

# Activation Function: Cell Recognition Based on YoLov5s/m

Zuomin Yang

College of Electromechanical Engineering, Qingdao University of Science and Technology, Qingdao, China

Email: zmy202006@gmail.com

**How to cite this paper:** Yang, Z.M. (2021) Activation Function: Cell Recognition Based on YoLov5s/m. *Journal of Computer and Communications*, 9, 1-16.  
<https://doi.org/10.4236/jcc.2021.912001>

**Received:** November 21, 2021

**Accepted:** December 19, 2021

**Published:** December 22, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Activation functions play a critical role in neural networks. The paper mainly studies activation functions with four activation functions that were the selection for reference and comparison. The Mish activation function was expanding as the Mish\_PLUS activation function, the Sigmoid activation function, and the Tanh were combined to obtain a new Sigmoid\_Tanh activation function. We used the recently popular YoLov5s and YoLov5m as the basic structure of the neural network. The function realized in this article was the recognition function of red blood cells, white blood cells, and platelets. Through the role and comparison of different activation functions in the neural network structure, the test results show that, in this paper, the training precision curve under the Sigmoid\_Tanh activation function was better than that under the action of other activation functions. That means that the accuracy of cell recognition under the activation function was higher.

## Keywords

Mish\_PLUS, Sigmoid\_Tanh, Cell Recognition

## 1. Introduction

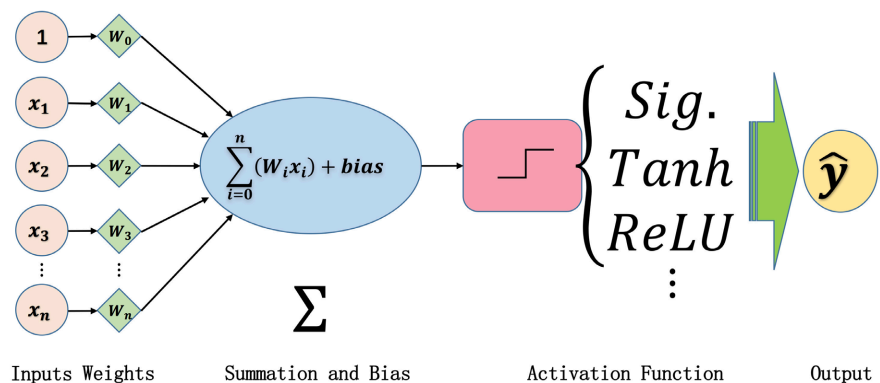
The blood of an organism contains many components. It was one of the tasks of biomedical research to find out the cells in the blood, for instance, using some methods rapidly and accurately screening of red blood cells, white blood cells, platelets, etc. [1] [2] [3]. The application of artificial intelligence technology in the biomedical field provided a critical research theory for the development of biomedical. Using artificial intelligence recognition technology is one of the methods to screening out different types of cells in biological blood for example, adopting machine learning and deep learning methods to screen diverse cells types [4] [5]. The previous means have an application value for screening and

identifying different cells types, but it was not perfect. How to screen diverse cells quickly, efficiently, and accurately is one of the research fields of biomedicine.

Activation functions were a critical part of the design of a neural network. It helps the neural network to learn the complex patterns in the data, just like the neuron-based model in the human brain. The activation function obtains information from the front neuron and then transmits it to the next neuron [6]. As shown in **Figure 1**, in the neuron, Input is also applied to a function after weighting and summing. This function was the activation function. The activation function was an introduction to increase the nonlinearity of the neural network model. Each layer without the activation function was equivalent to matrix multiplication. However, the neuron information of the previous layer cannot transfer to the neuron information of the next layer. The function can realize the transmission of neuron information, so the activation function was an indispensable part of an artificial neural network. To improve the computational performance of neural networks, people have studied the activation functions in neural networks. Common activation functions include Sigmoid [7], Tanh [8], SiLU [9], Hardswish [10], Mish [11], MemoryEfficientMish, etc.

The SiLU activation function was also called the Swish activation function. The Swish was an activation function proposed by the Google team in recent years and was composed of the previous activation function. The expression of the Swish is  $f(x) = x * \sigma(x)$ , and  $\sigma(x)$  is the Sigmoid. Because the saturation of the Sigmoid tends to cause the gradient to disappear, learn from the effect of ReLU, when  $x \rightarrow +\infty$ , then  $f(x) \rightarrow x$ , but when  $x \rightarrow -\infty$ , then  $f(x) \rightarrow 0$ , the general trend of the function is similar to ReLU but more complicated than ReLU [12]. It is a reflection in the function by adding a few hyper-parameters and then showing more characteristics. Adding a hyperparameter to Swish makes the function expression  $f(x) = x * \sigma(\beta x)$ , and we say that  $\beta$  can be a constant or a trainable parameter.

Although this Swish nonlinearity improves accuracy, its cost is non-zero in an embedded environment. It is much more expensive to calculate the Sigmoid function on a mobile device. The author of MobileNetV3 used Hardswish & Hard-sigmoid to replace the Sigmoid layer in ReLU6 & SE-block. But only in the latter



**Figure 1.** Summary diagram of activation function.

half of the network did ReLU6 be replaced with Hardswish, because the author found that the Swish can only use in a deeper network layer to reflect its advantages.

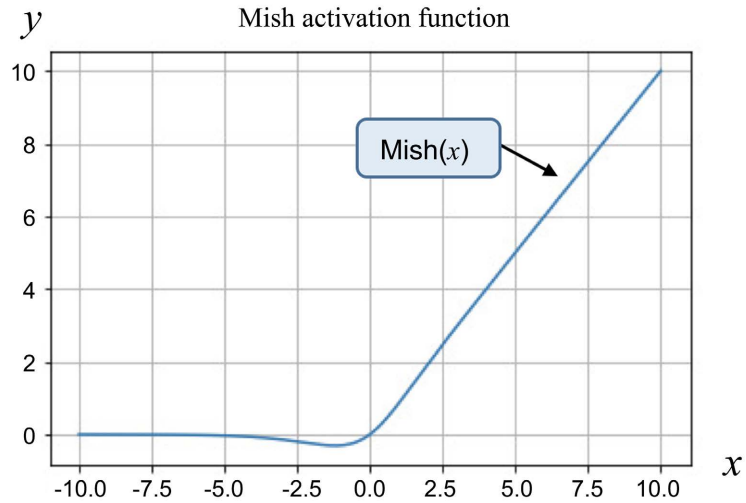
The Mish activation function is a new activation function proposed by Diganta Misra *et al.* It surpasses ReLU in some tasks. From YoLov1 to YoLov5, the accuracy has been improving. For example, YoLov4 has a qualitative leap compared to YoLov3 with mAP. One of the reasons is to replace LeakyReLU with the Mish activation function. In the paper, Diganta Misra described that the widely used activation functions are ReLU, Tanh, Sigmoid, Leaky ReLU, and Swish. For instance, in Squeeze Excite Net-18 for CIFAR 100 classification, the network with Mish had an increase in Top-1 test accuracy by 0.494% and 1.671% as compared to the same network with Swish and ReLU, respectively. It shows that the Mish activation function has certain advantages over others. The formula of the Mish activation function is  $\text{Mish}(x) = x \cdot \tanh[\text{Softplus}(x)] = x \cdot \tanh[\ln(1 + e^x)]$ . Softplus is proposed by Yoshua Bengio. In the paper [13],  $\text{Softplus}(x) = \ln(1 + e^x)$ , and the value range is  $(0, +\infty)$ . Softplus can regard as the smoothness of ReLU. The Mish activation function image is shown in **Figure 2**.

The MemoryEfficientMish activation function was the first derivative of the Mish activation function, and its formula was

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \quad (1)$$

$$f_x = \tanh[\ln(1 + e^x)] \quad (2)$$

$$\begin{aligned} \text{MemoryEfficientMish}(x) &= \text{Mish}'(x) = \{x \cdot \tanh[\ln(1 + e^x)]\}' \\ &= \tanh[\ln(1 + e^x)] + [1 - \tanh[\ln(1 + e^x)]^2] \cdot \frac{x \cdot e^x}{1 + e^x} \\ &= f_x + (1 - f_x^2) \cdot x \cdot \text{sigmoid}(x) \end{aligned} \quad (3)$$



**Figure 2.** Image of Mish activation function.

In this paper, the activation function was the research object mainly. Two kinds of neural network morphology in YoLov5 are used as an experimental basis to conduct experiments on different activation functions and explore the accuracy influence of different kinds of activation functions on cell recognition.

## 2. Methods

### 2.1. Mish\_PLUS Activation Function

Given the excellent performance of the Mish activation function in YoLo, this article expands the Mish activation function to obtain the Mish\_PLUS activation function and use the Mish\_PLUS activation function to the YoLov5s and YoLov5m neural network structures. The formula of the Mish\_PLUS activation function is like (6). In the Mish\_PLUS activation function  $F(x)$ , the original Mish formula is changed, and  $x \cdot \tanh[\omega]$  is addition, where  $\omega$  means  $\text{Mish}(x)$ . The image of  $F(x)$  is **Figure 3**.

$$\text{Softplus}(x) = \ln(1 + e^x) \quad (4)$$

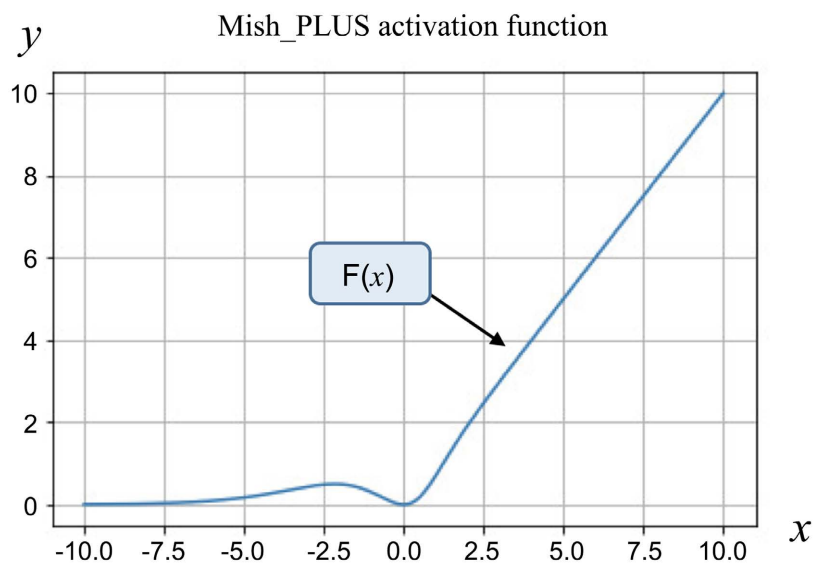
$$\text{Mish}(x) = x \cdot \tanh[\text{Softplus}(x)] = x \cdot \tanh[\ln(1 + e^x)] \quad (5)$$

$$F(x) = x \cdot \tanh\{\text{Mish}(x)\} = x \cdot \tanh\{x \cdot \tanh[\ln(1 + e^x)]\} \quad (6)$$

It can be seen from the comparison of **Figure 2** and **Figure 3** that when  $x$  approaches zero to the Mish activation function approaches zero from a negative value. The Mish\_PLUS activation function approaches zero from a positive value.

### 2.2. Sigmoid\_Tanh Activation Function

At present, there are dozens of activation functions widely used in neural network structures, and different activation functions have different roles in neural



**Figure 3.** Image of Mish\_PLUS activation function.

networks. And their advantages and disadvantages are also so. Commonly used activation functions include: Sigmoid, Tanh, ReLU [14], Leaky ReLU [15], ELU [16], SELUs [17], GELUs [18], PreLU [19], MaxOut [20], RReLU [21], etc., and some activation functions are expanded based on the original activation function to obtain a variant of the activation function. In this article, the classic Sigmoid and Tanh are mainly studied.

The Sigmoid formula and derivative are:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - \text{Sigmoid}(-x) \quad (7)$$

$$\begin{aligned} \text{Sigmoid}'(x) &= \frac{0 - 1 \cdot (-e^{-x})}{(1 + e^{-x})^2} = \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{e^{-x}}{1 + e^{-x}} \cdot \frac{1}{1 + e^{-x}} \\ &= \frac{1 + e^{-x} - 1}{1 + e^{-x}} \cdot \sigma(x) = (1 - \sigma(x)) \cdot \sigma(x) \end{aligned} \quad (8)$$

The Tanh formula and derivative are:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{2}{1 + e^{-2x}} - 1 \quad (9)$$

$$\begin{aligned} \tanh'(x) &= \left( \frac{e^x - e^{-x}}{e^x + e^{-x}} \right)' = \frac{e^x + e^{-x}}{e^x + e^{-x}} - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2} \\ &= 1 - \left( \frac{e^x - e^{-x}}{e^x + e^{-x}} \right)^2 = 1 - \tanh^2(x) \end{aligned} \quad (10)$$

The source of formula (7)-(10) can be obtained from references [7] [8]. The Sigmoid function was proposed by Jun Han *et al.* It is a mathematical logic function, such as formula (7). Since the visualized curve is S-shaped, called S-curve, as shown in **Figure 4**. Used in the neural network for the output of hidden layer neurons, the value range is (0, 1). It can map a real number to the interval of (0, 1), so it was two classifications in training. The difference in image features is more complicated or the difference is not special large, the effect was better. The advantage of Sigmoid as an activation function is that the curve presented is relatively smooth, and it was easy to derive during the calculation process, as shown in formula (8). The disadvantage is that the activation function has a large amount of calculation. When backpropagating to find the error gradient, the derivation involves division. When backpropagating, the gradient disappears easily, and the training of the deep network cannot complete. The hyperbolic tangent function is a kind of it, called Tanh. In the neural network, Tanh is used as the activation function of the neuron to transmit information. It is a non-linear function. The formula is shown in (9), and its derivative is shown in (10). The Tanh function converts the final result of the fitted curve to the interval of (-1, 1). The maximum negative number is infinitely close to -1, and the maximum positive number is infinitely close to 1. Tanh solves whether the output of Sigmoid is zero centers, but there is still a saturation problem (**Figure 5**).

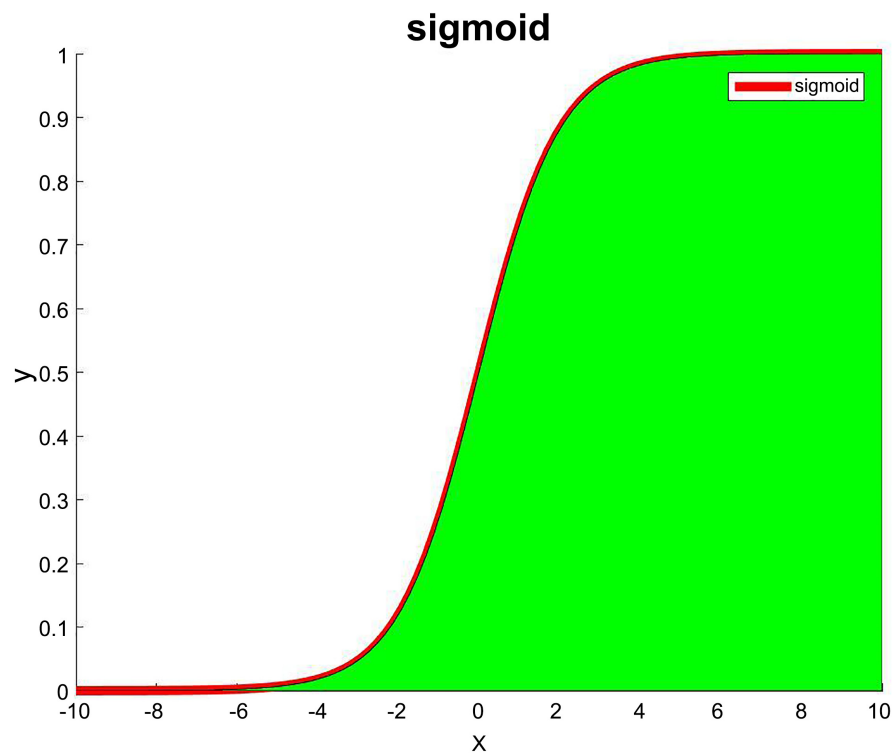


Figure 4. Image of Sigmoid activation function.

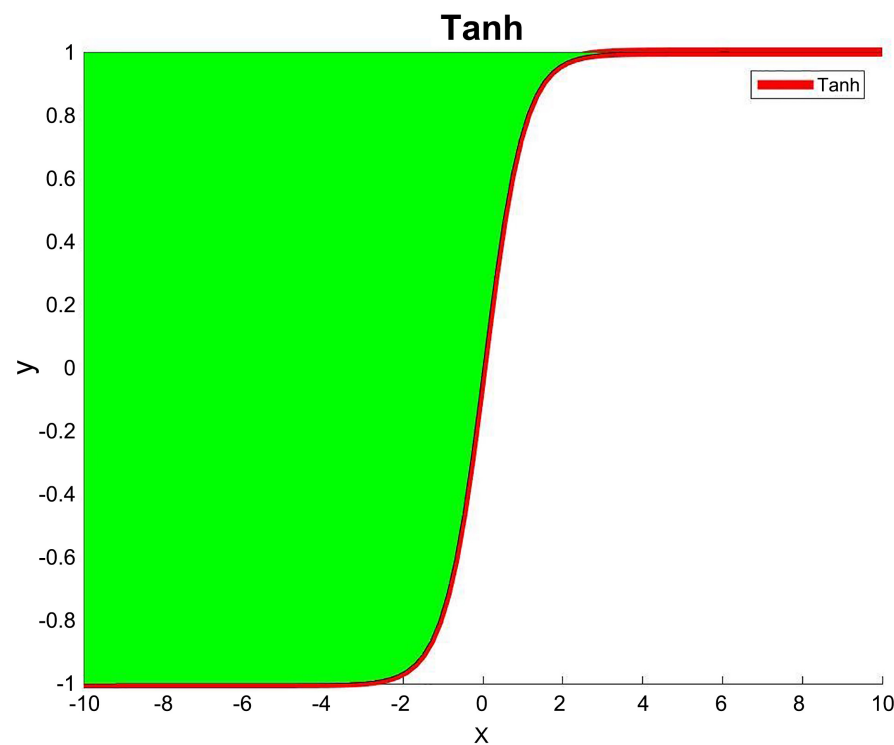


Figure 5. Image of Tanh activation function.

Given the characteristics of the Sigmoid activation function and the Tanh activation function, in this article, the Sigmoid activation function and the Tanh

activation function are combined for experimentation. That is to say, the Sigmoid activation function and the Tanh activation function are multiplication to obtain a new activation function, the formula as shown in (11), its value range is  $(-1, 1)$ , and the obtained visualization curve in **Figure 6**. We can see from **Figure 6**,  $x \rightarrow -\infty$ ,  $W(x) \rightarrow 0$ ;  $x \rightarrow +\infty$ ,  $W(x) \rightarrow 1$ .

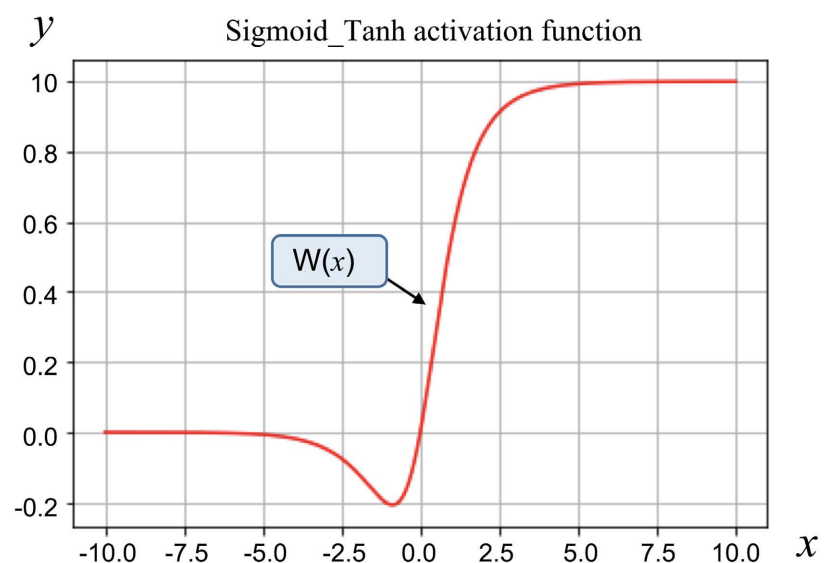
$$\begin{aligned} W(x) &= \text{Sigmoid}(x) \cdot \tanh(x) = \frac{1}{1+e^{-x}} \cdot \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ &= \frac{e^x}{e^x + 1} \cdot \frac{(e^x + 1) \cdot (e^x - 1)}{e^{2x} + 1} = e^x \cdot \frac{e^x - 1}{e^{2x} + 1} = \frac{e^{2x} - e^x}{e^{2x} + 1} \end{aligned} \quad (11)$$

### 3. Results

In this paper, on the one hand, inspired by the Mish activation function, the Mish activation function is extended based on the Mish activation function; on the other hand, the Sigmoid activation function and the Tanh activation function choose from the widely used activation functions. Function as the research object, multiply the Sigmoid activation function and Tanh activation function to get a new activation function for the experiment. The codes of Mish\_PLUS activation function and Sigmoid\_Tanh activation function are in **Table 1**.

In this paper, YoLov5s and YoLov5m as the neural network structure, and the function realized was the recognition function of white blood cells, red blood cells, and platelets. There is a dataset of blood cells photos, originally open-sourced by <https://github.com/cosmicad/dataset>. In this paper, the dataset was exported via roboflow.ai on February 23, 2021. There are 874 images across three classes: WBC (white blood cells), RBC (red blood cells), and Platelets. The following sections were the parameter results obtained under the conditions of different activation functions.

**Table 2** shows the results of different activation functions under the neural



**Figure 6.** Image of Sigmoid\_Tanh activation function.

**Table 1.** Mish\_PLUS activation function and Sigmoid\_Tanh activation function code. (a) Mish\_PLUS activation function code; (b) Sigmoid\_Tanh activation function code.

(a)	
1.	Class Mish_PLUS(nn.Module):
2.	def forward(x):
3.	M=x*F.softplus(x).tanh()
4.	return x*torch.tanh(M)
(b)	
1.	Class Sigmoid_Tanh(nn.Module):
2.	def _init_(self):
3.	super(Sigmoid_Tanh,self)._init_()
4.	def forward(self, x):
5.	return torch.sigmoid(x) * torch.tanh(x)

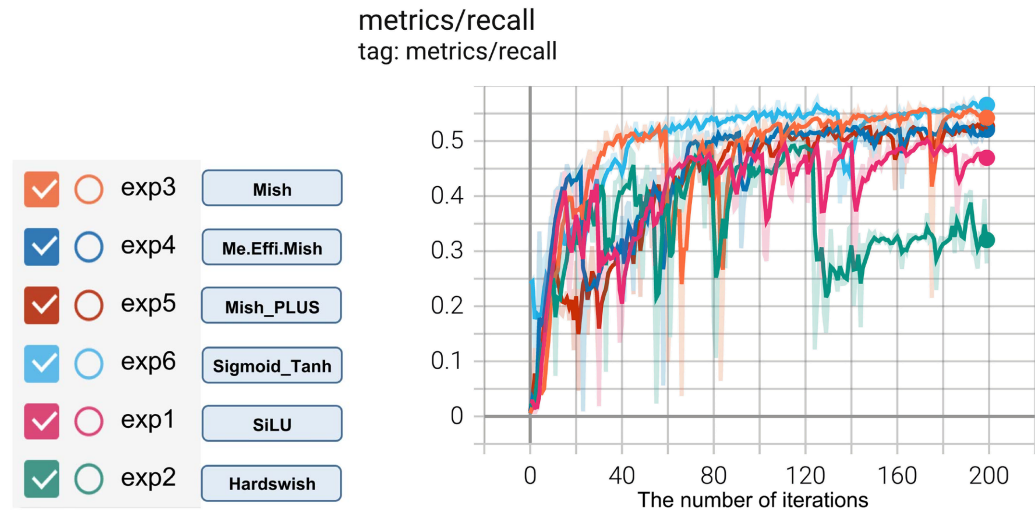
**Table 2.** Parameter results based on the structure of YoLov5s neural network.

Model	Layers	File Name	Activation Function	Params	GFLOPS
YoLov5s	283	exp1	SiLU	7,068,936	16.4
	283	exp2	Hardswish	7,068,936	16.4
	283	exp3	Mish	7,068,936	16.4
	283	exp4	MemoryEfficientMish	7,068,936	16.4
	283	exp5	<b>Mish_PLUS</b>	7,068,936	16.4
	283	exp6	<b>Sigmoid_Tanh</b>	7,068,936	16.4

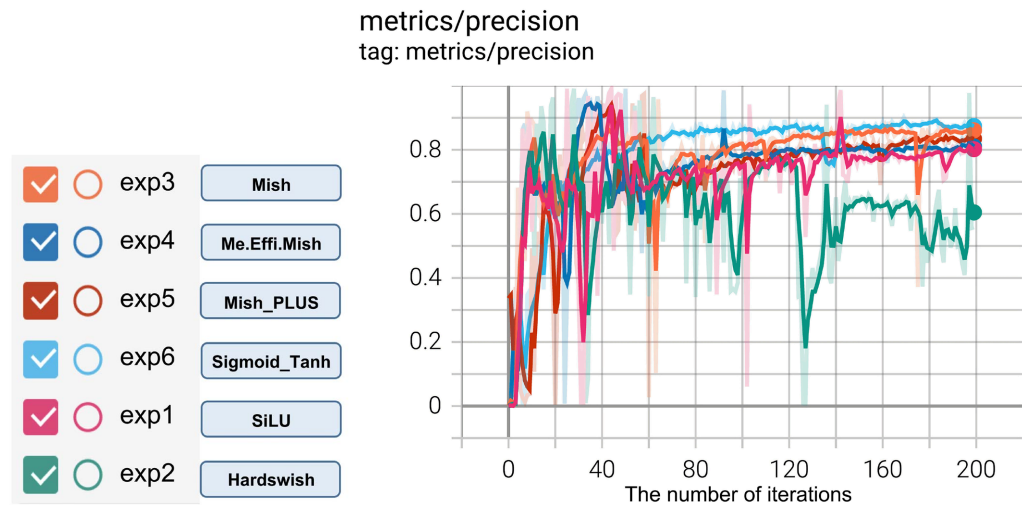
network structure YoLov5s. It can be seen from **Table 1** that using YoLov5s as the network structure of this article, the neural network has a total of 283 layers, and the activation functions are SiLU function, Hardswish function, Mish function, MemoryEfficientMish function, Mish\_PLUS function, and Sigmoid\_Tanh function. Each training has a total of 7,068,936 parameters, and the number of floating-point operations is 16.4GFLOPS. We have a comparison of the results of different activation functions under the neural network structure YoLov5s.

We use different activation functions to identify cell types, and the recall and precision are representation, by curves with different colors. Each activation function is training 200 times. Knowledge about recall and precision can obtain from [22]. **Figure 7** and **Figure 8** are the curves of recall and precision under different activation functions. It can see from **Figure 7** that the green curve under the Hardswish activation function shows the worst effect, and the curve change is unstable. When the training reaches 128 times, it displays a turning point in the recall, and the recall increases before the 128 times training. The recall dropped sharply after 128 training sessions and did not change much. The blue curve represents the Sigmoid\_Tanh activation function, which presents the best effect. The curve maintains a steady upward trend. When the training reaches





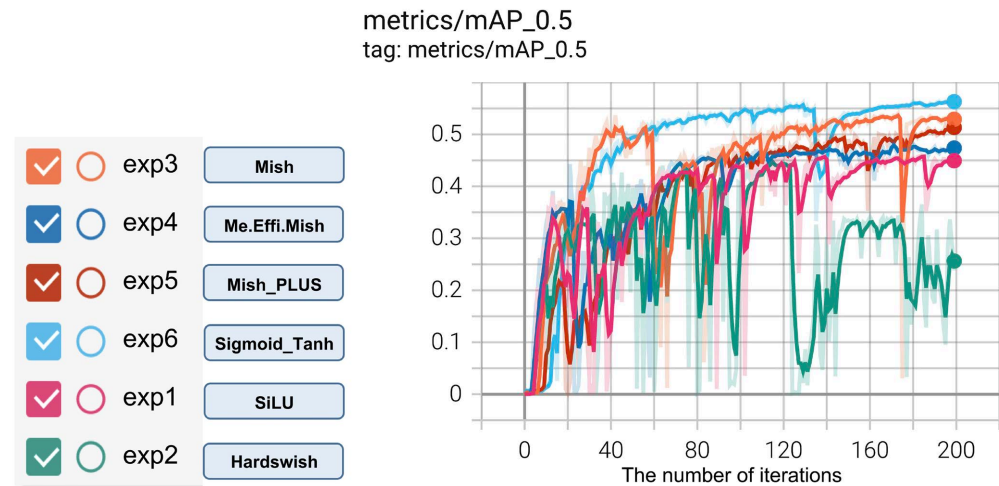
**Figure 7.** The recall curve changes under different activation functions based on YoLov5s.



**Figure 8.** Precision curve changes under different activation functions based on YoLov5s.

140 times, the recall fluctuates, but after 140 times, the recall still shows an upward trend. The recall under the other five activation functions is better. It can see from **Figure 8** that the green curve under the Hardswish activation function shows the worst effect, and the curve change is unstable. When the training reaches 128 times, it displays a turning point in the precision, and the precision increases before the 128 times training. The accuracy rate drops and fluctuates unstable after 128 times. The blue curve represents the Sigmoid\_Tanh activation function, which presents the best effect. The curve maintains a steady upward trend. When the training reaches 140 times, the precision rate fluctuates, but after the 140 times training, the precision still shows an upward trend. The precision rate is better under the other five activation functions.

**Figure 9** shows the mean average precision (mAP) of different activation functions for different cell types under the YoLov5s neural network structure. It can see from **Figure 9** that the blue curve under the action of the Sigmoid\_Tanh



**Figure 9.** Comparison of average precision under different mean average precision (mAP) based on YoLov5s.

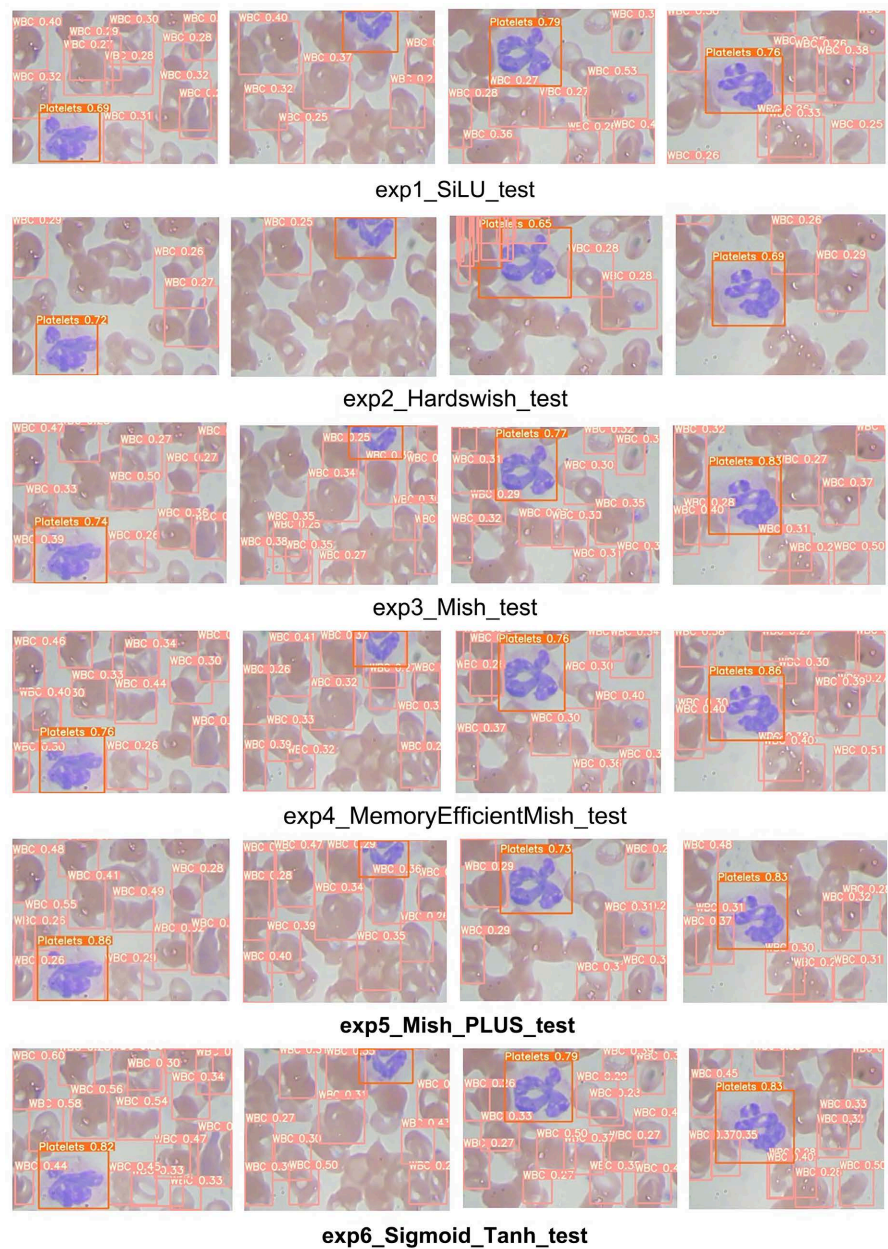
activation function presents a better effect, and the mAP of different types of cell recognition obtains better scores. The second is the result obtained by the orange curve under the action of the Mish\_PLUS activation function. The green curve under the activation function Hardswish presents a poor effect, and the mAP of different types of cell recognition obtains a poor score.

### 3.1. Cell Recognition Results

Under different activation functions, we conduct experiments on the identification of cell types. **Figure 10** shows the cell recognition results under activation functions based on the YoLov5s neural network structure. It can be seen from the sparse degree of cell-type recognition in **Figure 10** that the cell type recognition effect under the Hardswish activation function was the worst, and the cell type recognition effect under the Sigmoid\_Tanh activation function was the best. The cell type recognition under the Mish\_PLUS activation function can also achieve better results, but it was not as good as the cell type recognition under the MemoryEfficientMish activation function.

**Table 3** shows the parameter results of different activation functions under the neural network structure YoLov5m. It can be seen from **Table 3** that using YoLov5m as the neural network structure of this article, the neural network has a total of 391 layers, and the activation functions are SiLU function, Hardswish function, Mish function, MemoryEfficientMish function, Mish\_PLUS function, and Sigmoid\_Tanh function. There are 21,064,488 parameters in each training, and the number of floating-point operations was 50.4GFLOPS. We have a comparison of the results of different activation functions under the neural network structure YoLov5m.

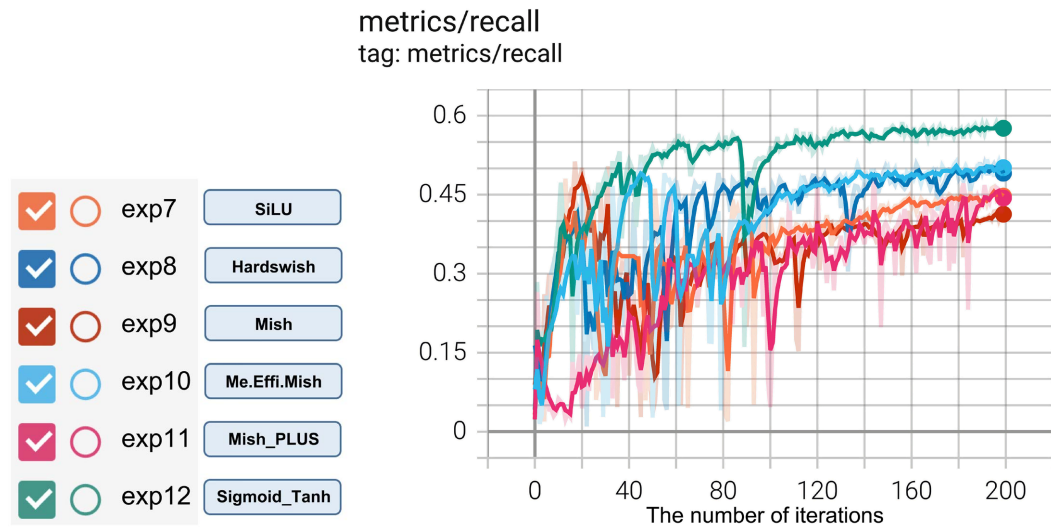
For the diverse cell types recognition, each activation function had has been trained 200 times to obtain the recall and precision of the types. **Figure 11** and **Figure 12** are the curves of recall and precision under different activation functions. It can be seen from **Figure 11** that the red curve represents the result under the



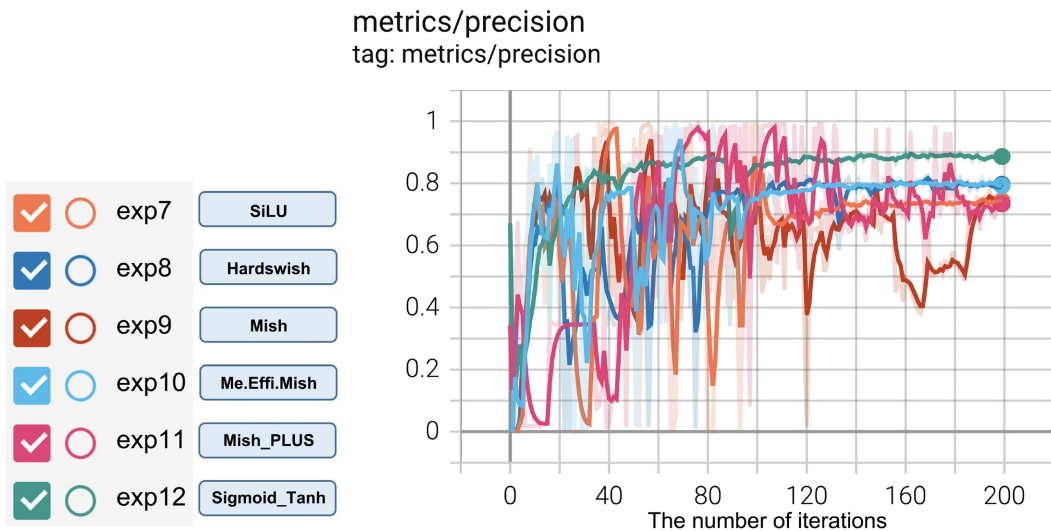
**Figure 10.** Cell recognition results under different activation functions based on YoLov5s.

**Table 3.** Parameter results based on the structure of YoLov5m neural network.

Model	layers	File Name	Activation	Params	FLOPS
YoLov5m	391	exp7	SiLU	21,064,488	50.4 G
	391	exp8	Hardswish	21,064,488	50.4 G
	391	exp9	Mish	21,064,488	50.4 G
	391	exp10	MemoryEfficientMish	21,064,488	50.4 G
	391	exp11	<b>Mish_PLUS</b>	21,064,488	50.4 G
	391	exp12	<b>Sigmoid_Tanh</b>	21,064,488	50.4 G



**Figure 11.** The change of recall curve under different activation functions based on YoLov5m.

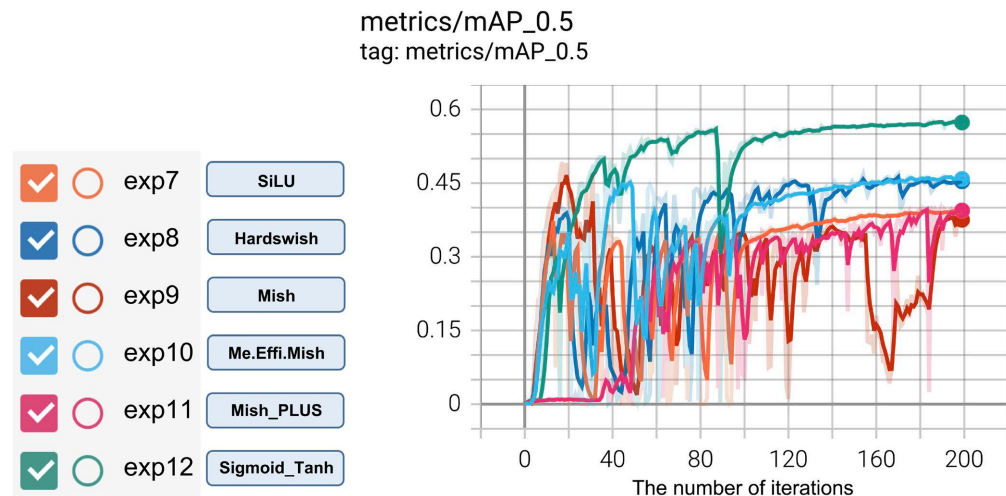


**Figure 12.** Precision curve changes under different activation functions based on YoLov5m.

Mish activation function, and the effect was the worst and unstable. The green curve represents the result under the Sigmoid\_Tanh activation function, and the result was the best. The curve maintains a steady upward trend. When the training reaches 90 times, the recall rate fluctuates badly, but after 90 times, the recall rate still shows an upward trend. In general, the recall rate was better than the other five activation functions. It can see from **Figure 12** that the red curve under the Mish activation function has the worst effect, and the curve change was unstable. The green curve under the Sigmoid\_Tanh activation function presents the best result. The curve maintains a steady upward trend which was generally better than the precision of the other five activation functions.

**Figure 13** shows the mAP of different activation functions for different cell types under the YoLov5s neural network structure. It can see from **Figure 13** that the green curve under the action of the Sigmoid\_Tanh activation function





**Figure 13.** Comparison of mAP under different activation functions based on YoLov5m.

presents a better effect, and the mAP obtains a better score. The red curve under the Mish activation function gets a poor result and the mAP of different types of cell recognition to a poor score.

### 3.2. Cell Recognition Results Based on YoLov5m

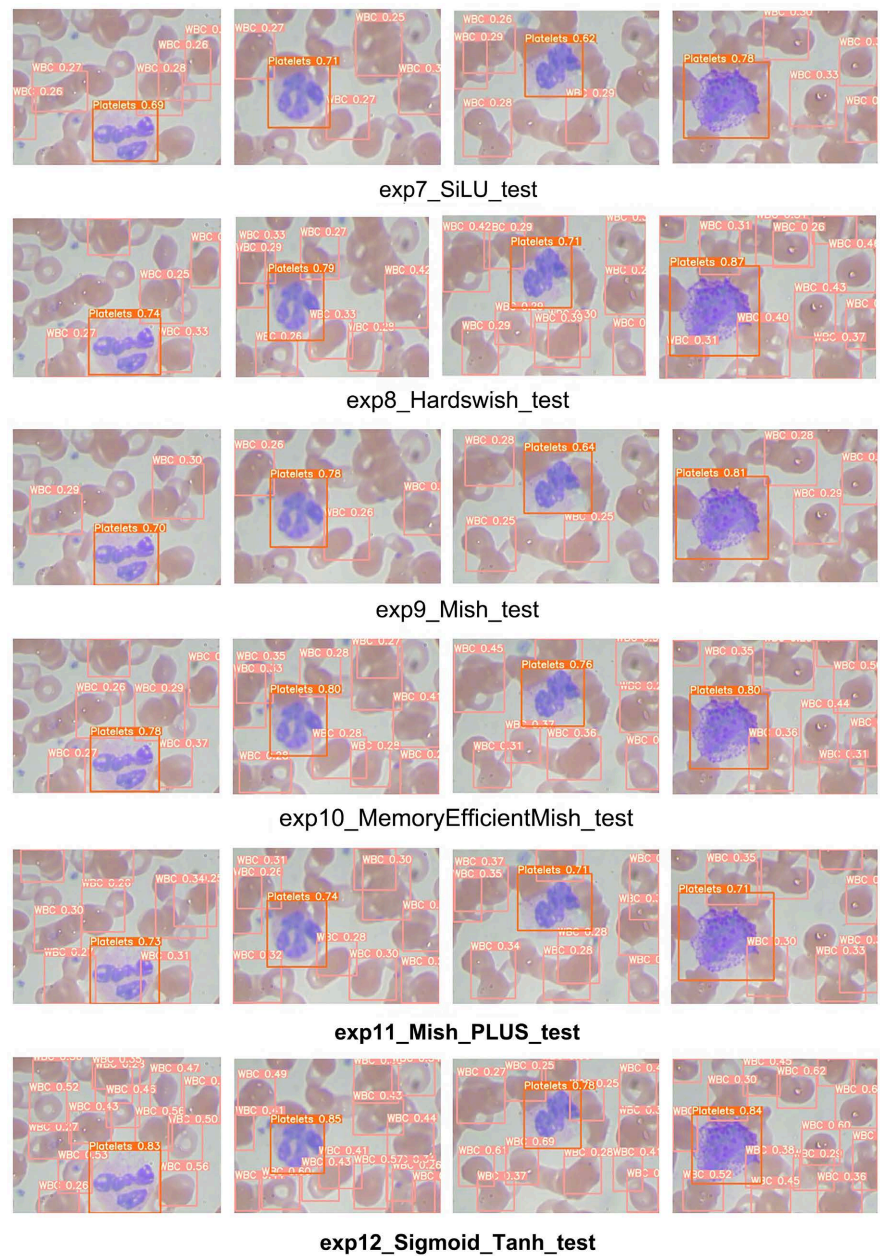
**Figure 14** shows the cell recognition results under different activation functions based on the YoLov5m neural network structure. It can see from the sparse degree of cell-type recognition in **Figure 14** that the cell type recognition effect under the Mish activation function was the worst, and the cell type recognition effect under the Sigmoid\_Tanh activation function was the best. The cell type recognition under the Mish\_PLUS activation function can also achieve better results, but it was not as good as the cell type recognition under the Sigmoid\_Tanh activation function.

## 4. Discussion

As a necessary condition in the neural network structure, the activation function directly affects results on certain functions. In recent years, the research on the activation function has been intensification. Common activation functions include: Sigmoid, Tanh, ReLU, LReLU, PReLU, Swish, etc. The activation function was also widely used. For example, the activation function used to mobile robots as a robot recognition and the algorithm for understanding the scene; in drone technology, the drone needs to face a complex environment, and the activation function algorithm in machine learning plays an important role in the understanding of the drone scene.

This paper mainly studies the activation function, but there were limitations in cell recognition. Mainly as follows:

1) This paper takes cells as recognition objects and tests the performance of recognition cells through different activation functions. If you replace other recognition objects, maybe not be able to achieve the desired effect.



**Figure 14.** Cell recognition results under different activation functions based on YoLov5m.

2) There are many common activation functions. There were only compares and tests of the extended activation function with the four activation functions tapes, and does not compare with other common activation functions.

3) Due to the limitation of experimental equipment, there was only a simple training, which has certain restraints for the training results.

4) The research in this paper focuses more on recognition, which needs to experiment under a high-power microscope in practical application.

Finding an efficient and suitable activation function is the subject of future research. In the future, the activation function needs further research.

## 5. Conclusion

This paper focuses on finding a suitable activation function for research. Using YoLov5s and YoLov5m as the basic structure of the neural network, the activation functions SiLU, Hardswish, MemoryEfficientMish, Mish, Mish\_PLUS, Sigmoid\_Tanh were a test. The Mish\_PLUS activation function was an improved form of Mish, and Sigmoid\_Tanh combines the Sigmoid activation function and Tanh activation function to a new Sigmoid\_Tanh activation function. The data set used in this article was the BCCD.v4-416x416 data set, which realizes the function of identifying red blood cells, white blood cells, and platelets. The test results show that the Sigmoid\_Tanh activation function obtained in this paper can play a positive role in cell recognition.

## Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

## References

- [1] Parab, M.A. and Mehendale, N.D. (2021) Red Blood Cell Classification Using Image Processing and CNN. *SN Computer Science*, **2**, Article No. 70. <https://doi.org/10.1007/s42979-021-00458-2>
- [2] Liu, M., Liu, Y., Qian, W. and Wang, Y. (2021) DeepSeed Local Graph Matching for Densely Packed Cells Tracking. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **18**, 1060-1069. <https://doi.org/10.1109/TCBB.2019.2936851>
- [3] Xie, Y., Liu, M., Zhou, S. and Wang, Y. (2021) A Deep Local Patch Matching Network for Cell Tracking in Microscopy Image Sequences without Registration. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. <https://doi.org/10.1109/TCBB.2021.3113129>
- [4] Lavitt, F., Rijlaarsdam, D.J., van der Linden, D., Weglarz-Tomczak, E. and Tomczak, J.M. (2021) Deep Learning and Transfer Learning for Automatic Cell Counting in Microscope Images of Human Cancer Cell Lines. *Applied Sciences*, **11**, Article No. 4912. <https://doi.org/10.3390/app11114912>
- [5] Ryu, D.H., Kim, J., Lim, D., Min, H.-S., Yoo, I.Y., Cho, D. and Park, Y.K. (2021) Label-Free White Blood Cell Classification Using Refractive Index Tomography and Deep Learning. *BME Frontiers*, **2021**, Article ID: 9893804, 9 p. <https://doi.org/10.34133/2021/9893804>
- [6] Agostinelli, F., Hoffman, M., Sadowski, P. and Baldi, P. (2015) Learning Activation Functions to Improve Deep Neural Networks. <http://arxiv.org/abs/1412.6830>
- [7] Han, J. and Moraga, C. (1995) The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning. *International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, Torremolinos, 7-9 June 1995, 195-201. [https://doi.org/10.1007/3-540-59497-3\\_175](https://doi.org/10.1007/3-540-59497-3_175)
- [8] Ryck, T.D., Lanthaler, S. and Mishra, S. (2021) On the Approximation of Functions by Tanh Neural Networks. *Neural Networks*, **143**, 732-750. <https://doi.org/10.1016/j.neunet.2021.08.015>
- [9] Ramachandran, P., Zoph, B. and Le, Q.V. (2017) Swish: A Self-Gated Activation Function. arXiv:1710.05941.

- [10] Howard, A., Sandler, M., Chu, G., Wang, W., Chen, L.-C., Tan, M., *et al.* (2019) Searching for MobileNetV3. 2019 *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 27 October-2 November 2019, 1314-1324. <https://doi.org/10.1109/ICCV.2019.00140>
- [11] Misra, D. (2019) Mish: A Self Regularized Non-Monotonic Neural Activation Function. <http://arXiv.org/arXiv:1908.08681>
- [12] Nair, V. and Hinton, G.E. (2010) Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair. 2010 *International Conference on International Conference on Machine Learning*, Haifa, 21-24 June 2010, 807-814.
- [13] Dugas, C., Bengio, Y., Bélisle, F., *et al.* (2001) Incorporating Second-Order Functional Knowledge for Better Option Pricing. *Advances in Neural Information Processing Systems*, **13**, 472-478.
- [14] Glorot, X., Bordes, A. and Bengio, Y. (2011) Deep Sparse Rectifier Neural Networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research*, **15**, 315-323.
- [15] Maas, A.L., Hannun, A.Y. and Ng, A.Y. (2013) Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, 16-21 June 2013, 3.
- [16] Clevert, D.-A., Unterthiner, T. and Hochreiter, S. (2016) Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *Paper Presented at the Meeting of the 4th International Conference on Learning Representations (Poster)*, San Juan, 2-4 May 2016. <http://arxiv.org/abs/1511.07289>
- [17] Klambauer, G., Unterthiner, T., Mayr, A. and Hochreiter, S. (2017) Self-Normalizing Neural Networks. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 17)*, Long Beach, 4-9 December 2017, 972-981. arXiv:1706.02515.
- [18] Hendrycks, D. and Gimpel, K. (2016) Gaussian Error Linear Units (GELUs). arXiv:1606.08415.
- [19] He, K., Zhang, X., Ren, S. and Sun, J. (2015) Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1026-1034. <https://doi.org/10.1109/ICCV.2015.123>
- [20] Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A. and Bengio, Y. (2013) Maxout Networks. arXiv:1302.4389.
- [21] Xu, B., Wang, N., Chen, T. and Li, M. (2015) Empirical Evaluation of Rectified Activations in Convolutional Network. arXiv:1505.00853.
- [22] Davis, J. (2006) The Relationship between Precision-Recall and ROC Curves. *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, 25-29 June 2006, 233-240. <https://doi.org/10.1145/1143844.1143874>