electronics

MDPI

Article

Detection of Liquid Retention on Pipette Tips in High-Throughput Liquid Handling Workstations Based on Improved YOLOv8 Algorithm with Attention Mechanism

Yanpu Yin [1], Jiahui Lei [2] and Wei Tao [2,*]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; yanpu.yin@alumni.sjtu.edu.cn
[2] School of Sensing Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; makuragami@sjtu.edu.cn
* Correspondence: taowei@sjtu.edu.cn

Abstract: High-throughput liquid handling workstations are required to process large numbers of test samples in the fields of life sciences and medicine. Liquid retention and droplets hanging in the pipette tips can lead to cross-contamination of samples and reagents and inaccurate experimental results. Traditional methods for detecting liquid retention have low precision and poor real-time performance. This paper proposes an improved YOLOv8 (You Only Look Once version 8) object detection algorithm to address the challenges posed by different liquid sizes and colors, complex situation of test tube racks and multiple samples in the background, and poor global image structure understanding in pipette tip liquid retention detection. A global context (GC) attention mechanism module is introduced into the backbone network and the cross-stage partial feature fusion (C2f) module to better focus on target features. To enhance the ability to effectively combine and process different types of data inputs and background information, a Large Kernel Selection (LKS) module is also introduced into the backbone network. Additionally, the neck network is redesigned to incorporate the Simple Attention (SimAM) mechanism module, generating attention weights and improving overall performance. We evaluated the algorithm using a self-built dataset of pipette tips. Compared to the original YOLOv8 model, the improved algorithm increased mAP@0.5 (mean average precision), F1 score, and precision by 1.7%, 2%, and 1.7%, respectively. The improved YOLOv8 algorithm can enhance the detection capability of liquid-retaining pipette tips, and prevent cross-contamination from affecting the results of sample solution experiments. It provides a detection basis for subsequent automatic processing of solution for liquid retention.

Keywords: object detection; handling workstation; liquid retention detection; YOLOv8; attention mechanism

Citation: Yin, Y.; Lei, J.; Tao, W. Detection of Liquid Retention on Pipette Tips in High-Throughput Liquid Handling Workstations Based on Improved YOLOv8 Algorithm with Attention Mechanism. *Electronics* **2024**, *13*, 2836. https://doi.org/10.3390/electronics13142836

Academic Editors: Haibin Wu, Aili Wang and Yuji Iwahori

Received: 18 June 2024
Revised: 13 July 2024
Accepted: 17 July 2024
Published: 18 July 2024

1. Introduction

The discovery of the double helix structure of DNA has greatly promoted the development of molecular biology and spawned new disciplines such as molecular genetics and diagnostics. In particular, molecular diagnosis can detect and treat diseases earlier and more accurately. In vitro diagnostic technology has become an indispensable part of the medical and biological fields, and molecular diagnostics have become an important research direction. The steps of molecular diagnostics generally include sample preprocessing, nucleic acid extraction and purification, amplification, and detection. Each step involves liquid extraction, distribution, mixing, sample transfer and handling, making the entire process very complex and tedious. With the outbreak of COVID-19, a large number of samples need to be tested. The complexity of molecular diagnostics, high technical requirements for personnel, and the need to handle a large number of samples makes traditional manual extraction and detection methods slow, with limited sample, throughput

and potential human error, unable to meet current demands. Therefore, high-precision, high-throughput liquid handling workstations have become a very urgent need.

In sample processing, there is a desire to handle a large number of samples in one go while eliminating errors caused by manual extraction. Thus, liquid handling workstations are widely used in disease diagnosis, virus detection, biopharmaceuticals, chemistry, and other fields. It can save labor costs and time, improve the accuracy of experimental results, avoid human error, and achieve efficient operations. For example, Langer et al. [1] rapidly produced and recovered cell spheroids using a high-throughput liquid workstation; Coppola et al. [2] established an automated process for isolating and purifying peripheral blood mononuclear cells using a liquid handling workstation; Annona et al. [3] used a high-throughput liquid handling workstation to evaluate the accuracy of molecular biology experimental procedures.

In liquid handling operations, pipette tips, as the core component of liquid handling workstations, directly affect the accuracy and efficiency of experiments. During pipetting, differences in sample solution concentration, hydrophobicity of tips from different manufacturers, aspiration and dispense speeds, and pipetting volumes can lead to liquid retention (droplet hanging) on the tips. This causes liquid residue on the outside or top of the tips, resulting in incomplete sample transfer and cross-contamination, thus affecting the accuracy of experimental results. Traditional methods for detecting liquid retention often rely on visual inspection or simple sensor technologies, which have low detection accuracy, poor real-time performance, and difficulty handling complex situations.

In order to improve the accuracy and efficiency of liquid retention detection, a method of applying object detection technology to liquid handling workstations is proposed. By integrating object detection technology, high-resolution cameras need to be deployed in the pipette tip area, ensuring sufficient lighting inside the workstation to capture images of the pipetting process in real time. Then, a pre-trained model is used to analyze the images, automatically identifying whether there is liquid retention on the pipette tips. The object detection model can accurately determine whether liquid retention has occurred by analyzing the contours, colors, and morphological features of the liquid in the images. This process can not only be performed after pipetting is completed but also enables real-time monitoring during the operation, promptly detecting and correcting liquid retention issues to avoid affecting subsequent experimental steps. Applying object detection technology to high-throughput liquid handling workstations can significantly improve the accuracy and efficiency of sample solution extraction and detection. It also enhances the level of automation and data management capabilities in the experimental process, promoting the development of laboratory automation in a more efficient and intelligent direction.

2. Related Work

With the continuous development of deep learning, computer vision tasks have been increasingly refined, and object detection has become one of the main tasks in the field of computer vision. Its purpose is to identify and locate objects in images and videos. Due to issues such as different poses, features, brightness levels, sizes, light intensity interference, occlusion, and noise, object detection remains a significant and challenging task in computer vision. Object detection has been applied across various fields, including building surveillance and fire protection systems, autonomous driving, intelligent logistics, and medical image processing. For instance, Gautam et al. [4] used object detection to analyze videos in intelligent building surveillance systems, addressing the problem of personnel identification; Xie et al. [5] applied object detection for target tracking in logistics warehouses, distinguishing between people and goods, and established a target tracking evaluation system; Meda et al. [6] used object detection methods to identify rickets and normal wrists in pediatric wrist X-rays.

Early object detection relied heavily on manually designed feature extraction and classifiers [7]. This involved using a sliding window to scan every pixel block in the image, extracting features and classifying each window. Albadawi et al. [8] utilized the Histogram

of Oriented Gradients face detector to extract facial landmarks. HOG (Histogram of Oriented Gradients) is one of the commonly used image feature descriptors, calculating the gradient in horizontal and vertical directions to extract feature information from the image, while also handling geometric and optical transformations of the image. Hütten et al. [9] proposed using deep learning methods to enhance deformable part models (DPM) for object detection tasks, the paper explores combining DPM with convolutional neural networks (CNNs) to improve the accuracy and robustness of detecting complex deformable objects. DPM also improves HOG features by decomposing the target object into multiple parts, giving each part specific shape and positional relationships, thus showing strong robustness to object deformation.

Object detection driven by deep learning has become the current mainstream method. The object detection algorithm based on deep learning is divided into one-stage object detection algorithms and two-stage object detection algorithms [10,11], both algorithms use convolutional neural networks (CNNs) in their architectures. CNNs have strong feature representation capabilities and better robustness, making them a typical model architecture in current object detection. Two-stage object detection algorithms generate regions of interest for predicted objects and then classify and predict their positions, whereas one-stage algorithms directly extract features from the network to classify and predict object positions. Two-stage object detection algorithms are slower but more accurate. He et al. [12] proposed the Mask R-CNN object detection and segmentation algorithm, their approach efficiently detects objects while simultaneously generating a high-quality segmentation mask for each instance.

One-stage object detection algorithms, characterized by fast processing speed, lightweight model architecture, and ease of deployment, have been widely applied in industry and various fields. The algorithms for one-stage object detection are mainly YOLO (You only look once) series and SSD series. The SSD [13] algorithm simultaneously predicts the locations and categories of multiple objects in a single forward pass through the neural network, using multiple convolutional layers to predict bounding boxes of different scales and aspect ratios. The YOLO series object detection algorithms have been widely used and rapidly developed, becoming typical algorithms for one-stage object detection. Jiang et al. [14] briefly describes the development process of the YOLO algorithm and summarizes the methods of target recognition and feature selection. The YOLO object detection algorithm takes the entire image as input and directly regresses the bounding box locations and categories at the output layer. Bochkovskiy et al. [15] updated the YOLO architecture in 2020 to YOLOv4, adding ResNet (residual networks) [16] and FPN (feature pyramid networks) [17] for feature fusion. In 2022, the YOLOv7 [18] algorithm improved both speed and accuracy. In 2024, Wang et al. [19] introduced YOLOv9, incorporating the concept of Programmable Gradient Information (PGI) to handle the various transformations required for detecting multiple objects with deep networks. These developments demonstrate that the YOLO series algorithms have become the current mainstream object detection algorithms.

This study aims to detect liquid retention on pipette tips in high-throughput liquid handling workstations, identifying whether the tips have retained liquid to avoid inaccuracies in pipetting and cross-contamination in experiments. The liquid retained on the external and top parts of the tips poses detection challenges due to the small and variable volumes, different concentrations and colors of solutions (e.g., transparent liquids), and the varied colors of the pipette tips (e.g., black and transparent tips). Cao et al. [20] proposed a multi-scaled deformable convolutional object detection network to address the challenges of detecting small, dense objects and those with random geometric transformations. Additionally, the high-throughput nature means numerous tips with narrow gaps between them, significantly increasing the complexity and difficulty of detecting liquid retention on pipette tips. In the field of biomedical engineering, liquid detection primarily focuses on large-volume targets. For instance, Hwang et al. [21] used deep learning to monitor residual liquid in intravenous drug administration, reducing injection-related accidents and improving patient safety in hospitals. However, detecting micro-volume liquids presents

higher challenges compared to large-volume liquid detection, and research in this area is insufficient, further research in this direction needs to be improved and enhanced.

3. Materials and Methods

3.1. Yolov8 Algorithm

YOLOv8 [22] is a representative algorithm in object detection and is widely used in the fields of object detection and image segmentation. The YOLOv8 network architecture primarily consists of three parts: the backbone, the neck, and the head. The backbone is composed of multiple repeated convolutional and residual modules, downsampling the input image. Meanwhile, it introduces the C2f module to enhance image feature extraction. The backbone ends with three max-pooling layers to extract and fuse features. The image features are extracted by the backbone and used for analysis and processing in subsequent network modules. The neck utilizes the FPN-PAN feature fusion method, combining feature maps of different sizes through upsampling and downsampling, effectively integrating extracted features. It adopts the decoupled-head structure in head models to compute losses separately for regression and categories. The YOLOv8 network architecture is shown in Figure 1.

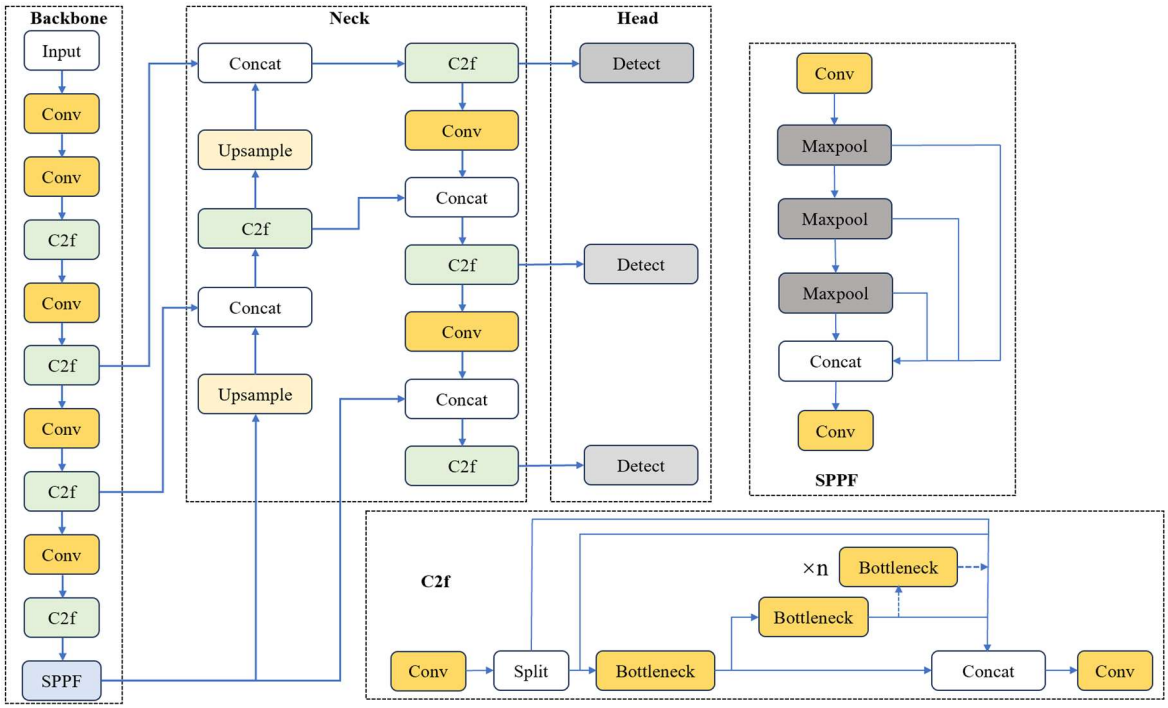

Figure 1. YOLOv8 network architecture.

3.2. Improvement Model

The detection of liquid retention on pipette tips in high-throughput liquid handling workstations faces the following challenges:

(1) The liquid retained on the external and top parts of the pipette tips is small in volume and variable in size.
(2) The solutions have different concentrations and colors, such as transparent liquids, and the pipette tips also come in different colors, such as black and transparent tips.
(3) The high throughput nature results in numerous pipette tips with narrow gaps between them, potentially causing occlusion issues.

These challenges significantly increase the difficulty [20] of object detection and the likelihood of missed or false detections. To mitigate these issues, this paper proposes an improved YOLOv8 network architecture. We selected the lightweight YOLOv8n as

the baseline network architecture and made improvements based on it. The improved network architecture is shown in Figure 2. The model incorporates the attention mechanism modules GCNet [23] and LSK [24] into the backbone, as well as the GCNet module into the C2f module. This enhances the network's ability to capture and understand global context information. The LSK module dynamically adjusts the receptive field during feature extraction, enhancing the network's ability to understand different background information and improving the model's ability to adapt to backgrounds of different sizes. In the neck network, the attention mechanism SimAM [25] module is added between the feature fusion and C2f_GC during the feature fusion process. This module designs an energy function to calculate attention weights, allowing better focus on the primary target.

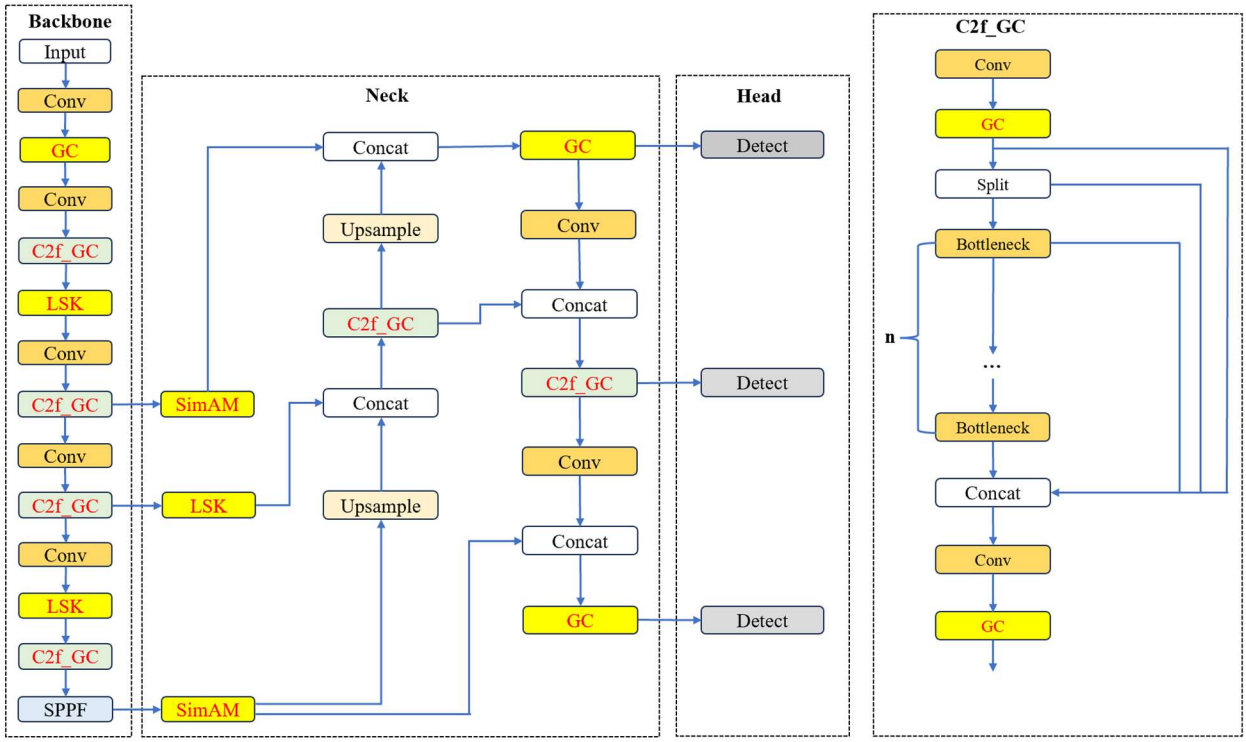

Figure 2. Improved YOLOv8 network structure.

3.2.1. Introducing the Attention Mechanism GCNet Module

GCNet combines the Non-local network [26] and SE [27] network structures, simplifying and updating based on them. In the non-local network, the contextual information captured for different query positions in the image is consistent, so GCNet creates a query-independent network structure, reducing parameters and computational load. It integrates with the SE module to form a multi-head attention mechanism, making GCNet a lightweight attention mechanism module. When capturing contextual information, the GCNet module disregards the query position, allowing all query positions to share one attention map and removing the query convolution operation $W_q x_i$, simplifying the non-local network module. The specific simplified calculation is shown in Equation (1). Here, "i" is the position to be calculated in the input feature, "j" is the index of all possible related positions of x_i, N_p is the total number of pixels in the feature map, and W_k and W_v represent linear transformation matrices.

$$z_i = x_i + \sum_{j=1}^{N_p} \frac{exp(W_k x_j)}{\sum_{m=1}^{N_p} exp(W_k x_m)} (W_v * x_j) \quad (1)$$

To further reduce computation and parameters, W_v convolution is moved before the attention. The GCNet module can be summarized in three processes: First, the input image

or feature map undergoes 1×1 convolution W_k and the Softmax function to calculate attention weights, followed by global average pooling to obtain global contextual feature vectors. Next, 1×1 convolution W_v and the ReLU activation function are used for feature transformation to obtain new global contextual features. Finally, the new global contextual features are fused into the features of each position through weighted fusion. To optimize training parameters, the 1×1 convolution in the feature transformation part is replaced by a bottleneck transform module, reducing the parameters from C·C to 2·C·C/r. We can clearly see the process of the GCNet module in Table 1. In the original YOLOv8 network structure, the C2f module divides the feature map into two parts along the first dimension, improving the model's non-linear representation capabilities. The C2f module consists of multiple bottleneck blocks, each block contains two convolutional layers. First, the input feature map undergoes preliminary transformation through the first convolutional layer (cv1). The output feature map is divided into two parts, each processed by different convolutional layers. Subsequently, these two parts are merged and processed by the second convolutional layer (cv2), resulting in an enhanced feature map.

Table 1. GCNet module process.

Process	Description
Attention Weights Calculation	1. The input image or feature map undergoes 1×1 convolution W_k. 2. The Softmax function is applied to calculate attention weights. 3. Global average pooling is used to obtain global contextual feature vectors.
Feature Transformation	1. 1×1 convolution W_v and the ReLU activation function to obtain new global contextual features. 2. To optimize training parameters, the 1×1 convolution is replaced by a bottleneck transform module, reducing the parameters from C·C to 2·C·C/r.
Feature Fusion	The new global contextual features are fused into the features of each position through weighted fusion.

The improved YOLOv8 network structure incorporates the GCNet attention mechanism module in the backbone and C2f modules. After the input feature map passes through the first convolutional layer (cv1), a GCNet module is added to capture the global contextual information of the input feature map. The feature map is then processed by different convolutional layers, and after merging the feature maps, another GCNet module is added post the second convolutional layer (cv2). This enables the network to capture long-range dependencies, and the integration of global contextual information, it improves the network's robustness to various interfering factors. The improved C2f_GC network structure is shown in Figure 3. In the C2f_GC module, if the input tensor x is (h,w,c), where c is the number of channels, h is the height of image, w is the width of image. After one convolution operation, then it passes through GCNet module, if the attention mechanism is used, the convolution module will be used to perform an operation to change the number of channels c to 1, generating an attention map. Then, a reshape operation is performed to change the size of tensor to (1,h $\times$ w), followed by Softmax function to calculate attention weights, transforming it to (1,h $\times$ w,1). Then, multiplicative fusion with the input tensor is performed by matrix multiplication, and the final output is (c,1,1). If the additive fusion is used, the output will be (c,h,w). The attention mechanism fusion in the C2f_GC module adopts multiplicative fusion.

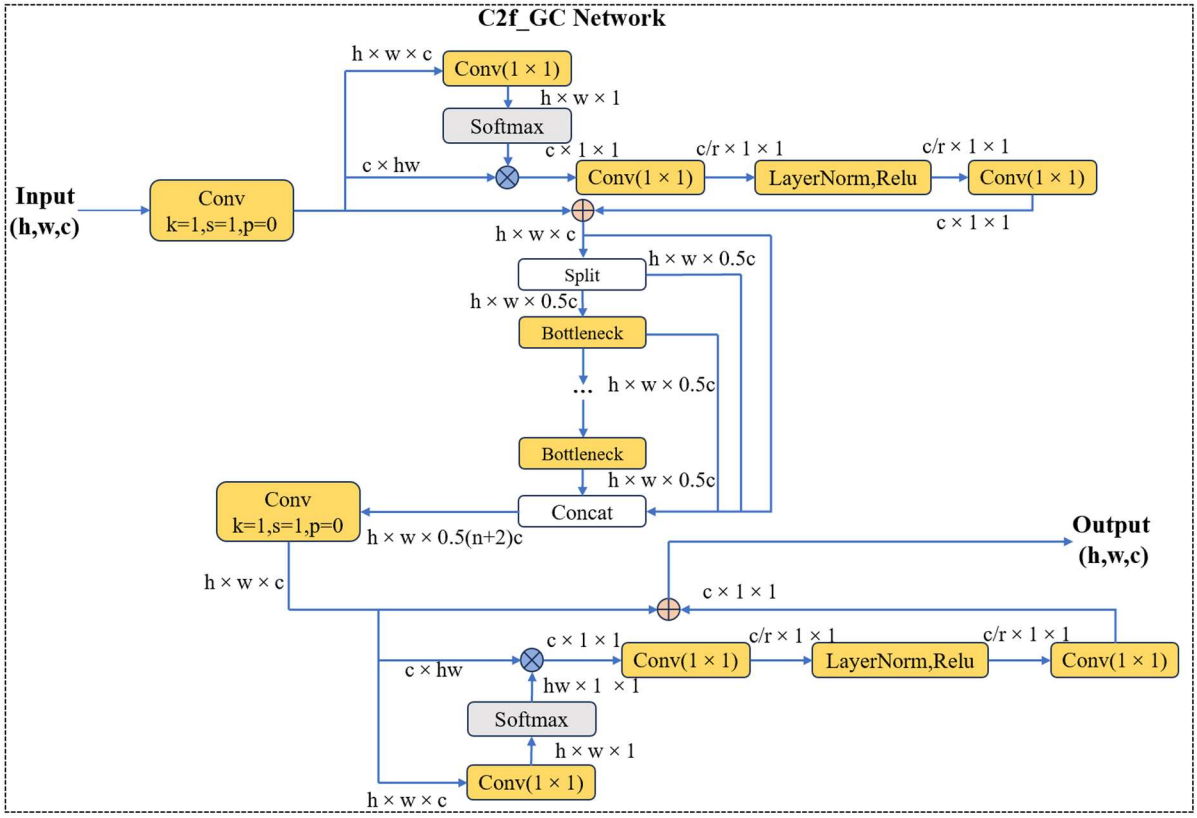

Figure 3. Improved C2f_GC network structure.

3.2.2. Introducing Attention Mechanism LSK Module Fusion

LKSNet is used for object detection in remote sensing images. To address the need for different backgrounds for various objects and improve the recognition capability of background information, LKSNet dynamically adjusts the receptive field of the extracted features, handling the differences in background required for different objects. LKSNet includes two residual sub-blocks, Large Kernel Selection (LK Selection) and Feed-forward Network (FFN). LK Selection dynamically adjusts the network's receptive field, and FFN is used for channel and feature fusion. The LKS module comprises a large kernel convolution and a spatial kernel selection mechanism, as shown in Figure 4.

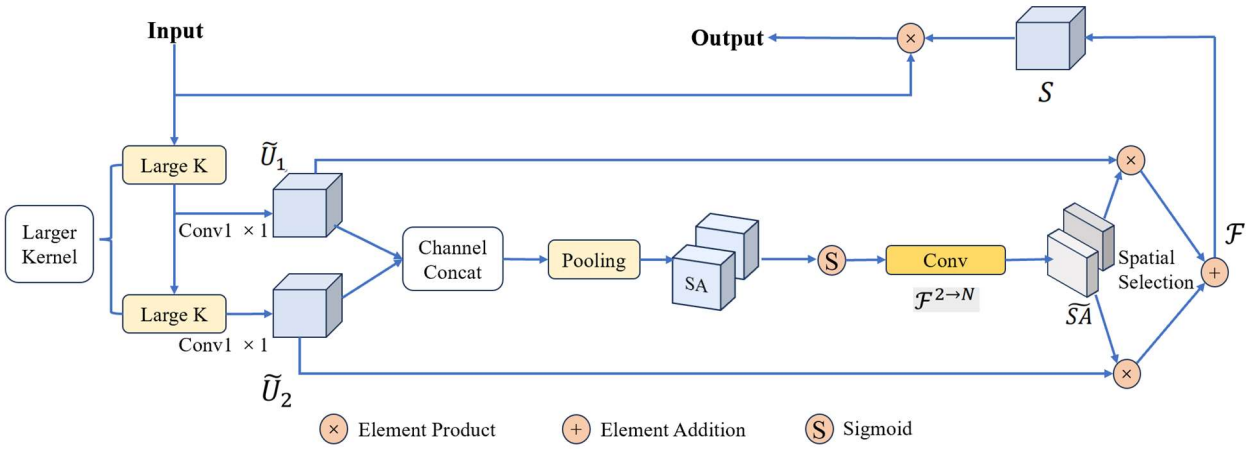

Figure 4. LSK module network architecture.

Since different targets have different background information, the model needs to automatically select the appropriate background range size. Large kernel convolutions con-

tinuously change the Depth-wise convolution, thereby continuously adjusting the receptive field. Depth-wise convolution satisfies the relationship as shown in Equations (2) and (3), where k is the size of the i-th depth-wise convolution kernel, d is the dilation rate, RF is the receptive field. The change in the convolution kernel and the increase in the dilation rate will obviously increase the receptive field.

$$k_{i-1} \leq k_i, d_i = 1, d_{i-1} < d_i < RF_{i-1} \tag{2}$$

$$RF_1 = k_1, RF_i = d_i(k_i - 1) + RF_{i-1} \tag{3}$$

In order to obtain more background information features in different regions of the input data x, a series of decoupled depth-wise convolution kernels with different receptive fields can be used. If there are n decoupled convolution kernels, each convolution operation needs to be followed by a 1×1 convolution kernel for feature fusion. First, the convolution kernels with different receptive fields are concatenated and then undergo average pooling and max pooling. The features after average pooling (SA_{avg}) and max pooling (SA_{max}) are concatenated, converting the two channels of pooling features into N spatial attention feature maps. The activation function Sigmoid is used to calculate the mask for each spatial attention feature map, and finally, it is weighted with the features extracted by the decoupled large convolution kernels.

This study integrates the LKS module into the backbone network and combines it with the previously improved C2f_GC module. First, the C2f_GC module captures the global contextual information of image features, and then the dynamic convolution kernels in the LSK module continually adjust the required target background information, enhancing the capabilities of feature fusion and feature extraction. In the LSK module, the dimensions of input image x is (c,h,w), where the c is number of channels, h is the height of the input image, and w is the width. First, the input image is passed through the depth-wise convolution and the output is attention1, the number of image channels and size remain unchanged, the kernel size of depth-wise convolution is 5, and the padding is 2. This is followed by a depth-wise convolution where the kernel size is 7, padding is 9, dilation rate is 3, the output is attention2, and attention2 also maintains the same dimensions and number of channels. Each of these is then processed through a 1×1 convolution, the dimensions of output are (c/2,h,w), the outputs are concatenated, followed by average pooling and max pooling, and the dimensions of the output are changed to be (1,h,w). Then, use the 7×7 convolution kernel for feature concatenation to change the number of channels to 2. Finally, after weighted and concatenation of attention1 and attention2, the number of channels c is restored after passing through the 1×1 convolution kernel. The specific structure of the C2f_GC + LKS module is shown in Figure 5.

3.2.3. Introducing Attention Mechanism SimAM Module

In the field of computer vision, attention mechanism modules focus primarily on channel attention and spatial attention. However, in the human brain, these two attention mechanisms exist simultaneously. The SimAM module uses the neuroscience method to make these two attention mechanisms work simultaneously. The weight of a single neuron is calculated through the feature map in a layer. The SimAM attention mechanism calculates the linear separability between the target neuron and other neurons to determine which neurons have higher priority. The energy function for neurons in SimAM is defined in Equation (4), where t is the target neuron, x_i represents other neurons, i is the spatial index, N (equal to h $\times$ w) is the total number of neurons in the channel, and w_t and b_t are the weights and biases of the linear transformation. This energy function is minimized to calculate the mean and variance of all neurons; the lower the energy, the more distinct the neurons are, and thus it has a greater impact on visual processing.

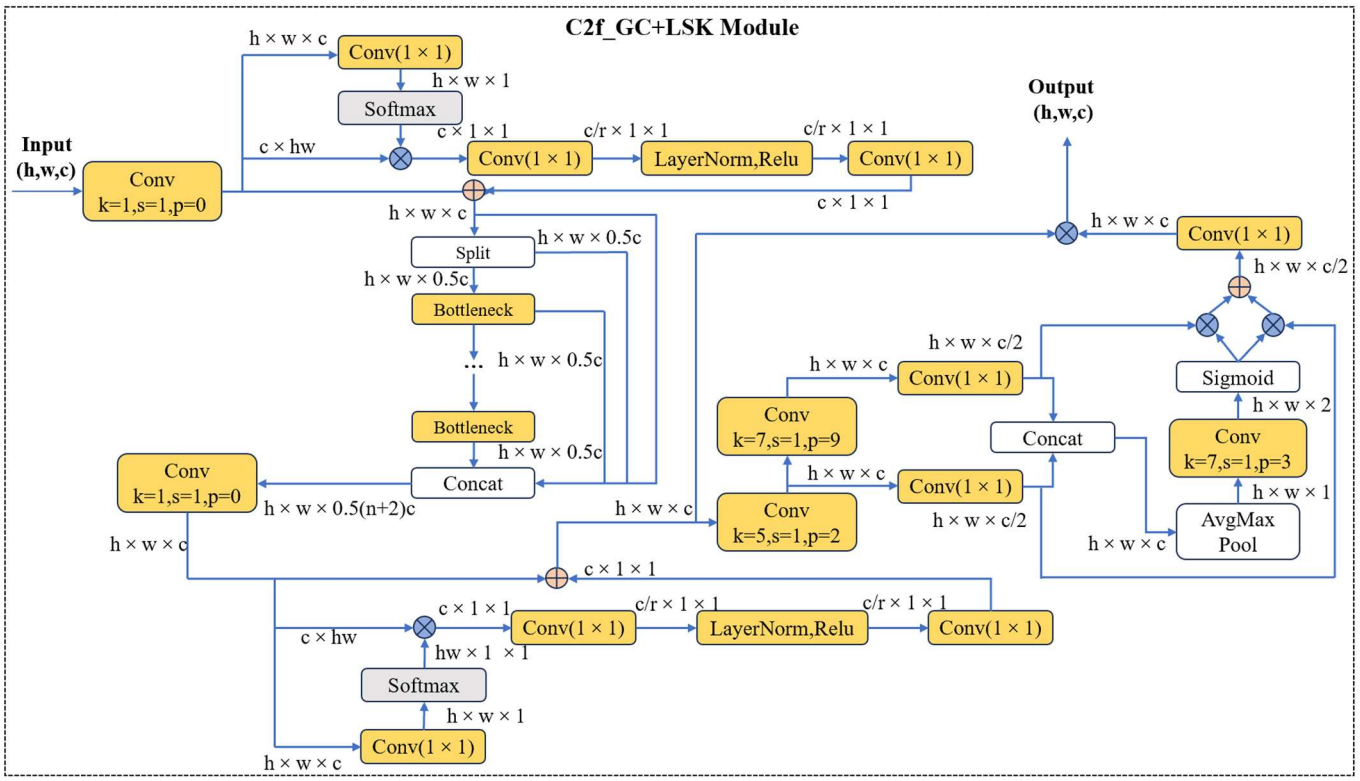

Figure 5. C2f_GC+LKS module network architecture.

$$g_t(w_t, b_t, y, x_i) = (y_t - \hat{t})^2 + \frac{1}{N-1}\sum_{i=1}^{N-1}(y_o - \hat{x_i})^2 \tag{4}$$

SimAM adjusts the extraction ability of feature maps by calculating the attention weight of each channel. When the input image has dimensions $H \times W \times C$, the attention weight calculation follows Equation (5), where w_c is the weight for channel C, F_{ijc} is the feature value at position in channel C, and the spatial dimension size is N, and N equals to $H \times W$.

$$w_c = \frac{1}{N}\sum_{i=1}^{H}\sum_{j=1}^{W} F_{ijc} \tag{5}$$

Then, the weights are normalized by the mean and standard deviation of all channel weights, and finally, the original feature map is adjusted by the normalized weight α_c. The normalization calculation is shown in Equation (6), where μ_ω is the mean of all channel weights and σ_ω is the standard deviation of all channel weights. The normalized weight α_c is then used to adjust the original feature map.

$$\alpha_c = \frac{\omega_c - \mu_\omega}{\sigma_\omega} \tag{6}$$

In the neck network, the feature maps output by SPPF first pass through a SimAM module before upsampling. After the first upsampling, the feature maps are concatenated with the LKS module's output from stage layer 4, followed by a second upsampling and concatenation with the SimAM output from stage layer 3. Adding the SimAM attention mechanism in the neck network involves the following: Assuming the input image x has dimensions (c,h,w), with c as the number of channels, h as the height, and w as the width, the number of other neurons in the spatial dimension is h × w−1, calculating the mean μ of all input neurons, and then find the square of the difference between the neuron and the mean. Then, calculate the attention weight y; the specific calculation is shown in

Equation (7), and the output attention weight y is passed through the sigmoid activation function and then weighted with the original input x to obtain the final output Y.

$$y = \frac{(x-\mu)^2}{4(\frac{\sum(x-\mu)^2}{n} + \gamma)} + 0.5 \tag{7}$$

By adding SimAM attention, the feature representation capability is improved, key features are enhanced, and the noise and redundant information of the feature map are reduced, so that the network can focus on important feature areas more effectively without adding additional parameters, ensuring the light weight of the model.

4. Experiments and Analysis

4.1. Experimental Setting

The computer operating system used in this experiment is Windows 10 64-bit, equipped with a 12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz, a GPU NVIDIA RTX 3080 (10 GB), and a Pytorch 2.0.0 framework used in the deep learning environment, configured with Cuda11.1. In the experimental environment parameters, the optimizer uses SGD, the learning rate is 0.01, the weight decay is 0.0005, the batch size is set to 16, and the number of training epochs is set as 1200. The configuration environment and parameters during the experiment are shown in Tables 2 and 3.

Table 2. Experimental configuration.

Name	Configure
Operating system	Windows 10
CPU	12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz
GPU	NVIDIA RTX 3080 (10 GB)
Run memory	64 GB
Deep-learning Framework	Pytorch2.0.0

Table 3. Training parameters.

Parameter	Value
Learning Rate	0.01
Batch Size	16
Epochs	1200
Weight Decay	0.0005
Optimizer	SGD

4.2. Experimental Dataset

The dataset used in this paper was obtained through independent collection. The dataset mainly comes from the pipette tips used on the liquid handling platform. It is collected indoors by a high-resolution camera. By collecting images of the normal tips and liquid-hanging conditions of the pipette tips, different colors of pipette tips were used in the collection process. At the same time, different solutions were used to simulate the pipette tips hanging liquid. The dataset contains two types of targets, one is the normal pipette tips without hanging liquid and liquid retention, and the other is the pipette tips with hanging liquid and liquid retention. The entire dataset has a total of 1286 images, including 736 images of pipette tips with hanging liquid and 550 images of normal pipette tips without hanging liquid. Both normal and hanging liquid pipette tips are annotated in the images. There are 923 training set images, 263 validation set images, and 100 test set images. The detailed information of the dataset is shown in Figure 6, including the midpoint coordinates, height, and width of each ground truth.

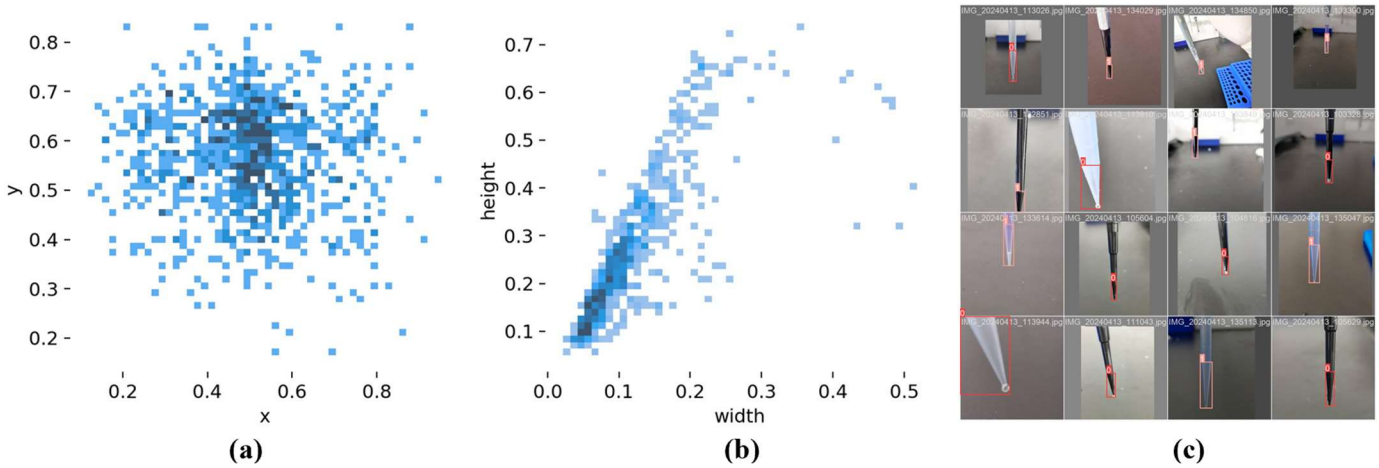

Figure 6. Detailed information of the pipette tip dataset; (**a**) the center point coordinates of each ground truth; (**b**) the height and width of each ground truth; (**c**) some examples and pictures of the ground truth in datasets.

4.3. Model Evaluation Indicators

In order to evaluate the performance and effectiveness of the model, the evaluation indicators used in this study are precision (P), mean average precision (mAP), F1 score, GFLOPS, and FPS.

(1) Precision: Precision is the proportion of the actual positive value among all samples classified as positives in the model's prediction results. It can measure the accuracy of the algorithm. The calculation of precision is represented by Equation (8). *TP* represents the true case, *FP* represents the false positive case, and *FN* represents the false negative case.

$$P = \frac{TP}{TP + FP} \tag{8}$$

(2) Mean average precision (*mAP*): *mAP* is a key evaluation indicator of the object detection algorithm. It is the mean average precision. The higher the *mAP* (mean average precision) value is, the better the performance of the algorithm is. mAP@0.5 represents the precision calculated when the *mAP* is at an IoU threshold of 0.5. That is, when the IoU between the detected object and the real object exceeds 0.5, it is considered that the correct object is detected. The specific calculation formula is shown in Equation (9).

$$mAP = \frac{\sum_{i=1}^{m} AP_i}{m} \tag{9}$$

(3) F1 score is the harmonic mean of precision and recall, and is an evaluation indicator that comprehensively considers precision and recall. The specific calculation is shown in Equation (10).

$$\mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}} \tag{10}$$

(4) GFLOPS is the abbreviation for billion floating-point operations per second. FPS is typically the frequency (rate) at which consecutive images (frames) are processed.

4.4. Ablation Experiments

In order to verify the effectiveness of the model improvement, this study conducted an ablation experiment. The same training environment and training parameters were used during the experiment. The baseline network selected in this study was YOLOv8. The experimental results are shown in Table 4. In Table 4, we compared the impact of adding the improved module on the detection results.

Table 4. Ablation experiment. '√' indicates that the corresponding improvement is used in the model for detection.

Serial Number	C2F_GC	GC	LKS	SimAM	mAP@0.5	F1 Score	Precision	FPS	GFLOPS
1	-	-	-	-	0.875	0.81	0.783	385	8.1
2	√				0.862	0.79	0.756	417	8.3
3		√			0.828	0.78	0.744	417	18.1
4	√	√			0.888	0.83	0.780	400	18.3
5	√	√	√		0.886	0.82	0.782	345	18.6
6	√	√	√	√	0.892	0.83	0.800	278	19.0

According to the ablation experiments, the mAP@0.5 (mean average precision) of the baseline model is 87.50%. The mAP@0.5 of the second experiment is 86.20%, which shows that when the attention mechanism GCNet module is added only to the C2f module, the performance of the model decreases slightly. In the third experiment, the GCNet module was added only to the backbone network, and the performance of the model also decreased. However, in the fourth experiment, the attention GCNet module was added to both the backbone and the C2f module, which increased mAP@0.5 by 1.3% and F1 by 2%. Through experiments 2, 3 and 4, it is shown that the GCNet module has a synergistic effect. The addition of the GCNet module to the backbone and the C2f module simultaneously makes the model mutually reinforced, and the overall effect is better. The fifth and sixth experiments show that the mAP@0.5 increases by 1.1% and 1.70%, respectively. The mAP@0.5 of the improved algorithm reaches 89.20%, which is 1.7% higher than the baseline model.

After adding the attention module GCNet to the C2f module and the backbone network, the model detection performance increased significantly. Adding the attention mechanism module to the backbone significantly enhanced the model's detection performance, allowing the model to better understand the global image structure and improving the ability to deal with non-target features that interfere with the network, thereby focusing more on the target features. On this basis, adding the LSK attention mechanism module to the backbone and the SimAM attention mechanism module to the neck network further improved small target detection performance. The LSK module adaptively extracts information from different target backgrounds, and the SimAM module generates attention weights by calculating the self-similarity of feature maps, enabling the model to focus more on key areas of the image. According to the experimental results in Table 4, the improved Yolov8 shows superior performance compared to the original baseline model Yolov8, with the mAP@0.5 increased by 1.7% and the F1 score improved by 2%. These results further enhance the model's accuracy and demonstrate its effectiveness. After training, the F1 score curve and PR (precision–recall) curve of the improved model and the baseline model were plotted based on the training data, as shown in Figures 7 and 8.

The coverage of the F1 score curve and PR curve of the improved model has been significantly improved, indicating that the accuracy of the detection results has been improved.

4.5. Comparison of Different Algorithms

In order to evaluate the detection ability of the improved algorithm reasonably and effectively, this study compares the algorithm with the mainstream models of object detection. We used five common object detection models: YOLOv3, YOLOv5s, YOLOv6, YOLOv7-tiny, and YOLOv8n, comparing them with the improved YOLOv8+C2f_GC+GC+LKS+SimAM model on a self-constructed dataset. Consistency in experimental datasets, environments, and parameters was ensured. The comparison results are shown in Table 5, and the PR curves of the comparative experiments are shown in Figure 9. According to the experimental results in Table 5 and Figure 9, the improved algorithm model outperformed other object detection algorithms in terms of mean average precision (mAP@0.5), F1 score, and precision. Although the detection speed of the improved YOLOv8 algorithm slightly decreased, its detection performance was improved.

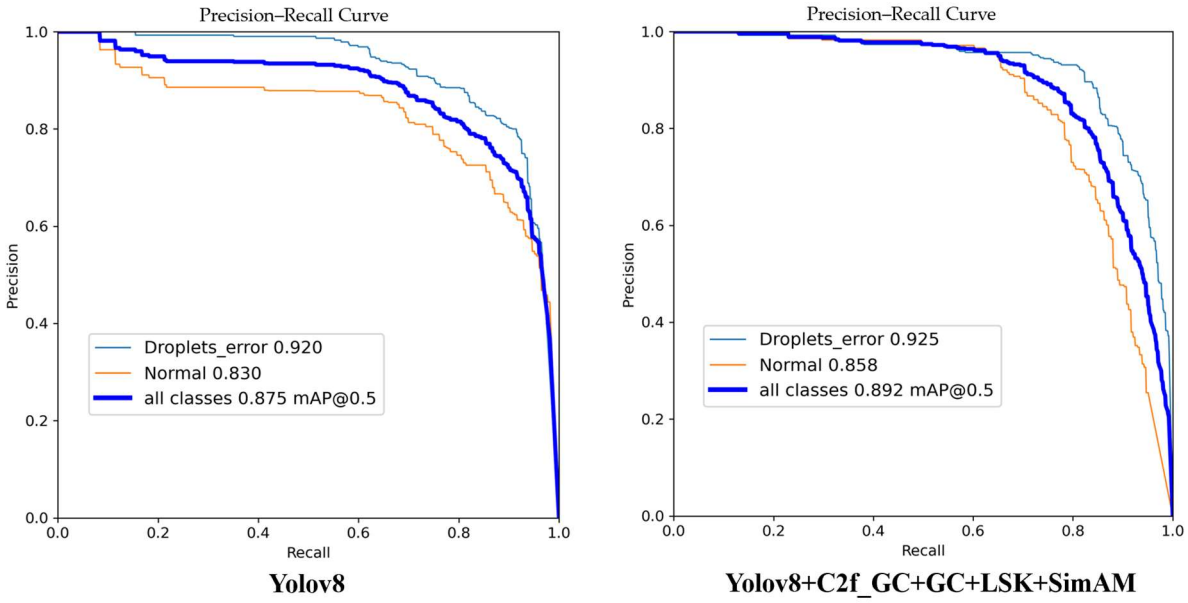

Figure 7. PR curve.

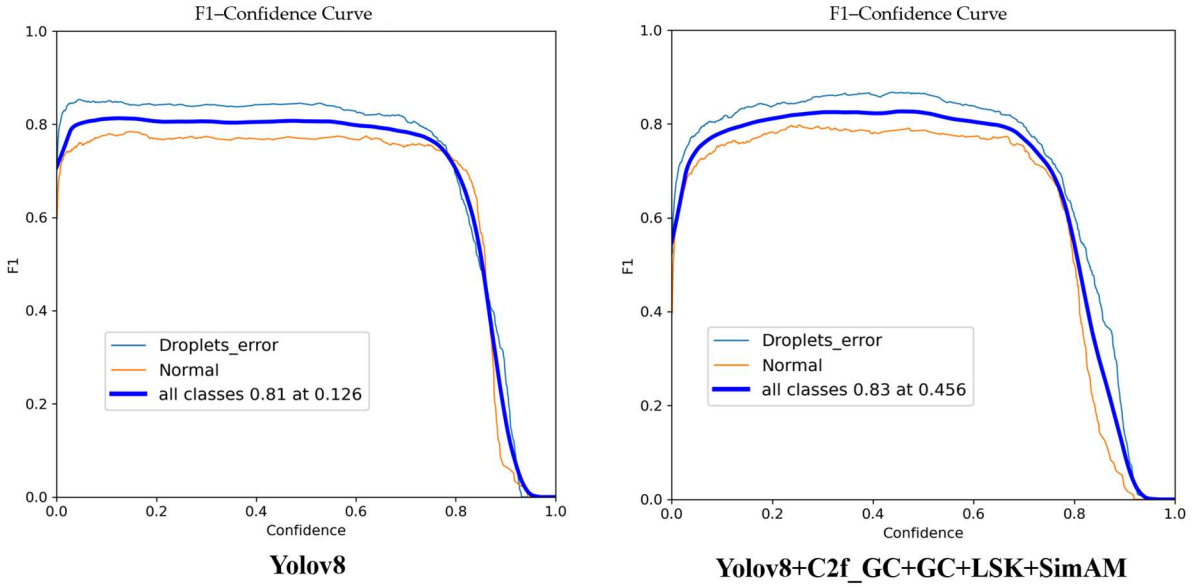

Figure 8. F1 score curve.

Table 5. Different algorithm model comparison.

Algorithm	mAP@0.5	F1 Score	Precision	FPS	GFLOPS
YOLOv3	0.843	0.770	0.780	114	282.2
YOLOv5s	0.877	0.820	0.770	357	23.8
YOLOv6	0.883	0.810	0.731	435	13.1
YOLOv7-tiny	0.876	0.830	0.770	142	13.0
YOLOv8n	0.875	0.810	0.783	385	8.0
Ours	0.892	0.830	0.800	278	19.0

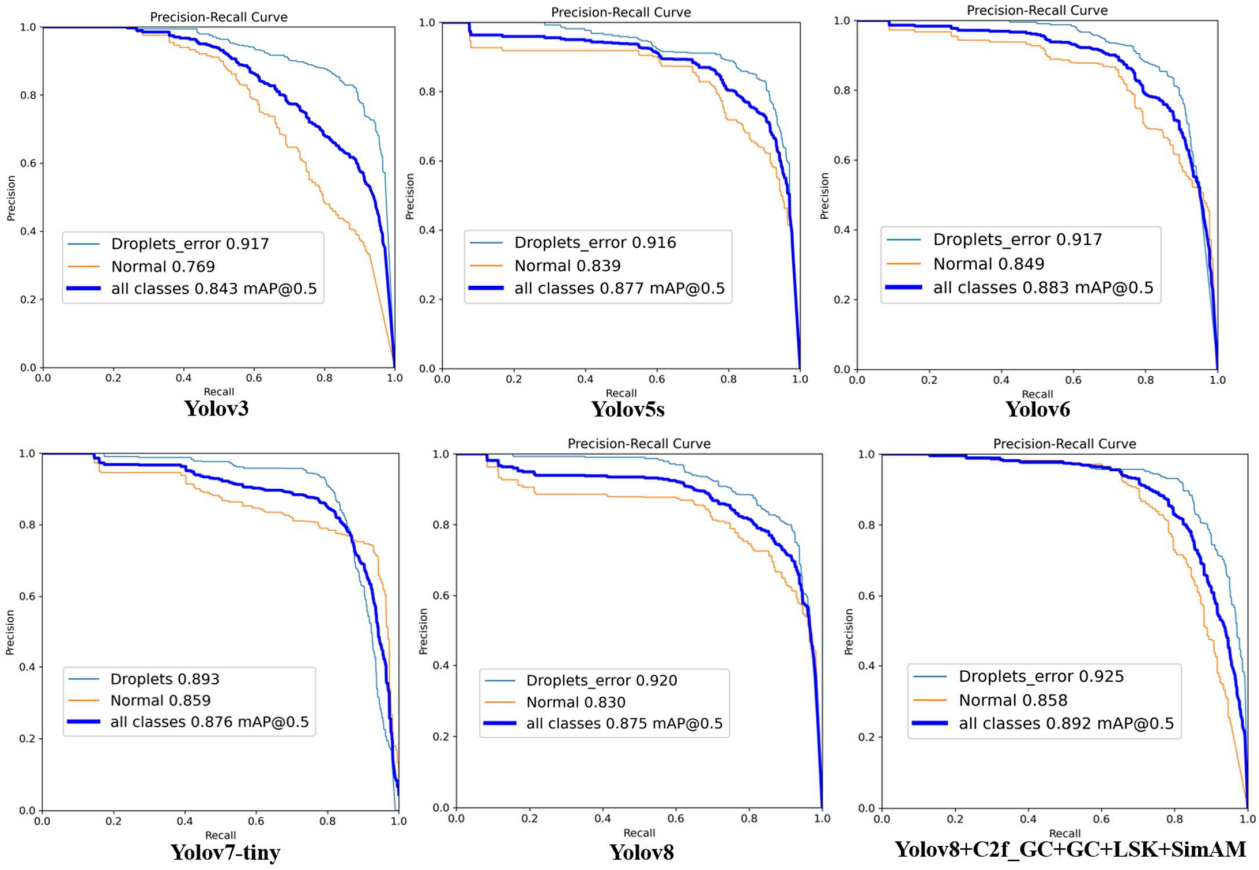

Figure 9. PR curves of different algorithm comparison experiments.

Based on the results of the comparative experiments, we plotted the F1 score, mAP@0.5, and precision parameters of the comparison results, as shown in Figure 10. This verified that the performance of the improved algorithm on our self-constructed pipette tip hanging liquid and liquid retention dataset was enhanced. Compared to other object detection algorithms, our algorithm demonstrated higher precision and recall while maintaining speed, significantly reducing the likelihood of missed detections and false detections. Based on the results of ablation and comparative experiments, the improved YOLOv8 algorithm performed better than other algorithms in the high-throughput pipette tip hanging liquid detection task.

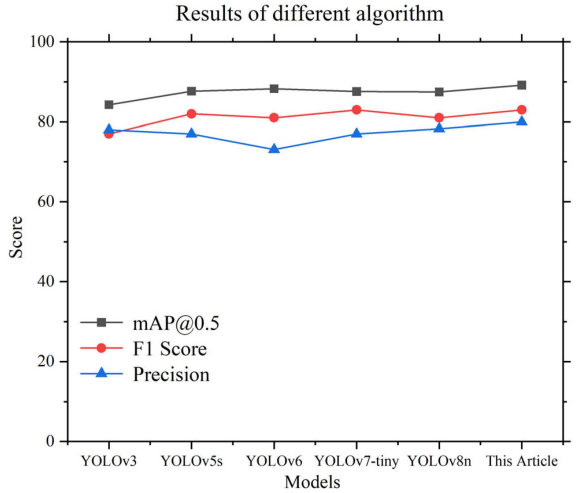

Figure 10. Results of different algorithm.

4.6. Test Results of Image Datasets

Based on the previous ablation experiments and comparative experiments, the improved YOLOv8 model was verified. We tested 12 identical test images with the baseline model YOLOv8 model and the improved YOLOv8 model and observed the test results. The test results are shown in Figure 11, where (a) is the original annotated image, (b) is the result predicted by the baseline model YOLOv8, and (c) is the result predicted by the improved YOLOv8 model.

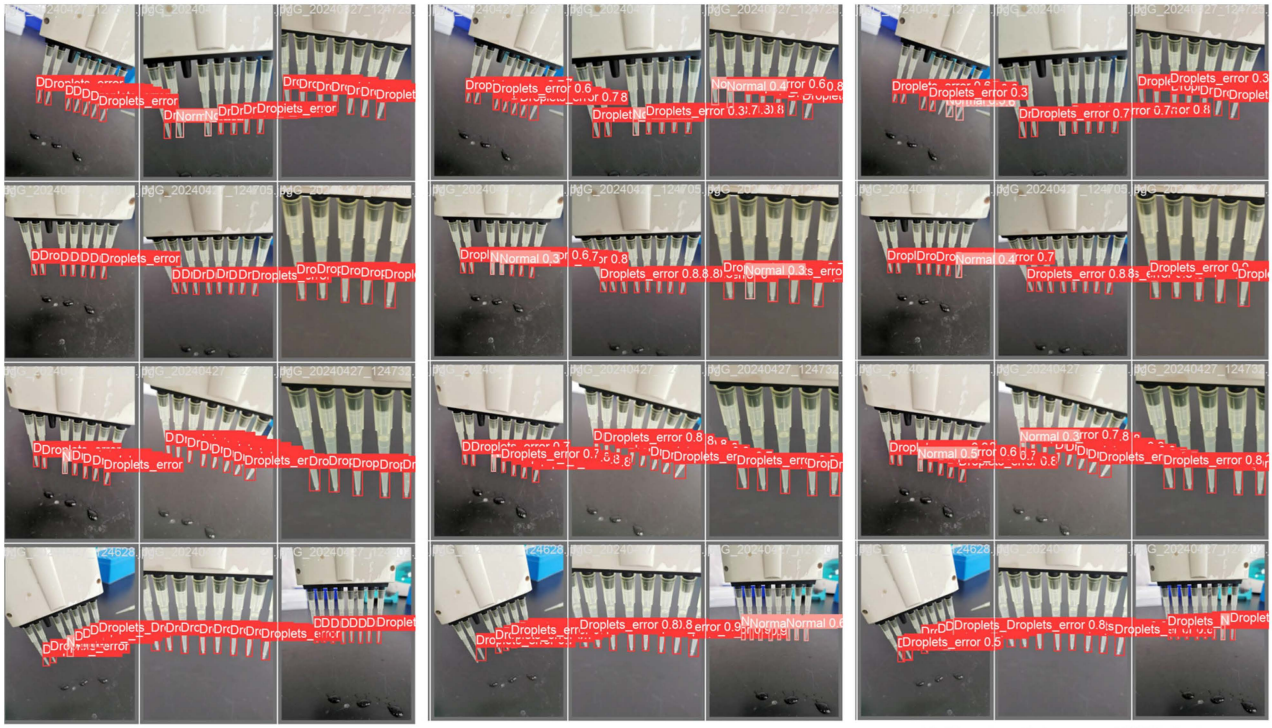

Figure 11. (**a**) Labels of pictures, where rectangles represent the ground truth boxes. (**b**) The results predicted by the YOLOv8, rectangles represent the predicted boxes and the numbers indicate the predicted probabilities. (**c**) The results predicted by the improved YOLOv8.

By comparing Figure 11, it can be seen that the improved algorithm can more accurately locate the pipette tips with liquid retention and the normal pipette tips without liquid hanging and retention, and at the same time, it improves the confidence to a certain extent, while increasing the detection ability and accuracy of small targets and object. The improved YOLOv8 model performs better.

5. Conclusions

In this paper, we proposed significant enhancements to the YOLOv8 algorithm to address the detection challenges of liquid retention on pipette tips in high-throughput liquid handling workstations. It also provides the basis for the automatic solution of the liquid hanging on the pipette tip in the subsequent pipetting workstation. Our contributions can be summarized as follows.

(1) Incorporation of attention mechanisms:

We integrated the GC attention mechanism into the backbone network and C2f modules of YOLOv8, this enhancement allows the model to better capture global image structures and focus more precisely on target features, thereby improving robustness against various interference factors. To dynamically adjust the extraction of background information for different targets, we integrated the LSK module into the backbone network, this module

enhances the fusion capability for small target features, which is critical for detecting pipette tips and liquid retention. We redesigned the neckbone network of YOLOv8 by incorporating SimAM attention mechanism modules between the up and down sampling processes and during the feature map fusion process of the backbone network. By calculating the self-similarity of the feature maps to generate attention weights, our model focuses more effectively on key regions of the image, thus improving overall detection performance.

(2) Performance improved on datasets:

We validated the effectiveness of our improved algorithm through ablation and comparative experiments. Our results demonstrated significant performance gains over the baseline YOLOv8 model, with increases in mAP@0.5, F1 score, and precision by 1.7%, 2%, and 1.7%, respectively. We tested and compared our improved algorithm on a high-throughput pipette tip dataset, achieving higher accuracy and enhancing overall detection performance compared with the baseline model. These contributions collectively enhance the detection capabilities and reliability of the improved YOLOv8 algorithm for applications in high-throughput liquid handling workstations, particularly in detecting small targets such as pipette tips with liquid residue. It provides the basis for the automatic solution of the liquid hanging on the pipette tip in the subsequent pipetting workstation.

However, further work is needed to lightweight this model, aiming to reduce parameters and floating-point calculations while maintaining a high level of detection accuracy. In future research, we will use model pruning methods to remove unimportant weights to reduce the number of parameters. At the same time, we will try to use quantization methods to convert the weights and activations in the model. This conversion can significantly reduce the storage requirements and computational complexity of the model, thereby improving the inference speed. These would improve prediction inference speed, achieve higher detection speed and accuracy, and increase deployment efficiency and speed on other devices such as embedded systems.

Author Contributions: Paper direction, W.T. and Y.Y.; data collection, Y.Y. and J.L.; software, Y.Y. and J.L.; algorithm improvements, Y.Y., J.L. and W.T.; training and validation, Y.Y. and J.L.; writing—original draft preparation, Y.Y. and J.L.; writing—review and editing, Y.Y.; supervision, W.T.; Y.Y. and J.L. contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Derived data supporting the findings of this study are available from the corresponding author on request.

Acknowledgments: The authors would like to thank the support of the reviewers as well as the editors for their insightful comments.

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

References

1. Krzysztof, L.; Haakan, N.J. Rapid Production and Recovery of Cell Spheroids by Automated Droplet Microfluidics. *SLAS Technol.* **2020**, *25*, 111–122. [CrossRef]
2. Coppola, L.; Smaldone, G.; Cianflone, A.; Baselice, S.; Mirabelli, P.; Salvatore, M. Purification of viable peripheral blood mononuclear cells for biobanking using a robotized liquid handling workstation. *J. Transl. Med.* **2019**, *17*, 371. [CrossRef] [PubMed]
3. Annona, G.; Liberti, A.; Pollastro, P.; Spagnuolo, A.; Sordino, P.; Luca, P.D. Reaping the benefits of liquid handlers for high-throughput gene expression profiling in a marine model invertebrate. *BMC Biotechnol.* **2024**, *24*, 4. [CrossRef] [PubMed]
4. Gautam, K.S.; Thangavel, S.K. Video analytics-based intelligent surveillance system for smart buildings. *Soft Comput.* **2019**, *23*, 2813–2837. [CrossRef]
5. Xie, T.B.; Yao, X.F. Smart Logistics Warehouse Moving-Object Tracking Based on YOLOv5 and DeepSORT. *Appl. Sci.* **2023**, *13*, 9895. [CrossRef]
6. Meda, K.C.; Milla, S.S.; Rostad, B.S. Artificial intelligence research within reach: An object detection model to identify rickets on pediatric wrist radiographs. *Pediatr. Radiol.* **2021**, *51*, 782–791. [CrossRef]
7. Hong, Q.; Dong, H.; Deng, W.; Ping, Y.H. Education robot object detection with a brain-inspired approach integrating Faster R-CNN, YOLOv3, and semi-supervised learning. *Front. Neurorobot.* **2023**, *17*, 1338104. [CrossRef] [PubMed]

8. Albadawi, Y.; AlRedhaei, A.; Takruri, M. Real-Time Machine Learning-Based Driver Drowsiness Detection Using Visual Features. *J. Imaging* **2023**, *9*, 91. [CrossRef] [PubMed]
9. Hütten, N.; Alves Gomes, M.; Hölken, F.; Andricevic, K.; Meyes, R.; Meisen, T. Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open-Access Papers. *Appl. Syst. Innov.* **2024**, *7*, 11. [CrossRef]
10. Deng, J.; Xuan, X.; Wang, W.; Li, Z.; Yao, H.; Wang, Z. A review of research on object detection based on deep learning. *J. Phys. Conf. Ser.* **2020**, *1684*, 012028. [CrossRef]
11. Hebbache, L.; Amirkhani, D.; Allili, M.S.; Hammouche, N.; Lapointe, J.-F. Leveraging Saliency in Single-Stage Multi-Label Concrete Defect Detection Using Unmanned Aerial Vehicle Imagery. *Remote Sens.* **2023**, *15*, 1218. [CrossRef]
12. He, K.M.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
13. Morera, Á.; Sánchez, Á.; Moreno, A.B.; Sappa, Á.D.; Vélez, J.F. SSD vs. YOLO for Detection of Outdoor Urban Advertising Panels under Multiple Variabilities. *Sensors* **2020**, *20*, 4587. [CrossRef] [PubMed]
14. Jiang, P.Y.; Ergu, D.J.; Liu, F.Y.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
15. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
16. He, F.X.; Liu, T.L.; Tao, D.C. Why ResNet Works? Residuals Generalize. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 5349–5362. [CrossRef]
17. Toan, V.Q.; Min, Y.K. Feature pyramid network with multi-scale prediction fusion for real-time semantic segmentation. *Neurocomputing* **2023**, *519*, 104–113. [CrossRef]
18. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [CrossRef]
19. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9:Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616. [CrossRef]
20. Cao, D.Y.; Chen, Z.X.; Gao, L. An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks. *Hum. Centric Comput. Inf. Sci.* **2020**, *10*, 14. [CrossRef]
21. Hwang, Y.J.; Kim, G.H.; Kim, M.J.; Nam, K.W. Deep learning-based monitoring technique for real-time intravenous medication bag status. *Biomed. Eng. Lett.* **2023**, *13*, 705–714. [CrossRef]
22. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972. [CrossRef]
23. Cao, Y.; Xu, J.R.; Lin, S.; Wei, F.Y.; Hu, H. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond. *arXiv* **2019**, arXiv:1904.11492. [CrossRef]
24. Li, Y.X.; Hou, Q.B.; Zheng, Z.H.; Cheng, M.M.; Yang, J.; Li, X. Large Selective Kernel Network for Remote Sensing Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 16748–16759. [CrossRef]
25. Yang, L.X.; Zhang, R.Y.; Li, L.D.; Xie, X.H. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 11863–11874.
26. Cui, W.X.; Liu, S.H.; Jiang, F.; Zhao, D.B. Image Compressed Sensing Using Non-Local Neural Network. *IEEE Trans. Multimed.* **2023**, *25*, 816–830. [CrossRef]
27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]