Proceedings of the 2nd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/42/20240673

LiqD: Dynamic Liquid Level Detection under Containers

Yukun Ma[1], Zikun Mao[1], Yexin Li[1], Xinyu Chen[1], Hang Zhou[1,2,*]

[1]Beijing Jiaotong University, School of Electronic and Information Engineering, Beijing, 100044, China

[2]hangzhou@bjtu.edu.cn
*corresponding author

Abstract. In daily life and industrial production, it is crucial to accurately detect changes in liquid level in containers. Traditional contact measurement methods have some limitations, while emerging non-contact image processing technology shows good application prospects. This paper proposes a container dynamic liquid level detection model based on U²-Net. This model uses the SAM model to generate an initial data set, and then evaluates and filters out high-quality pseudo-label images through the semi-supervised framework to build an exclusive data set. The model uses U²-Net to extract masking images of containers from the data set, and uses morphological processing to compensate for masking defects. Subsequently, the model calculates the grayscale difference between adjacent video frame images at the same position, segments the liquid level change area by setting a difference threshold, and finally uses a lightweight neural network to classify the liquid level state and achieves the accuracy of 91.39%. This approach not only mitigates the impact of intricate surroundings, but also reduces the demand for training data, showing strong robustness and versatility. A large number of experimental results show that the proposed model can effectively detect the dynamic liquid level changes of the liquid in the container, providing a novel and efficient solution for related fields.

Keywords: Detection, data augmentation, semi-supervised learning, image processing.

1. Introduction

Liquid level detection technology in containers plays a vital role in daily life. It not only prevents liquid overflow in home kitchens and ensures cooking safety, but also monitors the amount of liquid in storage tanks and reactors in the industrial field to ensure smooth and safe production processes. Also in construction, liquid level detection is used to monitor liquid level in tunnels and underground facilities to prevent flooding and structural damage. Scenarios like this are widely used.

To accurately monitor liquid levels, traditional contact measurement methods like float gauges and pressure transmitters [1,2] offer high measurement accuracy but have certain limitations. These methods require the measuring element to be directly immersed in the liquid, making them unsuitable for harsh environments involving highly corrosive substances, extreme temperatures, or high pressures.

In recent years, some non-contact remote measurement technologies have rapidly advanced, such as liquid level measurement systems based on radar and sonar principles [3,4]. These novel techniques eliminate the necessity for physical contact with the liquid being measured, offer a wide measurement

DOI: 10.54254/2753-8818/42/20240673

range, and highly adapt to different environments. However, they also face challenges like relatively high system costs and strict requirements on environmental conditions like temperature and pressure.

With continuous advancements in computer vision and image processing, image-based liquid level detection methods are increasingly emerging and attracting widespread attention in the field. Traditional image algorithms have proposed numerous liquid level detection methods using image shooting and processing to obtain liquid level conditions through spatial mathematical relationships [5]. These methods achieved convincing results over a decade ago. However, the application of deep learning in image processing has ushered in a new era. For liquid level detection in large scenes like lakes and reservoirs, substantial advancements have been achieved [6,7,8,9,10,11]. For example, Fang et al. [8] used YOLOv4 to accurately locate liquid gauge scale characters, then used DeepLabv3+ to precisely segment the junction area between the gauge and liquid body, and finally extracted liquid level and calculated actual values using image processing techniques. Sun et al. [6] achieved high-precision, real-time liquid level monitoring through steps like image pre-processing, edge detection, affine transformation correction, keyword positioning, and edge projection. Xia et al. [7] improved the superpixel and graph cutting algorithm, then performed liquid level detection based on the semantic segmentation network technology of U-net. Zhang et al. [10] proposed a liquid level height difference prediction method based on digital image processing by using a digital camera to capture a top view of the container, then performing image pre-processing, edge detection, and ellipse fitting to calculate the liquid level and distance from the container top. Bobovnik et al. [11] processed the images of the liquid surface and the measuring scale to detect the positions of the scale marks and the liquid level, correcting for lens distortion and parallax effects. Then, the machine vision system ultimately converts the liquid level position into a volume reading.

These methods have improved accuracy, generalization ability, and environmental adaptability but still face challenges and bottlenecks. Firstly, existing research mainly focuses on large liquid bodies, lacking relevant technology accumulation for tricky container scenarios. Secondly, most algorithms have high training data requirements, resulting in poor generalization capabilities when applied to different environments. Furthermore, complex environments introduce interference like lighting and occlusion, affecting detection accuracy, which means that mitigating the influence of environmental factors remains a critical challenge. Finally, for dynamically changing liquid level, accurate and stable detection is challenging due to factors like fluctuations, and existing methods lack modeling and analysis of dynamic processes. All these challenges await further breakthroughs and research.

Based on the above analysis, this study proposed a new visual processing method for dynamic liquid level changes in containers, greatly addressing issues of high sample requirements, complex environmental influences, and limited detection scene sizes. Our main contributions are threefold:

- We construct a dedicated dataset using the SAM model and evaluate it through the semi-supervised framework to obtain a standardized and specialized dataset.
- By employing U²-Net for salient object extraction, we obtain the container masking, focusing the analysis solely on the liquid surface within the container image. This greatly mitigates interference from external environments and shifts the detection emphasis toward subtle changes in small scale features within the image.
- We adopt image morphological compensation methods to significantly improve the quality of suboptimal masking, resulting in more distinct and smooth boundaries.

2. Related Works

2.1. SAM Model

Segment Anything Model (SAM) [12] represents an innovative deep learning architecture designed to efficiently segment arbitrary image content through a prompt-based segmentation task. As shown in Fig. 1, this model can generate precise segmentation masking in real time, without the need for specific task training, by utilizing flexible prompts such as points, bounding boxes, and text. SAM relies on a large-

DOI: 10.54254/2753-8818/42/20240673

scale dataset named SA-1B, which includes over 1.1 billion auto-generated masking, ensuring the model's generalization across diverse scenes. The zero- shot transfer learning capabilities of SAM have demonstrated remarkable performance across multiple downstream tasks, marking a significant breakthrough in the field of image segmentation.

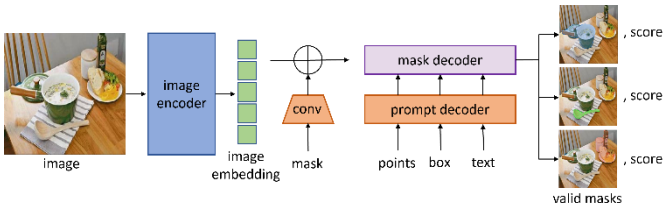


Figure 1. SAM architecture

It's noteworthy that SAM has learned a universal ability for object recognition and segmentation, thus its exceptional performance is not confined to specific object categories. Whether dealing with a single target or multiple targets of the same or different categories, SAM accurately segments them. This versatility positions SAM for a wide range of applications [13,14], such as interactive image editing, general object segmentation, and visual question answering, among others. Beyond segmentation quality, another major advantage of the SAM model is its computational efficiency. With no need for time-consuming task-specific fine-tuning, SAM can respond to user prompts in real time, rapidly producing segmentation outcomes, thereby facilitating downstream visual tasks and offering an excellent user interaction experience.

SAM's image segmentation capabilities and prompt adapt- ability guide the container masking creation, creating a foundational dataset for model training. Its scene generalization allows the use of various container types, broadening the method's scope. While SAM presents real-time interaction, this study uses it for data creation, not full liquid level detection. To improve dataset reliability, we also integrate SemiReward [15] for masking quality refinement.

2.2. U²-Net

U²-Net [16] architecture is a deep learning framework specifically tailored for salient object detection (SOD) tasks. Its core innovation lies in the unique nested U-shaped structure, which effectively captures rich contextual information at different scales. The architecture utilizes Residual U-blocks (RSUs) at each stage to extract multi-scale features while maintaining high-resolution feature maps. The clever design of the RSUs enhances the network's depth without significantly increasing computational costs, allowing U²-Net to be trained from scratch without relying on pre-trained image classification backbones. This design not only improves SOD performance but also computational efficiency, providing a novel and efficient solution for the SOD domain.

Unlike traditional methods that depend on pre-trained back- bones [17], U²-Net's ability to train from zero showcases performance comparable to or even better than the current state of the art. And the training loss L from [16] is defined as:

$$L = \sum_{m=1}^{M} w_{side}^{(m)} l_{side}^{(m)} + w_{fuse} l_{fuse} \tag{1}$$

where M is the number of side-output saliency maps, $w_{side}^{(m)}$ is the weight of the mth side-output loss, $l_{side}^{(m)}$ is the loss of the mth side-output saliency map, w_{fuse} is the weight of the fusion output loss, and l_{fuse} is the loss of the final fusion output saliency map. Each side-output loss $l_{side}^{(m)}$ is computed using the binary cross-entropy loss from U²-Net as shown below:

DOI: 10.54254/2753-8818/42/20240673

$$l = \sum_{(r,c)}^{(H,W)} \left[P_{G(r,c)} log P_{S(r,c)} + \left(1 - P_{G(r,c)}\right) log\left(1 - P_{S(r,c)}\right) \right] \quad (2)$$

where (r, c) is the pixel coordinates and (H, W) is image size of height and width. $P_{G(r,c)}$ and $P_{S(r,c)}$ denote the pixel values of the ground truth and the predicted saliency probability map, respectively. The training process tries to minimize the overall loss L. In the testing process, here choose the fusion output l_{fuse} as final saliency map.

U²-Net's hierarchical U-shaped architecture and RSUs in- form the approach, allowing LiqD to enhance container segmentation precision without increasing computational de- mands. Its train-from-zero approach enables to create models tailored for specific container data, deviating from U²-Net's general SOD focus. We've adapted U²-Net for container segmentation by adjusting training data, loss functions, and adding morphological processing to better suit liquid level detection tasks.

2.3. Bottleneck in Hand-crafted Design

2.3.1. Morphological Compensation

In the process of image analysis, defective images are commonly encountered. To address the issue like what this study encountered, Vizilter et al. [18] employed morphological image analysis to solve the problems of change detection and shape matching in images, which is similar to the idea of using morphological operations for image restoration as described by Raid et al. [19]. By adopting these methods, defects can be compensated for by filling holes and connecting broken regions in the image.

Firstly, a structuring element needs to be defined, which specifies the shape and size of the morphological operation. In this study, here choose to use an elliptical structuring element with a size of 5×5 pixels. The morphological closing operation, which consists of dilation followed by erosion, is then applied to the current binary image to fill small holes and connect broken regions. Based on this theory, the following equation can be derived:

$$A \oplus B = \left\{ x, y \middle| (B)_{xy} \cap A \neq \varnothing \right\} \quad (3)$$

where $(B)_{xy}$ denotes the translation of the structuring element B such that its origin is at (x, y). The output pixel (x, y) is set to 1 if the intersection of the translated B with the set A is non-empty, otherwise it is set to 0.

Erosion can shrink the target region, essentially causing the image boundaries to contract. It can be used to eliminate small, insignificant targets. The equation for erosion from [19] is expressed as:

$$A \ominus B = \left\{ x, y \middle| (B)_{xy} \subseteq A \right\} \quad (4)$$

where $(B)_{xy}$ denotes the translation of the structuring element

B such that its origin is at (x, y). The output pixel (x, y) is set to 1 if the translated B is completely contained within the set A, otherwise it is set to 0. This equation represents the erosion of A by the structuring element B.

2.3.2. Grayscale Value Conversion

Most of the images in this study are in color format, but the color information is not highly relevant. Therefore, it is crucial to introduce grayscale conversion to obtain meaningful numerical values. In terms of grayscale conversion methods, Saravanan [20] proposed a novel algorithm that addresses the contrast, sharpness, shadows, and structure of the image. This algorithm approximates, reduces, and adds to the chromaticity and luminosity of the RGB values. The formula from [20] is as follows:

DOI: 10.54254/2753-8818/42/20240673

$$Y = 0.299R + 0.587G + 0.114B$$
$$U = 0.565(B - Y)$$
$$V = 0.713(R - Y)$$
$$I_1 = \frac{\frac{R}{3} + \frac{G}{3} + \frac{B}{3} + U + V}{4} \quad (5)$$

where Y represents luminance, while U and V represent chrominance. The calculation of Y is based on the weighted sum of RGB components, while the calculation of U and V is based on the differences between red, green, blue, and luminance. The intensity value I_1 is computed by taking the average of the RGB components, adding the U and V components, and dividing the sum by 4.

Traditional grayscale image algorithms are not specifically tailored for classification purposes. In the context of image classification, Güneş, et al. [21] proposed a novel color-to- grayscale conversion method based on Genetic Algorithm (GA). By utilizing GA, the conversion coefficients for color images are optimized to generate grayscale images with enhanced discriminative features, aiming to reduce errors in image classification problems. The formula from is as follows:

$$r' = \frac{r}{r + g + b}$$
$$g' = \frac{g}{r + g + b}$$
$$b' = \frac{b}{r + g + b}$$
$$I_2 = r'R + g'G + b'B \quad (6)$$

Integrating the above two methods, the final intensity value I is obtained by adding I_1 and I_2 through the weighted proportional coefficients α and β using the equation:

$$I = \alpha \cdot I_1 + \beta \cdot I_2 \quad (7)$$

where α and β are weighting coefficients satisfying $\alpha+\beta = 1$. I_1 takes into account visual factors such as luminance, chrominance, and contrast, while I_2 emphasizes discriminative power for classification. The two methods are complementary to each other. By employing a weighted fusion approach, the visual quality can be enhanced while simultaneously taking classification performance into account.

3. Methodology

Based on the algorithm analysis mentioned above, this study proposes the whole architecture of LiqD as illustrated in Fig. 2. The algorithm consists of four core modules: Data engine construction, salient object extraction from the container, morphological compensation of the container shape, and calculation of the height difference in the container for liquid level detection.

3.1. Construct Data Engine

For the data engine approach, the core key is how to evaluate and filter labels and how to generate more label candidates of different qualities. SemiReward (SR) [15] has proposed an effective pseudo-label screening method for classification and regression tasks in the past. We modified this method to make it a method that can evaluate the generated masking. We use common Masking evaluation as the metric that can be learned allows the trained SR model to start evaluating and screening Masking. At the same time, data amplification is performed using methods such as noise addition and the most advanced mix-up [22] to ultimately seek the possibility of traversing Masking as much as possible. Through the data engine, we found that this is a very resource-saving method to achieve better training purposes. Combined with many of the most advanced methods, it greatly improves the sample quality during training.

DOI: 10.54254/2753-8818/42/20240673

3.2. Salient Target Extraction

Using the U²-Net based salient object extraction algorithm to focus on container images. Initially, the SAM was employed for image collection and processing, resulting in a substantial dataset of container images along with their corresponding masking images for subsequent analysis. These images, along with the masking, ccc were fed into the U²-Net for training, resulting in a salient object detection model specifically designed for extracting containers from common images.

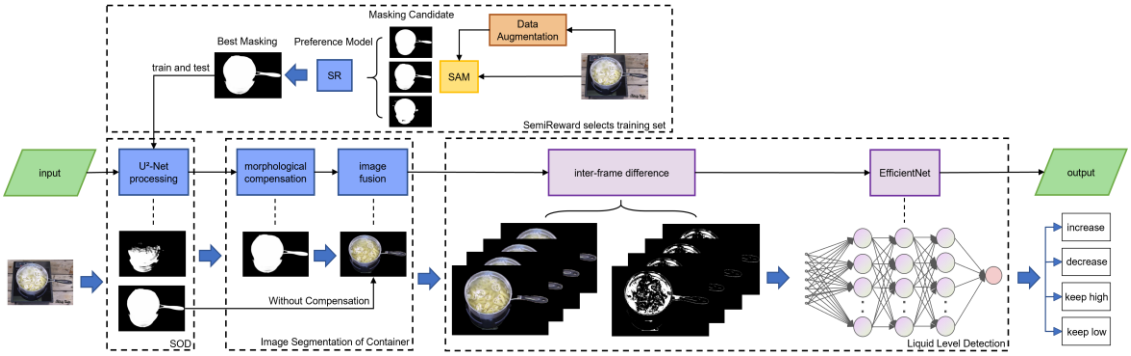


Figure 2. The architecture of LiqD. It consists of four core parts. **(1)** SemiReaward selects training set: SAM generates plenty of masking candidates after data augmentation and then SemiReward selects the high-quality masking to form the dataset; **(2)** SOD: After training from the specific dataset, use U²-Net to get the masking. The masking will include the complete and incomplete shape; **(3)** Image Segmentation Container: For the masking without complete shape, use morphological compensation to gain the complete shape. And then, fuse the original image with masking to obtain the image with only the container part; **(4)** Liquid Level Detection: Uniform binarized images are obtained by inter-frame difference of the obtained fusion images, and these images are sent to neural networks EfficientNet for training with four liquid level state labels.

3.3. Container Morphology Compensation

Following the application of the U²-Net model, certain images exhibited containers with colors closely resembling the surrounding environment, making them difficult to separate, which is shown in the left of Fig. 3. This resulted in discontinuity between adjacent segmented images. To address this issue, morphological operations were applied to the images to fill in the gaps and obtain complete images, ensuring a stable and continuous segmentation of the images, which is shown in right of Fig. 3. After being processed by the trained U²-Net salient object detection model, an image containing only the location in the image is obtained, and then fused with the original image to obtain an image containing only the container.

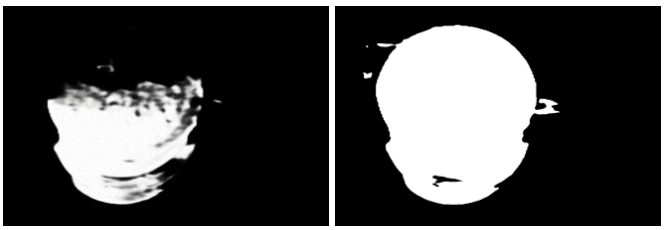


Figure 3. Contrast for the compensation

DOI: 10.54254/2753-8818/42/20240673

3.4. Dynamic Liquid Level Detection

In motion object detection using frame differencing, the goal is to detect the changing parts by eliminating the static regions and retaining the areas with variations in the difference image. Zhan et al. [23] divided the edge difference image into several small blocks and determined whether they were motion regions by comparing the number of non-zero pixels with a threshold. By applying this method, it is possible to extract information about the changing liquid levels within the container.

3.4.1. Threshold Division

According to the gray-level conversion method in this paper, the RGB values of each adjacent frame image are converted to grayscale values that are more conducive to image classification. Assuming the external environment and the container remain unchanged, the grayscale value changes at the corresponding positions between consecutive frames indicate the subtle dynamic changes in the liquid level inside the container. Then, a threshold is set for the magnitude of these value differences, and the optimal threshold is determined through experimentation. As shown in Fig 4, pixels with differences greater than the threshold are marked in white, while those below the threshold are marked in black. This process captures the minute changes in the liquid level in the container and assigns different labels to different liquid level states: rising liquid level, falling liquid level, low liquid level unchanged, and high liquid level unchanged. The labelled images are then input into a neural network for image classification.

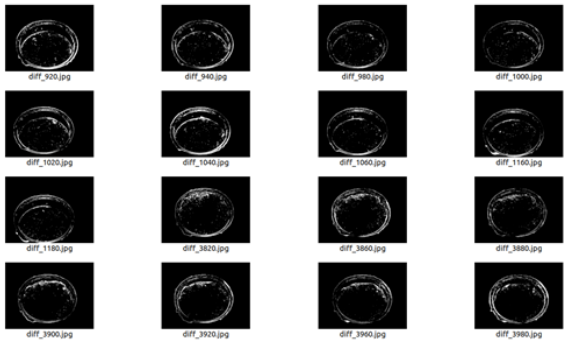
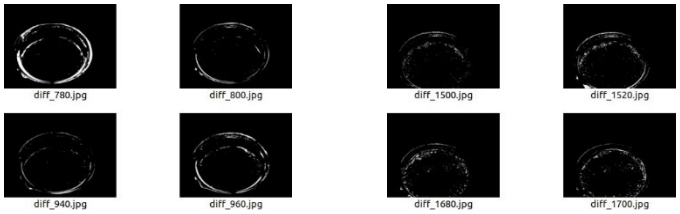


Figure 4. Threshold division

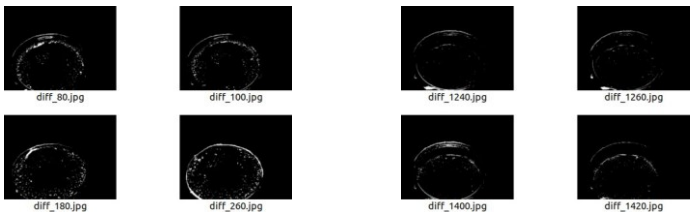
3.4.2. Liquid level detection

After processing, since the detected objects are all uniform binary images, there is no need to use a complex neural network for image classification. Therefore, this paper selects a lightweight neural network, EfficientNet-B0 [24], for training, testing, and validation. The images shown in Fig 5 and their corresponding labels are input into the neural network for image classification, thereby obtaining a neural network that can detect this specific task. By utilizing this neural network, this paper achieves the final detection of dynamic liquid level changes.



(a) increase b) decrease

DOI: 10.54254/2753-8818/42/20240673



(c) keep low (d) keep high

Figure 5. Four labels of liquid level

4. Experiment and Result Analysis

4.1. Test for SOD

Following the implementation of U²-Net for dynamic liquid level detection, here compared its performance with several well-known semantic segmentation models to benchmark its effectiveness. These models include U-Net [25], DeepLabV3+ [26], Mask R-CNN [27], F3Net [28], HRNet [29], and PSPNet [30]. The evaluation metrics [31] employed in the comparison were Accuracy (Acc), Precision (P), Recall (R), F1-score, Mean Absolute Error (MAE).

Table 1. SOD model comparison

Model	Acc	P	R	F1-score	MAE
U-Net	0.77	0.79	0.97	0.86	17.26
DeepLabV3+	0.72	0.75	0.95	0.82	22.95
Mask R-CNN	0.80	0.86	0.93	0.88	14.44
F3Net	0.62	0.66	0.89	0.7	30.24
HRNet	0.78	0.93	0.82	0.87	12.28
PSPNet	0.75	0.83	0.90	0.85	16.73
U²-Net	**0.90**	**0.95**	**0.94**	**0.94**	**5.52**

As indicated in Table 1, U²-Net shows superior performance compared to the other models evaluated. It achieves the highest accuracy at 0.90, significantly higher than the next best- performing model, Mask R-CNN, which has an accuracy of 0.80. U²-Net's precision and recall scores, 0.95 and 0.94 respectively, highlight its effectiveness in correctly classifying salient areas in the images.

The F1-score for U²-Net is 0.94, confirming its robustness and the effective balance it strikes between precision and recall. In terms of error metrics, U²-Net records the lowest values with a mean absolute error of 5.52, emphasizing its precision and reliability in predicting liquid level changes.

These comparative results underscore the potential of U²- Net for practical deployment in scenarios where accurate liquid level detection is paramount, such as in industrial control systems. The evaluation suggests that U²-Net could serve as a reliable model for similar segmentation tasks that demand high accuracy and consistency.

4.2. Test for morphological compensation

To evaluate the impact of morphological compensation on the performance of the liquid level detection processing, here conducted a set of ablation experiments. In these experiments, here used the same dataset and evaluation metrics as in the main experiments, with the only difference being the removal of the morphological processing step after the masking generation. Here compared the masking quality and liquid level detection performance with and without morphological compensation, and the results are shown in Table 2.

DOI: 10.54254/2753-8818/42/20240673

Table 2. Morphological compensation evaluation

Evaluation metric	With compensation	Without compensation
Masking IoU	0.95	0.76
Masking smoothness*	0.92	0.69
Extraction accuracy	0.93	0.58

*Masking Smoothness is measured using the edge smoothness metric, which calculates the variance of the curvature of the masking edges. Higher values indicate smoother edges.

The experimental results in Table 2 demonstrate that using morphological compensation can significantly improve the quality of making and the performance of liquid level detection. Morphological compensation increases the Masking IoU and Masking Smoothness scores for the masking. Regarding liquid level detection performance, morphological compensation enhances the accuracy of liquid level change region extraction.

The ablation experiments prove the necessity and effective- ness of morphological compensation in the proposed liquid level detection algorithm. Without morphological compensation, the masking generated by U²-Net suffers from defects such as discontinuities and jagged edges, resulting in lower masking quality scores and interfering with the subsequent liquid level change region extraction. However, by applying morphological compensation as a post-processing strategy, these defects are effectively repaired, and the masking be- comes more complete and smoother. This improves the integrity and smoothness of the masking generated by U²- Net, compensating for some shortcomings of deep learning models in semantic segmentation tasks, and provides a more reliable foundation for the subsequent liquid level change region extraction, significantly enhancing the performance of liquid level detection.

4.3. Test for threshold

The threshold range is between 0 and 255. However, according to the research and general principles of threshold proposed by Robertson et al. [32], the optimal threshold range should not exceed half of the maximum range. To comprehensively test the thresholds, a preliminary threshold ranges from 0 to 127 was set, divided into intervals of length 8. The initial segmentation results are shown in Fig. 6.

As the threshold increases beyond 80, the binary information in the image significantly decreases, and the degree of variation between the images decreases greatly. The distinct characteristics among the four types of labels gradually become blurred, and the similarity between the images increases. When the threshold is further increased to 120, the images become predominantly black, and the four types of label information almost have the same black features. If training is forcibly conducted at this stage, it would be equivalent to performing object detection on images with the same features after being segmented into four labels, leading to a phenomenon where the results are extremely close to being all correct, which is an inappropriate behaviour. When the threshold is set too low, the resulting image after differentiation will capture numerous minor variations. These variations may be influenced by atmospheric haze or uneven flickering of lights, which can lead to erroneous detection of changes in areas where no actual changes occurred during the camera capture process. Therefore, setting an excessively small threshold is also unreasonable.

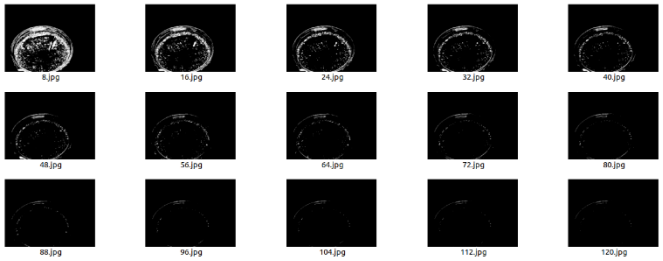


Figure 6. Change for threshold

DOI: 10.54254/2753-8818/42/20240673

Based on the above considerations, values below 10 or above 80 for the threshold are not within the range of consideration. By comparing the accuracy of the threshold, the model can select the optimal threshold, and the resulting difference images using various threshold values within this range shown in Fig. 7.

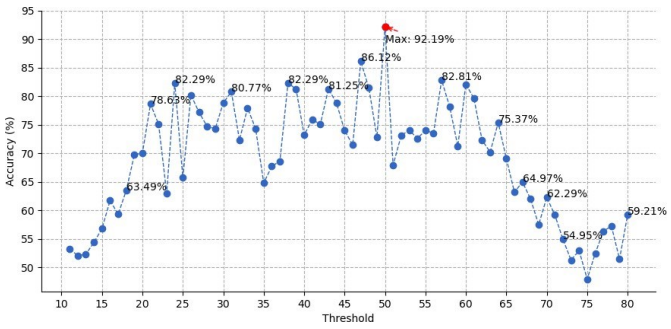


Figure 7. Threshold result comparison

The data presented in Fig 7 illustrates the comparative results of the threshold experiments. The graph shows the accuracy of detecting neighbouring differences for each tested threshold value. As the threshold changes from 11 to 80, the accuracy reaches a peak of 92.19% at a threshold value of 50.

These results suggest that setting the threshold too low may lead to the detection of noise, while setting it too high may cause important differences to be missed. The optimal thresh old value of 50 achieved the best balance between detecting meaningful differences and filtering out noise, resulting in the highest accuracy. The experiment demonstrates the importance of carefully selecting an appropriate threshold to maximize performance, and the results provide a valuable reference for similar tasks. By setting the threshold to 50, we were able to clearly distinguish significant differences, enabling more accurate and reliable analysis in this liquid level detection system.

4.4. Test for overall process

According to the process in Fig 8, following the steps of salient target extraction, partial incomplete image compensation, original image and incomplete image fusion, inter-frame difference and liquid level classification, 18 images with a total duration of 31 minutes and 30 frames per second were processed.

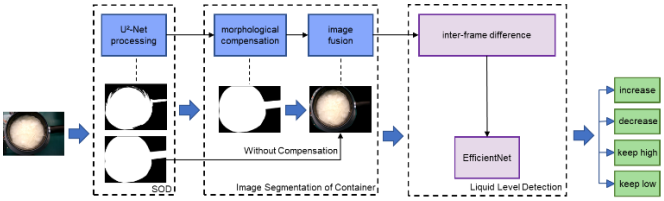


Figure 8. Change for threshold

In this detection, video frames are extracted for detection at 3 frames per second, that is, one frame is extracted from every 10 frames in the video for subsequent processing. After U²-Net extracts salient targets, the morphology of the image is used. Make up to ensure the integrity of the images of adjacent frames, then perform image fusion and gray value conversion, perform inter-frame difference according to the threshold of 50, and finally send these differentiated pictures to a large number of trained EfficientNet networks for processing Liquid level status classification was completed to complete the final test detection. The liquid level image classification accuracy obtained through the EfficientNet network reached 91.39%.

DOI: 10.54254/2753-8818/42/20240673

5. Conclusion

In this study, we developed a novel approach for liquid level state detection by combining image differencing and binarization techniques. This model demonstrated strong robustness against variations in container types and environ mental conditions. By simplifying input images into binary representations focusing on the target object, here achieved accurate classification using a straightforward neural network architecture, without the need for complex network designs.

A key advantage of this model is its reduced reliance on large training datasets, made possible by leveraging the SemiReward framework to generate high-quality pseudo labelled images using the SAM model. The resulting dedicated dataset enabled efficient training and generalization of this model.

Integrating image differencing and object-focused binarization simplifies complex visual information into manageable representations. By focusing on essential features of the target object, this model effectively captures and analyzes relevant changes while being resilient to background variations.

However, this method also has some limitations. A major drawback is the inability to take into account the influence of environmental factors such as aerosol. In some application scenarios, aerosol may be generated in the container, obscuring the liquid level and interfering with image analysis. It is difficult to eliminate the influence of aerosol by relying on computer vision alone, and auxiliary means such as external blowing may be required. Future work could incorporate dehazing algorithms to overcome this limitation.

It is the liquid level detection model that offers a robust, efficient, and generalizable approach for analyzing small changes in static object environments. By simplifying complex images and leveraging high-quality pseudo-labelled data, here have demonstrated the potential for solving a wide range of similar problems with reduced data requirements and computational complexity.

References

[1] Yadvendra Singh, Kumar Sanjeev Raghuwanshi, and Soubir Kumar. Review on liquid-level measurement and level transmit- ter using conventional and optical techniques. IETE Technical Review, 2018.

[2] Georgi Nikolov and Boyanka Nikolova. Virtual techniques for liquid level monitoring using differential pressure sensors. Recent, 9(2):49, 2008.

[3] Pankaj Mohindru. Development of liquid level measurement technology: A review. Flow Measurement and Instrumentation, 89:102295, 2023.

[4] Kam W Lo and Brian G Ferguson. Automatic detection and tracking of a small surface watercraft in shallow water using a high-frequency active sonar. IEEE Transactions on Aerospace and Electronic Systems, 40(4):1377–1388, 2004.

[5] Ti-Ho Wang, Ming-Chih Lu, Chen-Chien Hsu, Cheng-Chuan Chen, and Jia-Dong Tan. Liquid-level measurement using a single digital camera. Measurement, 42(4):604–610, 2009.

[6] Weiya Sun, Da Wang, Shuai Xu, Jingye Wang, and Zhanyu Ma. Water level detection algorithm based on computer vision. Journal of Applied Sciences, 40(03):434–447, 2022.

[7] Ping Xia, Feng Wang, Bangjun Lei, and Dongxia Shi. Intelligent visual water level recognition based on super pixel and graph cut algorithm. computer simulation, 38(03):430–436+441, 2021.

[8] Aiyin Fang, Yongxian Wang, Ximeng Yin, Peng Wang, Zhongyi Li, and Zhi Liu. Detsegnet: A high-precision water level detection network based on detection and segmentation. In Proceedings of the 2023 (11th) China Symposium on Water Conservancy Information Technology, pages 56–73, 2023.

[9] Yan-Ting Lin, Yi-Chun Lin, and Jen-Yu Han. Automatic water- level detection using single-camera images with varied poses. Measurement, 127:167–174, 2018.

[10] Xiaoyuan Zhang, Tangyou Liu, and Futing Yu. Prediction of height difference between liquid level and container top based on ellipse detection. Computer and Information Technology, 30(03):19–23, 2022.

DOI: 10.54254/2753-8818/42/20240673

[11] Gregor Bobovnik, Tim Musˇicˇ, and Jozˇe Kutin. Liquid level detection in standard capacity measures with machine vision. Sensors, 21(8):2676, 2021.
[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, C Alexander Berg, Wan-Yen Lo, et al. Segment anything. arXiv preprint arXiv:2304.02643, 2023.
[13] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In A. Oh, T. Neumann, A. Globerson, K. Saenko,M. Hardt, and S. Levine, editors, Advances in Neural Infor- mation Processing Systems, volume 36, pages 29914–29934. Curran Associates, Inc., 2023.
[14] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. Medical Image Analysis, 89:102918, 2023.
[15] Siyuan Li, Weiyang Jin, Zedong Wang, Fang Wu, Zicheng Liu, Cheng Tan, and Stan Z. Li. Semireward: A general reward model for semi-supervised learning. In The Twelfth International Conference on Learning Representations, 2024.
[16] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood De- hghan, R Osmar Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. Pattern recognition, 106:107404, 2020.
[17] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. Computational visual media, 5:117–150, 2019.
[18] V Yuriy Vizilter, P Yuriy Pyt'ev, I Alexey Chulichkov, and M Leonid Mestetskiy. Morphological image analysis for com- puter vision applications. Computer Vision in Control Systems- 1: Mathematical Theory, pages 9–58, 2015.
[19] AM Raid, WM Khedr, MA El-Dosuky, and Aoud Mona. Image restoration based on morphological operations. International Journal of Computer Science, Engineering and Information Technology, 4(3):9–21, 2014.
[20] C Saravanan. Color image to grayscale image conversion. In 2010 Second International Conference on Computer Engineer- ing and Applications, volume 2, pages 196–199. IEEE, 2010.
[21] Ali Güneş,, Habil Kalkan, and Efkan Durmus¸. Optimizing the color-to-grayscale conversion for image classification. Signal, Image and Video Processing, 10:853–860, 2016.
[22] Zicheng Liu, Siyuan Li, Di Wu, Zihan Liu, Zhiyuan Chen, Lirong Wu, and Stan Z. Li. Unveiling the power of mixup for stronger classifiers. In European Conference on Computer Vision, 2022.
[23] Chaohui Zhan, Xiaohui Duan, Shuoyu Xu, Zheng Song, and Min Luo. An improved moving object detection algorithm based on frame difference and edge detection. In Fourth international conference on image and graphics (ICIG 2007), pages 519–523. IEEE, 2007.
[24] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR, 2019.
[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention– MICCAI 2015: 18th international conference, Munich, Ger- many, October 5-9, 2015, proceedings, part III 18, pages 234–241. Springer, 2015.
[26] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV), pages 801–818, 2018.
[27] Kaiming He, Georgia Gkioxari, Piotr Dolla´r, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international confer- ence on computer vision, pages 2961–2969, 2017.
[28] Jun Wei, Shuhui Wang, and Qingming Huang. F3net: fusion, feedback and focus for salient object detection. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 12321–12328, 2020.

DOI: 10.54254/2753-8818/42/20240673

[29] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high- resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5693–5703, 2019.
[30] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In Proceed- ings of the IEEE conference on computer vision and pattern recognition, pages 2881–2890, 2017.
[31] Jianlong Zhou, Amir H Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explana- tions: A survey on methods and metrics. Electronics, 10(5):593, 2021.
[32] Stephen Robertson and Stephen Walker. Threshold setting in adaptive filtering. Journal of documentation, 56(3):312–331, 2000.