electronics

Article

FL-YOLOv8: Lightweight Object Detector Based on Feature Fusion

Ying Xue [1,2], Qijin Wang [2,3,*], Yating Hu [4], Yu Qian [1,2], Long Cheng [1,2] and Hongqiang Wang [3]

[1] School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei 230601, China; yiyi07102421@stu.ahjzu.edu.cn (Y.X.); qyu21490@stu.ahjzu.edu.cn (Y.Q.); longcheng@stu.ahjzu.edu.cn (L.C.)
[2] School of Big Data and Artificial Intelligence, Anhui Xinhua University, Hefei 230088, China
[3] Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China; hqwang@ustc.edu
[4] School of Information and Network Engineering, Anhui Science and Technology University, Bengbu 233030, China; huyating@ahstu.edu.cn
* Correspondence: qjwang@mail.ustc.edu.cn

Abstract: In recent years, anchor-free object detectors have become predominant in deep learning, the YOLOv8 model as a real-time object detector based on anchor-free frames is universal and influential, it efficiently detects objects across multiple scales. However, the generalization performance of the model is lacking, and the feature fusion within the neck module overly relies on its structural design and dataset size, and it is particularly difficult to localize and detect small objects. To address these issues, we propose the FL-YOLOv8 object detector, which is improved based on YOLOv8s. Firstly, we introduce the FSDI module in the neck, enhancing semantic information across all layers and incorporating rich detailed features through straightforward layer-hopping connections. This module integrates both high-level and low-level information to enhance the accuracy and efficiency of image detection. Meanwhile, the structure of the model was optimized and designed, and the LSCD module is constructed in the detection head; adopting a lightweight shared convolutional detection head reduces the number of parameters and computation of the model by 19% and 10%, respectively. Our model achieves a comprehensive performance of 45.5% on the COCO generalized dataset, surpassing the benchmark by 0.8 percentage points. To further validate the effectiveness of the method, experiments were also performed on specific domain urine sediment data (FCUS22), and the results on category detection also better justify the FL-YOLOv8 object detection algorithm.

Keywords: object detection; YOLOv8s; feature fusion; lightweight shared convolution

Citation: Xue, Y.; Wang, Q.; Hu, Y.; Qian, Y.; Cheng, L.; Wang, H. FL-YOLOv8: Lightweight Object Detector Based on Feature Fusion. *Electronics* **2024**, *13*, 4653. https://doi.org/10.3390/electronics13234653

Academic Editor: José Carlos Castillo

Received: 29 September 2024
Revised: 19 November 2024
Accepted: 21 November 2024
Published: 25 November 2024

1. Introduction

Object detection is a crucial component of deep learning and a trending subject in recent years, which aims to precisely locate and classify objects during image or video processing based on predefined categories in the dataset. Currently, object detection finds extensive application across various industries such as transportation [1], artificial intelligence [2], medical imaging [3], autonomous driving [4], and others.

With the rapid development of deep learning and computer vision technology, object detection algorithms have evolved into anchor-based and anchor-free methods. Examples of anchor-free methods include CornerNet [5], FCOS [6], and the YOLO series. The YOLOv8 model, in particular, stands out for its exceptional accuracy and speed, facilitating real-time object detection in any scenario. The backbone network of this model is based on the Darknet [7] architecture and incorporates ideas from CSP [8]. Typically, the model conducts feature fusion operations on extracted backbone features. For instance, YOLOv3 [9] employs a feature pyramid network and NAS-FPN [10] utilizes an adaptable feature pyramid network architecture for feature fusion; Tan et al. [11] noted that the feature fusion network of the FPN class performs only basic fusion and does not account for the varying

contributions of features across different layers. Therefore, they proposed a weighted bi-directional feature pyramid network that assigns weights to the contributions of different layers, significantly improving model accuracy. Similarly, Peng et al. [12] believe that when high-level features and low-level features are fused, they will be interfered by noise and the limitation of dataset size; in response to this, YOLOv8 adopts the network structure of PANet [13] in feature fusion of the neck, and through the introduction of the aggregation network of top-down and bottom-up paths, it can adequately extract the feature maps of different levels and improve the detection ability of the object, but it also brings some problems, including information loss, distortion, and large memory occupation when the high-level and low-level features are passed and information is spliced and fused, which is not favorable for the detection of vague objects and small-scale objects; so, we first propose the FSDI module to be applied in the field of object detection, which fuses the high-level features and the low-level feature information by hopping the layer connection, which in turn acquires more richly detailed information and improves the object detection accuracy.

As the field of object detection grows rapidly, research is increasingly focusing on adopting more efficient detection heads. For instance, Feng et al. [14], in research related to the detection of heads, observed that using decoupled heads causes a misalignment between the classification and regression detection tasks due to their differing spatial feature distributions. While the detection head of the YOLOv8 model adopts the design of decoupled head [15], which separates the detection head of classification and regression and can flexibly use the detection of objects in different scenarios, the model also uses the DFL strategy in the integral form representation for regression, which further enhances the performance of the model; however, the number of channels of the prediction head changes under different datasets, which results in the model needing to traverse all channels during detection, which also contributes to the increase in the computation of the detection head of the model. In order to better solve the limitations brought by the decoupled head, this paper uses the lightweight shared convolution to reduce computational and parametric complexities in the detection head [16] and at the same time uses group normalization GN (Group Normalization) [17] to enhance the performance of the model classification and localization. In summary, the main contributions of this paper can be summarized as follows:

1. A Fully Semantics and Detail Infusion (FSDI) module is developed to replace conventional feature splicing during the feature fusion process. Additional shallow layers are added to the fusion stage to improve feature integration. Then, the corresponding feature information is processed using the Hadamard product to obtain more richly detailed features;
2. The Lightweight Shared Convolutional Detection (LSCD) module is designed to retain classification and localization features within the detection head, it utilizes shared convolution to reduce parameters and computational load, and it might lead to decreased accuracy. Then, after the convolution process, group normalization is applied to offset any potential loss in accuracy. Additionally, scale layer scaling is employed to facilitate the detection of objects across varying sizes.

Besides performing experiments under the generalized COCO dataset, the above two approaches were combined to conduct relevant tests and analyses on domain-specific urinary sediment datasets, with a significant decrease in the number of parameters and computation and an improvement in the detection of different types of cells.

2. Related Work

2.1. Multi-Scale Feature Fusion

Multi-scale refers to the process of sampling signals at various levels of granularity [18]. Different features can be observed across different scales to facilitate object detection at various levels. Feature fusion is a crucial technique in deep learning, enabling the combination of features from different branches or layers. In image processing, conventional multi-scale fusion often involves directly summing or splicing them together, typically

through a straightforward operation. However, this approach may not always be optimal. Effectively fusing multi-scale information to predict objects of varying sizes across different scales can significantly enhance overall model performance.

Multi-scale feature fusion encompasses parallel multi-branch networks and serial layer-hopping connectivity structures. Parallel multi-branch networks like SPPNet [19] utilize multiple parallel convolutional layers to extract features from various receptive fields, which are then processed and fused in separate branches before passing the final results to the next layer, thereby balancing model performance and computational load more effectively. In contrast, serial structures integrate features from different layers; for instance, Liu et al. enhanced FPN [20] on SSD [21] by integrating deeper network advanced semantic information into shallower networks for recognition. Also, Peng et al. improved the U-Net Semantic Segmentation Network on FCN [22] incorporating an auxiliary decoder through skip connections to enhance semantic features and improve network representation. PANet introduces a bottom-up path aggregation module on FPN, providing precise semantic information for high-level features while reducing the number of convolutional layers required for information flow from high-level to bottom feature maps. PANet demonstrates more significant effectiveness compared to FPN, but the computation and number of parameters are also obviously increased.

2.2. Lightweight Convolutional Structures

When using convolution for feature extraction, the model will utilize multiple convolution layers to obtain more useful information features. For example, He et al. proposed Resnet [23], which is a classical convolutional neural network that provides thoughts for training deeper networks by introducing residual connections. Still, in practical applications, this deep neural network leads to higher computational cost and network depth. The model training and inference speeds are slow when dealing with larger datasets. while lightweight convolution requires fewer parameters and less computation in some specific domains and scenarios only by sacrificing some accuracy to obtain better real-time performance. For instance, YOLOv7 [24] features the ELAN module, which incorporates grouped convolution and grouping input feature maps to perform independent convolution operations within each group before merging the results. Depthwise Separable Convolution (DWConv) [25], introduced by Google, is a widely adopted convolutional network structure that convolves channels using a convolution kernel, promoting inter-channel information exchange. This approach substantially reduces computational complexity and network parameters, albeit with a potential performance degradation. One of them, MobileNet [26], uses deeply separable convolution with multiple layers which greatly reduces the number of model parameters and computations with a small reduction in accuracy. For example, Iandola et al. proposed SqueezeNet [27], which simplifies the network complexity by compressing the convolution kernel size and reducing the number of intermediate layer channels to achieve model compression; the comparison of the number of parameters before and after is shown in Table 1. In natural language processing, the BERT [28] model is commonly used as a teacher model to improve the performance of lightweight models by employing the concept of knowledge distillation [29] to transfer knowledge from a complex model to a lightweight model. Lightweight models are particularly useful in mobile devices, embedded systems, and resource-constrained environments.

Table 1. Before and after comparison experiments of lightweight model.

Architecture	Approach	#Params
MobileNet	Full Convolution	29.3 M
MobileNet	Depthwise Separable	4.2 M
SqueezeNe	None	4.8 M
SqueezeNet	Deep Compression	0.66 M

2.3. Decoupled Detector Head

In object detection, the detection head of a model plays a crucial role in recognizing features extracted from the backbone network. Traditional object detection models in deep learning typically employ a single detection head to predict the category and position of bounding boxes simultaneously. For example, YOLOv3 and YOLOv4 [30] utilize a coupled detection head, where category and position predictions are consolidated. This method may suffer from mutual interference as these tasks require distinct loss functions. Conversely, models like FCOS, YOLOv6 [31], and YOLOv7 resolve this issue by adopting a decoupled head design. Category prediction involves fully connected layers to determine category probabilities, while location prediction calculates bounding box coordinates using convolutional layers. This approach separates category and location prediction, each processed by distinct branches within the network. For instance, adopting a decoupled head design in YOLOX [32] notably enhances model convergence, resulting in a 4.2% increase in average precision compared to models with coupled detection heads. While decoupled head design improves model accuracy, it also increases the number of parameters and computational load, potentially reducing generalization performance and limiting applicability across different domains.

3. Method

This paper focuses on enhancing the detection performance of the YOLOv8s model by refining the neck and head components. Figure 1 illustrates the improved FL-YOLOv8 model, integrating the novel FSDI and LSCD modules. The feature fusion method in the neck is redesigned with a hopping connection approach, and the FSDI module incorporates semantic and detail injections to enrich feature extraction, enhancing the perceptual ability and average accuracy of the model. Furthermore, the model employs a decoupled head in the detection module to segregate classification and regression tasks. An LSCD module is reconstructed to minimize the number of parameters and computation of the model, which improves the detection accuracy and efficiency of the model. These methodological enhancements have yielded notable results on urine sediment data, facilitating more precise urine detection.

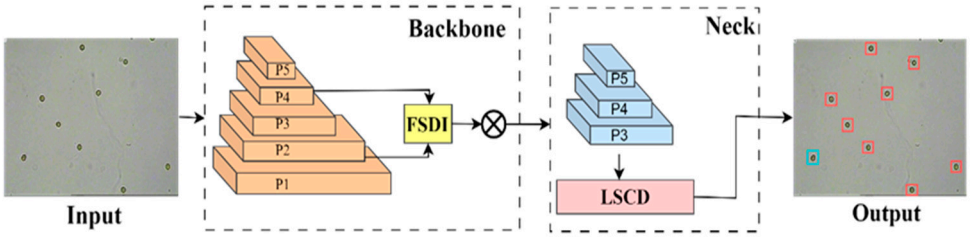


Figure 1. Flowchart of the FL-YOLOv8 detector.

3.1. FSDI Module

The SDI module is primarily employed in image segmentation. U-NET V2 integrates the SDI module into both the encoder and decoder to facilitate feature fusion through skip connections. Initially, the encoder extracts features from the input image. Subsequently, in the decoder, low-level features are integrated with high-level features to enrich the detailed information from low-level features and the semantic information from high-level features. This approach enhances the capability of the model for segmentation and image reconstruction.

Since simply fusing the features by concatenating them would rely heavily on the learning ability of the network, and since experimental results on model training in the field of object detection vary due to differences in dataset sizes, the FSDI module was chosen and applied to the field of object detection. Since the shallow layer contains rich detail information, the relevant information will be lost in the process of constant convolutional optimization, which will cause certain difficulties for the detection of the

corresponding object, so an FSDI module, a more comprehensive semantic and detail injection module, is constructed in the neck of YOLOv8, which fuses more comprehensive image features by jumping layer connections. Firstly, the FSDI module optimizes the feature maps extracted from the backbone network, and the number of channels is downscaled by 1×1 convolution, while the hopping layer connection fuses the shallower C2F module which together perform the feature extraction of information. Then, interpolation algorithm and constant mapping and pooling operation are applied to the feature maps of different feature layers to unify the size of the feature maps, and deeper and shallower features are fused to fully extract more semantic and detailed information, and then, all the results are multiplied using the Hadamard algorithm.

In FSDI for feature fusion, with FSDI of the P4 layer as an example, as shown specifically in (b) in Figure 2, due to the different resolutions of the C2F of the layer P3 and the C2F of the P4 layer, as well as the image resolution of the up-sampling layer, the size of the FSDI passed on to the next layer is $40 \times 40 \times 512$. The C2F of the P4 layer with the same resolution is regarded as the i layer and represents the features from different layers as j, and f_{ij} denotes the feature transformation from layer j to layer i. Equation (1) is as follows: when $j = i$, the corresponding features are obtained via constant mapping, denoted by Ⓘ; for $j > i$ and $j < i$, features are obtained via bilinear interpolation and pooling, denoted by Ⓓ and Ⓤ, respectively. Then, the images after the uniform resolution size are multiplied by the Hadamard code to obtain the final result. The principle of FSDI at layer P2 follows a similar approach.

$$f_{ij} = \begin{cases} Ⓓ\, f_j & j > i \\ Ⓘ\, f_j & j = i \\ Ⓤ\, f_j & j < i \end{cases} \tag{1}$$

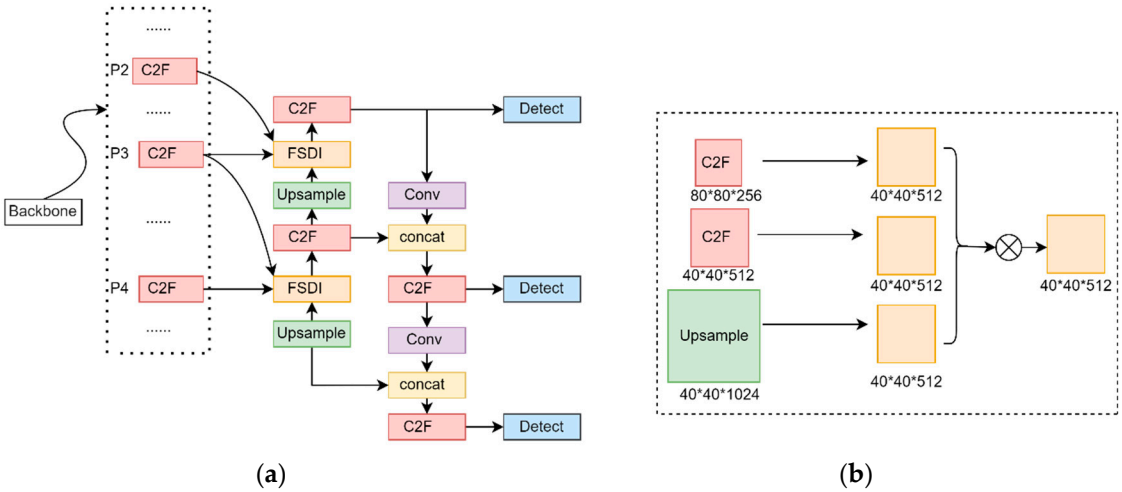


Figure 2. (**a**) represents the fusion of FSDI modules in the PAFPN structure, and (**b**) represents the FSDI specific fusion process, in which ⊗ represents the Hadamard product.

3.2. LSCD Module

The detection head plays a pivotal role in object detection. YOLOv8 utilizes the prevalent decoupled head detection structure, which splits object detection into two distinct tasks: classification and regression. Information is extracted through respective convolutions, followed by loss computation for each task. The decoupled head design makes each sub-task responsible for a specific category of object, which can realize the detection of different categories of objects individually, focusing on their respective independent tasks for detection and performance improvement, thereby significantly enhancing its flexibility and stability. But through comprehensive analysis and research, it was found that the detection head occupies 20% of total computation resources. Increased computation affects both

memory usage and speed. To optimize efficiency and reduce model complexity, we propose the Lightweight Shared Convolutional Detection (LSCD) head. Our approach focuses on using a lightweight design structure to better coordinate the accuracy and efficiency of the network and reduce the number of parameters and computation of the model.

To construct a lightweight shared convolutional detection head, firstly, the number of channels is changed by downscaling and upscaling the convolutional layers of different feature maps, and secondly, the form of shared convolution is adopted, as shown in (b) in Figure 3 below, so that the same convolution kernel can be used to extract features in layers P2, P3, and P4, which can significantly reduce the number of parameters and computation, but also accompanied by a decrease in the accuracy of the model along with a reduction in the amount of computation. In this regard, we choose to further optimize the detection head of the model, the features extracted after convolution are grouped on the channel, and the way of normalization within the group can improve the performance of the detection head classification and localization, which optimizes the disadvantage of the BN that is ineffective when the batch size is small, so as to make up for the missing accuracy of the shared convolution and to balance the accuracy and efficiency of the model and then conduct classification and regression operations. The classification uses a binary cross entropy loss function, where N denotes the total number of samples, y_i denotes the label corresponding to sample i, and p_i denotes the probability that sample i is predicted to be a positive class, so the function of the classification loss can be described as follows:

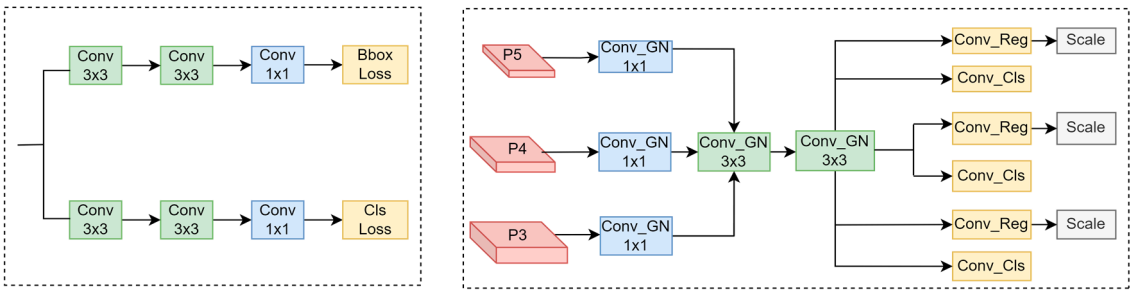
$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -[y_i \times log(p_i) + (1 - y_i) \times log(1 - p_i)] \tag{2}$$

The loss function that includes both DFL and $CIoU$ in the regression process can be described as follows:

$$DFL(S_i,\ S_{i+1}) = -((y_{i+1} - y)log(S_i) + (y - y_i)\,log(S_{i+1})) \tag{3}$$

$$CIoU = IoU - \frac{\rho^2(b, b_{gt})}{c^2} - \alpha v \tag{4}$$

with IoU denoting the intersection and merger ratio, b and b_{gt} denoting the centroids of the two rectangular boxes, respectively, ρ denoting the Euclidean distance between the two rectangular boxes, c denoting the diagonal distance between the closed regions of the two rectangular boxes, v used to measure the congruence of the relative proportions of the two rectangular boxes, and α denoting the weight coefficient. Finally, the features are scaled through the scale layer to solve the problem of inconsistent scaling of the object detected via each detection head in order to facilitate the detection of large, medium, and small objects, and, therefore, to be able to maximize the detection accuracy of the object and improve the computational efficiency of the model with less parameter counts and computational amount.



(**a**) (**b**)

Figure 3. (**a**) depicts the original detection head configuration, and (**b**) outlines the general structure of the LSCD module, green indicates shared convolution, and P2, P3, and P4 denote specific layers within the neck.

3.3. Urinary Sediment Dataset

The MS COCO dataset is widely recognized in deep learning, contributing significantly to object detection and segmentation. However, its large size presents challenges, and its effectiveness diminishes in scenarios with limited resources. Additionally, the uneven distribution of images across categories limits its applicability to recognizing a wide range of categories. Therefore, this paper conducts experiments on a specific urinary sediment dataset from the Hefei Institute of Physical Sciences of the Chinese Academy of Sciences.

In clinical medicine, disease diagnosis often relies on urine sediment tests to identify conditions such as hepatobiliary diseases, hematological disorders, renal tubular diseases, and others. Detection of urine sediment components offers insights into potential diseases and guides treatment with the physical condition of the patient. Earlier, healthcare professionals relied solely on experience for urine sediment analysis, which was inefficient. Sun et al. [33] addressed this by using aggregated channel features to detect urinary sediment complexities. Liang et al. produced [34] a USE dataset containing 5377 images and 42,759 instances. However, Tuncer et al. [35] analyzed the urine deposits of some patients in the hospital, and the lack of data caused a certain degree of limitation to the study of deep learning, which in turn affects the detection of the models. The later dataset was photographed with a microscopic imager of urinary sediment, and the relevant images of urinary sediment were obtained from the Intelligent Machine Research for Physical Science Research in Hefei, Chinese Academy of Sciences.

The FCUS (Formed Components in Urine Sediment) dataset was finally produced, which covers 36 formed components of urinary sediment with 30,754 images and 148,163 label instances. Training challenges include overlapping objects, poor image quality, and non-object backgrounds, especially the serious imbalance of instances, which will significantly impact detection results. Figure 4 shows that over one-third of categories exceed 7000 labels, with cocryst at 28,207 instances, whereas 17 images of cholesterol crystals include 20 labels, and 9 images of talcum powder particles have only 11 labels. This paper excludes categories with fewer than 100 instances and focuses on 22 categories from the urine sediment dataset named FCUS22 for object detection. The FCUS22 urinary sediment dataset comprises 28,484 images and 125,170 labels, serving as training and testing data for deep learning networks. Figure 5 illustrates the specific morphology and cell names; it mainly displays eight main types: eryth, leuko, crystal, epithelial, cast, sperm, yeast, and bact, detailing label counts and color differences from the background in Figure 6. Through the bar chart, we can find that the number of bleuko and epith instances are 8372 and 9882, respectively, while the number of labeled instances of eryth, leuko, and cocryst is more than 20,000, and more seriously, the number of labeled instances of mapcryst and wacast is less than 200, which fully demonstrates that there is a large disparity in the number of cells in different categories. In the object detection task, the average value of the color difference of all instances of each category is taken as the color difference value of the category, and the closer the color difference between the object category and the background, the higher the color similarity between the category and the background that is indicated, and it is difficult for the model to judge the category and the background accurately in practical applications. From the Figure 6 line graph, it can be seen that the cryst class of the cell color difference is large, in which lcryst can reach 224.04. The sperm and bact color difference is small at only 33.88 and 23.43, and in the practical application it is not easy to be identified. At the same time, calculating the relative proportion of the image for each object instance accounted for the whole image through the relative proportion of the average value of the description of the relative scale of each category. From Figure 7, it can be found that the number of small-scale objects is large, and the relative scales of the objects are mainly distributed between 0 and 0.03, of which the largest one is cast, with a relative scale of 0.0276, and there are 2708 object instances; the smallest one is speryth, with a relative scale of 0.002, and there are 1755 object instances.

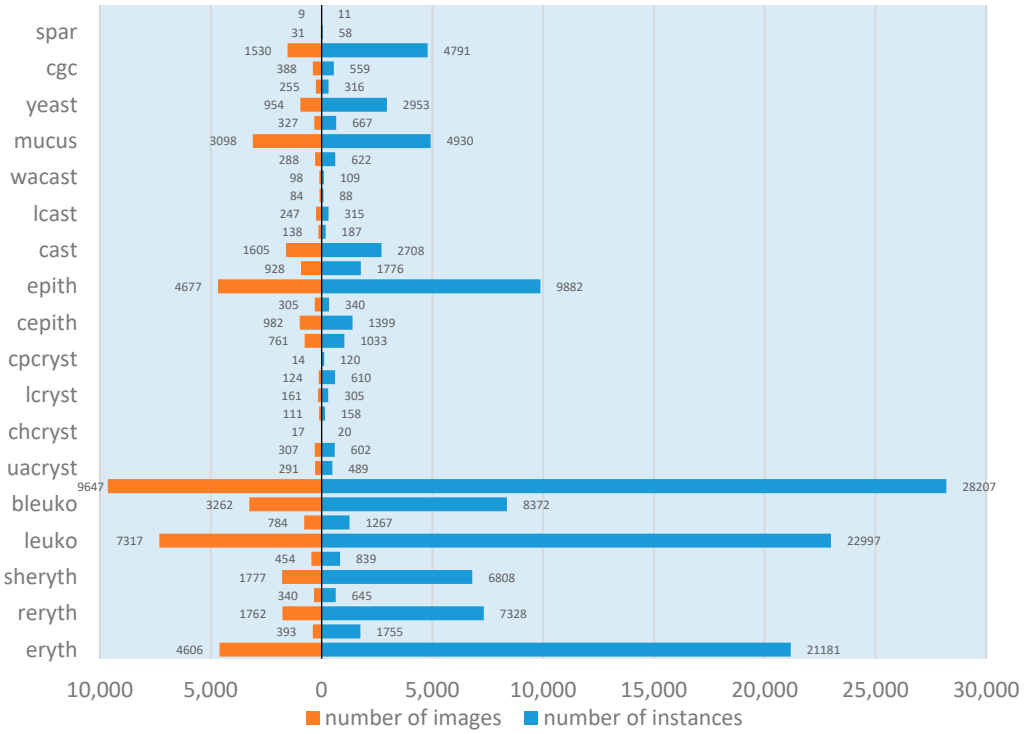


Figure 4. Orange bars indicate the number of images, while blue bars represent the number of instances.

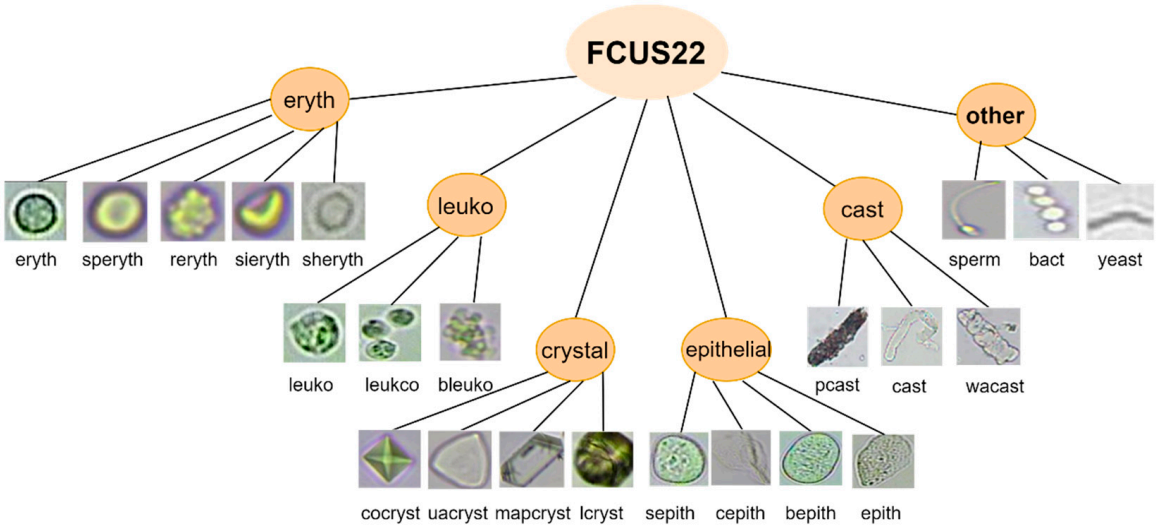


Figure 5. Types of FCUS22 dataset categories.

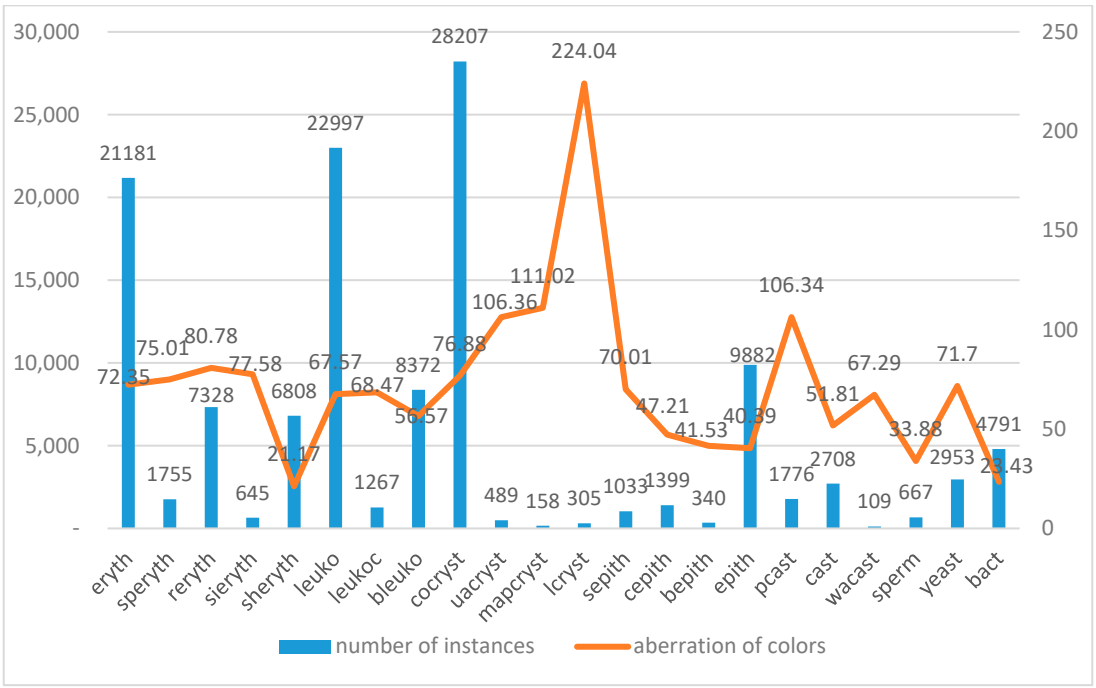


Figure 6. The number of labels and the color difference of corresponding categories in the FCUS22 dataset, the bar graph represents the number of labels in the 22 categories, while the line graph represents the color difference between the object and the background of the corresponding category.

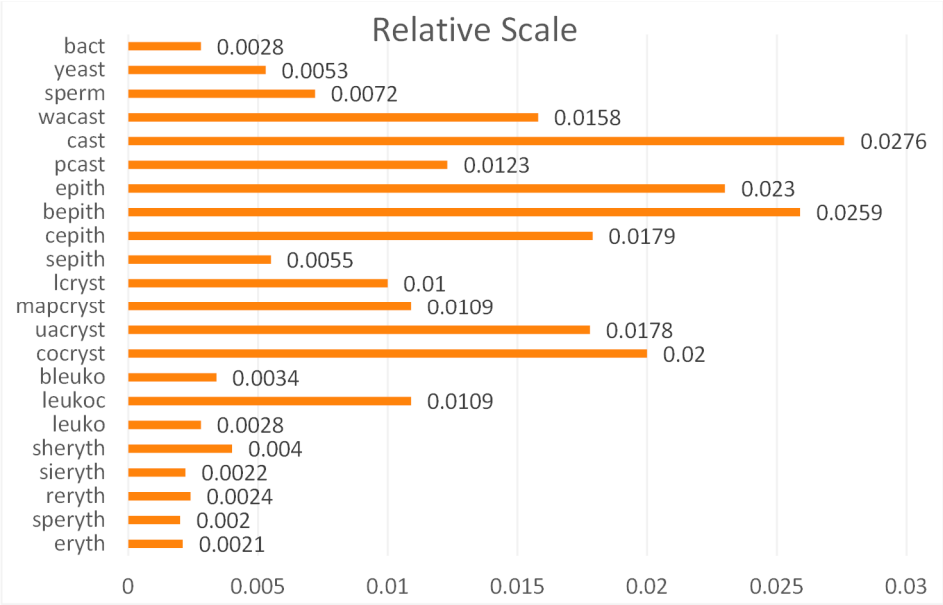


Figure 7. The relative scale of categories in the FCUS22 dataset.

4. Experiment

4.1. Environment and Evaluation of the Experiment

All experiments were conducted on a DELL PowerEdge 640 server featuring a GeForce RTX 3090 GPU with 24 GB of video memory. The software environment included Ubuntu 20.04, CUDA 11.3, Python 3.9, and PyTorch 1.10.2. The models were optimized using the SGD optimizer with a batch size of 32, an initial learning rate of 0.01, a learning rate decay of 0.0005, and an input image size set to 640 × 640 pixels. This paper explores improvements

to related methods using the MS COCO 2017 dataset, with further validation on the Urinary Sediment dataset.

4.2. Evaluation of Indicators

Performance evaluation metrics in object detection encompass four perspectives: accuracy, speed, memory footprint, and model size. The accuracy of the classification and detection of the model on the test set is quantified through metrics such as precision, recall, and F1 score. Speed can be judged by calculating the inference time of the model, and memory footprint refers to the amount of memory required by the model for inference; model size is typically assessed by the number of parameters and the size of the model file.

This paper adopts MS COCO evaluation metrics including IoU which indicates the degree of overlap between the predicted bounding box and the real bounding box; AP (mean Average Precision with IoU from 0.5 to 0.95), AP_{50} (AP with IoU $= 0.5$), AP_s (the average accuracy of small object), AP_m (the average accuracy of medium object), and AP_l (the average accuracy of large object) are used as the evaluation indexes, precision denotes the proportion of samples predicted by the model to be in the positive category that are actually in the positive category, as shown in Equation (5), FP denotes a false positive example, meaning that a negative example in the sample is incorrectly recognized as a positive example, and FN denotes a false negative example, meaning that a positive example in the sample is incorrectly recognized as a negative example. Additionally, there is Recall (proportion of true positive samples correctly categorized), Params (number of model parameters), and GFLOPs (algorithmic complexity).

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

4.3. Experimental Results and Analysis

4.3.1. Comparison with Other SOTA Methods

This paper evaluates the FL-YOLOv8 model on the MS COCO dataset against other state-of-the-art object detectors. Each detector is compared using backbone networks of similar size and capacity. The RetinaNet, TridentNet, FCOS, YOLOF, and Deformable-DETR all utilize ResNet50 as the backbone network; YOLOv3, YOLOv5, and YOLOv8 all employ S-level training models, while EfficientDet-D1 adopts the backbone network EfficientNet-B1. As shown in Table 2, the FL-YOLOv8 object detector achieves the highest accuracy of 45.5% with only 9.6 million parameters. The comprehensive model evaluation demonstrates its efficacy in image processing.

Table 2. Comparison with SOTA method on MS COCO dataset.

Method	#Params	GFLOPs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
YOLOv3	61 M	58.9	31.8	55.3	31.8	14.2	34.1	46.4
RetinaNet	38 M	201	35.9	55.7	38.5	18.9	39.5	48.2
TridentNet	33 M	90	36.6	57.3	39.5	18.3	41.4	52.3
FCOS	32 M	179	38.6	57.4	41.4	22.3	42.5	49.8
YOLOv5	7.2 M	16.4	37.4	58.0	40.4	21.9	43.0	48.8
YOLOF	44 M	86	37.2	56.4	40.1	18.4	42.1	52.3
EfficientDet-D1	6.6 M	6.1	39.6	58.6	42.3	17.9	44.3	56.0
PP-YOLOE	40 M	173	43.9	62.8	47.8	26.1	47.4	58.0
YOLOv8	11.2 M	28.6	44.7	61.6	48.5	25.8	49.8	61.2
FL-YOLOv8	9.6 M	27.2	45.5	62.0	49.6	26.6	50.1	60.8

4.3.2. Ablation Experiments

Currently, the YOLOv8 model stands as an advanced technology in object detection, widely recognized and influential. In this study, we adopt S as the benchmark detector model and evaluate its performance on the MS COCO 2017 dataset, achieving an average

accuracy of 44.7%. Furthermore, we demonstrate the efficacy of enhanced methods in the domain by testing the urine sediment dataset, Table 3 displays the modifications in the model resulting from the different methods. Initially, the semantic and detail injection module significantly enhances feature fusion, which makes the FL-YOLOv8 model slightly improve object detection under different thresholds. Introducing the lightweight shared convolutional detector head reduces model parameters by 19% and computation by 10%. The implementation of group normalization in the LSCD module enabled the model to decrease computation while improving accuracy. When both methods were combined, the result was a reduction in the number of parameters and computations, along with an improved model with an AP of 45.5%.

Table 3. Ablation experiments on the MS COCO dataset.

Method	#Params	GFLOPs	*AP*	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Baseline	11.2 M	28.6	44.7	61.6	48.5	25.8	49.8	61.2
+FSDI	11.4 M	29.8	45.6	62.2	49.4	27.8	50.8	61.3
+LSCD	9.4 M	25.9	44.5	61.3	48.3	26.2	49.6	60.1
FL-YOLOv8	9.6 M	27.2	45.5	62.0	49.6	26.6	50.1	60.8

4.3.3. Comparison Experiments in the LSCD Module

The optimized design of the detection head part of the different scales of the object is in the form of lightweight shared convolution. The feasibility of the module is illustrated through the comparison experiments of varying levels of the YOLOv8 model, as shown in Table 4. When the LSCD structure is added to the YOLOv8-N model, the overall parameter counts and computation amount of the model decrease by 25% and 24%, respectively. Comparison experiments on the YOLOv8-M model demonstrated a notable reduction in the number of parameters and computations. For the YOLOv8-S model, the ablation experiment shows almost no change in accuracy, but the number of parameters decreases by 19%, and the amount of computation decreases by 10%. The size of the model will lead to different parameter counts and computation of the model due to the variability in the number of layers. For small models, the contextual information they can provide is limited, especially for anchor-free detection of the object, and the lack of accuracy in the size of the centroid and the predicted offsets affects the shape of the detection frame, whereas a large model with larger parameter counts and computation can extract richer features, and the detection frame can be closer to the boundary of the object, which improves the accuracy of detection and can be applied in other fields.

Table 4. Ablation experiments using LSCD for different detection algorithms.

Method	+LSCD	#Params	GFLOPs	*AP*	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
YOLOv8-N	×	3.2 M	8.7	37.0	52.1	40.2	18.2	40.6	52.3
YOLOv8-N	√	2.4 M	6.6	36.1	51.2	39.3	17.2	39.5	51.0
YOLOv8-S	×	11.2 M	28.6	44.5	61.0	48.5	25.6	49.5	61.0
YOLOv8-S	√	9.4 M	25.9	44.4	61.1	48.2	25.5	49.2	59.8
YOLOv8-M	×	25.9 M	78.9	49.7	66.6	54.1	31.0	55.4	66.7
YOLOv8-M	√	23 M	75.3	49.5	66.4	53.8	32.1	54.6	65.6

4.3.4. Experiments with FSDI Combinations at Different Locations Under the COCO 2017 Dataset

Implementing semantic and detail injection modules at various neck locations significantly enhances the fusion of information features across upper and lower layers, enabling the network to capture diverse layers of specific information. As shown in Table 5 below, firstly, the semantic and information injection module is used for the neck FPN structure, and at the same time, it is connected with the shallow C2F module to obtain richer feature information, and it is found that the comprehensive performance of the model is improved

by 0.9 percentage points through the experiments. While acting on the neck PAN structure using the semantic and information injection module, the effect is slightly improved. This is mainly due to the different means of feature fusion between the path aggregation network and the feature pyramid network in the PAFPN network, which leads to the difference in FSDI detection performance between the [N1] and [N2] layers. Therefore, we choose to perform feature fusion for the information at the location of [N1].

Table 5. Adoption of semantic and information injection modules for different locations.

Input Layer	#Params	GFLOPs	*AP*	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
-	11.2 M	28.6	44.7	61.6	48.5	25.8	49.8	61.2
[N_1]	11.4 M	29.8	45.6	62.2	49.4	27.8	50.8	61.3
[N_2]	11.0 M	28.4	44.8	61.1	48.7	25.7	49.0	61.2

Note: The horizontal line indicates the baseline model approach, [N_1] means that the FSDI structure is applied to the FPN structure of the neck, and [N_2] means that the FSDI structure is applied to the PAN structure of the neck.

4.4. Experimental Comparisons Under the Urine Sediment Dataset

The MS COCO dataset is a standard dataset in deep learning, widely used for tasks such as image segmentation and object detection. Due to the diverse objects and complex scenarios, there are challenges in recognizing and understanding domain-specific scenarios. Further experiments are conducted on the urine sediment dataset (FCUS22), and the number of parameters and computation of the model decreased by 14.3% and 5.3%, respectively, The comprehensive performance decreased slightly mainly because the number of wacast and mapcryst cell instances were 109 and 158, respectively, and the accuracy of eryth, leuko, and cocryst, which have more cells, were improved, and at the same time, the relative scale of bepith cells was larger than 10 times that of sieryth, and the color difference of sheryth and bact cells was only more than 20, only one-tenth that of lcryst. The uneven distribution of data volume, the large difference in size and scale, and the obvious multiplicity of color difference seriously lower the detection results of the model. Table 6 displays the results of the tests performed on the first 11 categories, and Table 7 corresponds to the results of the tests performed on the last 11 categories. The changes in these categories are more significant, and they closely match the real situation of urinary detection. For example, lcryst and mapcryst have more similar cells, making them easy to be mistakenly detected. The cast and wacast have cells that are too close to each other in color, which can only be detected by the morphology of the cells. Additionally, the number of instances of the object in the image is relatively small, leading to insufficiently rich feature extraction and affecting the detection effect. Whereas the bepith cells are relatively large in scale, their detection accuracy is improved by 1.8 percentage points. A different number of categories were selected for testing on the COCO dataset as shown in Table 8 and analyzed in comparison with the results in Tables 6 and 7. It can be found that when the categories correspond to a higher number of images, it can provide more data sources for the model pretraining, which can improve the detection results of the model, and on the contrary, it will affect the overall performance of the model, and it also shows that the FL-YOLOv8 model has a significant improvement in small object detection on both the MSCOCO and FCUS22 datasets.

Table 6. Comparison of accuracy for the first 11 categories.

Method	Reryth	Sieryth	Leukoc	Uacryst	Lcryst	Sepith	Cocryst	Cast	Sperm	Bact	Wacast
YOLOv8	37.8	23.7	29.4	59.9	33.5	40.7	43.5	23.5	35.2	22.9	12
Ours	37.8	23.1	29.6	57.3	31.4	42.5	43.6	22.9	36.8	22.9	6.3
vs	0	−0.6	**+0.2**	−1.9	−2.1	**+1.8**	**+0.1**	−0.6	**+1.6**	0	−5.7

Table 7. Accuracy comparison of the last 11 categories.

Method	Eryth	Speryth	Bleuko	Sheryth	Leuko	Bepith	Mapcryst	Epith	Cepith	Yeast	Pcast
YOLOv8	43	27	35.1	25.5	50.2	25.4	26	63.7	34.2	55.9	39.1
Ours	43.5	27.7	34.1	24.8	49.7	27.2	22.8	63.2	32.5	54.5	39.1
vs	**+0.5**	**+0.7**	−1.0	−0.7	−0.5	**+1.8**	−3.2	−0.5	−1.7	−0.6	0

Note: vs denotes the comparison of YOLOv8 to FL-YOLOv8, and bold denotes the corresponding category.

Table 8. Detection results for selected MSCOCO categories.

Class	Person	Car	Chair	Hot Dog	Bear	Zebra	Bird	Bottle	Book
Instances	10,777	1918	1771	125	71	266	427	1013	1129
YOLOv8	57.3	43.8	33.3	40.1	75.6	72.7	33.9	38.4	13.5
Ours	57.8	44.9	33.9	39.4	75.2	72.0	35.1	39.7	14.0
vs	0.5	+1.1	+0.6	−0.6	−0.4	−0.7	+1.2	+1.3	+0.5

Note: In the categories, the first three categories represent the categories with a higher number of instances, the middle three categories represent the categories with a lower number of instances, and the last three categories represent the detection results in the small-scale categories.

The detection results of each category are analyzed through Tables 6 and 7. For the first three categories with poor detection results, which are lcryst, mapcryst, and wacast, it is found that most of the images in the crystal category have the aggregation phenomenon after dividing and detecting the images, as shown in Figure 8 below. After using both the conventional model and the FL-YOLOv8 model, lcryst cells in the first column were not clearly detected, the detection result of mapcryst in the second column decreased from 0.67 to 0.64, and the detection of tubular cells in the third column erroneously detected wacast as cast cells. The above poor results are mainly because the object aggregation makes it difficult to locate the object, decreases the detection effect of the object, and confuses the categories, with detection of small-scale direction more difficult, which makes it easy to miss the detection situation. Therefore, object aggregation may reduce the performance of the object detection model in accurately identifying and localizing object and increase the difficulty and complexity of the task. As for wacast cells, the number of images of their object is too small, which will lead to insufficient feature extraction and affect the detection effect of this category. Given the complexity and specificity of the urinary sediment dataset and the need to consider the cell aggregation situation, the coexistence and balance of the two is also a direction worthy of further research.

However, for the detection of epithelial cells as well as sperm cells with lower color difference, some improvement can be achieved on the FL-YOLOv8 model, as shown in Figure 9, in contrast to the crystalline, sepith, bepith, and sperm cells which are uniformly distributed, and it can be found through the results of the heatmap detection that the results of the detection are relatively dispersed in the sampling area using the conventional YOLOv8, but adopting the FL-YOLOv8 model showed better results in relatively concentrated areas. The model can detect the image better by adopting the heatmap, enhancing the localization ability to capture the object area accurately, adopting the object centrally, and retaining the relevant semantic information to a greater extent, which indicates that the improved FL-YOLOv8 model not only enriches the spatial sampling information of the feature map but also strengthens the recognition ability of the feature map. This result is of great significance for object detection tasks.

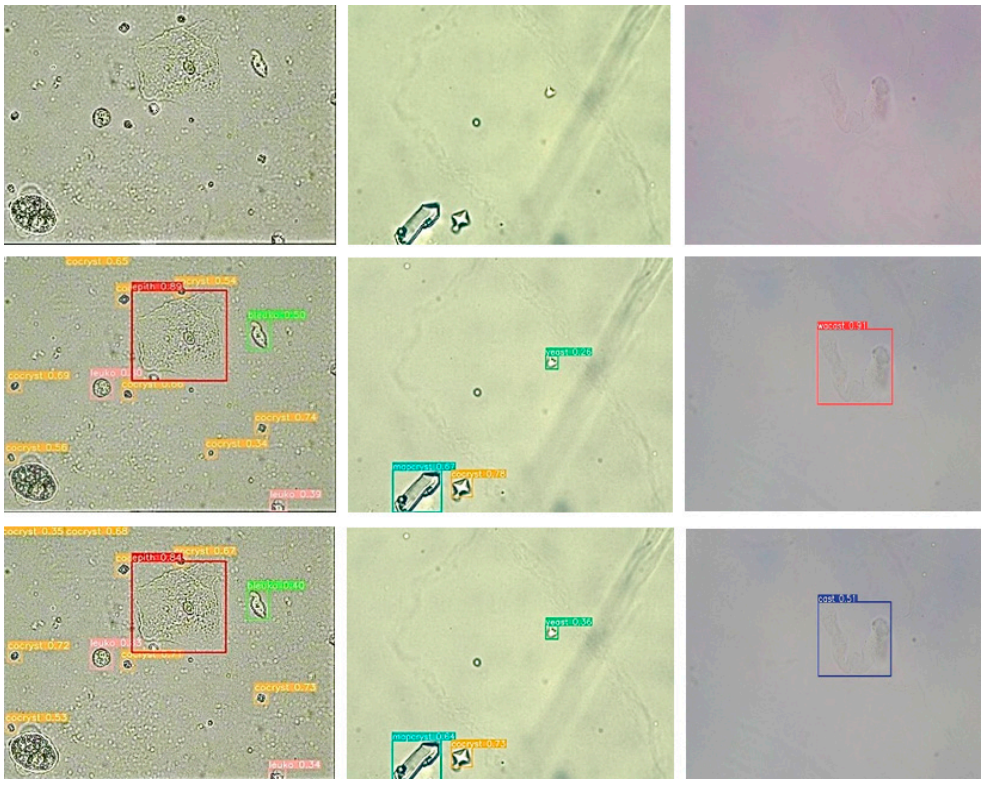


Figure 8. The first row represents the original lcryst, mapcryst, and wacast class of cells, respectively, and the second and third rows represent the results of YOLOv8 and FL-YOLOv8 detection of the classes.

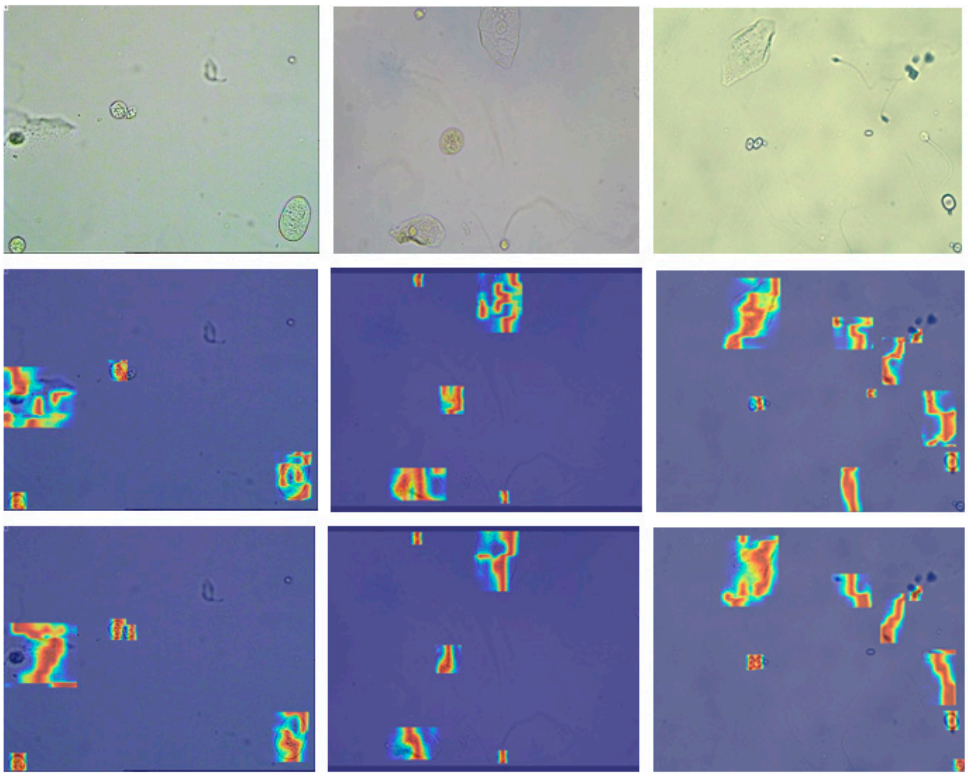
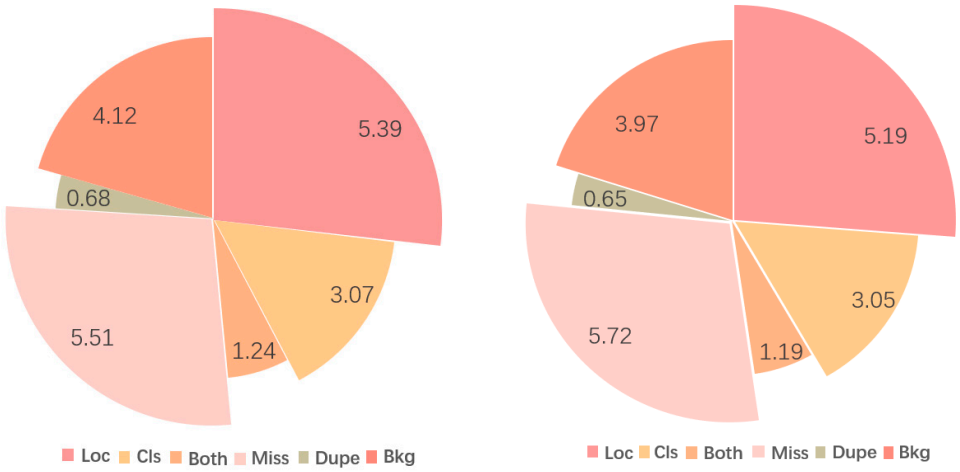


Figure 9. The first row represents bepith, sepith, and sperm in the original figure, respectively, and the second and third rows represent the heatmap detection results of the corresponding YOLOv8 and FL-YOLOv8 algorithms.

4.5. TIDE Error Analysis

Performance evaluation of the model may only focus on the average accuracy. Still, some industrial and medical fields need to understand the comprehensive detection results, which cannot be found from the mAP, to see its specific errors. To analyze better the contribution of each type of error to the model, we use the TIDE error analysis method from the study [36], as shown in Figure 10. The fan chart clearly shows the contribution of the six types of errors. Starting from noon and going clockwise in the figure, there is Loc: mislocalization, Cls: classification error, Both: both classification and localization error, Miss: omission error, Dupe: repeated prediction error, and Bkg: background error.



(**a**) (**b**)

Figure 10. (**a**) shows the error analysis of YOLOv8; (**b**) shows the error analysis of FL-YOLOv8.

According to the TIDE error analysis method, there is no significant change in the classification and repeated prediction errors. Still, the percentage of localization errors and background errors in the model is reduced, which indicates that the FL-YOLOv8 model can significantly improve the localization ability for object detection and can also better distinguish the object from the background based on the extracted feature information, mainly because the layer-hopping connection through the FSDI module will be used to connect to the high-level. The main reason is that the fusion of high-level and low-level information through the hopping connection of the FSDI module reduces the localization error. Using shared convolution to group the features can better distinguish the background information. It also means that the model has more accurate results in object localization and background prediction, and our method achieves a reduction of 0.48 in FP, which further proves that the method improves the feature extraction of the model while lightening the shared convolution.

5. Conclusions

This paper proposes the FL-YOLOv8 object detector by enhancing the YOLOv8-S model to address challenges in object detection without increasing computational overhead. To enhance feature extraction and capture richer semantic and detailed information, this paper introduces the FSDI model, significantly boosting model precision and accuracy. At the same time, adopting the LSCD model with a lightweight shared convolutional detector head reduces parameters and computational load, further enhancing model efficiency. The YOLOv8 object detector demonstrates improved accuracy and efficiency over previous models.

Since FL-YOLOv8 is a lightweight model structure and the FCUS22 dataset is smaller than the common dataset, the accuracy and computation are not well balanced in the urine sediment dataset. Through the study of the sediment dataset, it is found that there are cells

with similar color, shape, and internal structure in its data, such as eryth and reryth cells, and the object detection also found that there are a large number of aggregation phenomena of the object, like lcryst and mapcryst, which brings a certain degree of difficulties to the object detection. At present, there is no better way to improve the effect of detection; this series of problems has not been effectively solved. In the future, the urine sediment data can be filtered, or deep learning methods can be combined with other fields to make up for the shortcomings of the urine sediment data, to yield more accurate localization and detection, and we hope that the urine sediment dataset can help and contribute to object detection and the medical field. It will play a greater value in the field of deep learning.

The FL-YOLOv8 model has a significant effect in detection of objects with small scales, while the number of parameters and computation are reduced by 19% and 10%, respectively, which can provide patients with specific diagnostic analyses of related diseases and promote the research of urinary sediment datasets in the medical field; in addition, we can enhance the detection performance of the model by constructing rich urinary sediment detection datasets and exploring new methods and techniques.

Author Contributions: Conceptualization, Q.W.; methodology, Y.X. and Y.H.; writing—original draft preparation, Y.X.; writing—review and editing, Q.W.; data collection, Y.X., Y.Q., L.C. and H.W.; analysis and interpretation of results, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Anhui Province Quality Engineering Project (No.2014zytz035, No. 2021sqyrgx01) and the Academic funding project for top talents of disciplines in colleges and universities of Anhui Province (No. gxbjZD2020096).

Data Availability Statement: The urinary sediment dataset can be researched by contacting the authors directly.

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

References

1. Guerrero-Ibañez, J.; Contreras-Castillo, J.; Zeadally, S. Deep learning support for intelligent transportation systems. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4169. [CrossRef]
2. Chen, Y.; Wang, C.; Lou, S. Edge artificial intelligence camera network: An efficient object detection and tracking framework. *J. Electron. Imaging* **2022**, *31*, 033030. [CrossRef]
3. Yang, R.; Yu, Y. Artificial convolutional neural network in object detection and semantic segmentation for medical imaging analysis. *Front. Oncol.* **2021**, *11*, 638182. [CrossRef] [PubMed]
4. Parmar, Y.; Natarajan, S.; Sobha, G. Deeprange: Deep-learning-based object detection and ranging in autonomous driving. *IET Intell. Transp. Syst.* **2019**, *13*, 1256–1264. [CrossRef]
5. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
6. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully convolutional one-stage object detection. *arXiv* 2019, arXiv:1904.01355.
7. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
8. Wang, C.Y.; Liao HY, M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
9. Farhadi, A.; Redmon, J. Yolov3: An incremental improvement. In *Computer Vision and Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 1804, pp. 1–6.
10. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
11. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
12. Peng, Y.; Sonka, M.; Chen, D.Z. U-Net v2: Rethinking the skip connections of U-Net for medical image segmentation. *arXiv*, 2023; arXiv:2311.17791.
13. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9197–9206.

14. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; IEEE Computer Society: Washington, DC, USA, 2021; pp. 3490–3499.
15. Lv, Y.; Li, M.; He, Y.; Li, S.; He, Z.; Yang, A. Anchor-intermediate detector: Decoupling and coupling bounding boxes for accurate object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 6275–6284.
16. Lu, Y.; Lu, G.; Zhou, Y.; Li, J.; Xu, Y.; Zhang, D. Highly shared convolutional neural networks. *Expert Syst. Appl.* **2021**, *175*, 114782. [CrossRef]
17. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
18. Zhang, Q.; Yang, Y.; Cheng, Y.; Wang, G.; Ding, W.; Wu, W.; Pelusi, D. Information fusion for multi-scale data: Survey and challenges. *Inf. Fusion* **2023**, *100*, 101954. [CrossRef]
19. Xu, Q.; Kuang, W.; Zhang, Z.; Bao, X.; Chen, H.; Duan, W. Sppnet: A single-point prompt network for nuclei image segmentation. In Proceedings of the International Workshop on Machine Learning in Medical Imaging, Vancouver, BC, Canada, 8–12 October 2023; Springer Nature: Cham, Switzerland, 2023; pp. 227–236.
20. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Part I 14; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
22. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Wang, C.Y.; Bochkovskiy, A.; Liao HY, M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17 June–24 June 2023; pp. 7464–7475.
25. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
26. Howard, A.G. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017; arXiv:1704.04861.
27. Iandola, F.N. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* 2016, arXiv:1602.07360.
28. Kenton JD MW, C.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, p. 2.
29. Hinton, G. Distilling the Knowledge in a Neural Network. *arXiv* 2015, arXiv:1503.02531.
30. Bochkovskiy, A.; Wang, C.Y.; Liao HY, M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* 2020, arXiv:2004.10934.
31. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* 2022, arXiv:2209.02976.
32. Ge, Z. Yolox: Exceeding yolo series in 2021. *arXiv* 2021, arXiv:2107.08430.
33. Sun, Q.; Yang, S.; Sun, C.; Yang, W. Exploiting aggregate channel features for urine sediment detection. *Multimed. Tools Appl.* **2019**, *78*, 23883–23895. [CrossRef]
34. Liang, Y.; Tang, Z.; Yan, M.; Liu, J. Object detection based on deep learning for urine sediment examination. *Biocybern. Biomed. Eng.* **2018**, *38*, 661–670. [CrossRef]
35. Tuncer, T.; Erkuş, M.; Çınar, A.; Ayyıldız, H.; Tuncer, S.A. Urine Dataset having eight particles classes. *arXiv*, 2023; arXiv:2302.09312.
36. Bolya, D.; Foley, S.; Hays, J.; Hoffman, J. Tide: A general toolbox for identifying object detection errors. In *Computer Vision–ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Part III 16; Springer International Publishing: Cham, Switzerland, 2020; pp. 558–573.