nutrient content prediction, the user selects the bowl type and inputs a top-down image of the liquid or food in the bowl, and the platform outputs the predicted liquid volume or the food category together with its estimated nutrient content.

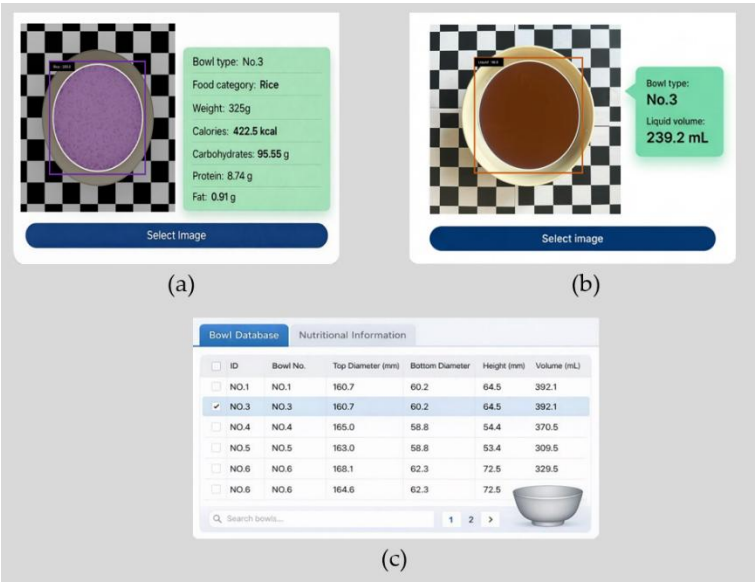


Figure 13. Estimation and Visualization Platform for the Volume of Liquid in Bowls：(a) Food nutrient content estimation page. (b) Liquid volume prediction page. (c) Bowl type and parameter data page.

The platform can be applied to standardized portioning and output management in food retail, dietary intake monitoring in hospitals or elderly-care settings, calorie logging for fitness and weight management, as well as quantitative liquid dispensing and container capacity assessment under laboratory conditions. Looking ahead, the system can be extended to more vessel types and more complex food geometries, and multi-view or depth cues can be incorporated to improve robustness to uneven surfaces and occlusions. In addition, developing finer-grained prediction modules and

 doi:10.20944/preprints202512.2802.v1

providing AI-driven dietary recommendations could further enhance the generalization capability and interpretability of the nutritional assessment.

3. Conclusions

To address the low accuracy of conventional methods for predicting the liquid volume and food nutrient content in bowl-type tableware, as well as the tool dependence and time-consuming nature of manual measurements, this study proposes a new approach that integrates 3D reconstruction with deep learning–based segmentation to predict the liquid volume in a bowl and the nutrient content of the contained food with relatively high accuracy.

Section 2.1 ensures accurate bowl dimension estimation by establishing a pixel-to-world mapping based on the pinhole camera model. Zhang's calibration method is adopted, where 15 images of a 25 mm checkerboard are captured for corner detection to estimate camera intrinsics, extrinsics, and distortion parameters, followed by nonlinear least-squares refinement. The mean reprojection error is 0.23 pixels, and a scale factor of $P_x = 0.3438mm/pixel$ at a resolution of 640×640 is obtained for subsequent conversion. Section 2.2 targets the requirements of contour fitting. We first compare Gaussian, mean, and median filtering, and select median filtering to balance denoising and edge preservation. We then construct a two-stage pipeline comprising image enhancement and edge extraction, and evaluate 40 combinations formed by seven enhancement methods and four edge-detection techniques. The results show that the combination of CLAHE and Canny most stably distinguishes the inner and outer rim contours, providing high-quality inputs for wall-thickness prediction.

Section 2.3 presents Bowl Thick Net for bowl wall-thickness prediction. Using Conv NeXt-Tiny as the backbone with ResNet-18 as a baseline, the model introduces a Transformer-based encoder–decoder architecture and incorporates spatial positional encoding, Hungarian matching, and Circle NMS to improve the stability of detecting the two circles corresponding to the inner and outer rims in top-down images. The model outputs the circle centers and diameters; after de-normalization and conversion using the scale factor P_x, the wall thickness is computed. The model is trained on 363 images augmented to 1089, and evaluated on 218 test images. Bowl Thick Net achieves recognition accuracies of 97.2% for C_1 and 95.4% for C_2 with 93.6% of images having both circles correctly identified. The MPE of the predicted diameters are 0.64 mm for C_1 and 0.78 mm for C_2, and the wall-thickness prediction MPE is 0.75 mm, with an $MWTPA$ of 73.3%. Capacity measurement experiments on multiple bowls further show that, using the wall-thickness predictions from this model, the capacity difference falls within 1.5%–4%, demonstrating its effectiveness and accuracy in practical applications.

Section 2.4 takes frontal-view bowl images as input and proposes a mask segmentation and capacity estimation pipeline tailored to a single-bowl scenario with high geometric precision. Built on Mask R-CNN, the method retains the two-stage architecture and mask branch, while simplifying RPN proposals for the single-object setting to reduce redundant computation and improve contour delineation. Three backbones—Res Net-50, Res NeXt-50 (32×4d), and Res Net-C4—are compared. In addition, we improve the FPN on the C3–C5 features by proposing GFPN, which incorporates GPM geometric priors and CBAM attention to enhance multi-scale boundary responses. Using 363 images augmented to 1089, we evaluate performance with $mIOU$, mAP, and $Boundary\ F1\ Score$. Res NeXt-50 with GFPN achieves the best results and outperforms U-Net, HTC, and B Mask R-CNN. Section 2.5 defines 10 keypoints $(P_1 \sim P_6, S_1 \sim S_4)$on the high-quality masks. We construct the foreground and contour sets, locate extremal points for the rim and base, and use $P_5$and $P_6$to exclude the base and obtain the effective height. The rim diameter, base diameter, and height are then computed and converted to millimeters. Next, we uniformly sample the arc length of the right-side contour to fit the outer-wall function. Combined with the wall thickness predicted by Bowl Thick Net, the outer-wall curve is corrected to the inner-wall curve. Finally, bowl capacity and the reconstructed model are obtained via axisymmetric integration and validated through error analysis, with the minimum bowl-volume prediction error reaching 3.41%.

Section 2.6 stores each bowl's rim diameter, base diameter, effective height, capacity, and inner-wall function in a database indexed by bowl ID after the geometric parameters and reconstructed

 doi:10.20944/preprints202512.2802.v1

model are obtained. During prediction, the user selects the bowl type and captures a top-down image of the liquid or food. Under consistent camera parameters and viewing angle, the visible upper surface is segmented and circularly fitted to obtain the surface radius. The corresponding height is then inferred from the inner-wall function, and the liquid or food volume is computed via axisymmetric integration. For food nutrient content prediction, a classification head is further retained to recognize the eight dish categories. The estimated volume is converted to weight using dish-specific density, and calories, carbohydrates, protein, and fat are computed according to the per-unit-mass nutrient table for each dish. Experimental results show a mean prediction error of 9.24% for liquid volume and a mean prediction error of 11.49% for nutrient content by weight across the eight food categories.

4. Discussion

In this subsection, we discuss several important issues, including the assumptions, limitations, and applicability of our method for bowl capacity prediction, liquid volume estimation in bowls, and nutrient content prediction for food contained in bowls. The proposed approach simplifies the bowl cavity as an axisymmetric solid of revolution about the central axis and represents its geometry using a continuous inner-wall function. The top-down mask of the liquid or food surface is fitted with an equivalent circle radius, which is then used to infer the corresponding height and compute volume via axisymmetric integration. This assumption is effective for regular, circular bowls; however, for vessels with non-circular rims, eccentric structures, handles, or pronounced local irregularities on the inner wall, geometric consistency is violated, leading to systematic bias. In this study, scale conversion relies on camera calibration and the pixel-to-physical mapping. Therefore, during inference, the camera parameters, mounting height, top-down viewing angle, and resolution should be kept as close as possible to the experimental setup to ensure that the mask radius and the inner-wall function are defined in the same geometric coordinate system. If the capture distance changes, the lens differs, or the viewing angle deviates, the pixel-to-physical mapping must be recalibrated; however, recalibration is unnecessary when the system is deployed as a fixed device mounted on a stand or tabletop. Moreover, liquid volume estimation assumes that the liquid surface can be approximated as a flat, level plane, and it is primarily validated under homogeneous liquid conditions. When solid particles, foam, wall adhesion, high-viscosity fluids, or substantial surface fluctuations occur, the equivalent circular radius from the top-down view no longer corresponds to a unique and stable height, and the inferred volume may be distorted. Similarly, food estimation assumes a relatively flat upper surface; stacked peaks, concavities, or local collapses can introduce mapping errors and lead to larger prediction deviations. Finally, nutrient prediction is based on a chained assumption: volume is converted to weight using density, and nutrients are then linearly scaled using per-100 g nutrition tables. Because the density of a dish is not constant—varying with oil usage, moisture content of sauces, ingredient composition, and packing/compaction during serving—weight errors can propagate to estimates of calories and macronutrients. Despite these limitations, the key advantage of our method is that it unifies frontal-view and top-down information into a physical-scale representation and enables a closed-loop, interpretable inference pipeline from images to geometric parameters and then to volume estimation. The workflow does not require depth sensors and can be stably reused under controlled imaging conditions after a one-time calibration. Axisymmetric integration gives the volume and nutrient estimates clear physical meaning, facilitating error analysis and practical deployment. In addition, database management enables rapid switching among multiple bowls and supports batch prediction.

Supplementary Materials: Conceptualization, X J. and Y.F.; methodology, X J.; validation, X.J.; investigation, X J., H L., K S., H Z., Y.F.; resources, L.S.; data curation, X J., H L.; writing—original draft preparation, X J.; writing—review and editing, Y.F., X.J., H L., L.S., K S., H.Z.; visualization, H L.; supervision, Y.F.; funding acquisition, L.S.; project administration, Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Research Project of Liaoning Provincial Applied Basic Research Program Project(2025JH2/101330041) and the National Key R&D Program of China (2018YFD0400800).

 doi:10.20944/preprints202512.2802.v1

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article/Supplementary Materials. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
CLAHE	Contrast Limited Adaptive Histogram Equalization
Circle NMS	Circle Non-Maximum Suppression
RGB	Red Green Blue
PE	Positional Encoding
FFN	Feed-Forward Network
SGD	Stochastic Gradient Descent optimizer
MPE	Mean Prediction Error
$MWTPA$	Mean Wall Thickness Prediction Accuracy
FPN	Feature Pyramid Network
GFPN	Geometry-aware Feature Pyramid Network
GPM	Geometry-aware Prior Modulation module
CBAM	Convolutional Block Attention Module
Conv	Convolution
RPN	Region Proposal Network
RoI	Region of Interest
RoIAlign	Region of Interest Align
IOU	Intersection over Union
$mIOU$	Mean Intersection over Union
AP	Average Precision
mAP	Mean Average Precision
MV	Measured Value
PV	Prediction Value

References

1. Lo, F.P.-W.; Qiu, J.; Jobarteh, M.L.; Sun, Y.; Wang, Z.; Jiang, S.; Baranowski, T.; Anderson, A.K.; Mccrory, M.A.; Sazonov, E. AI-enabled wearable cameras for assisting dietary assessment in African populations. *NPJ Digital Medicine* **2024**, *7*, 356.
2. Lee, D.-s.; Kwon, S.-k. Amount estimation method for food intake based on color and depth images through deep learning. *Sensors* **2024**, *24*, 2044.
3. Yan, R.; Luo, H.; Lu, J.; Liu, D.; Posluszny, H.; Dhaliwal, M.P.; MacLeod, J.; Qin, Y.; Yang, C.; Hartman, T.J. DietAI24 as a framework for comprehensive nutrition estimation using multimodal large language models. *Communications Medicine* **2025**, *5*, 458.
4. Dehais, J.; Anthimopoulos, M.; Shevchik, S.; Mougiakakou, S. Two-view 3D reconstruction for food volume estimation. *IEEE transactions on multimedia* **2016**, *19*, 1090–1099.
5. Lo, F.P.-W.; Sun, Y.; Qiu, J.; Lo, B. Food volume estimation based on deep learning view synthesis from a single depth map. *Nutrients* **2018**, *10*, 2005.
6. Jia, W.; Ren, Y.; Li, B.; Beatrice, B.; Que, J.; Cao, S.; Wu, Z.; Mao, Z.-H.; Lo, B.; Anderson, A.K. A novel approach to dining bowl reconstruction for image-based food volume estimation. *Sensors* **2022**, *22*, 1493.
7. Gutiérrez-Moizant, R.; Boada, M.J.L.; Ramírez-Berasategui, M.; Al-Kaff, A. Novel Bayesian Inference-Based Approach for the Uncertainty Characterization of Zhang's Camera Calibration Method. *Sensors* **2023**, *23*, 7903.

 doi:10.20944/preprints202512.2802.v1

8. Hao, Y.; Tai, V.C.; Tan, Y.C. A systematic stereo camera calibration strategy: Leveraging latin hypercube sampling and 2k full-factorial design of experiment methods. *Sensors* **2023**, *23*, 8240.
9. Liang, L.; Chen, J.; Shi, J.; Zhang, K.; Zheng, X. Noise-Robust image edge detection based on multi-scale automatic anisotropic morphological Gaussian Kernels. *PLoS One* **2025**, *20*, e0319852.
10. Guan, S.; Liu, B.; Chen, S.; Wu, Y.; Wang, F.; Liu, X.; Wei, R. Adaptive median filter salt and pepper noise suppression approach for common path coherent dispersion spectrometer. *Scientific Reports* **2024**, *14*, 17445.
11. Julia, A.; Iguernaissi, R.; Michel, F.J.; Matarazzo, V.; Merad, D. Distortion correction and denoising of light sheet fluorescence images. *Sensors* **2024**, *24*, 2053.
12. Buriboev, A.S.; Khashimov, A.; Abduvaitov, A.; Jeon, H.S. CNN-Based Kidney Segmentation Using a Modified CLAHE Algorithm. *Sensors* **2024**, *24*, 7703.
13. Ahmad, M.S.Z.; Aziz, N.A.A.; Lim, H.S.; Ghazali, A.K.; Latiff, A.A. Impact of Image Enhancement Using Contrast-Limited Adaptive Histogram Equalization (CLAHE), Anisotropic Diffusion, and Histogram Equalization on Spine X-Ray Segmentation with U-Net, Mask R-CNN, and Transfer Learning. *Algorithms* **2025**, *18*, 796.
14. El Houby, E.M. Acute lymphoblastic leukemia diagnosis using machine learning techniques based on selected features. *Scientific Reports* **2025**, *15*, 28056.
15. He, Y.; Kang, S.; Li, W.; Xu, H.; Liu, S. Advanced enhancement technique for infrared images of wind turbine blades utilizing adaptive difference multi-scale top-hat transformation. *Scientific Reports* **2024**, *14*, 15604.
16. Cao, S.; Zhao, C.; Dong, J.; Fu, X. Ship detection in synthetic aperture radar images under complex geographical environments, based on deep learning and morphological networks. *Sensors* **2024**, *24*, 4290.
17. Zhong, X.; Liang, G.; Meng, L.; Xi, W.; Gu, L.; Tian, N.; Zhai, Y.; He, Y.; Huang, Y.; Jin, F. Automated Particle Size Analysis of Supported Nanoparticle TEM Images Using a Pre-Trained SAM Model. *Nanomaterials* **2025**, *15*, 1886.
18. Xie, W.; Zhou, D.; Zhang, W.; Wang, W. EAFormer: Edge-Aware Guided Adaptive Frequency-Navigator Network for Image Restoration. *Sensors* **2025**, *25*, 5912.
19. Li, Y.; Zhang, D. Toward Efficient Edge Detection: A Novel Optimization Method Based on Integral Image Technology and Canny Edge Detection. *Processes* **2025**, *13*, 293.
20. Avendaño, J.C.; Leander, J.; Karoumi, R. Image-based concrete crack detection method using the median absolute deviation. *Sensors* **2024**, *24*, 2736.
21. Wu, Y.; Li, Q. The algorithm of watershed color image segmentation based on morphological gradient. *Sensors* **2022**, *22*, 8202.
22. Choi, C.-Y.; Lee, S.-W. NeXt-DETR: A scalable and efficient transformer-based detector for resource-constrained systems. *Information Sciences* **2025**, 122913.
23. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022; pp. 11976–11986.
24. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European conference on computer vision, 2020; pp. 213–229.
25. Zhang, H.; Liang, P.; Sun, Z.; Song, B.; Cheng, E. CircleFormer: Circular nuclei detection in whole slide images with circle queries and attention. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2023; pp. 493–502.
26. Girshick, R. Fast r-cnn. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2015; pp. 1440–1448.
27. Huang, Y.; Kruyer, A.; Syed, S.; Kayasandik, C.B.; Papadakis, M.; Labate, D. Automated detection of GFAP-labeled astrocytes in micrographs using YOLOv5. *Scientific Reports* **2022**, *12*, 22263.
28. Li, Y.; Bao, H.; Ge, Z.; Yang, J.; Sun, J.; Li, Z. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2023; pp. 1486–1494.
29. Zuo, L.a.; Ling, J.; Hu, N.; Chen, R. Establishment and validation of a population pharmacokinetic model for apatinib in patients with tumors. *BMC cancer* **2024**, *24*, 1346.

 doi:10.20944/preprints202512.2802.v1

30. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2017; pp. 2961–2969.
31. Zhao, K.; Wang, X.; Chen, X.; Zhang, R.; Shen, W. Rethinking mask heads for partially supervised instance segmentation. *Neurocomputing* **2022**, *514*, 426–434.
32. He, M.; He, K.; Huang, Q.; Xiao, H.; Zhang, H.; Li, G.; Chen, A. Lightweight mask R-CNN for instance segmentation and particle physical property analysis in multiphase flow. *Powder Technology* **2025**, *449*, 120366.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; pp. 770–778.
34. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 1492–1500.
35. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 2117–2125.
36. Xiu, J.; Li, Y.; Zhao, N.; Fang, H.; Wang, X.; Yao, A. Geometric Alignment and Prior Modulation for View-Guided Point Cloud Completion on Unseen Categories. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025; pp. 27435–27444.
37. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the Proceedings of the European conference on computer vision (ECCV), 2018; pp. 3–19.
38. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision* **2010**, *88*, 303–338.
39. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European conference on computer vision, 2014; pp. 740–755.
40. Cheng, B.; Girshick, R.; Dollár, P.; Berg, A.C.; Kirillov, A. Boundary IoU: Improving object-centric image segmentation evaluation. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021; pp. 15334–15342.
41. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical image computing and computer-assisted intervention, 2015; pp. 234–241.
42. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W. Hybrid task cascade for instance segmentation. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019; pp. 4974–4983.
43. Cheng, T.; Wang, X.; Huang, L.; Liu, W. Boundary-preserving mask r-cnn. In Proceedings of the European conference on computer vision, 2020; pp. 660–676.
44. Siswantoro, J.; Asmawati, E.; Siswantoro, M.Z. A rapid and accurate computer vision system for measuring the volume of axi-symmetric natural products based on cubic spline interpolation. *Journal of Food Engineering* **2022**, *333*, 111139.
45. Jia, W.; Li, B.; Xu, Q.; Chen, G.; Mao, Z.-H.; McCrory, M.A.; Baranowski, T.; Burke, L.E.; Lo, B.; Anderson, A.K. Image-based volume estimation for food in a bowl. *Journal of Food Engineering* **2024**, *372*, 111943.
46. Cheng, S.-T.; Lyu, Y.-J.; Teng, C. Image-Based Nutritional Advisory System: Employing Multimodal Deep Learning for Food Classification and Nutritional Analysis. *Applied Sciences* **2025**, *15*, 4911.
47. Shen, Y.; Choi, E.; Kleinberg, S. Predicting Postprandial Glycemic Responses With Limited Data in Type 1 and Type 2 Diabetes. *Journal of Diabetes Science and Technology* **2025**, 19322968251321508.