



# SDE-YOLO: A Novel Method for Blood Cell Detection

Yonglin Wu <sup>1</sup>, Dongxu Gao <sup>2,\*</sup>, Yinfeng Fang <sup>3</sup> , Xue Xu <sup>4</sup>, Hongwei Gao <sup>1</sup> and Zhaojie Ju <sup>2,\*</sup>

<sup>1</sup> School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110158, China; wuyonglin2000@sohu.com (Y.W.); ghw1978@sohu.com (H.G.)

<sup>2</sup> School of Computing, University of Portsmouth, Portsmouth PO13HE, UK

<sup>3</sup> School of Telecommunication Engineering, Hangzhou Dianzi University, Hangzhou 311305, China; yinfeng.fang@hdu.edu.cn

<sup>4</sup> China Tobacco Zhejiang Industrial Co., Ltd., Hangzhou 311500, China; xuxuehdck@sina.com

\* Correspondence: dongxu.gao@port.ac.uk (D.G.); zhaojie.ju@port.ac.uk (Z.J.)

**Abstract:** This paper proposes an improved target detection algorithm, SDE-YOLO, based on the YOLOv5s framework, to address the low detection accuracy, misdetection, and leakage in blood cell detection caused by existing single-stage and two-stage detection algorithms. Initially, the Swin Transformer is integrated into the back-end of the backbone to extract the features in a better way. Then, the  $32 \times 32$  network layer in the path-aggregation network (PANet) is removed to decrease the number of parameters in the network while increasing its accuracy in detecting small targets. Moreover, PANet substitutes traditional convolution with depth-separable convolution to accurately recognize small targets while maintaining a fast speed. Finally, replacing the complete intersection over union (CIoU) loss function with the Euclidean intersection over union (EIoU) loss function can help address the imbalance of positive and negative samples and speed up the convergence rate. The SDE-YOLO algorithm achieves a mAP of 99.5%, 95.3%, and 93.3% on the BCCD blood cell dataset for white blood cells, red blood cells, and platelets, respectively, which is an improvement over other single-stage and two-stage algorithms such as SSD, YOLOv4, and YOLOv5s. The experiment yields excellent results, and the algorithm detects blood cells very well. The SDE-YOLO algorithm also has advantages in accuracy and real-time blood cell detection performance compared to the YOLOv7 and YOLOv8 technologies.



**Citation:** Wu, Y.; Gao, D.; Fang, Y.; Xu, X.; Gao, H.; Ju, Z. SDE-YOLO: A Novel Method for Blood Cell Detection. *Biomimetics* **2023**, *8*, 404. <https://doi.org/10.3390/biomimetics8050404>

Academic Editors: Chaoran Cui and Xiaohui Han

Received: 30 July 2023

Revised: 24 August 2023

Accepted: 30 August 2023

Published: 1 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** blood cell testing; Swin Transformer; PAN; depth-separable convolution; EIoU

## 1. Introduction

Blood cell testing is a technique that analyses blood's cellular components using specialist instruments. In medicine, the cells circulating in the blood are divided into three categories: white blood cells, red blood cells, and platelets [1]. When the results for the cell counts of these three types are abnormally high or low, the body may already suffer from certain diseases. The accurate and efficient identification of these three types of cells can provide a patient's overall health status [2].

Previously, blood cell testing mainly used traditional image-processing techniques, most classically, manual photography and staining, followed by manual observation and classification with a microscope [3], which required the testers to have an extensive fundamental knowledge of cell morphology and to be skilled in morphological observation and discrimination through continuous study and repeated practice.

Due to the diversity of microscopic image data, traditional image-processing algorithms [4,5] can no longer meet the complex problems researchers encounter in cell detection and counting because of their low detection efficiency and low accuracy. Machine-learning algorithms for blood cell detection are later proposed [6–11]. Machine learning amalgamates fields like computer science, statistics, probability theory, etc. A mixture of computers and biological neural network structures enables computers to learn from vast data to

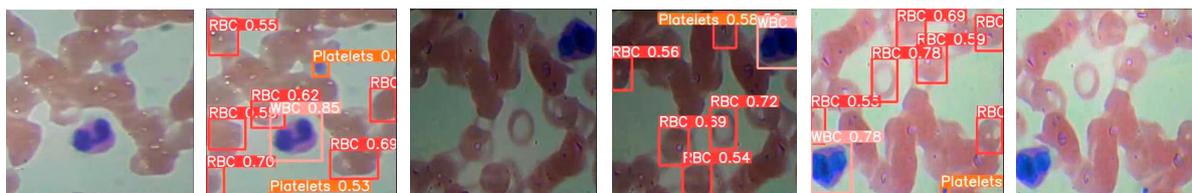
imitate or execute human learning actions [12]. Nevertheless, enhancing the precision of present machine-learning algorithms, which have notable operational restrictions, is challenging.

The emergence of deep learning [13–18] has enabled the possibility of automatic and precise cell detection. The existing target recognition algorithms based on deep neural networks can be divided into single-stage [19] and two-stage [20].

The two-stage algorithms are represented by the RCNN [21], Faster R-CNN [22], SPP-Net [23], and FPN [24], which extract the region of interest (ROI) through the region proposal network (RPN), and then classify the region of interest. The two-stage algorithms implementing convolutional networks instead of fully connected layers reduce the number of unnecessary calculations inherent to the traditional sliding window approach. The two-stage algorithms need to be revised due to their inability to detect small targets and the high rate of mistakes.

The single-stage algorithms are represented by SSD [25] and YOLO [26–28], which realize end-to-end detection by considering localization and classification as a regression problem. The single-stage algorithms have fast detection speeds but low accuracy.

The detection results of blood cells under single-stage and two-stage algorithms, depicted in Figure 1, show that algorithms such as Faster R-CNN, SSD, and YOLO are prone to misdetection and omission.



**Figure 1.** Comparison of blood cells under single-stage and two-stage algorithms.

Recently, some works [29–31] have begun to use the idea of an attention mechanism to improve the performance of computer vision tasks, such as image classification [32,33], object detection [34], and image segmentation [35]. Attention mechanisms in neural networks can divide into three types: channel attention mechanism [36], spatial attention mechanism [37], and mixed attention mechanism [38]. The channel attention mechanism focuses on more distinctive information by focusing on the channel relationship. In contrast, the spatial attention mechanism has advantages in extracting more detailed information, focusing on different spatial locations of feature images. The mixed attention mechanism combines the above two mechanisms and can simultaneously extract detailed and more distinctive information about feature species. Nevertheless, the attention mechanism usually requires more parameters. If the model is already complicated, adding the attention mechanism can result in overfitting and, thus, decrease the model's effectiveness on the test set.

In summary, in this paper, we incorporate the Swin Transformer attention mechanism into YOLOv5s, which we call SDE-YOLO, and SDE-YOLO can solve the problems of insufficient detection accuracy and misdetection and omission in both single-stage and two-stage algorithms, as well as the issue of parameter computation in the attention mechanism. The innovation of this paper is as follows:

- Fusing the Swin Transformer [39] with the end of the backbone network can improve the feature extraction of small targets;
- Deleting the  $32 \times 32$  down-sampling layer in PAN can reduce the number of network parameters while improving the detection rate;
- Replacing ordinary convolution with depth-separable convolution [40] in PANet can increase the detection of small targets while ensuring accuracy;
- Replacing the original CIOU loss function with the EIOU [41] loss function can better solve the problems of positive and negative sample imbalance while converging faster.

## 2. Related Work

### 2.1. YOLOv5s Model

YOLOv5 is divided into four models, s, m, l, and x, according to the depth and width [42], and, since blood cells are primarily small- and medium-sized targets, YOLOv5s is chosen as the basic framework for this paper. As shown in Figure 2, the network architecture of YOLOv5s is divided into four parts in total, as follows: Input, Backbone, Neck, and Prediction.

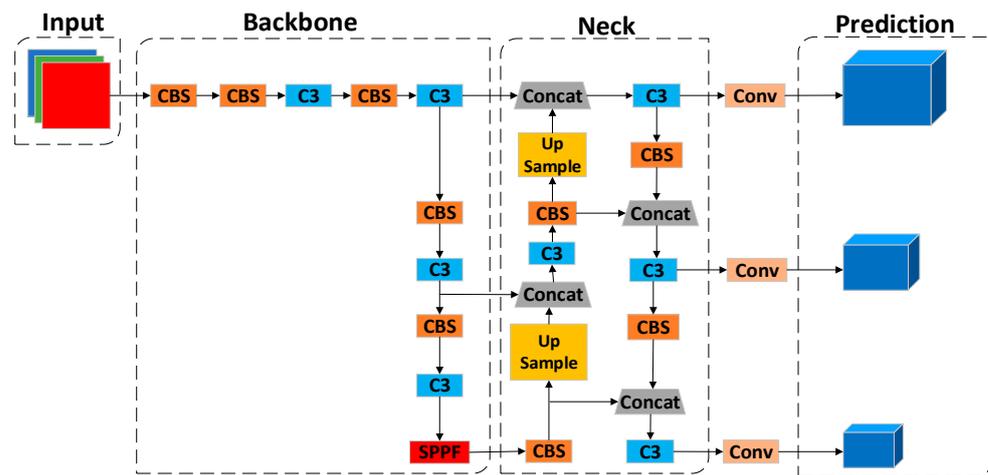


Figure 2. YOLOv5s network architecture.

#### 2.1.1. Input

On the Input side, different sizes of input samples are scaled to a fixed size by adaptive image and then fed into the Backbone networks for training. The data enhancement of YOLOv5s adopts Mosaic data enhancement to solve the problems of insufficient samples and uneven distribution in the dataset. Mosaic data enhancement splices four images by random cropping, random brightness adjustment, and flipping, which reduces GPU use and enhances the network’s robustness.

#### 2.1.2. Backbone

The Backbone networks of YOLOv5s mainly consist of the CS, SPPF, and CBS modules, as shown in Figure 3a. The current Backbone networks in the field of target detection are generally ResNet [43], AmoebaNet [44], Xception [45], and other networks.

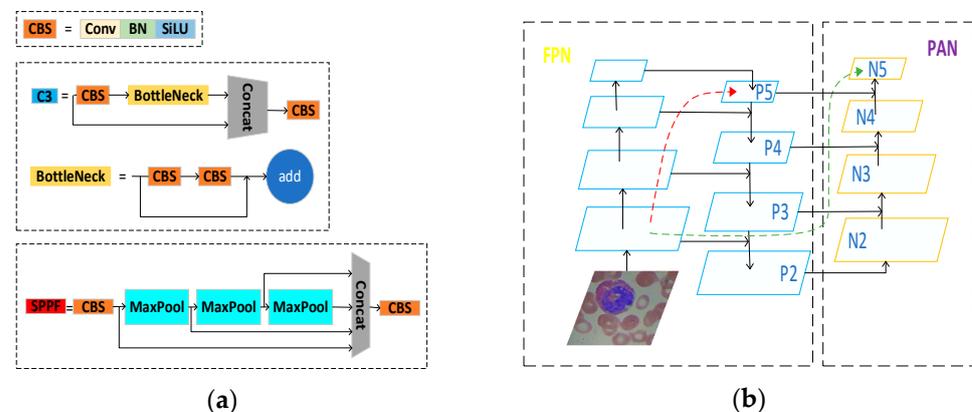


Figure 3. (a) Represents the main modules of YOLOv5s backbone network; (b) Representing the structure of the neck network.

### 2.1.3. Neck

The Neck parts of YOLOv5s adopt the structure of the feature pyramid network (FPN) [46] + the path aggregation network (PAN) [47], as shown in Figure 3b. The FPN structure passes the more vital semantic information from the more profound feature map to the shallower layer by up-sampling. The PAN structure transmits the location information from the superficial layer to the deep layer by down-sampling and performs multi-scale feature fusion.

### 2.1.4. Prediction

The head of YOLOv5s uses three  $1 \times 1$  convolutional layers to replace the fully connected layer (FC) for prediction and classification. It generates three different scales of feature maps, namely,  $20 \times 20$ ,  $40 \times 40$ , and  $80 \times 80$ , and the outputs are  $4 + 1 + C$ , which are the co-ordinates of the prediction frames, the target confidence, and the probability of the category values, respectively.

### 2.1.5. Loss

The loss function mainly comprises the confidence loss  $L_{obj}$ , classification loss  $L_{cls}$ , and position loss  $L_{obj}$  of the target and predicted frames. The confidence loss and classification loss are obtained from the cross-entropy loss function, which is shown in Equation (1). The metric above is computed using the CIOU loss function, as in Equation (2). The total loss function is the sum of the three loss functions, as shown in Equation (3).

$$L_{cls}(p, y) = -\frac{1}{N} \sum_{i=1}^N [y \log p + (1 - y) \log(1 - p)] \quad (1)$$

$$\begin{cases} L_{obj} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \\ v = \frac{4}{\pi^2} \left( \arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right) \\ \alpha = \frac{v}{1 - IOU(b, b^{gt}) + v} \end{cases} \quad (2)$$

$$L_{total} = L_{obj} + L_{cls} + L_{box} \quad (3)$$

## 3. Proposed Method

### 3.1. Integration of the Swin Transformer Module

The Swin Transformer is based on the transformer with a global information modelling capability to construct hierarchical feature maps while drawing on the idea of the locality to limit the self-attention computation to the non-overlapping window area and allow moving windows for feature interaction, with linear computational complexity for the image size, which can be used as the backbone network of YOLOv5s to extract image features more effectively.

As shown in Figure 4, the Swin Transformer model builds four Swin Transformer block groups {C1, C2, C3, and C4} according to the number of Swin Transformer blocks [2,2,2,6], and the down-sampling multiplicity is [4,8,16,32], respectively. For the input image (H, W, 3), the patch partition layer divides it into  $4 \times 4$  patches and converts the dimensionality of the image data into (H/4, W/4, 48). The linear embedding layer transforms the image data, the number of channels doubles, and the dimensions are (H/4, W/4, 96). In C1, the Swin Transformer block is based on the  $7 \times 7$  window to calculate the multi-head self-attention and extract image features, and the output feature map has the exact dimensions of (H/4, W/4, 96). In the subsequent process, each patch merging layer first divides the input feature map into  $2 \times 2$  patches, and then doubles the number of feature map channels to realize the down-sampling of the feature map. The dimension of the feature map transforms to (H/8, W/8, 192), (H/16, W/16, 384), and (H/32, W/32, 768), which is similar to the pooling operation in CNN. The Swin Transformer block groups of C2, C3, and C4 only extract the image data features without changing the feature map dimension.

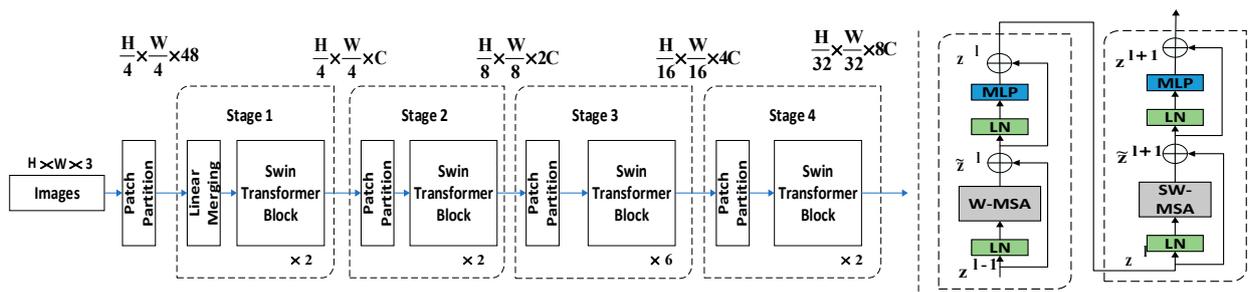


Figure 4. Swin Transformer network architecture.

The Swin Transformer block of the Swin Transformer consists of two modules: the window-based multi-head attention mechanism and mobile window-based multi-head attention mechanism. The information passes through the layer norm layer, window-based multi-head self-attention (W-MSA) unit, multilayer perceptron (MLP) layer, and jump connection of the left module to complete the multi-head self-attention calculation based on the window, and then passes through the Layer Norm layer, shifted window-based multi-head self-attention (SW-MSA) unit, MLP layer, and jump connection of the suitable module to complete the multi-head self-attention calculation based on the mobile window. The two modules have the same input and output information dimensions and are directly connected.

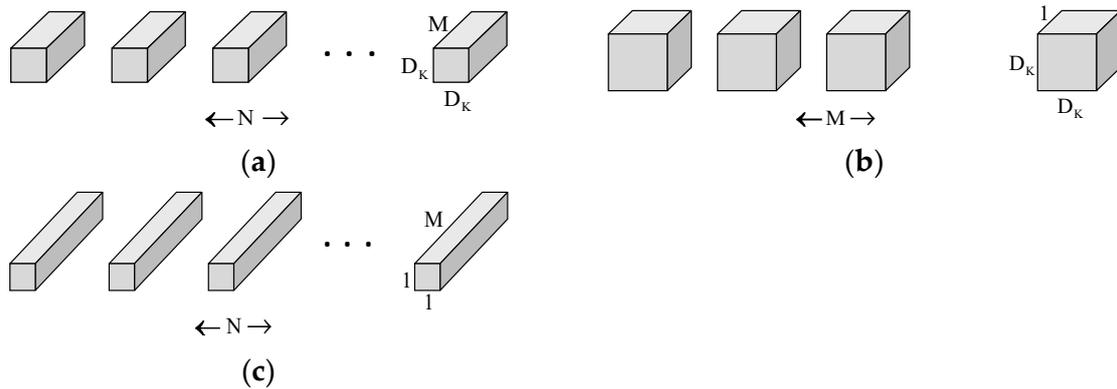
Window-based multi-head self-attention (W-MSA) models the dependency of a local window instead of the global one in multi-head self-attention (MSW), which helps to reduce the complexity of the model computation. W-MSA is similar to conv: it also has two parameters, the kernel size and stride, but W-MSA has the same kernel size and stride in the same stage, which leads to a problem—there is no difference between the receptive field of 1 W-MSA and n W-MSAs in a stage. A patch is applied to the W-MSA to get shifted window-based multi-head self-attention (SW-MSA) to alleviate this problem.

### 3.2. Improved the PANet Network Architecture

From the above, we know that the PANet network outputs predicted results by  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$  times, down-sampling for three small, medium, and large targets. Taking our blood cell BCCD dataset as an example, our size is  $640 \times 640$  when we perform  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$  down-sampling. The output prediction layers are  $80 \times 80$ ,  $40 \times 40$ , and  $20 \times 20$ , respectively. However, since the target sizes of our blood cell BCCD dataset are between  $8 \times 9$  and  $285 \times 442$ , and the small- and medium-size targets take up 95% of the total cells, when the image is down-sampled by  $32 \times$ , the blood cells smaller than  $102 \times 102$  in the image will compress one pixel, which will lead to our inefficiency in small cell detection. In this paper, we have opted not to employ the  $32 \times 32$  down-sampling correlation network layer, which will enable a faster cell detection process while reducing the number of parameters of the network model.

### 3.3. Improved Convolutional Structures in PAN

Deep separable convolution combines deep convolution and dot convolution ( $1 \times 1$  convolution). Deep convolution first performs a separate deep convolution for each input channel, and then mixes the output channels by dot convolution ( $1 \times 1$  convolution). The inputs are subject to filtering and combining operations in a standard convolution. Still, the deep separable convolution splits this operation by subjecting the information to a filtering operation in the first step, and then combining them in the second step. Composing a standard convolution reduces the network model’s computational cost and size. Figure 5 shows how a normal convolution operation decomposes a deep convolution and a dot convolution ( $1 \times 1$  convolution).



**Figure 5.** (a) Representing the filters for standard convolution; (b) Representing the filter for deep convolution; (c) Representing the filter for pointwise convolution.

A feature map  $F$  of size  $D_F \times D_F \times M$  is used as an input to a standard convolution. The output feature map  $G$  is  $D_G \times D_G \times N$ , where  $D_F$  is the spatial width and height of the input feature map  $F$ ,  $M$  is the number of input channels,  $D_G$  is the spatial width and size of the output feature map  $G$ , and  $N$  is the number of output channels.

The computational cost  $C_{conv}$  required for a standard convolutional layer is Equation (4), where the size of the convolution kernel is the square of  $D_K$ .

$$C_{conv} = D_K \times D_K \times M \times N \times D_F \times D_F \tag{4}$$

The computational cost of deep convolution  $C_{dconv}$  is Equation (5).

$$C_{dconv} = D_K \times D_K \times M \times D_F \times D_F \tag{5}$$

The computational cost of deep convolution  $C_{pconv}$  is Equation (6).

$$C_{pconv} = M \times N \times D_F \times D_F \tag{6}$$

A linear combination of depth convolution and point-by-point convolution ( $1 \times 1$  convolution) is called depth-separable convolution. The computational cost of the depth-separable convolution is the sum of the computational cost of the depth convolution and the  $1 \times 1$  point-by-point convolution as in Equation (7).

$$C_{dscnv} = D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \tag{7}$$

Splitting the standard convolution into two steps, filtering and combining, can drastically reduce the amount of parameter computation, the computational cost of the depth-separable convolution, and the computational cost of the standard convolution, two specific parameter pairs, shown in Equation (8).

$$\frac{C_{dscnv}}{C_{conv}} = \frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{8}$$

When we use a deeply separable convolution with a convolution kernel of  $3 \times 3$ , the computational effort is one-eighth to one-ninth that of a standard convolution using the same-size convolution kernel, with essentially no difference in accuracy. By substituting the usual convolution in the PANet network with a deeply separable convolution, the detection of small targets improves, while the accuracy and real-time detection are maintained.

### 3.4. Improved IOU Loss Function

From the classification loss function in Equation (1), it can see that when the predicted probability value ( $p \geq 0.5$ ) is large, the loss value of the positive samples is smaller than

that of the negative samples, which makes it difficult for the positive samples to influence the training parameters, thus affecting the detection accuracy. This paper adopts the EIOU loss function to solve the above problem, which is shown in Equation (9).

$$L_{EIOU} = L_{IOU} + L_{dis} + L_{asp} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \quad (9)$$

In the above equation:  $L_{IOU}$ ,  $L_{dis}$ , and  $L_{asp}$  are the overlap loss, centroid loss, and width-height loss, respectively.  $b$  and  $b^{gt}$  are the prediction box and the real box, respectively.  $\rho(b, b^{gt})$  is the Euclidean distance between the centroids of the two boxes,  $c$  is the diagonal distance of the smallest closure area that contains the two boxes, and  $c_w$  and  $c_h$  are the width and height of the smallest external box that covers the two boxes, respectively.  $w$  is the width of the prediction box, and  $w^{gt}$  is the width of the real box.  $h$  is the height of the prediction box, and  $h^{gt}$  is the height of the real box.

The overlap loss and center distance loss of the EIOU loss use the original method in CIoU. In contrast, the width–height loss directly minimizes the difference between the predicted and actual frames' width and height, making convergence faster. Specifically, this loss function acts as a guide during the optimization process during training, causing the dimensions of the predicted frame to gradually approach the dimensions of the real frame. By incorporating the difference between the width and height of the predicted frame and the width and height of the real frame into the loss function, the model is more strongly constrained during the optimization process to predict the location and size of the target more accurately. This paper uses EIOU as the loss function of the improved model, which can effectively improve the accuracy.

## 4. Experiments

### 4.1. Dataset

This paper makes use of the relatively small BCCD dataset [48]. The BCCD dataset (Blood Cell Count and Detection Dataset) is a dataset open to the public, which is frequently employed in cell image analysis and medical image processing to count and detect blood cells. The primary purpose of this dataset is to facilitate researchers in constructing and assessing algorithms for the identification and enumeration of blood cells, thus enhancing the precision of medical diagnosis and treatment.

The BCCD dataset includes three varieties of blood cells: WBCs (white blood cells), RBCs (red blood cells), and Platelets. The data associated with each image, such as the location and number of cells, serve as a reference point for assessing the algorithm's performance. The dataset consists of 768 images for training and 106 for testing, with the instances of three types of blood cells displayed in Figure 6.

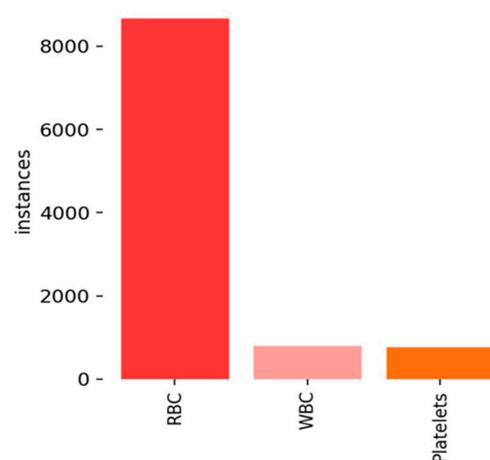


Figure 6. Instances of three types of blood cells.

#### 4.2. Experimental Configuration

This paper's experiments use Ubuntu 18.4.0 as the operating system. The experiments are conducted on the E-2136 CPU with a Quadro P5000 graphics card with 16 GB of video memory as its hardware configuration. The PyTorch 1.7.1 version is employed.

In Table 1, we can find the additional training parameters discussed in this paper.

**Table 1.** Training parameter settings for the BCCD dataset.

Parameter Name	Parameter Values	Parameter Name	Parameter Values
Batch size	16	Epoch	133
Learning rate	0.01	Momentum	0.937
Box loss	0.05	Cls loss	0.5
Anchors	(10, 13), (16, 30), 33, 23), (30, 61), (62, 45), (59,119)	Obj loss	1.0

#### 4.3. Evaluation Indicators

In this research, the accuracy of the algorithm is assessed based on three factors [49]: experimental detection precision, recall [50], and mean average precision (mAP) [51], as presented in Equations (10)–(12).

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (10)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

$$\text{mAP} = \frac{1}{N} \int_0^1 P_i(R) dR \quad (12)$$

TP represents the number of true positives—instances when the model's assessment is accurate, and the label is positive. FP represents the number of false positives—instances when the model's assessment is wrong, and the label is positive. FN represents the number of false negative—instances when the model's assessment is wrong, and the label is negative. TN represents the number of true negatives—instances when the model's assessment is accurate, and the label is negative. The precision of a blood cell category measures the accuracy of the predicted category out of the total number of predictions. At the same time, the recall is the accuracy of the predicted category out of the total number of actual categories. The number of distinct blood cell types represents N in Equation (12), and the mean average precision of all the categories is mAP.

#### 4.4. Results

The performance of SDE-YOLO for WBCs (white blood cells), RBCs (red blood cells), and Platelets is visualized in Figure 7 by plotting their respective P–R curves, and the mAP values of these blood cells obtained by computing the area beneath the curves.

The blood cell detection outcomes are presented in Figure 8b,c using the YOLO and SDE-YOLO algorithms, respectively. From the figure, we can see that the SDE-YOLO algorithm solves the problem of misdiagnosis of red blood cell as platelet in the first figure of Figure 8b, and the SDE-YOLO algorithm also solves the problem of missing red blood cells in the three figures of Figure 8b. In summary, the SDE-YOLO algorithm can solve the misdetection and omission problems in the original single-stage and two-stage algorithms.

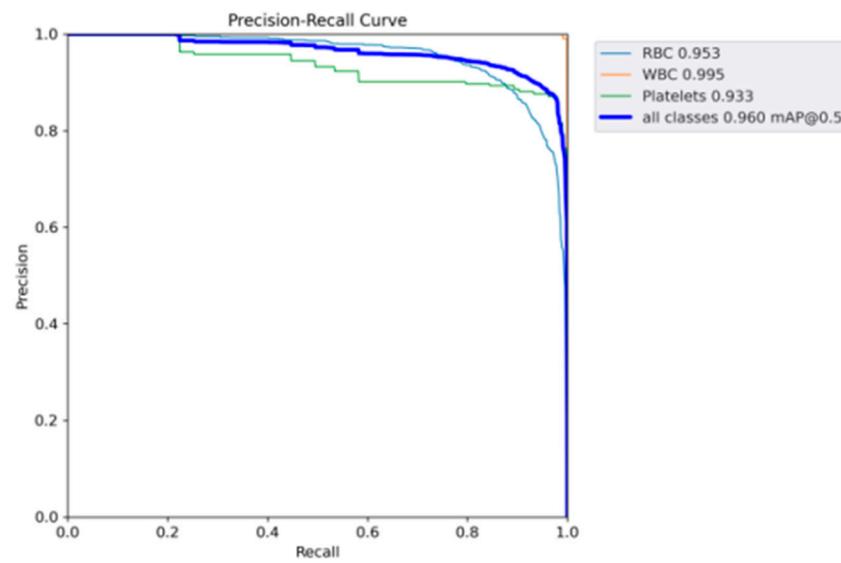


Figure 7. Representing the P–R curves of SDE-YOLO.

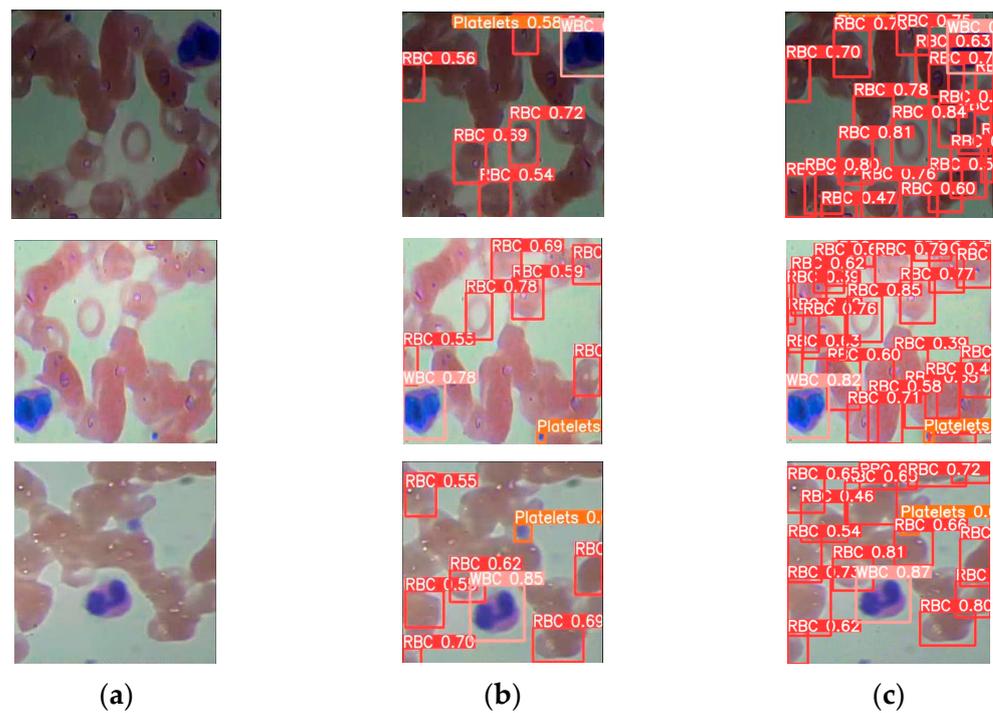
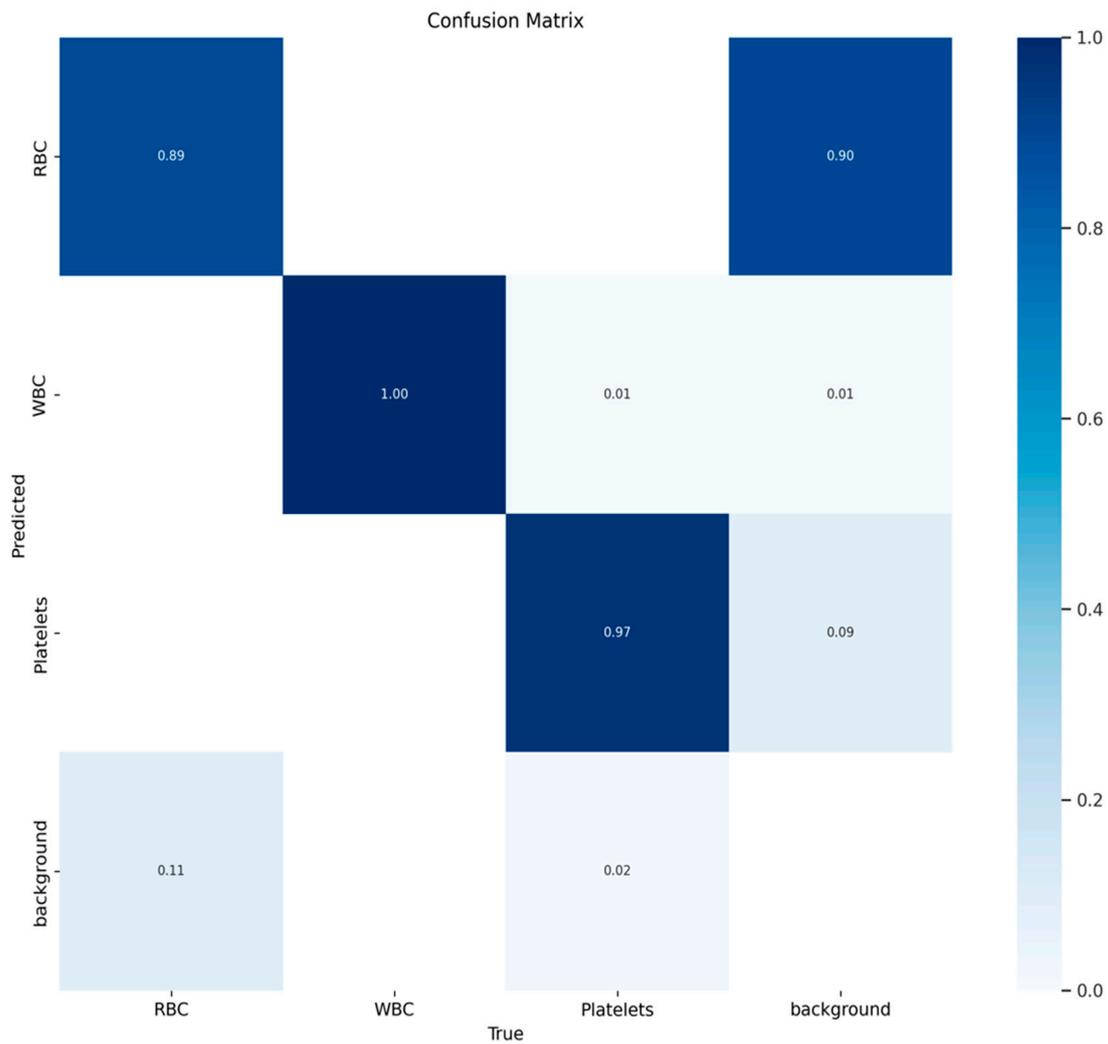


Figure 8. Representing the P–R curves of SDE-YOLO; (a) Original image of blood cells; (b) Blood cell images under other YOLO algorithms; (c) Blood cell images under SDE-YOLO detection algorithm.

The TP confusion matrix we generated is shown in Figure 9, from which we can see that, in addition to the background, red blood cells, white blood cells, and platelets form a diagonal matrix, which is ideal.



**Figure 9.** Confusion matrix results for blood cells.

Table 2 compares SDE-YOLO with some single-stage and two-stage detection algorithms.

**Table 2.** Comparative analysis experiments.

Method	Input Size	WBCs	RBCs	Platelets	mAP	FPS
Fast-RCNN [22]	1000 × 600	0.803	0.722	0.770	0.765	9.2
YOLOv3 [26]	608 × 608	0.914	0.829	0.774	0.839	34.5
YOLOv4 [27]	640 × 640	0.930	0.798	0.813	0.847	36.1
TE-YOLOF-B3 [52]	416 × 416	0.987	0.873	0.898	0.919	43
ISE-YOLO [53]	416 × 416	0.965	0.927	0.896	0.857	34.5
YOLOv5s [42]	640 × 640	0.977	0.838	0.873	0.896	56.7
SDE-YOLO	640 × 640	0.995	0.953	0.933	0.960	43.4

We can see from Table 2 that we achieved higher mAP and FPS than all other detection algorithms except YOLOv5s. SDE-YOLO has a higher mAP than Faster-R-CNN [22], YOLOv3 [26], YOLOv4 [27], TE-YOLOF-B3 [52], and ISE-YOLO [53], by 19.5, 12.1, 11.3, 4.1, and 10.3 percentage points, respectively. SDE-YOLO has a higher FPS than Faster-R-CNN, YOLOv3, YOLOv4, TE-YOLOF-B3, and ISE-YOLO, by 34.2, 8.9, 7.3, 0.4, and 8.9, respectively.

SDE-YOLO has a lower FPS than YOLOv5s [42] due to the Swin Transformer’s integration and deep separable convolution incorporation, resulting in a 13.3 FPS difference. But SDE-YOLO has the mAP that is 6.4 percentage points greater than YOLOv5s.

The experimental results indicate that the SDE-YOLO algorithm is the most effective of the models examined, as it improves the mAP without compromising the detection speed.

The graph in Table 3 shows the performance of SDE-YOLO compared to YOLOv7 and YOLOv8.

**Table 3.** Experiments with the latest technology.

Method	Input Size	WBCs	RBCs	Platelets	mAP	FPS	F1_curve	Precision
YOLOv7	640 × 640	0.995	0.954	0.928	0.959	9.1	0.92	0.864
YOLOv8	640 × 640	0.995	0.960	0.945	0.967	54.8	0.93	0.807
SDE-YOLO	640 × 640	0.995	0.953	0.933	0.960	43.4	0.92	0.851

SDE-YOLO is compared with YOLOv7. The F1\_curve is the same as YOLOv7; both are 0.92. SDE-YOLO is 1.3% less than YOLOv7 in terms of precision, but SDE-YOLO is 0.1% more than YOLOv7 in terms of mAP, and 31.3 more in terms of FPS than YOLOv7.

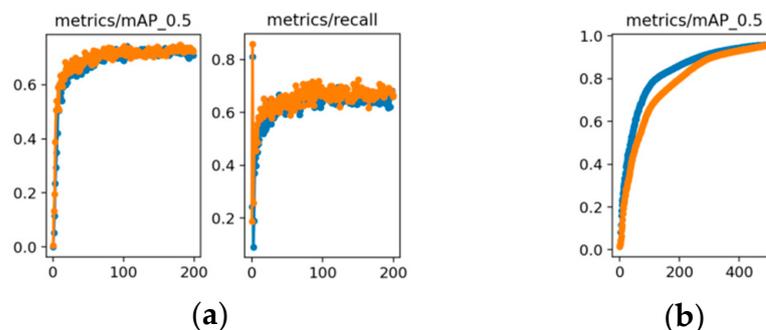
Compared with YOLOv8, SDE-YOLO is 0.7% less than YOLOv8 in mAP, 11.4% less than YOLOv8 in FPS, and 1% less than YOLOv8 in F1\_curve. However, SDE-YOLO is 1.3% higher than YOLOv8 in terms of accuracy.

SDE-YOLO has its advantages in performance over both YOLOv7 and YOLOv8, and all three models meet the accuracy and speed requirements.

### 5. Comparison with Other Datasets

We also assess SDE-YOLO on additional datasets to corroborate its generalization. The first dataset is the LIDC-IDRI set [54], a medically related collection of lung nodules, and we obtained 3144 images for the training set and 349 images for the test set through annotation. The TT100K dataset [55] comprises traffic signs, most of which are small- or medium-sized, and annotates 8816 images from the training set and 1712 images from the test set. The graphical representation of the two datasets’ performance metrics is in Figure 7.

The experimental results show that SDE-YOLO achieves a mAP of 72% on the LIDC-IDRI dataset, which is higher than the 70% of YOLOv5s. As illustrated in Figure 10a, SDE-YOLO exhibits a significant performance improvement compared to YOLOv5s when applied to the LIDC-IDRI dataset.



**Figure 10.** (a) Comparison of the YOLOv5s algorithm and the SDE-YOLO algorithm on the LIDC-IDRI dataset; (b) Comparison of the YOLOv5s algorithm and the SDE-YOLO algorithm on the TT100K dataset. The blue line represents the original YOLOv5s algorithm and the orange line represents the SDE-YOLO algorithm.

The mAP of SDE-YOLO on the TT100K dataset is comparable to YOLOv5s, which are both 96%, and Figure 10b shows that the final curve of SDE-YOLO is gradually approaching YOLOv5s.

In conclusion, SDE-YOLO has demonstrated its effectiveness across various datasets.

## 6. Ablation Studies

We run ablation experiments to assess the impact of each component on SDE-YOLO, with YOLOv5s as the foundation. A represents the combination of Swin Transformer, B represents the excision of the  $32 \times 32$  network layer, C signifies the implementation of depth-separable convolution, and D indicates the use of the EIOU loss function to raise the effectiveness. The findings of the experiment are displayed in Table 4.

**Table 4.** Ablation studies.

Method	Input Size	WBCs	RBCs	Platelets	mAP	F1_curve	Precision
YOLOv5s [42]	640 × 640	0.977	0.838	0.873	0.896	0.90	0.858
YOLOv5s + A	640 × 640	0.995	0.949	0.959	0.968	0.93	0.922
YOLOv5s + B	640 × 640	0.991	0.938	0.953	0.961	0.92	0.892
YOLOv5s + C	640 × 640	0.995	0.959	0.945	0.966	0.92	0.865
YOLOv5s + D	640 × 640	0.995	0.950	0.967	0.971	0.93	0.888
YOLOv5s + A + B	640 × 640	0.957	0.899	0.955	0.937	0.91	0.889
YOLOv5s + A + C	640 × 640	0.995	0.957	0.958	0.970	0.93	0.898
YOLOv5s + A + D	640 × 640	0.995	0.950	0.953	0.966	0.93	0.920
YOLOv5s + B + C	640 × 640	0.995	0.953	0.954	0.967	0.92	0.970
YOLOv5s + B + D	640 × 640	0.995	0.945	0.954	0.965	0.92	0.861
YOLOv5s + C + D	640 × 640	0.995	0.937	0.945	0.959	0.91	0.888
YOLOv5s + A + B + C	640 × 640	0.995	0.948	0.953	0.965	0.92	0.862
YOLOv5s + A + C + D	640 × 640	0.995	0.948	0.955	0.966	0.93	0.909
YOLOv5s + B + C + D	640 × 640	0.995	0.939	0.958	0.964	0.92	0.904
SDE-YOLO	640 × 640	0.995	0.953	0.933	0.960	0.92	0.902

Table 4 indicates that individual enhancement points contribute to the model's performance index, and that amalgamating multiple enhancement points is advantageous for the model. Compared to YOLOv5s, mAP improved by 4.1% to 7.5%, F1\_curve was enhanced by 0.1% to 0.3%, and accuracy improved by 0.2% to 6.4%.

## 7. Conclusions and Discussion

This paper presents a new algorithm, SDE-YOLO, which improves upon YOLOv5s. To begin with, we present an overview of the YOLOv5s architecture. Next, this paper outlines four new ideas: the integration of the Swin Transformer, the deletion of the  $32 \times 32$  target layer in PANet, the substitution of standard convolution with depth-separable convolution in PANet, and the use of the EIOU loss function in place of the CIOU loss function. This paper evaluates the single-stage and two-stage algorithms of the past, as well as YOLOv7 and YOLOv8, respectively, by conducting comparison experiments. The SDE-YOLO algorithm has demonstrated its effectiveness on other datasets, and finally, we test the impact of its components through ablation experiments. The run shows that SDE-YOLO is accurate and can be implemented swiftly.

However, during the implementation of blood cell testing, one may encounter the following challenges: 1. Quality and quantity of datasets: Blood cell image datasets may be affected by image quality, resolution, cueing conditions, and other factors. Obtaining large-scale and diverse blood cell datasets is challenging and crucial for model training and generalization; 2. Category imbalance: There may be an imbalance in the number of different types of cells in the blood cell images, resulting in a degradation of the performance of a few categories in the model. Strategies are needed to balance different categories of samples to improve the detector's ability to recognize various blood cells; 3. Geometric variations: The sizes and shapes of blood cells may have more significant variations in the

image. The model must adapt to cells with various geometries and can detect targets with different geometries; 4. Overlap and duplication: Blood cells may overlap and duplicate, etc., which makes target detection more urgent.

Future research will explore the fusion of multiple data sources to improve detection accuracy. Utilizing multimodal information can allow for a more comprehensive detection of blood cells and enhance the ability of the model to capture targets at different scales.

**Author Contributions:** Conceptualization, Y.W. and D.G.; methodology, Y.F.; software, H.G.; validation, Z.J., X.X. and D.G.; formal analysis, X.X.; investigation, D.G.; resources, Y.F.; data curation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, D.G.; visualization, H.G.; supervision, Z.J.; project administration, Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is funded by the National Natural Science Foundation of China (grant No. 52075530).

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The data used to support the findings of this study are included in this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chen, Y. Study on the value of blood smear analysis in routine blood tests. *China Pharm. Guide* **2018**, *16*, 118–119.
2. Chan, L.; Lavery, D. Accurate measurement of peripheral blood mononuclear cell concentration using image cytometry to eliminate RBC-induced counting error. *J. Immunol. Methods* **2013**, *388*, 25–32. [[CrossRef](#)] [[PubMed](#)]
3. Lehmann, T.M.; Guld, M.O. Content-based image retrieval in medical applications. *Methods Inf. Med.* **2004**, *43*, 354–361. [[PubMed](#)]
4. Patil, P.R.; Sable, G.S. Counting of WBCs and RBCs from blood images using gray thresholding. *Int. J. Res. Eng. Technol.* **2014**, *3*, 391–395.
5. Reddy, V.H. Automatic red blood cell and white blood cell counting for telemedicine system. *Int. J. Res. Advent Technol.* **2014**, *2*, 294–299.
6. Xiao, Q.; Huang, Y.B. Identification of diagnostic biomarkers for ulcerative colitis based on bioinformatics and machine learning. *Chin. J. Immunol.* **2023**, *39*, 1713–1718.
7. Guo, J.B.; Zhang, T.H. A machine learning model based on gadoteric acid disodium-enhanced MRI predicts microvascular invasion in hepatocellular carcinoma. *J. Mol. Imaging* **2023**, *46*, 736–740.
8. Liu, Y.; Tian, J.; Hu, R. Improved feature point pair purification algorithm based on SIFT during endoscope image stitching. *Front. Neurobot.* **2022**, *16*, 840594. [[CrossRef](#)]
9. Guo, H.; Ding, J.Y. Research progress on the application of machine learning in the diagnosis and treatment of novel coronavirus infection. *PLA Med. J.* **2023**, *48*, 863–870.
10. Zhuang, Y.; Jiang, N.; Xu, Y. Progressive distributed and parallel similarity retrieval of large CT image sequences in mobile telemedicine networks. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1–13. [[CrossRef](#)]
11. Wang, F.; Yan, J.H. Application of machine learning in leukaemia. *Med. Rev.* **2022**, *28*, 1928–1934.
12. Guo, R.; Song, Z.J. Machine learning: The future of artificial intelligence. *Electron. World* **2018**, *15*, 33–35.
13. Zhu, Y.; Huang, R.; Wu, Z. Deep learning-based predictive identification of neural stem cell differentiation. *Nat. Commun.* **2021**, *12*, 2614. [[CrossRef](#)] [[PubMed](#)]
14. Yu, J.H.; Gao, H.W. Adaptive spatiotemporal representation learning for skeleton-based human action recognition. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *14*, 1654–1665. [[CrossRef](#)]
15. Ao, J.; Shao, X.; Liu, Z. Stimulated Raman scattering microscopy enables gleason scoring of prostate core needle biopsy by a convolutional neural network. *Cancer Res.* **2023**, *83*, 641–651. [[CrossRef](#)]
16. Yu, J.H.; Gao, H.W. Deep temporal model-based identity-aware hand detection for space human–robot interaction. *IEEE Trans. Cybern.* **2021**, *52*, 13738–13751. [[CrossRef](#)]
17. Li, M.; Mu, C.C. Application of deep learning in differential diagnosis of enamel cell tumour and odontogenic keratocyst based on surface lamellar slices. *J. Chin. Acad. Med. Sci.* **2023**, *45*, 273–279.
18. Yu, J.H.; Gao, H.W.; Sun, J.; Zhou, D.L.; Ju, Z.J. Spatial cognition-driven deep learning for car detection in unmanned aerial vehicle imagery. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *14*, 1574–1583. [[CrossRef](#)]
19. Zhu, H.; Zhou, S.Y. A review of single-stage target detection algorithms based on deep learning. *Ind. Control. Comput.* **2023**, *36*, 101–103.
20. Ye, J.X. A review of two-stage target detection algorithms based on deep learning. *Internet Wkly.* **2023**, *18*, 16–18.

21. Wen, X.L.; Bai, T. Design of audio emotion recognition and classification model based on improved multimodal RCNN. *Mod. Electron. Technol.* **2023**, *46*, 114–118.
22. Ren, S.Q.; He, K.; Girshick, R. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
23. Yang, H.Z.; Li, D. YOLOv4 target detection based on improved SPPnet. *Electron. Prod.* **2021**, *12*, 52–54.
24. Qiao, Y.M.; Shi, J.K. Enhanced FPN underwater small target detection with improved loss function. *J. Comput. Aided Des. Graph.* **2023**, *35*, 525–537.
25. Chen, D.H.; Sun, S.R.; Wang, Y.C. Research on improved SSD algorithm for small target detection. *Sens. Microsyst.* **2023**, *42*, 65–68.
26. Yan, W.J.; Dai, J.H. Traffic sign recognition based on improved YOLOv3. *J. Wuhan Eng. Vocat. Technol. Coll.* **2023**, *35*, 31–35.
27. Chen, Y.F.; Yan, C.C.; Zhou, C. Improved YOLOv4-based vehicle detection for autonomous driving scenarios. *Autom. Instrum.* **2023**, *38*, 59–63.
28. Wang, Y.W.; Jiao, L.B. Remote Sensing target detection based on improved YOLOv5s. *Comput. Meas. Control.* **2023**, *31*, 70–76.
29. Wang, F.X.; He, L.H. Research on weakly supervised melanoma image segmentation based on attention mechanism. *Comput. Knowl. Technol.* **2023**, *19*, 12–14.
30. Ren, X.L. A study of attentional mechanisms and their role in medical visual tasks. *Imaging Technol.* **2023**, *35*, 76–80.
31. Chen, C.Y.; Xu, B.W. A review of research on attention mechanisms in medical image processing. *Comput. Eng. Appl.* **2022**, *58*, 23–33.
32. Yu, E.Z.; Zuo, X. Remote sensing ship image classification based on attention mechanism. *Softw. Guide* **2023**, *22*, 138–143.
33. Fang, Z.C.; Wu, S.J. Application of image classification algorithm based on soft attention mechanism in defect detection. *Mod. Inf. Technol.* **2023**, *7*, 151–154.
34. Fu, C.; Zhang, M. Anomalous brain MRI segmentation method incorporating texture features and attention mechanism. *Print. Digit. Media Technol. Res.* **2023**, *12*, 203–211.
35. Fang, Y.; Jiang, Z. Insulator defect detection based on attention mechanism and feature fusion. *Mod. Electron. Technol.* **2023**, *46*, 49–54.
36. Ren, H.S.; Zhou, Z.Y. Indoor scene depth map complementation based on channel attention mechanism. *J. Zhejiang Univ. Technol.* **2023**, *49*, 344–352.
37. Wang, X.J.; Qian, R.R. A low-light image enhancement method combining spatial attention mechanism and multi-scale resolution fusion. *Laser J.* **2023**, *44*, 86–91.
38. Wang, R.Q.; Wang, H.Q. Fusion of detail features and hybrid attention mechanism for fire smoke detection. *Liq. Cryst. Disp.* **2022**, *37*, 900–912. [[CrossRef](#)]
39. Zheng, C.W.; Lin, H. A YOLOv5 helmet wearing detection method based on Swin Transformer. *Comput. Meas. Control.* **2023**, *31*, 15–21.
40. Li, J.Q.; Zhang, L. An improved method for expression recognition based on depth-separable convolution. *Intell. Comput. Appl.* **2023**, *13*, 58–63.
41. Zhang, Y.F.; Ren, W.; Zhang, Z. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [[CrossRef](#)]
42. Ouyang, D.; Huang, H.; Li, J. Improved yolov5s model for aerial image target detection algorithm. *Fujian Comput.* **2023**, *39*, 7–15.
43. He, K.; Zhang, X.; Ren, S. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–7 July 2016; pp. 770–778.
44. Shah, S.A.R.; Wu, W.; Lu, Q. AmoebaNet: An SDN-enabled network service for big data science. *J. Netw. Comput. Appl.* **2018**, *119*, 70–82. [[CrossRef](#)]
45. Chollet, F. Xception: Deep learning with depth wise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
46. Lin, T.Y.; Dollar, P.; Girshick, R. 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.
47. Liu, S.; Qi, L.; Qin, H. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
48. Available online: [https://github.com/Shenggan/BCCD\\_Dataset](https://github.com/Shenggan/BCCD_Dataset) (accessed on 24 February 2018).
49. Wang, Y.F.; Li, D.H. Improved blood cell detection algorithm for YOLO framework. *Comput. Eng. Appl.* **2022**, *58*, 191–198.
50. Ahmad, M.; Qadri, S.F.; Ashraf, M.U. Efficient liver segmentation from computed tomography images using deep learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 2665283. [[CrossRef](#)] [[PubMed](#)]
51. Qadri, S.F.; Lin, H.; Shen, L. CT-Based Automatic Spine Segmentation Using Patch-Based Deep Learning. *Int. J. Intell. Syst.* **2023**, *2023*, 2345835. [[CrossRef](#)]
52. Xu, F.; Li, X.; Yang, H. TE-YOLOF: Tiny and efficient YOLOF for blood cell detection. *Biomed. Signal Process. Control.* **2022**, *73*, 103416. [[CrossRef](#)]
53. Liu, C.; Li, D.; Huang, P. ISE-YOLO: Improved Squeeze-and-Excitation Attention Module based YOLO for Blood Cells Detection. In Proceedings of the 2021 IEEE International Conference on Big Data, Orlando, FL, USA, 15–18 December 2021; pp. 3911–3916.

- 
54. Available online: <https://luna16.grand-challenge.org> (accessed on 20 October 2020).
  55. Huang, S.A. Traffic sign detection based on improved YOLO model. *Sci. Technol. Innov.* **2021**, *18*, 194–196.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.