symmetry

MDPI

Article

An Efficient Asymmetric Nonlinear Activation Function for Deep Neural Networks

Enhui Chai [1,†], Wei Yu [2,*], Tianxiang Cui [3,*], Jianfeng Ren [3,*] and Shusheng Ding [2]

[1] Baotou Teachers College, Inner Mongolia University of Science and Technology, Baotou 014030, China; chai1309787302@163.com
[2] School of Business, Ningbo University, Ningbo 315000, China; dingshusheng@nbu.edu.cn
[3] School of Computer Science, University of Nottingham Ningbo China, Ningbo 315000, China
* Correspondence: yuwei1@nbu.edu.cn (W.Y.); tianxiang.cui@nottingham.edu.cn (T.C.); jianfeng.ren@nottingham.edu.cn (J.R.)
† Current Address: School of Information Science and Technology, Northwest University, Xi'an 710127, China.

Abstract: As a key step to endow the neural network with nonlinear factors, the activation function is crucial to the performance of the network. This paper proposes an Efficient Asymmetric Nonlinear Activation Function (EANAF) for deep neural networks. Compared with existing activation functions, the proposed EANAF requires less computational effort, and it is self-regularized, asymmetric and non-monotonic. These desired characteristics facilitate the outstanding performance of the proposed EANAF. To demonstrate the effectiveness of this function in the field of object detection, the proposed activation function is compared with several state-of-the-art activation functions on the typical backbone networks such as ResNet and DSPDarkNet. The experimental results demonstrate the superior performance of the proposed EANAF.

Keywords: the neural network; the activation function; asymmetry; self-regular; non-monotonic; backbone network

Citation: Chai, E.; Yu, W.; Cui, T.; Ren, J.; Ding, S. An Efficient Asymmetric Nonlinear Activation Function for Deep Neural Networks. *Symmetry* **2022**, *14*, 1027. https://doi.org/10.3390/sym14051027

Academic Editor: José Carlos R. Alcantud

Received: 20 April 2022
Accepted: 15 May 2022
Published: 17 May 2022

1. Introduction

Deep neural networks have been widely used in many applications, e.g., handwritten digit recognition [1], style transfer [2], speech recognition [3], etc. As one of the most fundamental but important building blocks, the activation function plays an important role for deep neural network models [4].

One illustrative example is shown in Figure 1. Each neuron in the neural network generates a weighted summation of the outputs of previous neurons. The weighted summation is then added with the offset as defined in Equation (1), where x_i is the input from the i-th neuron of the previous layer, w_i is the corresponding weight, and b is the bias. Then, a non-linear activation function, e.g., the Sigmoid function [5] defined in Equation (2), is often applied to derive the output of this neuron. This output value will be then used as the input of the next layer. The quality of the activation function will determine the performance of the neural network.

$$f(x) = \sum_{i=1}^{m} w_i x_i + b \tag{1}$$

$$g(x) = Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{2}$$

The most commonly used activation functions are discussed in [6]. And a possible taxonomy is proposed to separate the trainable activation functions into two main categories: fixed shape and trainable shape. The authors of [7] evaluated the commonly used additive functions, such as Swish, Rectified Linear Unit (ReLU) and Sigmoid. The particular formula application recommendations of the activation functions are summarized based

on their properties. The authors of [8] proposed a technique for automatically designing novel, high-performing, parametric activation functions, PANGAEA, with the ability to discover general activation functions that perform well across architectures, and specialized functions that take advantage of a particular architecture.

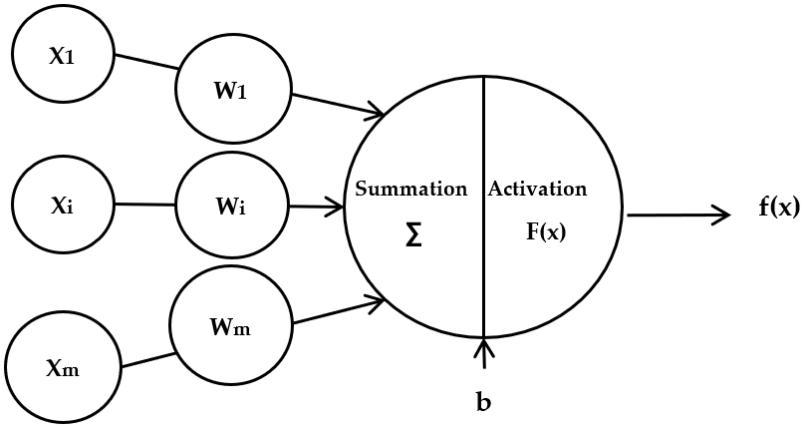


Figure 1. A typical neural network architecture.

In this work, we propose an Efficient Asymmetric Nonlinear Activation Function (EANAF) that combines the characteristics that activation functions should have, which are smoothness, asymmetry, soft saturation on the left, non-saturation on the right, non-monotonicity and self-regularization. As shown later in Section 3, by integrating these four desired characteristics into one activation function, the proposed EANAF could better capture the characteristics of input features and hence boost the classification performance of deep neural network.

To evaluate the proposed EANAF, we integrate it with several state-of-the-art deep convolutional neural networks for object detection, e.g., the ResNet [9], the backbone of RetinaNet [10], and the CSPDarkNet [11], the backbone of YOLO v4 [11]. We compared it with several state-of-the-art activation functions for object detection tasks on the large benchmark dataset, MS COCO [12]. The proposed EANAF consistently and significantly outperforms all the compared activation functions.

2. Related Work

Many activation functions have been developed in literature. Among them, the Sigmoid function [5], Tanh function [13] and Softsign function [14] were the first to be widely used. These functions are smooth and centrosymmetric. However, they suffer from vanishing gradients and monotonicity, which may lead to poor efficiency for deep convolutional neural networks. Subsequently, researchers developed the piecewise function ReLU [15,16], which has a strong generalization ability with fast convergence. However, it still has the problem of vanishing gradients. The ReLU6 function is a variant of the ReLU function with a restricted maximum value [17]. It remains as a piecewise linear function and has the advantage of less computational cost and the disadvantage of non-smoothness. This function was originally proposed in the MobileNet v1 network. Its main application scenario is the mobile platform under low-precision situations. It is shown that ReLU6 function could help improve neural networks' numerical resolution [17].

Some existing work tries to assign the negative part of ReLU to a function with a non-zero slope, such as the Leaky Rectified Linear Unit (LeakyReLU) function [18], in which the concept of "learning rate" is designed to solve the problem when the weights cannot be updated. The key idea of the ReLU function and its variants could be summarized as finding a suitable "learning rate" during the network training phase. The essence of this operation is to multiply the linear function in the negative direction of x with a non-zero slope. The "learning rate" in the LeakyReLU function is manually assigned by prior knowledge. Similarly, the "learning rate" in Parametric Rectified Linear Unit (PreLU) [19]

is changed according to the training data, but the learning rate in the Randomized Rectified Linear Unit (RReLU) function [20] is a value randomly generated by Gaussian distribution during the network training process. This value is averaged in the test phase as the new "learning rate" [20]. This function solves the problem of vanishing gradients in the negative direction, but it still has the problem of non-smoothness. The authors of [21] proposed a new activation function named the Parametric Rectified Nonlinear Unit (PRenu). This function is similar to ReLU. It solves the problem of vanishing gradients in the positive direction, but still has the problem of non-smooth and monotonicity.

To tackle the non-smoothness problem, many approaches have been developed. One of them is to combine the ReLU function and the Sigmoid function [5] to obtain the Exponential Linear Unit (ELU) function [22]. Numerous variants of the ELU function have been developed. ELU's variants are optimized in terms of independent variables and function parameters. For example, the Continuously Differentiable Exponential Linear Unit (CELU) function [23] replaces the x argument in the ELU function with x/a. To increase the convergence speed, the SELU function [24] is multiplied to the function value of the ELU function by the standardized parameter scale. The Gaussian Error Linear Unit, GELU [25], developed in 2016, breaks through the traditional way to optimize the activation function, i.e., to improve the ReLU function, people could also resort to improving the generalization ability. Common techniques include the dropout and the zone-out. The GELU function is equivalent to a combination of the ReLU, the dropout and the zone-out. Both ReLU and dropout would return the output of a neuron by zero. The difference is that ReLU would deterministically multiply the input by zero or one, while the dropout randomly multiplies the input by zero. The GELU function is asymmetric, which has been mostly used in the transformer model in recent years. In 2019, a symmetrical GELU function, Scaled Exponential Linear Unit (SGELU) [26] was developed. This function incorporates symmetry into GELU. As a result, the SGELU function not only has the advantages of the GELU function but also has the property of bi-directional convergence. The GELU function is a piecewise function. The softplus function [27] is combined with an exponential logarithmic function to obtain a continuous function expression. This idea retains the advantage of non-saturation on the right side of the function. By combining the logarithmic function to reduce the gradient, the problem of overfitting can be minimized.

More recently, Prajit Ramachandran et al. [28] developed an activation function called Searching for Activation Functions (Swish) based on the Sigmoid function [5]. The function is demonstrated to have a high degree of fitness. To further improve the fitness, Digita Misra et al. [29] developed the Mish activation function based on the Tanh function [13]. They claimed that this function can achieve a better fitness compared with the Swish function. However, the Mish function is computationally intensive. Andrew Howard et al. [30] developed the h-swish function in the lightweight neural network MobileNet V2. This function comes with less computational effort than the Swish function. However, h-swish function is not smooth, which may result in a poor classification performance.

3. Proposed EANAF

3.1. Formulation

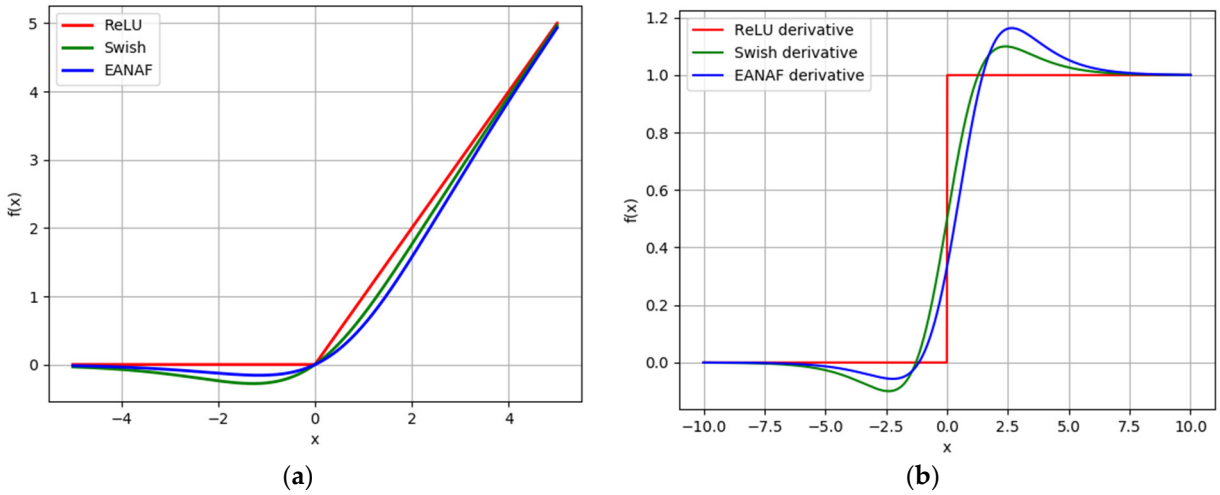
To tackle the challenges of existing activation functions, we develop an Efficient Asymmetric Non-linear Activation Function (EANAF). The proposed EANAF is computationally efficient and preserves the contribution to accuracy in the target detection domain. Mathematically, it is defined as the following:

$$\text{EANAF}(x) = x * g(h(x)) \tag{3}$$

where $h(x) = log(1 + e^x)$ is the Softplus function, and $g(x) = tanh(x/2) = \frac{e^{x/2} - e^{-x/2}}{e^{x/2} + e^{-x/2}} = \frac{e^x - 1}{e^x + 1}$ is the tanh function. The proposed EANAF function can be further simplified as,

$$\text{EANAF}(x) = \frac{xe^x}{e^x + 2} \tag{4}$$

We can see that the proposed EANAF looks like a combination of several activation functions, e.g., Softplus [27], tanh [13] and Swish function [28]. Indeed, the proposed EANAF inherits the advantages of these activation functions. In terms of mathematical properties, EANAF is similar to Swish, with approximately the same amount of computation. But EANAF has better fitness and contributes more to the network in terms of training efficiency compared with Swish. Figure 2 illustrates the EANAF function, the ReLU function and the Swish function with the corresponding first-order derivative curves. Although these three functions look similar at the first glance in Figure 2a, the proposed EANAF produces a larger response for the first-order derivative as shown in Figure 2b. Intuitively, the proposed EANAF could better handle the problem of gradient vanishing.



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

Figure 2. (**a**) The comparative graph and (**b**) the first-order derivatives of the ReLU function, the Swish function and the proposed EANAF.

Compared with typical activation functions in recent years, the proposed EANAF has the desired characteristics of excellent activation functions such as smoothness, asymmetry, soft saturation on the left, non-saturation on the right, non-monotonicity and self-regularization. In the next subsection, these desired properties will be illustrated in detail, along with how they may help the proposed EANAF achieve better performance.

3.2. Analysis of Proposed EANAF

3.2.1. Smoothness

A function is smooth if it is continuously derivable of infinite order in its domain. The smoothness property can bring many advantages to an activation function. Firstly, smoothness can ensure that there is no step change in the activated value of the activation function, which could help convergence. Secondly, the smooth function is often nonlinear, and it could fit better to complex patterns. Most importantly, the smooth function ensures that the function can be continuously derived, which facilitates the calculation and update of the gradient. Finally, smoothness can help specify more flexible gradient update rules to speed up model training. In literature, the Sigmoid function [5], Tanh function [13], Softsign function [14], ELU function [22], Softplus function [27] and Swish function [28] are all smooth functions. The ReLU function [15,16], ReLU6 function [17], LeakyReLU function [18] and h-swish function are all non-smooth functions.

The proposed EANAF has a smoothness property. It can be continuously differentiable, and its first-order derivative is shown in Equation (4).

$$\text{EANAF}(x)' = \frac{(e^x)^2 + 2(x+1)e^x}{(e^x+2)^2} \tag{5}$$

3.2.2. Asymmetry

A function is asymmetric if it is neither odd nor even symmetric. The asymmetric activation function has the advantage of soft saturation on the left and unsaturation on the right, which can help separate positive and negative samples more effectively [5]. If an activation function is a symmetric function, the function is either an odd function or an even function, i.e., f(−x) = −f(x) or f(−x) = f(x). Such a function would cause the weights in the neural network to be updated in only one direction. The activation function also needs soft saturation on the left side to improve robustness. Therefore, most activation functions are designed to be almost symmetric near the origin, but not completely centrosymmetric. For example, the Sigmoid function [5], Tanh function [13] and softsign function [14] are all centrosymmetric functions. The ReLU function [15,16] and its variants [17,18,22,25], ELU function [22], GELU function [25], Softplus function [27], Swish function [28] and h-swish function [30] are all asymmetric functions.

The non-centrosymmetric EANAF is reflected by the soft saturation on the negative interval of x and thus contributes more to the robustness of the neural network. This feature can help the neural network with the EANAF separate positive and negative samples more effectively.

3.2.3. Unsaturation

If an activation function f(x) satisfies $\lim_{n\to+\infty} h'(x) = 0$, we call it a right saturation function. In contrast, when an activation function f(x) satisfies $\lim_{n\to-\infty} h'(x) = 0$, we call it a left saturation function. In particular, when an activation function f(x) satisfies $\lim_{n\to+\infty} h'(x) \to 0$, we call it the right-hand soft saturation function and if an activation function f(x) satisfies $\lim_{n\to-\infty} h'(x) \to 0$, we call it the left-side soft saturation function. A function is a saturated function if it is both right-saturated and left-saturated. And a function is an unsaturated function if it is unsaturated on both sides, or one side is unsaturated.

Since the unsaturated function does not have a derivative of zero, this type of function does not have the problem of vanishing gradients during the training phase. The left-side soft-packet sum will be robust to noise for negative samples. Typical saturation functions include the Sigmoid function [5], Tanh function [13], ReLU6 function [17], and Softsign function [14] while typical non-saturating functions include the ReLU function [15,16], ELU function [22], GELU function [25], Softplus function [27], Swish function [28] and h-swish function [30].

The EANAF function is a right-hand unsaturated function. It solves the problem of vanishing gradients during the training stage. At the same time, EANAF also belongs to the left soft saturation function. Therefore, when extracting features for negative samples, it is highly robust to noise in the dataset.

3.2.4. Non-Monotonicity and Self-Regularity

The monotonicity of a function is also called the increase or decrease of a function. This property can qualitatively describe the relationship between the change of a function value and the change of an independent variable within a specified interval. When the value of the function increases continuously with the increase in the independent variable or the value of the function continues to decrease with the increase in the independent variable, we call the function monotonic. If the independent variable and the function value do not follow such relationships, we call the function non-monotonic. Meanwhile, the complex variable function that is differentiable everywhere in the domain is called the regular function. Functions with non-monotonicity do not have the problem of single directional weights. Non-monotonicity may increase the expressiveness of the function and hence improve the gradient flow. This desired property can also provide some robustness to different initializations and learning rates. The significance of self-regularity for activation functions is particularly important. Activation functions often face two major problems. One is underfitting, which is also called high bias. For example, in the case of $f(x) = \theta_0 + \theta_1 x$, the

value of θ_0 (bias) is high, and hence the fitness is low. It commonly exists in linear activation functions. To solve the underfit problem, nonlinear activation functions are developed. But this may also introduce another type of problem, which is overfit, also known as high variance. For example, in the case of $f(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 \cdots$, the function has a high fitness. But the values of high-order coefficients are sensitive to the noise in the data. To avoid overfitting, it is often desired to constrain the number of model parameters. Generally speaking, there are two approaches for addressing the overfitting problem, artificially reducing the number of variables and regularization. Among them, artificial reduction of variables may delete some effective variables and result in an unsatisfactory model. Therefore, the self-regularity of the activation function is favored since it can reduce the magnitude of feature variables while preserving them. L2-regularization is often used as shown in Equation (5), where $h_\theta\left(x^{(i)}\right)$ represents a polynomial with respect to x. It is used to fit the corresponding true value $y^{(i)}$. m represents the number of fitted terms. $\lambda \sum_{i=1}^{n} \theta_j^2$ is the regularization term. λ is the regularization coefficient and it is used to balance fitting the training objective and keeping the parameter values small.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} \theta_j^2\right] \tag{6}$$

As shown in Figure 2a, the proposed EANAF has the property of non-monotonicity. The non-monotonicity ensures the non-uniformity of the direction of the network with EANAF when the weights are updated in the training phase. From Equation (6), it can be seen that the proposed EANAF is a function that can be differentiated everywhere in its domain. Therefore, the proposed EANAF has self-regularity, which can help prevent the overfitting problem.

$$\frac{\mathrm{d}(\mathrm{EANAF}(x))}{dx} = \frac{(e^x)^2 + 2(x+1)e^x}{(e^x+2)^2}. \tag{7}$$

3.3. Discussions

Among them, the ReLU function [15,16] has some desired properties such as being less computationally intensive. Therefore, the ReLU function [15,16] is often used as the activation function in many neural networks. But the ReLU function has the problem of vanishing gradient when the input is negative [28]. Therefore, researchers developed the Swish function [28] to address this problem. In this work, we use the ReLU [15,16] and the Swish [28] as two benchmark activation functions for comparisons. We summarize the characteristics of commonly used activation functions in Table 1. The proposed EANAF is among the very few that have all the four desired properties: smoothness, asymmetry, unsaturation and non-monotonicity and self-regularity. Firstly, the proposed EANAF can be continuously derived. Therefore, there is no step change in the function, which can guarantee a good convergence during training. Secondly, the proposed EANAF uses an exponential function combined with a logarithmic function, which has a high fit for complex problems. The asymmetry of EANAF further helps to separate positive and negative samples efficiently. Thirdly, because of the non-saturation and non-monotonicity properties, the proposed EANAF solves the problem of vanishing gradient. Finally, the self-regularity of EANAF helps alleviate the problem of overfitting. One apparent drawback of the proposed EANAF is that its computational load is slightly higher than the ReLU function [15,16]. But it is comparable to the Swish function [28].

Table 1. The characteristics of common activation functions. Tick means the activation function satisfies certain property while cross means the opposite.

Property	ReLU [15,16]	LeakyReLU [18]	ReLU6 [17]	ELU [22]	GELU [25]	Sigmoid [5]	Tanh [13]	Softsign [14]	Softplus [27]	Swish [28]	h-Swish [30]	EANAF
Smoothness	×	×	×	√	√	√	√	√	√	√	×	√
Asymmetry	√	√	×	√	√	×	×	×	√	√	√	√
Unsaturation	√	√	×	√	√	×	×	×	×	√	√	√
Non-monotonicity and self-regularity	√	√	×	√	×	×	×	×	√	√	√	√

4. Experimental Results

Comparison experiments were carried out for the target detection on the MS COCO data set [12], which contains a rich set of object types and object sizes. We adopt the controlled variable method to conduct the following three sets of experiments. The input images are all resized to 608 × 608 pixels.

4.1. Experimental Settings

We selected RetinaNet [10] and YOLO-v4 [11] as two representative networks, with the backbone as ResNet [9] and the CSPDarkNet [11], respectively. More precisely, the backbone of the RetinaNet [10], ResNet [9], can be further divided into two parts. The first part consists of the convolution operation, standard layer, activation function and maximum pooling layer while the second part of the consists of four groups of the "Conv Block" and the "Identity Block" with different operation times. Figure 3 illustrates the backbone structure of ResNet (with the activation highlighted).

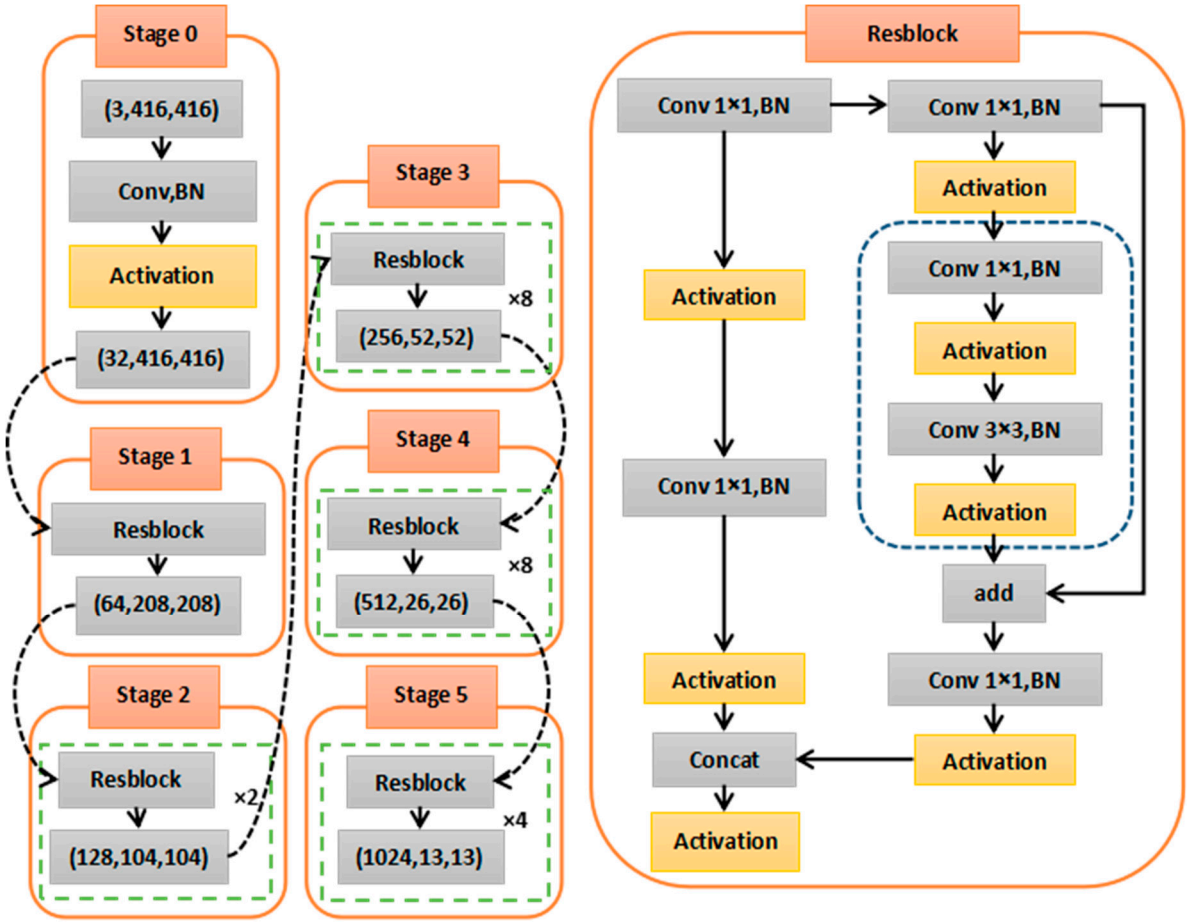


Figure 3. The structure diagram of the ResNet [9].

Figure 4 illustrates the structure diagram of CSPDarknet [11] (with the activation highlighted). It can be further divided into two parts. The first part sequentially performs convolution processing, normalization processing and activation function processing on the input image while the second part is composed of five groups of the "Resblock" modules with different layers. The three-layer output of YOLO v4 [11] is the feature layer processed by the deep "Resblock" module.

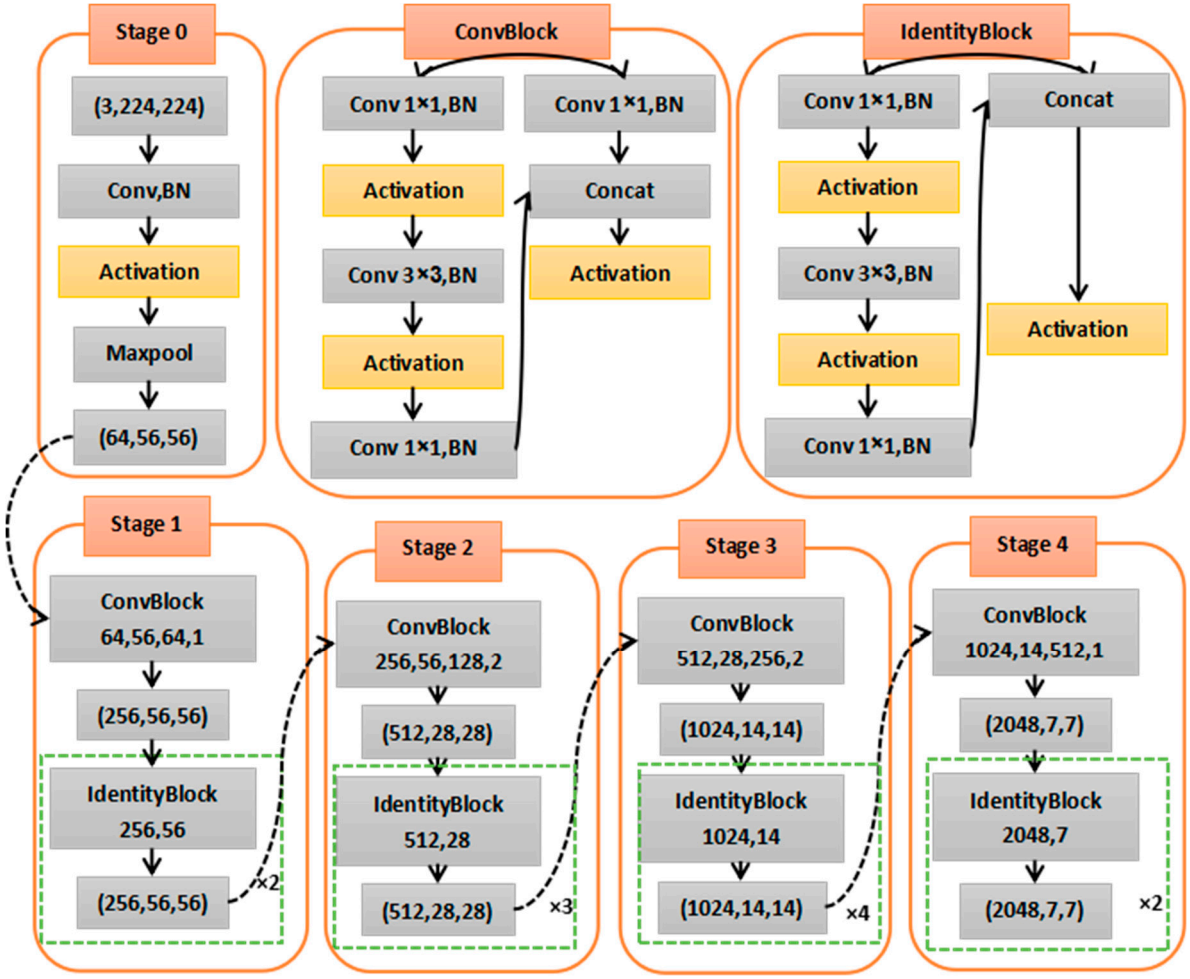
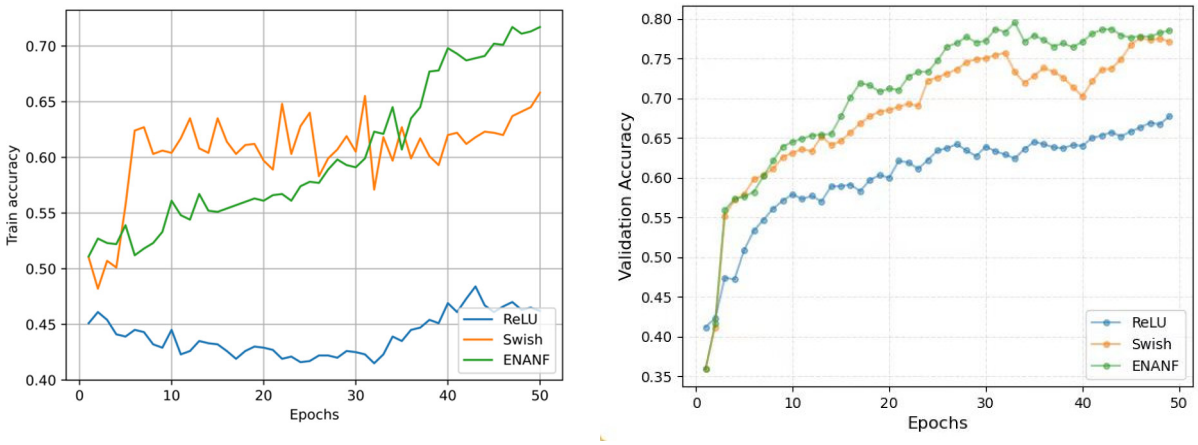


Figure 4. The structure diagram of the CSPDarknet [11].

We choose ReLU [15,16] and Swish [28] as two benchmark activation functions for comparison. Compared with the LeakyReLU function [18] and the ReLU6 function [17], ReLU [15,16] not only has less computation, but also satisfies the right-side non-saturation and stronger negative sample robustness. Compared with functions such as the ELU function [22] and GELU [25], the lower computational complexity of ReLU function [15,16] is a big advantage. The Sigmoid function [5], Tanh function [13] and Softsign function [14] are not suitable for the convolutional neural networks. The Swish function [28] is chosen because it has all the four desired properties for activation functions, while the non-smoothness of the h-swish function [30] makes it only suitable for light networks. Figure 5a shows the validation accuracy of the ReLU function, the Swish function and the proposed EANAF during the training phase and Figure 5b shows the validation accuracy of the ReLU function, the Swish function and the proposed EANAF during the testing phase. It can be easily seen that the proposed EANAF outperforms ReLU and Swish in both training and testing phases.



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

Figure 5. The training/test accuracies of different activation functions. (**a**) Training accuracy. (**b**) Test accuracy.

We conducted two sets of comparative experiments. The proposed EANAF is compared with ReLU [15,16] and Swish [28] on the ResNet [9], the backbone of RetinaNet [10], and the CSPDarknet [11], the backbone of YOLO v4 [11]. The evaluation metric is reflected by the detection accuracy of the model. In this work, we use mAP, AP50 and AP75 as our evaluation metrics. mAP represents the mean Average Precision (AP) across all object classes in MS COCO dataset. The formula for AP is shown in Equation (8), where r_i and p_i are the precision and recall at i-th threshold, n is the number of the threshold. Precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances among the retrieved instances. AP50 represents the average detection accuracy when the overlap ratio is 0.5 while AP75 represents the average detection accuracy when the overlap ratio is 0.75.

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_i \tag{8}$$

4.2. Comparison Experiments on ResNet

RetinaNet [10] has been widely used for object detection [9]. It uses the ResNet [9] network as the backbone. The network architecture of ResNet [9] is shown in Figure 3. We conduct comparison experiments for different activation functions, the ReLU function [15,16], Swish function [28] and the proposed EANAF. The other components of the network remain unchanged while only the activation functions are different. The detection accuracy on 24,868 images of the MS COCO dataset [12] is reported in Table 2. It can be seen that the proposed EANAF achieves the best performance. By utilizing the proposed EANAF activation function, the mAP increases from 35.7% to 37.1% compared with the second-best activation function, Swish on the backbone of RetinaNet [10], ResNet. The performance gain is most notable for AP50, where the proposed EANAF significantly outperforms Swish [28] by 6.9% and outperforms ReLU [15,16] by 9.1%. We can see that the proposed activation function can bring a significant and consistent performance gain over other state-of-the-art activation functions on the ResNet [9].

Table 2. The comparison of activation functions on the MS COCO [12] using ResNet [9] as the backbone.

Activation Function	mAP	AP50	AP75
ReLU [15,16]	32.5%	50.9%	34.8%
Swish [28]	35.7%	53.1%	36.8%
Proposed EANAF	**37.1%**	**60.0%**	**37.4%**

Figures 6–8 show the visualization results of three different activation functions when using ResNet [9] as the backbone network. Among them, Figure 6 shows the visual results when there are many types of objects in the image, and the occlusion between objects is serious. It can be seen that some objects are not detectable when using the ReLU function and the Swish function. Compared with the other two functions, more objects can be successfully detected when using the proposed EANAF. For instance, in the first image, only one "chair" is detected using the ReLU function. Although various objects are detected using the Swish function, some objects are not correctly detected (e.g.,"dining-table" is mistakenly detected as "clock"). In contrast, all the objects are correctly detected when using the proposed EANAF. In the second image, only six "person" can be detected using the ReLU function. All seven "person" are detected when using the Swish function and EANAF. However, the higher confidence level is obtained by using the proposed EANAF. In the third image, three "person", two "car" and one "bicycle" are detected using the ReLU function and eight "person", three "car" and one "bicycle" are detected using the Swish function. All objects (ten "person", three "car" and one "bicycle") are correctly detected when using the proposed EANAF.

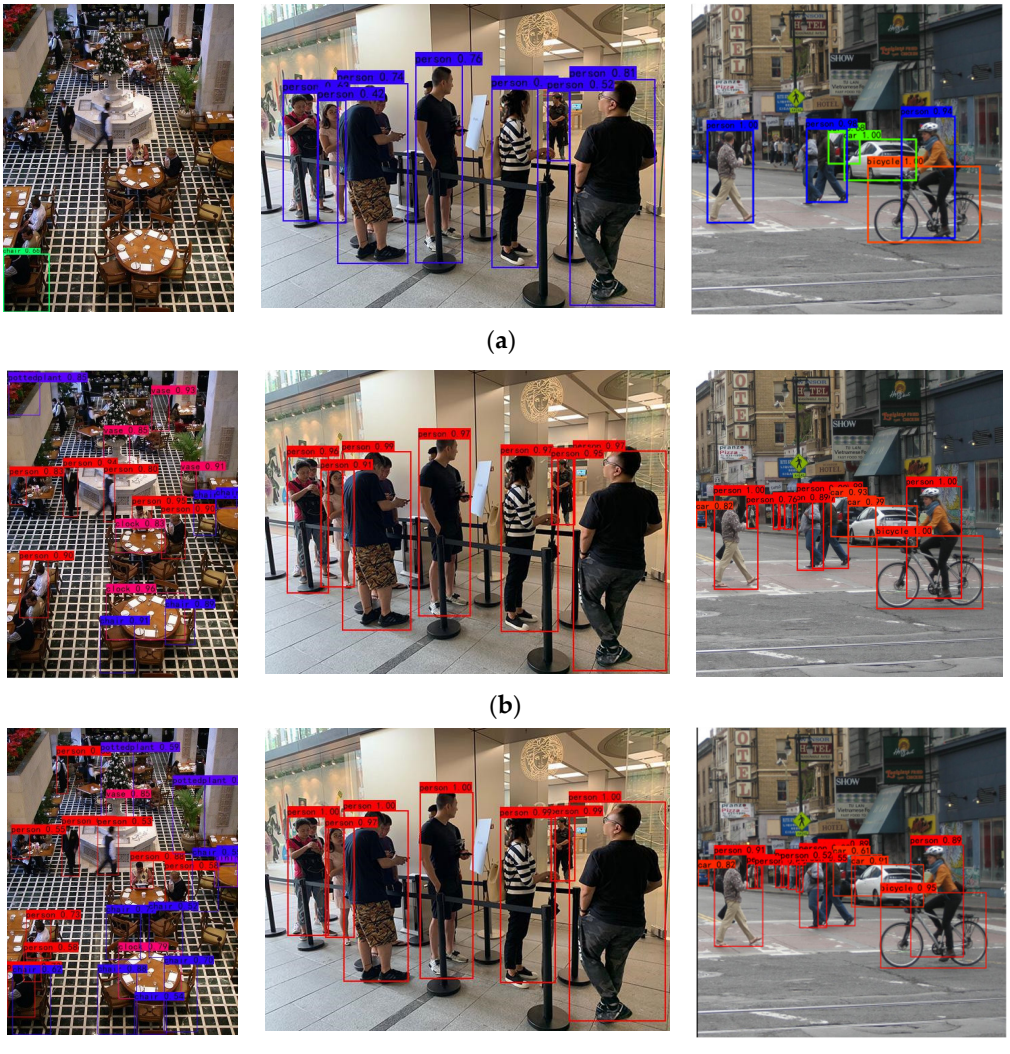


Figure 6. The visual results of using the ReLU function [15,16], the Swish function [28] and the proposed EANAF, respectively, as the activation function for ResNet [9] backbone of RetinaNet [10] when there are many types of objects and the occlusion is serious. (**a**) The visual results of using ReLU. (**b**) The visual results of using Swish. (**c**) The visual results of using EANAF.

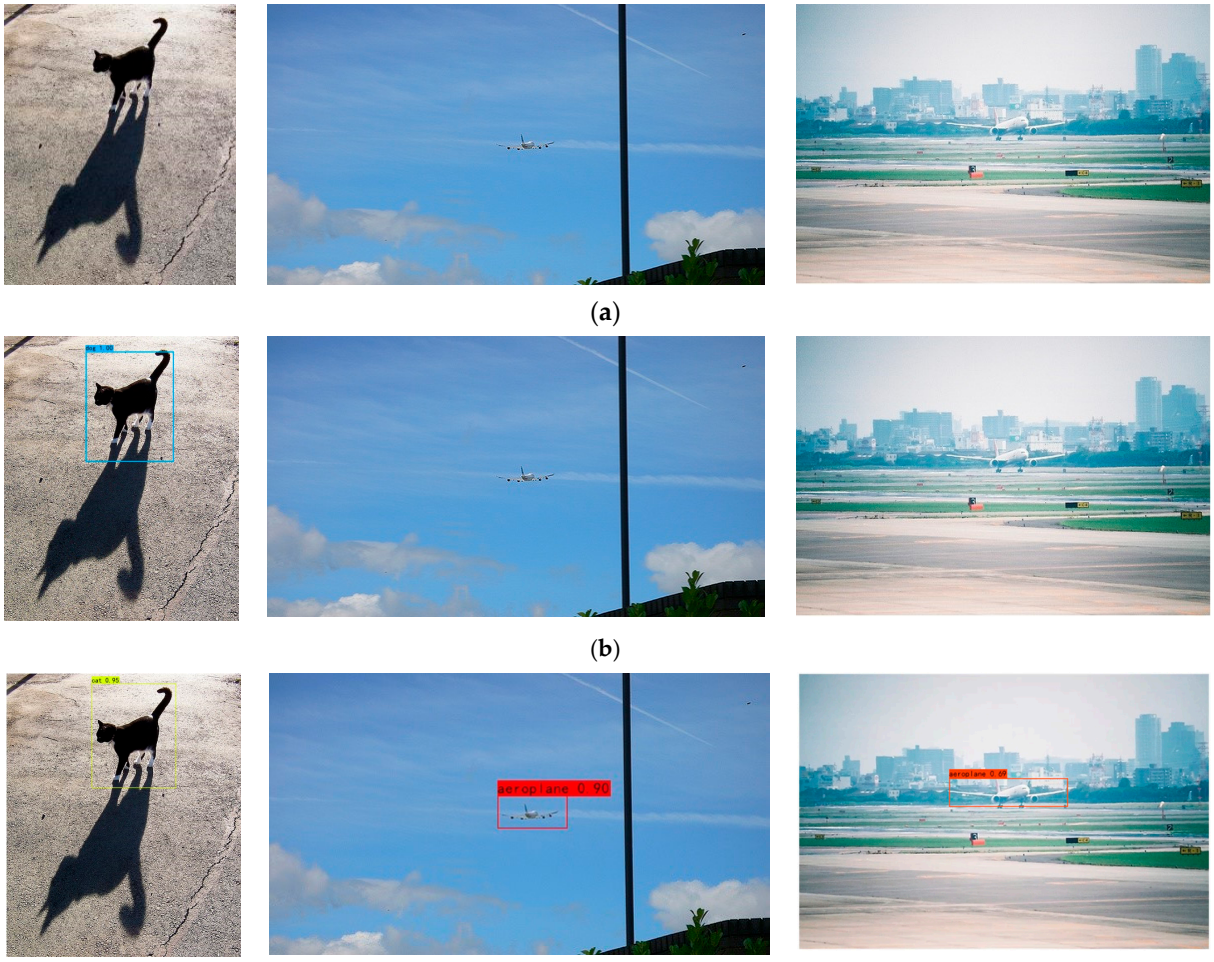


Figure 7. The visual comparisons of using the ReLU function [15,16], the Swish function [28] and the proposed EANAF respectively as the activation function for ResNet [9] backbone of RetinaNet [10] for small-object detection in a complex environment. (**a**) The visual results of using ReLU. (**b**) The visual results of using Swish. (**c**) The visual results of using EANAF.

Figure 7 shows the detection results when the illumination variation in the image may have a great influence on the detection results. It can be seen that the positive and negative samples are not well detected using both the ReLU function and the Swish function. As a result, no objects are correctly detected when using ReLU or Swish. The proposed EANAF shows a relatively robust performance, and all objects are correctly detected. Consequently, the proposed EANAF can help improve the feature extraction and effectively detect objects.

Figure 8 shows the detection results when the small objects are in a complex environment. We can see that there are some false detections and missed detections when using the ReLU function and the Swish function. In contrast, the proposed EANAF can detect more objects with a better accuracy. It can be seen that the proposed EANAF works well for small-object detection compared with ReLU and Swish.

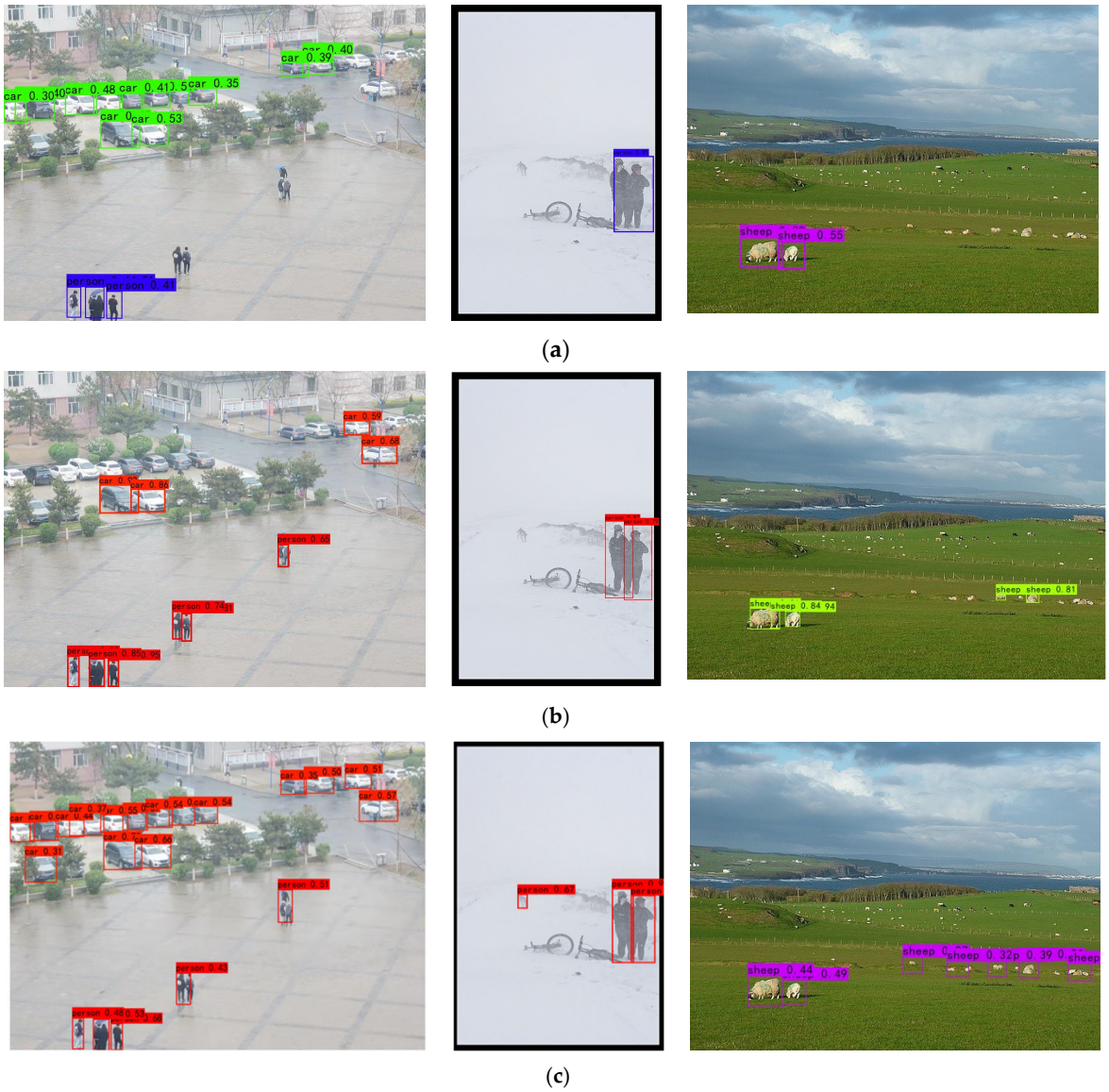


Figure 8. The visual comparisons of using the ReLU function [15,16], the Swish function [28] and the proposed EANAF respectively as the activation function for ResNet [9] backbone of RetinaNet [10] for small-object detection in a complex environment. (**a**) The visual results of using the ReLU function. (**b**) The visual results of using the Swish function. (**c**) The visual results of using the EANAF function.

4.3. Experimental Results on CSPDarknet

YOLO v4 [11] achieves state-of-the-art performance for various object detection tasks [9]. It utilizes the CSPDarkNet [11] as the backbone network. We conduct comparison experiments using ReLU function [15,16], Swish function [28] and the proposed EANAF as the activation function for CSPDarkNet [11]. Other components of the network remain unchanged. The detection results on the 24,868 images of the MS-COCO dataset are summarized in Table 3.

First of all, we can see that the performance on the YOLO v4 [11] using CSPDarkNet [11] as the backbone consistently outperforms the performance on the RetinaNet [10] using ResNet [9] as the backbone. Indeed, YOLO v4 [11] has achieved state-of-the-art performance on the MS-COCO dataset for object detection. Secondly, the proposed EANAF significantly outperforms the second-best activation function, Swish [28], by 2%, 1.9% and 2.0% in terms of mAP, AP50 and AP75, respectively. The consistent performance gain demonstrates the superiority of the proposed EANAF. Lastly, on both ResNet [9] and CSP-

DarkNet [11], which are the backbones for popular object detection models, RetinaNet [10] and YOLO v4 [11], respectively, the proposed EANAF consistently and significantly outperforms all the compared state-of-the-art activation functions, which demonstrates the effectiveness of the proposed EANAF.

Table 3. The comparison of activation functions on the MS COCO [12] using CSPDarkNet [11] as the backbone of YOLO v4 [11].

Activation Function	mAP	AP50	AP75
ReLU [15,16]	39.6%	58.6%	42.3%
Swish [28]	41.2%	63.8%	45.3%
Proposed EANAF	**43.2%**	**65.7%**	**47.3%**

The visual results of various activation functions using the CSPDarknet [11] as the backbone of the YOLO v4 model [11] are shown in Figures 9–11. Again, we can observe that some objects are not detected by other activation functions, or previously not by RetinaNet [10], but now can be well detected by YOLO v4 [11] using the proposed EANAF as the activation function. Figure 9 shows the visual results when there are many types of objects in the image. Although the overall structure of the CSPDarknet network has been improved with a better detection accuracy, the proposed EANAF still outperforms ReLU and Swish in terms of the number of the detected objects and the detection accuracy. For example, in the first image, only two objects are detected using the ReLU function. Six "person", two "diningtable", four "chair" and one "pottedplant" are detected using the Swish function. In contrast, seven "person", six "diningtable", nine "chair" and two "pottedplant" are detected using the proposed EANAF.

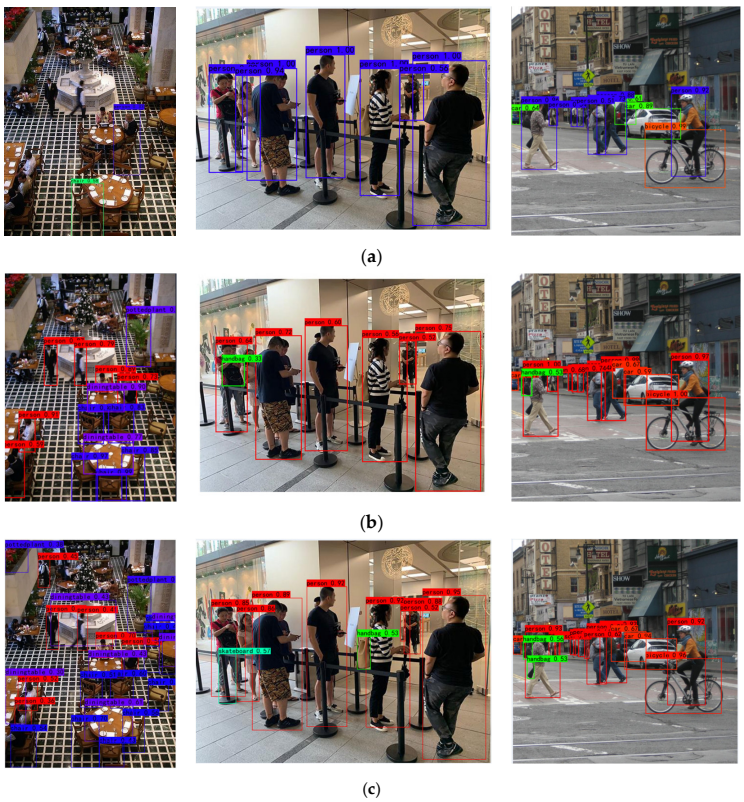


Figure 9. The visual comparison of using the ReLU function [15,16], the Swish function [28] and the proposed EANAF as the activation function for the CSPDarknet [11] backbone of YOLO v4 [11] when there are many types of objects, and the occlusion is serious. (**a**) The visual results of using ReLU. (**b**) The visual results of using Swish. (**c**) The visual results of using EANAF.

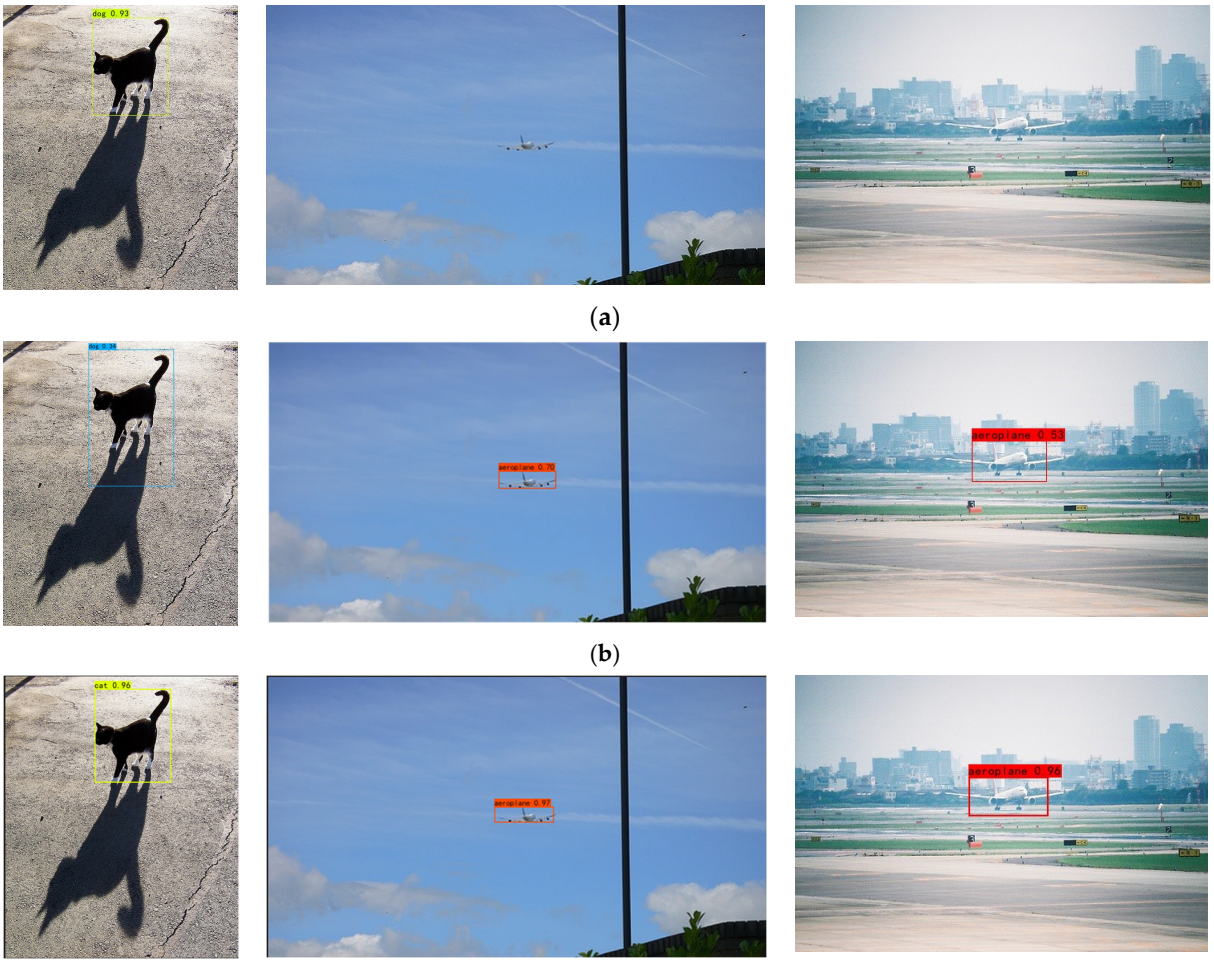


Figure 10. The visual comparisons of using the ReLU function [15,16], the Swish function [28] and the proposed EANAF as the activation function for the CSPDarknet [11] backbone of YOLO v4 [11] when the illumination variation has a great influence on detecting objects in the image. (**a**) The visual results of using ReLU. (**b**) The visual results of using Swish. (**c**) The visual results of using EANAF.

Figure 11. *Cont.*

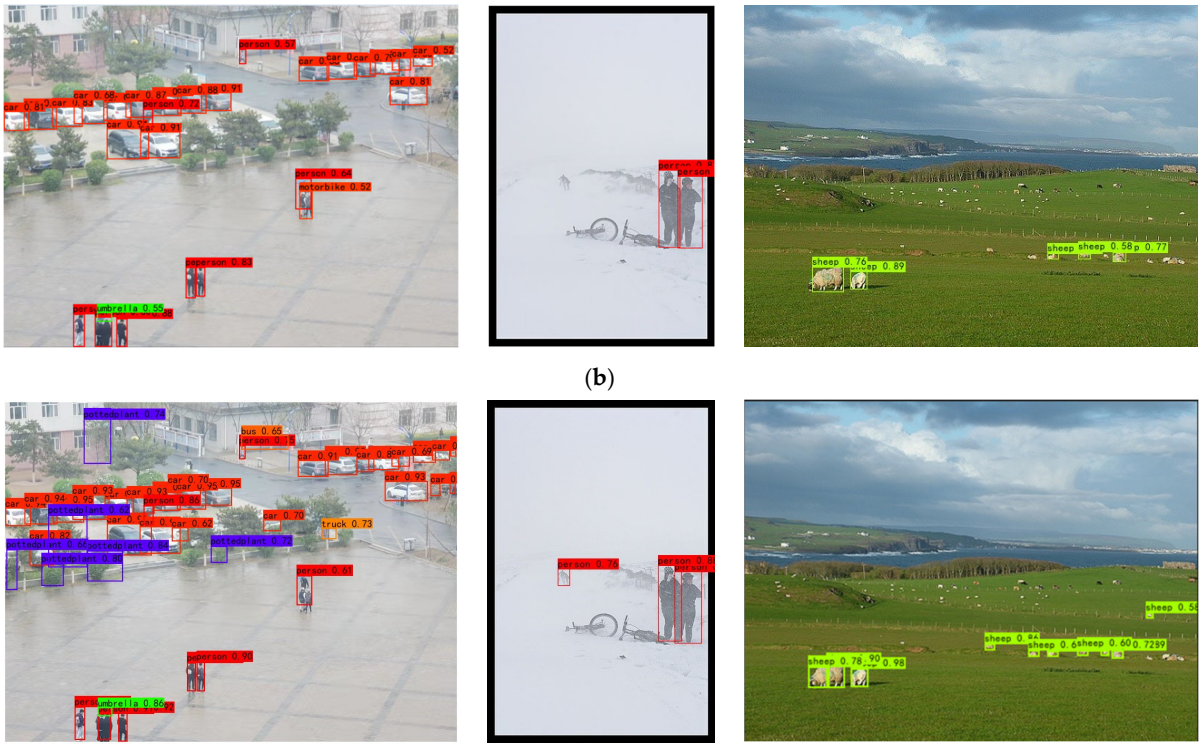


Figure 11. The visual results of using the ReLU function [15,16], the Swish function [28] and the proposed EANAF as the activation function for the CSPDarknet [11] backbone of YOLO v4 [11] for small-object detection in a complex environment. (**a**) The visual results of using ReLU. (**b**) The visual results of using Swish. (**c**) The visual results of using EANAF.

Figure 10 shows the detection results when the illumination variation in the image has a great influence on the detection results. Compared with the previous experiments on ResNet, some objects can be detected using the Swish function, but the confidence level is low. Our proposed EANAF can obtain a better detection accuracy with a higher confidence level. Figure 11 shows the detection of small objects in the complex background. We can clearly see that the objects detected by the proposed EANAF are more comprehensive.

4.4. Comparisons of Efficiency

Experiments were conducted to compare the computational impact of the activation functions on the target detection models. Table 4 shows the comparison of the forward and reverse transfer runtimes of ReLU [15,16], Swish [28] and the proposed EANAF activation functions for floating-point 32 data. ReLU [15,16] has the least runtimes, but it does not have all the desired properties of the activation functions, and it does not work very well as demonstrated previously. Compared with Swish, the proposed EANAF utilizes similar runtimes, but achieves a better detection performance, as demonstrated previously in Tables 2 and 3.

Table 4. The forward and reverse transfer run-time of various activation functions.

Activation	Data Type	Forward Pass	Backward Pass
ReLU	Fp32	224.2 µs ± 621.8 ns	419.3 µs ± 1.238 µs
Swish	Fp32	342.7 µs ± 1.026 µs	497.3 µs ± 1.357 µs
EANAF	Fp32	372.0 µs ± 1.852 µs	529.1 µs ± 1.882 µs

5. Conclusions

In this work, we propose the Efficient Asymmetric Nonlinear Activation Function (EANAF) for deep neural networks. The proposed EANAF has many desired properties such as smoothness, asymmetry, soft saturation on the left, non-saturation on the right, non-monotonicity and self-regularization. Because of smoothness, it is differentiable everywhere. Asymmetrical design makes it a better fit for a complex problem. The non-saturation and non-monotonicity, address the problem of vanishing gradient. Lastly, the proposed EANAF is self-regularized to minimize the risk of overfitting. Comparative experiments are conducted on two typical backbone networks, ResNet and DSPDarkNet, for object detection on the MS COCO dataset. The computational results demonstrate the superior performance of the proposed EANAF, compared with other state-of-the-art activation functions.

Subsequent work will be conducted with more detailed analysis and experiments on the proposed activation function in other image processing tasks. The combination of multiple activation functions will be another direction to be further explored. In future, our proposed activation function will be applied to more computer vision tasks such as object recognition and instance segmentation.

Author Contributions: Conceptualization, E.C. and T.C.; methodology, E.C. and T.C.; validation, E.C., T.C. and J.R.; formal analysis, W.Y. and J.R.; investigation, J.R. and S.D.; resources, W.Y.; data curation, E.C.; writing—original draft preparation, E.C.; writing—review and editing, T.C. and J.R.; visualization, S.D.; supervision, T.C. and J.R.; project administration, T.C.; funding acquisition, W.Y. and T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by University of Nottingham NRG grant (Grant Number: I03211200008). This work was also supported in part by the National Natural Science Foundation of China under Grant 72071116, and in part by the Ningbo Municipal Bureau Science and Technology under Grants 2019B10026.

Data Availability Statement: The MS-COCO dataset is publicly available from (https://cocodataset.org/#home) accessed on 17 January 2022.

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

References

1. Lecun, Y.; Bottou, L. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
2. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
3. Cheng, J.; Dong, L.; Lapata, M. Long Short-Term Memory-Networks for Machine Reading. *arXiv* **2016**, arXiv:1601.06733.
4. Bishop, C.M. Neural Networks for Pattern Recognition. In *Advances in Computers*; Clarendon Press: Oxford, UK, 1993; pp. 119–166.
5. Yukun, S.; Xiaohang, G.; Duoli, Z.; Gaoming, D. The piecewise non-linear approximation of the sigmoid function and its implementation in FPGA. *Appl. Electron. Technol.* **2017**, *43*, 49–51.
6. Apicella, A.; Donnarumma, F.; Isgrò, F.; Prevete, R. A survey on modern trainable activation functions. *Neural Netw.* **2021**, *138*, 14–32. [CrossRef]
7. Szandaa, T. Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. In *Bio-Inspired Neurocomputing*; Springer: Singapore, 2020; pp. 203–224.
8. Bingham, G.; Miikkulainen, R. Discovering Parametric Activation Functions. *Neural Netw.* **2022**, *148*, 48–65. [CrossRef]
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2016**, arXiv:1512.03385.
10. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 318–327. [CrossRef]
11. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
12. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision 2014, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
13. Huaguang, Z.; Zhiliang, W.; Ming, L.I.; Quan, Y.-B. Generalized Fuzzy Hyperbolic Model: A Universal Approximator. *J. Autom. Sin.* **2004**, *30*, 416–422.
14. Chang, C.H.; Zhang, E.H.; Huang, S.H. Softsign Function Hardware Implementation Using Piecewise Linear Approximation. In Proceedings of the 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Taipei, Taiwan, 3–6 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–2.

15. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Association for Computing Machinery: New York, NY, USA, 2012; pp. 1097–1105.
16. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; Omnipress: Madison, WI, USA, 2010; pp. 807–814.
17. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
18. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the ICML, Atlanta, GA, USA, 16–21 June 2013; p. 3.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv* **2015**, arXiv:1502.01852.
20. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853.
21. El Jaafari, I.; Ellahyani, A.; Charfi, S. Parametric rectified nonlinear unit (PRenu) for convolution neural networks. *Signal Image Video Process.* **2021**, *15*, 241–246. [CrossRef]
22. Clevert, D.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units. *arXiv* **2015**, arXiv:1511.07289.
23. Barron, J.T. Continuously Differentiable Exponential Linear Units. *arXiv* **2017**, arXiv:1704.07483v1.
24. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. *arXiv* **2017**, arXiv:1706.02515.
25. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
26. Chao, Y.; Su, Z. Symmetrical Gaussian Error Linear Units (SGELUs). *arXiv* **2019**, arXiv:1911.03925.
27. Dugas, C.; Bengio, Y.; Belisle, F.; Nadeau, C. Incorporating second order functional knowledge into learning algorithms. In *Advances in Neural Information Processing Systems 13, Proceedings of the 2000 Neural Information Processing Systems (NIPS) Conference, Denver, CO, USA, 28–30 November 2000*; MIT Press: Cambridge, MA, USA, 2000; pp. 472–478.
28. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
29. Misra, D. Mish: A Self Regularized Non-Monotonic Neural Activation Function. *arXiv* **2020**, arXiv:1908.08681.
30. Howard, A.; Sandler, M.; Chu, G.; Chen, L.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244.