IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 April 2023, accepted 3 May 2023, date of publication 15 May 2023, date of current version 19 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276298

RESEARCH ARTICLE

HcLSH: A Novel Non-Linear Monotonic Activation Function for Deep Learning Methods

HEBA ABDEL-NABI[1], GHAZI AL-NAYMAT[2], MOSTAFA Z. ALI[3], (Senior Member, IEEE), AND ARAFAT AWAJAN[1,4]

[1]Department of Computer Science, Princess Sumaya University for Technology, Amman 11941, Jordan
[2]Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates
[3]Faculty of Computer and Information Technology, Jordan University of Science & Technology, Irbid 22110, Jordan
[4]Computer Science Department, Mutah University, Karak 61710, Jordan

Corresponding author: Ghazi Al-Naymat (g.alnaymat@ajman.ac.ae)

ABSTRACT Activation functions are essential components in any neural network model; they play a crucial role in determining the network's expressive power through their introduced non-linearity. Rectified Linear Unit (ReLU) has been the famous and default choice for most deep neural network models because of its simplicity and ability to tackle the vanishing gradient problem that faces backpropagation optimization. However, ReLU introduces other challenges that hinder its performance; bias shift and dying neurons in the negative region. To address these problems, this paper presents a novel composite monotonic, zero-centered, semi-saturated activation function called Hyperbolic cosine Linearized SquasHing function (HcLSH) with partial gradient-based sparsity HcLSH owns many desirable properties, such as considering the contribution of the negative values of neurons while having a smooth output landscape to enhance the gradient flow during training. Furthermore, the regularization effect resulting from the self-gating property of the positive region of HcLSH reduces the risk of model overfitting and ensures learning more robust expressive representations. An extensive set of experiments and comparisons is conducted that includes four popular image classification datasets, seven deep network architectures, and ten state-of-the-art activation functions. HcLSH exhibited the Top-1 and Top-3 testing accuracy results in 20 and 25 out of 28 conducted experiments, respectively, suppressing the widely used ReLU that achieved 2 and 5, and the reputable Mish that achieved 0 and 5 Top-1 and Top-3 testing accuracy results, respectively. HcLSH attained improvements over ReLU, ranging from 0.2% to 96.4% in different models and datasets. Statistical results demonstrate the significance of the enhanced performance achieved by our proposed HcLSH activation function compared to the competitive activation functions in various datasets and models regarding the testing loss Furthermore, the ablation study further verifies the proposed activation function's robustness, stability, and adaptability for the different model parameter.

INDEX TERMS Activation function, convergence, deep learning, image classification accuracy, monotonicity, saturation.

I. INTRODUCTION

Since the breakthrough in 2006 [1], deep learning has dominated the tremendous successes in the machine learning field. Deep learning uses multi-layered deep neural networks to analyze, understand, learn and solve complex and complicated real-world tasks [2], [3], [4]. For instance, Convolutional Neural Networks (CNNs) [5] achieve remarkable recognition accuracies comparable to the human level in the computer vision field. The reasons for this revolution include the significant improvements in the computation power, the invention of new powerful models, and the development of new training methods and regularization techniques that can handle the increasing model depth. Finally, the introduction

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

of new activation functions that can handle the high number of layers.

Activation functions (AF) are point-wise functions responsible for creating a higher level of abstraction between layers by performing space folding; the input space of the neuron is mapped to a different space in the output, which adds a non-linear effect to that output. The activation functions determine whether the neurons should be activated or not based on their input [6].

The two main probability-inspired saturated activation functions that were intensively and traditionally used in early neural networks are the Sigmoid and Hyperbolic Tangent (Tanh) functions [7]. However, currently, the non-saturated counterparts, such as the rectified linear units-based functions become more dominant and more relevant by introducing an abrupt paradigm shift to overcome the suffered problems as the depth of the model increases [8] and to expedite the convergence speed and increase the generalizability of the model.

As can be depicted, the choice of the non-linearity provided by the activation function has a significant and imperative influence on the speed and the behavior of the neural network's learning process [9] and, consequently, on the strength of its expressive power. Therefore, the activation functions are the key players of the neural network, either the shallow or the deep ones, through their non-linearities that chain up the layers and transform a linear model into a model that can represent complex functions [10], [11].

At each neuron in the model, two operations are performed at each epoch in the forward direction: (1) The summation of the Hadamard product of the inputs and their associated weights. (2) The transformation of the resulting weighted sum and the bias according to a specific non-linearity. The role of the activation function, in general, can be mathematically realized as in (1). Where w_i is the weight of the inputs x_i to that neuron, b is the neuron bias, $f\ (z)$ is the activation function, z is the linear combination of neuron's weights, inputs and bias ($z = \sum_i w_i x_i + b$), and y is the neuron output.

$$y = f\ (z) = f\left(\sum_i w_i x_i + b\right) \tag{1}$$

The different designs for the activation functions in the literature are described by various properties and characteristics [7], [12] that often show diverse behavior across the tested task, the dataset, and the network configuration.

Monotonicity: The monotonic function follows a single rhythm in a given interval, either increasing or decreasing. In other words, if the function has different signs on that interval, it is said to be non-monotonic. The function is monotonically increasing on a particular interval if the function value increases as its independent variables increase, i.e., that is, if $\mathrm{x}_1 > \mathrm{x}_2$, then $\mathrm{f}(\mathrm{x}_1) > \mathrm{f}(\mathrm{x}_2)$. On the other hand, the function is monotonically decreasing if the value decreases as the independent variables increase on an interval, i.e., that is, if $\mathrm{x}_1 > \mathrm{x}_2$, then $\mathrm{f}(\mathrm{x}_1) < \mathrm{f}(\mathrm{x}_2)$.

Zero-Centered: The zero-centered function guarantees that its mean activation value is around zero. This characteristic is helpful if no effort is put into choosing the initial random weights of the model or processing the input data to normalize it [13], [14]. If the function is not zero-centered, i.e., if the data coming into a neuron is always positive, the weight gradient will have the same sign during the backpropagation training. This feature causes a bias shift in the successive layers, positive bias since the values do not cancel each other. The deeper the network, the larger the bias, which produces undesirable zigzagging dynamics in the weight update.

Saturation: The saturated function is a bounded function with finite upper and lower limits; therefore, it is called a squashing function. This characteristic directly affects the function gradients as they eventually get smaller and level out over time. In other words, the gradient vanishes, which leads to a weaker training process. On the other hand, training using a bounded function will be more stable than an unbounded function since the unbounded will affect most of the weights, while limited weights will be affected in the case of a bounded function.

However, the research on designing well-performing activation functions does not halt since the current ones still suffer from some problems. The most famous challenge facing activation functions is vanishing or exploding gradients. This problem is first introduced in [9]; such a problem can cause slow convergence and even cause the trained neural network to converge and get stuck in a poor local minimum [15]. This problem occurs when the activation function saturates at either of its tails. There are two possible cases: (1) if the function derivatives are less than 1, multiplying them many times during the gradient flow update will make the resulting value smaller and smaller until the gradient tends toward zero in the lower distant layers. Such a case makes them useless in representation learning, called the vanishing gradient. (2) If the values are larger than 1, the resulting value becomes bigger until they reach infinity and the gradients explode. The second problem is the dying neurons [16]; if not all the neurons are activated at each iteration in the training process, then some of the expressive power of the network is lost since these unattended neurons do not contribute to the learning.

Driven by the significance and the contribution of the effects of the different properties of the current activation functions; this paper introduces the design of a novel and effective monotonic composite piecewise activation function called Hyperbolic cosine Linearized SquasHing function (HcLSH). The aim is to accelerate and boost the learning process in deep learning-based models. The main contributions of this paper are:

1) Constructing the HcLSH activation function that blends different solutions to address multiple issues mentioned above. For instance, considering the importance of negative representation while providing partial gradient sparsity to include a portion of negative values in the feature learning without exhausting the model by incorporating all the negative values in the weights update process during training. This

unsymmetrical and separate treatment of positive and negative values solved the dying neuron problem while enhancing the gradient flow by introducing diversity in the expressiveness of the extracted features necessary to provide noise robustness in the learning process while preventing potential stuck in local solution. Furthermore, having sound regularization effects while being monotonic helped prevent the model equipped with HcLSH from overfitting the training data. On the other hand, the semi-saturated property due to being bounded below and unbounded above and the saturated first derivative of HcLSH efficiently addressed the exploding and vanishing gradient problem and allowed the model to learn more representative input features required to boost the performance.

2) Analyzing and discussing the properties of the proposed HcLSH that play a role in its good performance and success in deep network training, such as parameter-free nature, monotonicity, landscape smoothness, zero-centering, and semi-saturation.

3) Conducting an extensive experimental analysis to evaluate, validate and compare the performance of HcLSH against the popular existing activation functions using various image classification datasets. This shows that the proposed HcLSH improves classification accuracy and minimizes the loss of broad network architectures with different characteristics.

4) Conducting non-parametric statistical tests, such as the Wilcoxon signed rank test for pair-wise comparison and the Friedman ranking test for overall relative comparison, to prove the performance improvements achieved by HcLSH statistically.

5) Conducting an ablation study to observe the effects of hyperparameter finetuning on the performance of the deep model operated with HcLSH.

6) Verifying the compatibility of HcLSH with batch normalization to make the associated deep network less sensitive to parameter initialization.

The rest of the paper is organized as follows. An overview of the well-known state-of-the-art activation functions is presented in Section II. The proposed activation function (HcLSH) is introduced, and its derivatives and properties are discussed in detail in Section III. The experimental results of the proposed HcLSH activation function, including an evaluation of the average test accuracy and loss, a comparison against ten state-of-the-art activation functions, and an ablation study, are given in Section IV. Discussion of the finding is presented in Section V. The limitations of the proposed HcLSH are reported in Section VI. Finally, the main conclusions of the work are devoted in Section VII.

II. RELATED WORK

Designing new activation functions has been an active area in the literature. This section describes some of the famous and powerful activation functions. The equations, the derivatives' equations, and the different properties that describe the reviewed activation functions are listed in Table 1. Categorizing of the activation functions based on their properties is presented in Figure 1. Furthermore, the graphs of these activation functions are visualized in Figure 2, grouped into three categories for better illustration and clarity.

The sigmoid function, AKA the logistic function, is a continuous, non-linear saturated s-shaped function that takes an input and squashes it between 0 and 1. It is considered a direct probabilistic interpretation of the output. It was considered the popular choice for training shallow neural networks. However, it proved its unsuitability for training the deep networks [8] with random initialization for multiple reasons; toward its two ends, the sigmoid saturates; therefore, it suffers from the vanishing gradient problem, especially as the layers become deeper, which restricts the performance. Moreover, the sigmoid function is not zero-centered; consequently, it may introduce important singular values in the Hessian [14], which leads to a slower convergence of the network and thus finds a poorer local minimum.

On the other hand, to overcome the limitation of the sigmoid, the hyperbolic tangent (tanh) function squashed the input to the range [-1, 1] and produced a zero-centered output. However, it still suffers from the vanishing gradient problem because of its bounded limits. The tanh activation function is preferable over the sigmoid function [14] since its performance is more stable, especially since the derivatives of tanh are steeper than those of sigmoid functions. The penalized tanh [17] is an activation function that tackles another crucial issue for the training process: the nature of the slope of the activation function near the origin. It does improve the performance of tanh by penalizing it in the negative region, which enhances the saturation. The penalization aims to rescale the logistic sigmoid as follows

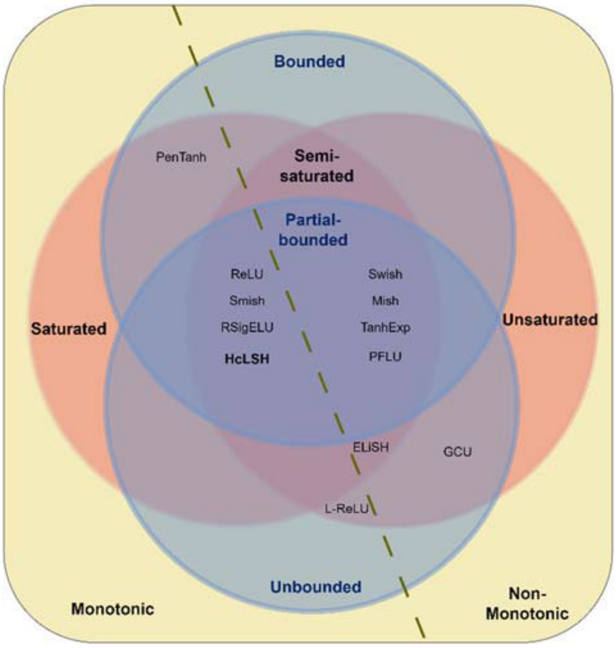


FIGURE 1. Categorization of different AFs based on the monotonicity, saturation, and bounding properties.

TABLE 1. Activation functions and derivative equations with classification according to their properties.

Function name	The function equation	The derivative equation	Monotonic	Zero-centered	Saturated	Range /bounded
ReLU	$f(x) = \max(0,x) = \begin{cases} x\,, & x \geq 0 \\ 0, & x < 0 \end{cases}$	$\acute{f}(x) = \begin{cases} 1\,, & x \geq 0 \\ 0, & x < 0 \end{cases}$	✓	✗	Partially saturated on -ve region	Bounded below and un-bounded above
L-ReLU	$f(x) = \min(0,\alpha x) + \max(0,x) = \begin{cases} x\,, & x \geq 0 \\ \alpha x, & x < 0 \end{cases}$	$\acute{f}(x) = \begin{cases} 1\,, & x \geq 0 \\ \alpha, & x < 0 \end{cases}$	✓	✓	✗	Un-bounded
Penalized tanh	$f(x) = \begin{cases} \tanh(x)\,, & x > 0 \\ 0.25\tanh(x)\,, x \leq 0 & \end{cases}$	$\acute{f}(x) = \begin{cases} \text{sech}^2(x)\,, & x > 0 \\ 0.25\,\text{sech}^2(x)\,, & x \leq 0 \end{cases}$	✓	✓	✓	bounded
Swish	$f(x) = x\,\sigma(\beta x)$	$\acute{f}(x) = \beta f(x) + \sigma(\beta x)\big(1 - \beta f(x)\big) = \dfrac{\beta(1 + e^{-x} + xe^{-x})}{(1 + e^{-x})^2}$	✗	✓	Partially saturated toward -∞	Bounded below and un-bounded above
ELiSH	$f(x) = \begin{cases} \dfrac{x}{1 + e^{-x}}, & x \geq 0 \\ \dfrac{e^x - 1}{1 + e^{-x}}, & x < 0 \end{cases}$	$\acute{f}(x) = \begin{cases} \dfrac{1}{1 + e^{-x}} + \dfrac{x\,e^{-x}}{(1 + e^{-x})^2}, & x \geq 0 \\ \dfrac{e^{-x}}{1 + e^{-x}} + \dfrac{e^{-x}(e^x - 1)}{(1 + e^{-x})^2}, & x < 0 \end{cases}$	✗	✓	Partially saturated toward -∞	Un-bounded
Mish	$f(x) = x\tanh(\ln(1 + e^x))$	$\acute{f}(x) = \tanh\ln(1 + e^x) + \dfrac{xe^x\,\text{sech}^2\ln(e^x + 1)}{e^x + 1}$	✗	✓	Partially saturated toward -∞	Bounded below and un-bounded above
Smish	$f(x) = x\tanh(\ln(1 + \sigma(x)))$	$\acute{f}(x) = \dfrac{e^x(15e^{3x} + (8x + 28)e^{2x} + (12x + 18)e^x + 4x + 4}{(5e^{2x} + 6e^x + 2)^2}$	✓	✓	Partially saturated toward -∞	Bounded below and un-bounded above
RSigELU	$f(x) = \begin{cases} \dfrac{\alpha x}{1 + e^{-x}} + x, & x > 1 \\ x, & 0 \leq\ x \leq 1 \\ \alpha(e^x - 1), & x < 0 \end{cases}$ $\alpha \in (0{,}1)$, Control the slope of positive and negative regions.	$\acute{f}(x) = \begin{cases} \dfrac{-\alpha x}{(e^x + 1)^2} + \dfrac{\alpha x - \alpha}{e^x + 1} + 1 + \alpha, & x > 1 \\ 1, & 0 \leq\ x \leq 1 \\ \alpha(e^x), & x < 0 \end{cases}$	✓	✓	Partially saturated toward -∞	Bounded below and un-bounded above
TanhExp	$f(x) = x\tanh(e^x)$	$\acute{f}(x) = \tanh e^x - xe^x\tanh^2 e^x + xe^x$	✗	✓	Partially saturated toward -∞	Bounded below and un-bounded above
PFLU	$f(x) = x.\dfrac{1}{2}.(1 + \dfrac{x}{\sqrt{1 + x^2}})$	$\acute{f}(x) = \dfrac{1}{2}.(1 + \dfrac{x}{\sqrt{1 + x^2}} + \dfrac{x}{(1 + x^2)\sqrt{1 + x^2}})$	✗	✓	Partially saturated toward -∞	Bounded below and un-bounded above
GCU	$f(x) = x\cos x$	$\acute{f}(x) = \cos x - x\sin x$	✗/ oscillating	✓	✗	Un-bounded
Proposed HcLSH (our)	$f(x) = \begin{cases} \ln\left(\cosh x + x\cos\dfrac{x}{2}\right), & x \geq 0 \\ \ln\left(\cosh x\right) + x, & x < 0 \end{cases}$	$\acute{f}(x) = \begin{cases} \dfrac{\sinh x - \dfrac{x}{2}\sin\dfrac{x}{2} + \cos\dfrac{x}{2}}{\cosh x + x\cos\dfrac{x}{2}} & x \geq 0 \\ \tanh x + 1 & x < 0 \end{cases}$	✓	✓	Partially saturated toward -∞	Bounded below and un-bounded above

(scaled sigmoid(x) = 4 $*$ sigmoid(x) – 2) to have twice larger the activation value of the tanh.

The Rectified Linear Unit (ReLU) [18] is a piecewise linear function that is considered one of the success factors and milestones of the breakthrough in training the deep learning neural network [19]. It became the popular default choice for training the deep neural network because of its superior performance, thanks to its simplicity, effectiveness, and the appealing advantage of enablement the computational calculation because of its non-sigmoidal nature. The negative part of ReLU is clipped to zero, while the identity function is used in its positive part. Therefore, the overall function is not linear, but it can be said that it is one-sided linear. However, ReLU is non-differentiable at zero [8]; it has a derivative of one for the positive inputs and zero otherwise; this raises both an advantage and a disadvantage at the same time. The zero derivatives at the negative side make the network sparse, which helps in reducing the representation dimensionality and decreasing the computational effort. However, these inactivated nodes will not improve learning since there will be no information flow; in other words, these neurons will not get adjusted during the gradient descent optimization. This results in a problem called dying ReLU, leading the model to underperform.

Moreover, ReLU is not zero-centered; it is lower-bounded at zero but has an unbounded upper, i.e., positive infinity, which may blow up the activation. ReLU alleviates the vanishing gradient problem because of the identity mapping and the one derivative on the positive side; it can preserve the magnitude of the error. Thus, it speeds up the learning and leads to a quicker convergence. To solve the dying ReLU problem, LeakyReLU (L-ReLU) [16] introduced a linear function with a small slope, controlled using a predefined constant (alpha), in the negative part of the ReLU. Consequently, this results in a small, non-zero, constant gradient that allows a small amount of information to flow in the negative part by re-activating some de-activated neurons. However, L-ReLU reduces the sparsity provided by ReLU. Since then, many AFs had been introduced in the literature with these functions' novelties represented in how they modeled their negative values differently.

As a result of using automated search techniques in [20], an activation function called Swish is proposed. Swish is unbounded above and bounded below as the ReLU. However, in contrast to ReLU, Swish is smooth and non-monotonic. The linear part in the Swish function helps avoid the vanishing gradient problem, while the sigmoid parts improve the information flow or propagation. This is achieved by introducing the "self-gating" property, in which a function is multiplied with its own input. The Beta parameter (β) can be either constant (usually 1) or a trainable that shapes the non-monotonic bump of the Swish. Building on that success, a piece-wise activation function called Exponential Linear Sigmoid SquasHing (ELiSH) function [21] used Swish in its positive part. In contrast, the negative part is the product of exponential and sigmoidal components. Another non-monotonic activation function but, at the same time, non-exponential is called Power Function Linear Unit (PFLU) [22], which is based on power functions. Inspired by Swish's self-gating property, Mish [23] is a self-regularized, non-monotonic activation function. Mish is unsaturated in the positive part and soft-saturated towards the negative infinity. A monotonic modification of Mish is introduced in [24], called Smish, replacing the exponential term with a sigmoid function. Another similarly behaved function that is non-monotonic and also utilizes the self-gating property is the Tanh Exponential activation function (TanhExp) [25]. However, it exhibited a steeper and bigger gradient near zero resulting in an acceleration of the parameters' updates in the network.

An activation function that behaved differently in three different regions is RSigELU [26] which exhibited an attitude that reflects a composition of ReLU and sigmoid functions, a linear function, and an ELU activation function [27] in its three activation regions: positive, linear and negative regions, respectively, as shown in Figure 2(b). On the other hand, an oscillatory activation function with an amplitude directly proportional to the input values is called the Growing Cosine Unit (GCU) [28]. GCU aimed to enhance the gradient flow by granting the neurons the ability to reverse their output sign inside the interior of neuronal hyperplane positive and negative half-spaces.

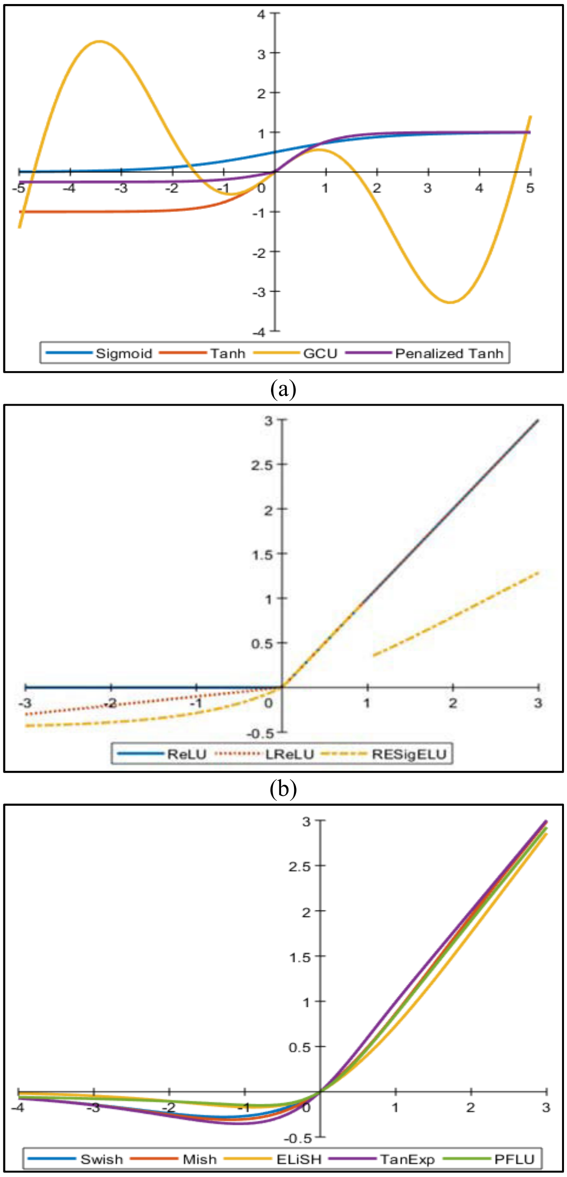
Although these new AFs introduce great success and advancements in the performance of deep learning models, the research does not halt due to rapid changes in the deep architectures, datasets, and tasks. The performance results remain unsatisfactory and need continuous improvements. Therefore, this paper aims to design a novel activation function that integrates multiple desirable characteristics to compete efficiently and provide generalizability and robustness across many models and datasets by addressing the challenges hindering the activation function performance.

III. THE PROPOSED HcLSH ACTIVATION FUNCTION

Inspired by the observations and insights from studying different activation functions' characteristics, properties, and performance, we propose HcLSH, a zero-centered, monotonic, and semi-saturated activation function. Thus, in this section, we present and analyze our Activation Function (HcLSH) and describe its properties.

A. GRAPH AND DERIVATIVES OF HCLSH

HcLSH, as visualized in Figure 3 (a) and (b) (the solid line), is a monotonic composite activation function. HcLSH is bounded below and unbounded above; HcLSH also extends below zero at the negative part. Therefore, HcLSH is unsaturated in the positive part and saturated in the negative part. Consequently, it is considered a semi-saturated function. HcLSH can be mathematically defined as in Eq.2. HcLSH has a range of $[-0.6931, +\infty)$, the minimum value of HcLSH is approximately -0.6931 (-ln2) with the corresponding input



(a)

(b)

(c)

FIGURE 2. Plots of the different activation functions divided into three groups for better illustrations. (a) Group 1: Sigmoid, Tanh, GCU and Penalized Tanh, (b) Group 2: ReLU, L-ReLU, and RSigELU, and (c) Group 3: Swish, Mish, ELiSH, TanhExp, and PFLU.

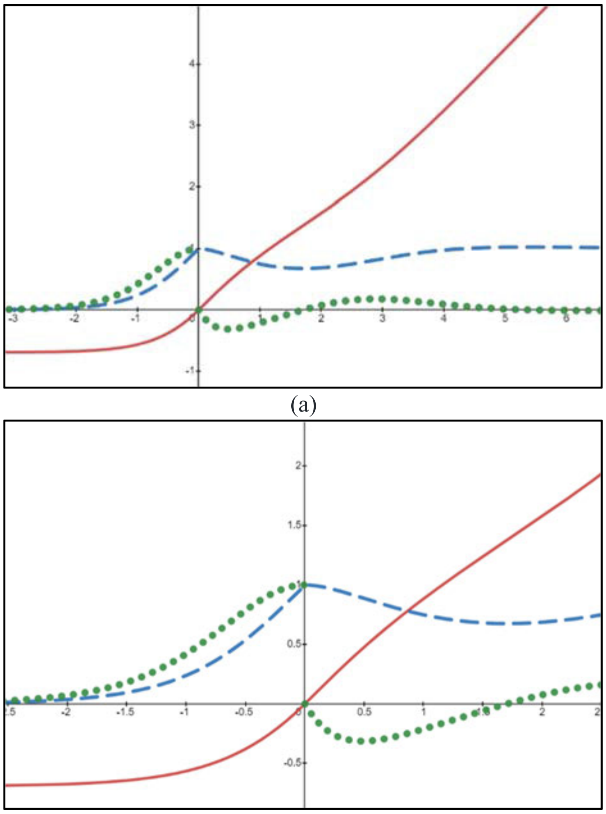
values of (-∞,-4.15). The first derivative of HcLSH with respect to the input can be calculated according to Eq.3. The visual representations of the first and second derivatives are shown in Figure 3 (a) and (b) (the dashed and dotted lines, respectively):

$$f(x) = \begin{cases} \ln(\cosh x + x\cos\dfrac{x}{2}), & x \geq 0 \\ \ln(\cosh x) + x, & x < 0 \end{cases} \tag{2}$$

$$\acute{f}(x) = \begin{cases} \dfrac{\sinh x - \frac{x}{2}\sin\frac{x}{2} + \cos\frac{x}{2}}{\cosh x + x\cos\frac{x}{2}} & x \geq 0 \\ \tanh x + 1 & x < 0 \end{cases} \tag{3}$$

Here, x represents the input to the activation function.

One appealing property of HcLSH is its parameter-free nature. No hyper-parameter is introduced in its definition, as can be seen in Eq.2. Thus, no undesirable consequences are encountered, such as searching for the optimal value suitable for the task at hand or increasing the complexity and capacity of the integrated model in case if this parameter being trainable.



(a)

(b)

FIGURE 3. The proposed HcLSH activation function graph. (a) a wider view, (b) closer view. The solid red line represents the activation function graph, the blue dashed line represents the first derivative graph, and the green dotted line represents the second derivative graph of the function.

B. PROPERTIES OF HCLSH

1) UNBOUNDED AND UNSATURATED POSITIVE PART

In order to overcome the vanishing gradient problem caused by saturation that results in slow training, HcLSH is unbounded above and unsaturated in the positive part. Motivated by the self-gating property of Swish, in the positive part, the non-modulated input is multiplied by the output of a cosine trigonometric function of half the input.

The intuition for using the self-gating property is to allow the network to keep the initial input distribution which gives stronger regularized effects represented by giving the function a linear-like behavior. Afterward, the contribution of the input's Hyperbolic Cosine function (cosh ()) is added to reduce the range of values and provide more stability. Finally, all this composition is suppressed with the natural logarithm (ln()). The choice of using the logarithmic operation for the positive inputs is to squash the input and reduce its numerical

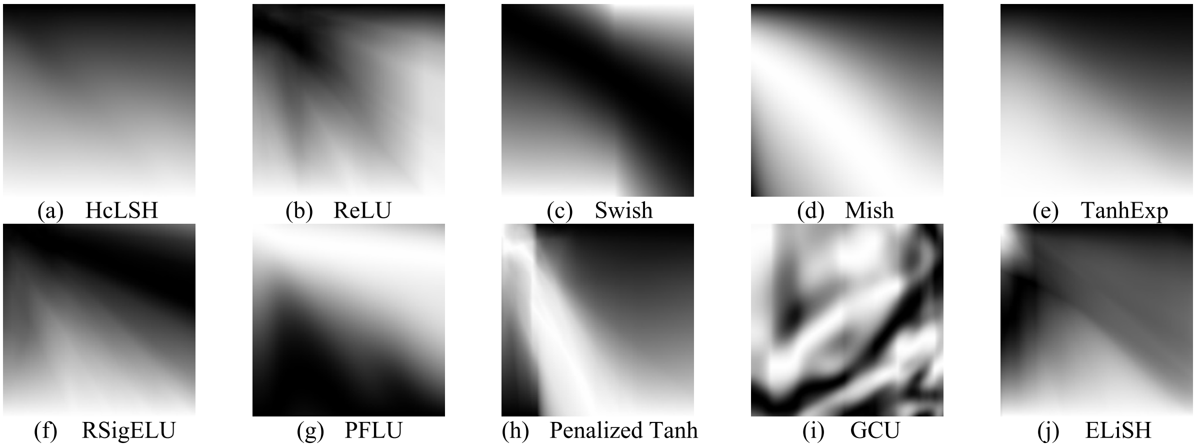


FIGURE 4. The output landscapes of the different competitive activation functions on a 5-layer randomly initialized neural network, in addition to the proposed HcLSH activation function (a).

range making the curve smoother and flatter. As the input goes towards zero, a small fluctuation can be seen in the positive part, proving the non-linearity of HcLSH.

2) BOUNDED AND SATURATED NEGATIVE PART

In order to address the dying neurons phenomenon without completely scarifying the sparsity in the negative part, HcLSH is bounded below and saturated in the negative part. A combination of Hyperbolic Cosine function and natural logarithmic operation are used to boost the smoothness and provide efficient range reduction. Furthermore, to force HcLSH to be zero-centered, the effect of non-modulated input is also added in the negative part. This zero-centered property is preferable to speed up learning [27].

The negative information is allowed to contribute to the learned representation of HcLSH, instead of zeroing it out as done in ReLU, to increase the expressive power and enhance the information flow. However, HcLSH also ensures a partial sparsity in its first derivative (gradients during backpropagation), as the value of the gradients tends to approach zero as the input decreases more toward negative infinity, but with a less de-activating probability rate than exists in ReLU. This partial sparsity, caused by being bounded below, and the resulting de-activated neurons, in which their role in learning is halted, enhance the regularization in the network in the negative part by introducing a similar behavior of the Dropout [29], [30].

Additionally, the proposed HcLSH activation function also guarantees negative activation values for negative inputs.

3) SMOOTH LANDSCAPE, DIFFERENTIABILITY, AND MONOTONICITY

Although that HcLSH is a monotonically increasing function, its first derivative is non-monotonic in the positive part. Moreover, HcLSH's first and second derivatives are continuously differentiable, as shown in Figure 3, a property that is preferable because it avoids singularities and, therefore, undesired side effects when performing the parameter updating during training using gradient-based backpropagation optimization.

Visualizing the output landscape of a 5-layer fully connected randomly initialized neural network with different activation functions, including HcLSH, ReLU, Penalized Tanh, Swish, ELiSH, Mish, PFLU, TanhExp, RSigELU, and GCU, is shown in Figure 4. As can be observed, the choice of the activation function plays an essential role in forming the output landscape smoothness. As demonstrated, HcLSH, Mish, and TanhExp activation functions have smooth profiles, which consequently indicate good gradient flow and thus learn more valuable information, obtaining smooth loss landscapes [31], and easing the optimization process. On the other hand, ReLU, Penalized Tanh, ELiSH, and GCU have rough profiles with sharp transitions, with GCU being the most disturbed and chaotic.

C. COMPARATIVE WITH OTHER ACTIVATION FUNCTIONS

In this section, we chose the most popular activation functions: Swish and Mish, to compare the attitude of the proposed HcLSH function within the negative and positive regions. For example, in the positive part, HcLSH initially obtains bigger values than Swish for the inputs that are less than x=1.533, which corresponds to the output value of 1.261, and bigger values than Mish for the inputs that are less than x=1.061 corresponds to the output value of 0.9292, after that, HcLSH obtains lower values than them. On the other hand, in the negative part, HcLSH obtains larger values and thus achieves steeper gradient values compared with Swish and Mish. Furthermore, HcLSH is monotonic in the negative region, unlike in Swish and Mish. Swish and Mish approach zero as the inputs decrease towards the negative infinity, while HcLSH saturates around x= -0.6931. Furthermore, the first derivatives of HcLSH, Swish, and Mish activation functions all reach 0 towards negative infinity, but the derivative of HcLSH reaches it faster at x=-4.15, while the derivative of Swish at x=-12.333 and the derivative of Mish at x=-9.8.

IV. EXPERIMENTAL ANALYSIS

In the following parts of this section, the benchmark datasets used in the study are introduced, information is given about the convolutional neural networks and the activation functions used in the study, and then the experimental results are presented, and the results are discussed. Finally, an ablation study is conducted to test the effects of different choices of the model's hyperparameters on the performance of the proposed activation function. The performance of the proposed HcLSH function relative to other activation functions is evaluated with four open datasets in the image classification task in computer vision using several popular deep learning models. Notably, our evaluation is based on test samples rather than training samples. The labels and examples of the classes in the used datasets are presented in Table 2, while the datasets descriptions are given next.

A. BENCHMARK DATASETS

1) FASHION MNIST DATASET

The Fashion Modified National Institute of Standards and Technology database (Fashion MNIST)[1] [32] is an image classification dataset that consists of 28 × 28 grey-scale images with ten different classes from real-world clothing and fashion items from Zalando's article images: T-shirts, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. The dataset is divided into 60k training and 10k testing images.

2) SVHN DATASET

The Street View House Numbers (SVHN)[2] [33] dataset is an image classification and object recognition dataset that contains real-world numerical numbers (0...9) of houses obtained from Google Street View images. The dataset consists of 32 × 32 colored images with a single digit in the center and some distractions on the sides. It is divided into 73,257 and 26,032 training and testing images.

3) CIFAR10 AND CIFAR100 DATASETS

The Canadian Institute for Advanced Research (CIFAR)[3] [34] dataset is an image classification dataset that consists of 32 × 32 colored images that are divided into 50k training and 10k testing images. CIFAR has two versions according to the number of classes; the CIFAR10 dataset with 10 classes and the CIFAR100 dataset with 100 classes.

B. EXPERIMENTAL SETUP

All the conducted experiments of all models performed on all datasets are trained with ADAM optimizer [35] for gradient backpropagation with default parameters of $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\epsilon = 10^{-7}$, fixed learning rate of 10^{-4}, and batch size of 128. These values were chosen without any search for optimal results or bias to favor any activation function, even the proposed one. The pixel values of the input images are normalized by scaling by 255. The categorical cross-entropy function is used as the loss function with SoftMax classification. No data augmentation is applied in any form in the experimental evaluation in all the experiments in this paper since the aim is to test the efficiency level of the different activation functions fairly. Also, the goal is not to achieve the optimal results and most advanced performance in the used datasets. Unless otherwise stated, these configuration values are used in all the experiments in this section to have a fair base for comparison.

For all experiments in this section, we only replaced all the activation functions in the architecture, except for the final SoftMax function, with the corresponding test activation function. All other training settings and architectures are kept unchanged at the same time. The performance evaluation experiments were implemented based on TensorFlow [36] backend with Keras Library and Google Colab computational resource platform. The number of training epochs for the Fashion MNIST, SVHN, CIFAR10, and CIFAR100 benchmark datasets is set to 20, 50, 50, and 100 epochs, respectively. Three runs of each experiment are performed to handle the uncertainty caused by the different weight initialization seeds.

The following deep learning models and architectures are used in the performance comparison in the following subsections: LeNet [37], MobileNetV1 [38], KerasNet [39], SqueezeNet [40], InceptionNetV4 [41], ResNetV2-20 [42], and ShuffleNetV2 [43]. These models vary in their parameters' number, layers' number, and connection types. They were used to test the effects of HcLSH against models with different depths of layers (shallow vs. deep), different connections (plain vs. residual vs. inception), different convolutions (regular vs. dilated vs. depthwise), and a different number of trainable parameters (light vs. heavy).

The comparative analysis is conducted against ten popular and state-of-the-art activation functions:

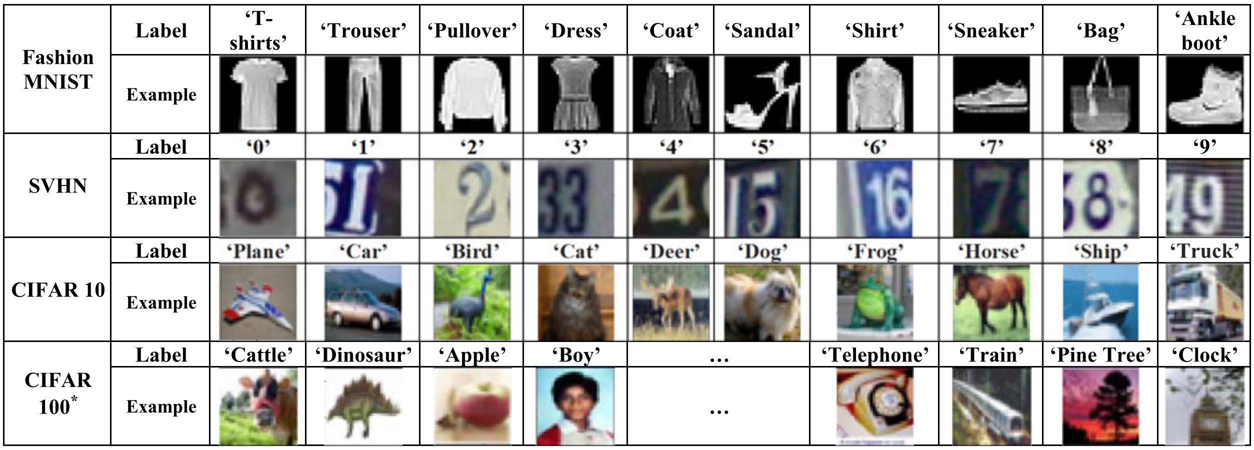
- ReLU [18]: Since it has been chosen as the default activation function in the deep learning community, this study treats ReLU as the baseline activation function for performance comparison purposes.
- Leaky ReLU (L-ReLU) [16]: The value of the parameter α is set to 0.1.
- Penalized Tanh [17]: denoted as (PenTanh).
- Swish [20]: The value of the parameter β is set to 1.
- ELiSH [21].
- Mish [23].
- PFLU [22].
- TanhExp [25].
- RSigELU [26]: The value of the parameter α is set to 0.45.
- GCU [28].

[1] Download link: https://github.com/zalandoresearch/fashion-mnist, Access Date: 6th March 6, 2023

[2] Download link: http://ufldl.stanford.edu/housenumbers, Access Date: 6th March 6, 2023

[3] Download link: https://www.cs.toronto.edu/~kriz/cifar.html, Access Date: 6th March 6, 2023

TABLE 2. Labels and examples from the Fashion MNIST, SVHN, and CIFAR 10 datasets.



Fashion MNIST	Label	‘T-shirts’	‘Trouser’	‘Pullover’	‘Dress’	‘Coat’	‘Sandal’	‘Shirt’	‘Sneaker’	‘Bag’	‘Ankle boot’
	Example										
SVHN	Label	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’
	Example										
CIFAR 10	Label	‘Plane’	‘Car’	‘Bird’	‘Cat’	‘Deer’	‘Dog’	‘Frog’	‘Horse’	‘Ship’	‘Truck’
	Example										
CIFAR 100*	Label	‘Cattle’	‘Dinosaur’	‘Apple’	‘Boy’	…		‘Telephone’	‘Train’	‘Pine Tree’	‘Clock’
	Example					…					

* for full and precise information about all the 100 class labels, more information is available in https://www.cs.toronto.edu/~kriz/cifar.html

C. EXPERIMENTAL RESULTS AND DISCUSSION

Ten popular and recent activation functions, ReLU [18], L-ReLU [16], PenTanh [17], Swish [20], ELiSH [21], Mish [23], RSigELU [26], TanhExp [25], PFLU [22], and GCU [28], are considered to compare the performance and validate the efficiency of HcLSH on four datasets: Fashion MNIST, SVHN, CIFAR10, and CIFAR100, and seven deep learning models on the image classification task: LeNet [37], MobileNetV1 [38], KerasNet [39], SqueezeNet [40], InceptionNetV4 [41], ResNetV2-20 [42], and ShuffleNetV2 [43].

1) AVERAGE TEST ACCURACY AND LOSS

Since the ReLU [18] activation function is the most widely used and preferable in deep learning models, it is considered the main baseline, and all average test accuracy results presented for all the comparative activation functions were normalized with respect to the accuracy obtained by the ReLU activation. Thus, a new metric called normalized accuracy (relative ratio in percentage) is also investigated. The performance was evaluated based on two metrics: average test classification accuracy and average test loss. The results, including average test loss and accuracy values with their standard deviations and the normalized accuracy ratio, that describe the performance and behaviors of different activation functions are demonstrated in Table 3 to Table 9.

In the simple 8-layer LeNet [37] model, as can be observed in Table 3, the performance of HcLSH in the SVHN and CIFAR100 datasets was better than in CIFAR10 and Fashion MNIST datasets, where it achieved the first, first, fourth, and third rank, respectively, in normalized accuracy, i.e., the accuracy ratio with respect to ReLU's accuracy. GCU outperformed the proposed HcLSH by 1.2% in test accuracy in the Fashion MNIST dataset. However, HcLSH ranked second and outperformed the third-ranking TanhExp by 0.41% and the rest of the activation functions by a good margin. In SVHN, HcLSH suppressed the testing accuracy of the second and third runners (ELiSH, ReLU, and PenTanh). In CIFAR10, the ReLU activation function achieved the top place by achieving an accuracy of 56.89%, followed by TanhExp, while Penalized Tanh (PenTanh) and GCU achieved the third rank in terms of testing accuracy. In CIFAR100, our proposed HcLSH was able to outperform all other activation functions; it enhanced the average accuracy by 0.25% and 0.27% compared to TanhExp, which achieved the second rank and penalized Tanh (PenTanh) that achieved the third rank, respectively. The enhancements introduced by the models with tested activation functions in the CIFAR100 dataset are minimal due to the dataset's complexity and the network's simplicity. The lowest test loss in the Fashion MNIST dataset of LeNet-based models was found to belong to GCU and was equal to 0.3465, followed by the loss value obtained by HcLSH, which was equal to 0.3876. On the other hand, ReLU achieved the smallest loss value in CIFAR10, followed by penalized Tanh (PenTanh), TanhExp, L-ReLU, and HcLSH. In SVHN, the lowest test loss values were 0.5382 and 0.5444, obtained by ELiSH and HcLSH, respectively. In contrast, HcLSH achieved the lowest test loss value in CIFAR100 dataset by obtaining a loss value of 2.9963.

The model complexity increased with the examined model being MobileNetV1 [38]. As can be observed from Table 4, the performance of the proposed HcLSH becomes more efficient by attaining the maximum normalized accuracy among all the compared activation functions. It enhanced the average accuracy by 0.74%, 0.3%, 0.83%, and 0.29% compared to the activation functions that achieved the second-best results, i.e., RSigELU, ELiSH/ L-ReLU, GCU, and GCU in Fashion MNIST, SVHN, CIFAR10, and CIFAR100 datasets, respectively. On the other hand, it is noticeable from Table 4 that replacing ReLU with our proposed HcLSH caused a significant boost of 3.5%, 2.1%, 9.5%, and 14.8% in the testing accuracy of Fashion MNIST, SVHN, CIFAR10, and CIFAR100 datasets, respectively. The MobileNetV1 models based on Swish and PFLU were the weakest in the four datasets under the adopted unified experiment settings.

Regarding the average test loss metric, the lowest values were found as 0.4307, 0.4394, 1.0822, and 2.8096 for the models based on MobileNetV1 across the four datasets, respectively, and they were achieved by the proposed HcLSH activation function.

With KerasNet architecture [39], our proposed HcLSH obtained the highest average test accuracy in CIFAR10 and CIFAR100 datasets outperforming the rest, as noted in Table 5. In SVHN, HcLSH obtained second place after ReLU and ELiSH. However, unfortunately, it fell behind ELiSH, GCU, and TanhExp activation functions by achieving an accuracy of 91.10% in the Fashion MNIST dataset. These functions turn out to have better performance of 91.65%, 91.47%, and 91.32%, respectively. However, as can be seen in Table 5, the proposed HcLSH outperformed other functions in terms of average test loss in KerasNet-based models by obtaining the lowest values of 0.6430 and 2.0248 in CIFAR10 and CIFAR100 datasets with enhancements of 0.005 and 0.1149 relative to the loss of the baseline ReLU, although that the improvements are not impressive, but HcLSH succeed by suppressing the test loss values obtained by the rest of the activation functions. In Fashion MNIST,, HcLSH obtained the second highest test loss value, after Swish, with a difference equal to -0.0012 compared to ReLU and -0.0157 compared to the ELiSH activation function, which has the lowest loss. Unfortunately, in SVHN, the test loss of HcLSH was suppressed by the values obtained by ReLU, L-ReLU, ELiSH and GCU.

When using SqueezeNet [40], as the testbed for the experiments summarized in Table 6, the models with the proposed HcLSH obtained the top results in Fashion MNIST, CIFAR10 and CIFAR100 datasets, as can be noted, with the most impressive performance belonging to the CIFAR100 dataset. HcLSH achieved the second rank in testing accuracy in the SVHN dataset. In the Fashion MNIST dataset, HcLSH was the only activation function that outperformed the ReLU baseline by increasing the accuracy by 0.6%, while ELiSH performed equally to ReLU. Furthermore, the enhancements introduced by HcLSH were massive in the CIFAR100 dataset compared with the ReLU baseline by its ability to achieve almost double the obtained test classification accuracy. RSigELU, TanhExp, ELiSH, and Mish activation functions performed relatively well with their ability to outperform ReLU accuracy but with a lower rate compared to HcLSH. In other words, HcLSH improves TanhExp accuracy in CIFAR100 by 2.26%, RSigELU accuracy by 4.12%, ELiSH accuracy by 4.48%, and Mish accuracy by 4.26%. Moreover, the improvement of the test loss in the Fashion MNIST dataset was significant by obtaining a value of 0.3566 with an enhancement of 0.0164 over ReLU. However, in the CIFAR10 dataset, the penalized Tanh (PenTanh) activation outperformed HcLSH in terms of loss value and obtained a value of 1.1192 with 0.1559 enhancement. On the other hand, the best test loss value in SVHN belongs to TanhExp, with a value of 0.4327, while HcLSH achieved a loss value of 0.5202. In the CIFAR100 dataset, Mish obtained the best loss value of 2.9348, followed directly by HcLSH, obtaining a loss of 3.0313.

In the InceptionNetV4 [41], Table 7, HcLSH yielded the best test accuracy results in CIFAR10 and CIFAR100 datasets, and the second best result in SVHN dataset. However, it was outperformed in the Fashion MNIST dataset, which gives an indication that HcLSH is more suitable for learning the representation of complex datasets. GCU performance was observed to be very poor with this network architecture by falling back with a large margin compared to the set of all compared activation functions. All the activation functions in all Fashion MNIST, CIFAR10 and CIFAR100 datasets, except for GCU and L-ReLU in Fashion MNIST, performed better on average test accuracy than the ReLU baseline with varying rates. On the other hand, the test accuracy results in SVHN of different AFs were similar and close. The activation functions with the lowest loss value belong to PFLU, with a 0.2213 loss value in the Fashion MNIST dataset, HcLSH with a 0.1796 loss value in the SVHN dataset, HcLSH with a 0.4835 loss value in the CIFAR10 dataset, and HcLSH with a 1.4817 loss value in the CIFAR100 dataset. The poor performance of the GCU activation function under this model regarding the loss metric is also observed.

As can be depicted from Table 8, accuracy improvements of 0.35%, 2.14%, and 3.65% were reported for ResNetV2-20 [42] with HcLSH for the Fashion MNIST, CIFAR10, and CIFAR100 datasets, respectively, over the second-best model in each with RSigELU, Swish and RSigELU activation functions. In the SVHN dataset, the residual model with Swish achieved the top testing accuracy by obtaining 102.3% normalized accuracy, followed directly by HcLSH and PFLU based models with normalized accuracy of 102.2%.Also, the ResNetV2-20 models based on GCU were underperforming, achieving the poorest results but with better performance than InceptionNet-based models. The proposed HcLSH demonstrates significant performance by achieving 2.2%, 2.2%, 16.4%, and 30.2% higher classification accuracy against ReLU accuracy in Fashion MNIST, SVHN, CIFAR10, and CIFAR100 datasets. The performance of HcLSH in terms of average test loss value for the residual-based models varies across the four different datasets, but with being able to outperform baseline ReLU in all of them. For example, the Fashion MNIST dataset, L-ReLU, ELiSH, Mish, and PFLU activation functions were able to obtain 0.0212, 0.0163, 0.001, and 0.0023 improvements in loss value compared to HcLSH, respectively. In SVHN, HcLSH obtained the second-best test loss value of 0.4441, suppressed by the loss value of PenTanh of 0.4351. However, in CIFAR10, HcLSH exhibited the lowest loss value with an enhancement of 0.0382 over the second-best loss that belongs to GCU. In the CIFAR100 dataset, HcLSH was outperformed by GCU with a 0.2562 difference, while HcLSH outperformed RSigELU, which obtained the third-best loss value, with a loss difference of 0.5526. With its opposite achieved results in the test loss and accuracy, GCU showed volatile performance.

TABLE 3. The mean (± standard deviation) of the test loss and test accuracy (%) for the LeNet [37] model obtained using different activation functions tested on the datasets: Fashion-MNIST, CIFAR 10, CIFAR 100, and SVHN. The red values denote the best result in the normalized accuracy, the green denotes the second-best result, and the blue denotes the third-best result. The underlined values denote the activation function with the poorest performance. Best viewed in color.

AF	Fashion MNIST			SVHN			CIFAR 10			CIFAR 100		
	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy
L-ReLU [16]	0.4261± 1.64E-02	84.64%± 7.08E-03	99.5%	0.5604± 1.63E-02	85.34%± 1.29E-03	99.8%	1.2179± 1.89E-02	56.89%± 6.29E-03	99.0%	3.0649± 1.70E-02	27.07%± 6.35E-03	95.8%
PenTanh [17]	0.4202± 1.68E-02	85.04%± 6.40E-03	100.0%	0.5574± 2.34E-02	85.44%± 6.63E-03	**100.0%**	1.2110± 2.89E-02	57.05%± 8.03E-03	**99.3%**	3.0084± 2.93E-02	28.37%± 6.29E-03	**100.4%**
Swish [20]	0.4408± 6.17E-03	84.16%± 3.28E-03	99.0%	0.6250± 1.46E-02	83.42%± 5.34E-03	97.6%	1.2713± 1.15E-02	55.06%± 2.82E-03	95.9%	3.0461 ±1.60E-02	27.30%± 9.59E-03	96.6%
ELiSH [21]	0.3965± 2.22E-03	85.57%± 4.16E-04	100.6%	0.5382± 2.11E-02	85.71%± 4.09E-03	**100.3%**	1.2397± 2.01E-02	56.34%± 6.95E-03	98.1%	3.0173± 3.91E-02	28.22%± 7.56E-03	99.9%
Mish [23]	0.4166± 8.05E-03	85.17%± 3.04E-03	100.2%	0.6036± 9.26E-03	84.15%± 4.04E-03	98.5%	1.2557± 2.04E-02	55.93%± 5.14E-03	97.4%	3.0064± 1.72E-02	28.17%± 4.82E-03	99.7%
RSigELU [26]	0.4237± 4.68E-03	84.85%± 1.93E-03	99.8%	0.5770± 1.51E-02	84.95%± 2.60E-03	99.4%	1.2367± 3.33E-02	56.65%± 1.18E-02	98.6%	3.0231± 5.76E-02	27.77%± 1.39E-02	98.3%
TanhExp [25]	0.3952± 2.38E-02	85.72%± 6.09E-03	**100.8%**	0.9466± 1.17E-01	71.91%± 3.68E-02	84.1%	1.2114± 1.62E-03	57.19%± 1.13E-03	**99.6%**	3.0185± 1.73E-02	28.39%± 6.49E-03	**100.5%**
PFLU [22]	0.4285± 2.98E-02	84.43%± 1.32E-02	99.3%	0.6032± 6.71E-03	84.05%± 1.93E-03	98.3%	1.2804± 1.17E-02	55.06%± 5.71E-03	95.8%	3.0627± 1.55E-02	27.57%± 5.10E-03	97.6%
GCU [28]	0.3465± 5.39E-03	87.33%± 1.31E-03	**102.7%**	0.5474± 1.28E-02	84.99%± 3.90E-03	99.4%	1.2532± 1.07E-02	57.02%± 2.35E-03	**99.3%**	3.0818± 5.01E-02	26.55%± 9.42E-03	94.0%
ReLU [18]	0.4226± 1.76E-02	85.04%± 7.96E-03	100.0%	0.5446± 2.09E-02	85.47%± 4.79E-03	**100.0%**	1.2105± 1.41E-02	57.44%± 4.67E-03	**100.0%**	3.0116± 4.11E-02	28.25%± 6.47E-03	100.0%
HcLSH (ours)	0.3876± 4.01E-03	86.13%± 9.07E-04	**101.3%**	0.5444± 6.68E-03	85.94%± 2.68E-03	**100.5%**	1.2198± 5.54E-03	56.89%± 2.41E-03	99.0%	2.9963± 1.67E-02	28.64%± 4.25E-03	**101.4%**

Regarding the models based on ShuffleNetV2 [43], as reported in Table 9, HcLSH also shows clear improvements by obtaining the best results with 3.9%, 11.7%, 23.5%, and 39.8% enhancement in average test accuracy compared with ReLU baseline in Fashion MNIST, SVHN, CIFAR10, and CIFAR100 datasets, respectively. Furthermore, HcLSH enhanced the test accuracy of RSigELU, the second-best activation function in terms of average test accuracy in the four datasets, by 0.73% in the Fashion MNIST dataset, by 2.05% in the SVHN dataset, by 4.54% in the CIFAR10 dataset, and by 3% in the CIFAR100 dataset. HcLSH achieved the best loss value in Fashion MNIST and SVHN datasets by suppressing the performance of the rest of the competing activation functions. In the CIFAR10 dataset, despite having the poorest accuracy value, GCU obtained the best loss value of 2.2417, followed by HcLSH with a loss value of 2.9582, then L-ReLU with a loss value of 3.1699. On the other hand, HcLSH obtained the third rank in terms of the test loss value in the CIFAR100 dataset, suppressed by GCU and RSigELU activation functions.

2) STATISTICAL ANALYSIS

Table 10 summarizes the detailed analysis of the performance of the proposed HcLSH and the other ten baseline activation functions in Table 3 to Table 9 concerning the normalized test accuracy value. The analysis tracks how well, equal, and bad is the performance of HcLSH compared to each baseline activation function individually in the 28 test scenarios, i.e., the experiments were conducted on four datasets with seven different deep neural architectures for each tested activation function. The terms "> Baseline", "= Baseline", and "< Baseline" are indicatives of better average test classification accuracy, equal accuracy, and worse accuracy, respectively. Overall, as observed, HcLSH consistently improves the classification accuracy over the different baseline functions by outperforming them in most cases. HcLSH consistently outperformed L-ReLU and Mish in 27 experiments, outperformed ReLU, Swish, PFLU, and RSigELU activation functions in 26 experiments, outperformed GCU in 25 experiments and finally outperformed penalized tanh, TanhExp, and ELiSH in 24 experiments.

Furthermore, Figure 5 offers another aggregation of the previously reported detailed results by counting the Top-1 (the best), and the Top-3 (the best three) results obtained by different activation functions applied to various models across multiple datasets. The superiority of the proposed HcLSH is witnessed compared to the other competing activation functions by acquiring 20 best results and 25 Top-3 results in the set of conducted experiments, whereas RSigELU is the second runner-up by succeeding to offer 14 Top-3 results; however, it fails to achieve any Top-1 result.

TABLE 4. The mean (± standard deviation) of the test loss and test accuracy (%) for the MobileNetV1 [38] model obtained using different activation functions tested on the datasets: Fashion-MNIST, CIFAR 10, CIFAR 100, and SVHN. The red values denote the best result in the normalized accuracy, the green denotes the second-best result, and the blue denotes the third-best result. The underlined values denote the activation function with the poorest performance. Best viewed in color.

AF	Fashion MNIST			SVHN			CIFAR 10			CIFAR 100		
	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy
L-ReLU [16]	0.4772± 2.64E-02	82.55%± 1.09E-02	101.4%	0.4446± 4.15E-02	86.80%± 1.07E-02	**101.8%**	1.1855± 1.67E-02	57.51%± 5.18E-03	101.7%	2.8407± 6.74E-03	32.07%± 4.42E-03	104.3%
PenTanh [17]	0.4852± 1.31E-02	82.61%± 8.04E-03	101.4%	0.4913± 1.06E-01	85.47%± 3.11E-02	100.2%	1.1874± 4.73E-03	57.49%± 2.89E-03	101.7%	2.8615± 6.32E-02	32.10%± 1.25E-02	104.4%
Swish [20]	0.5775± 2.21E-02	78.29%± 6.02E-03	96.1%	0.6383± 3.04E-02	80.87%± 1.09E-02	94.8%	1.3080± 2.97E-02	53.70%± 1.34E-02	95.0%	3.1517± 7.28E-02	30.11%± 1.39E-02	98.0%
ELiSH [21]	0.4704± 2.16E-02	82.91%± 8.97E-03	101.8%	0.4431± 8.80E-03	86.80%± 4.40E-03	**101.8%**	1.1701± 2.63E-02	58.41%± 1.22E-02	103.3%	2.9897± 2.79E-02	32.19%± 2.13E-03	104.7%
Mish [23]	0.5114± 2.50E-02	81.33%± 9.17E-03	99.9%	0.5728± 5.06E-02	82.98%± 1.50E-02	97.3%	1.2308± 2.15E-02	55.98%± 6.55E-03	99.0%	2.9911± 7.43E-02	32.02%± 6.48E-03	104.2%
RSigELU [26]	0.4554± 1.81E-02	83.51%± 7.80E-03	**102.5%**	0.5094± 5.65E-02	85.13%± 1.79E-02	99.8%	1.1589± 2.58E-02	58.76%± 8.59E-03	**104.0%**	2.8507± 3.98E-02	33.01%± 6.08E-03	**107.4%**
TanhExp [25]	0.4946± 3.00E-02	81.85%± 8.90E-03	100.5%	0.5113± 3.05E-02	84.90%± 6.63E-03	99.6%	1.1956± 3.32E-02	57.45%± 1.43E-02	101.6%	3.0406± 8.71E-02	32.62%± 5.37E-03	106.1%
PFLU [22]	0.5356± 6.36E-03	80.40%± 2.21E-03	98.7%	0.5701± 8.37E-02	82.85%± 2.41E-02	97.2%	1.2794± 1.77E-02	54.63%± 7.25E-03	96.7%	3.2279± 1.60E-01	29.78%± 9.04E-03	96.9%
GCU [28]	0.4549± 2.14E-02	83.31%± 1.25E-02	**102.3%**	0.4530± 2.34E-02	86.68%± 4.92E-03	**101.7%**	1.1025± 3.49E-02	61.06%± 1.21E-02	**108.0%**	2.8219± 1.99E-02	35.00%± 4.33E-03	**113.9%**
ReLU [18]	0.5126± 1.03E-02	81.44%± 6.41E-03	100.0%	0.4961± 8.73E-03	85.27%± 4.09E-0	100.0%	1.2145± 1.58E-02	56.52%± 6.73E-03	100.0%	2.8520± 1.81E-02	30.74%± 7.00E-03	100.0%
HcLSH (ours)	0.4307± 1.98E-03	84.25%± 3.18E-03	**103.5%**	0.4394± 1.97E-02	87.10%± 3.95E-03	**102.1%**	1.0822± 6.62E-03	61.89%± 4.80E-03	**109.5%**	2.8096± 4.16E-02	35.29%± 3.66E-03	**114.8%**

Apart from evaluating the performance based on the test accuracy, as shown in Table 10 and Figure 5, a non-parametric statistical analysis is also performed according to the metric, i.e., average test loss metric, since it is also a contributing factor in model performance. Two non-parametric tests were adopted; pair-wise comparison between the proposed HcLSH and each other activation function in the comparative set independently using the Wilcoxon signed rank test [44] and overall relative comparison using the Friedman ranking test [44].

Table 11 shows the average Friedman ranking of all the activation functions; it is distributed as a Chi-square distribution with 10 degrees of freedom, with statistics equal to 73.681818 and a p-value of 5.4006022E-11. HcLSH has the best performance as it has the best rank, with a rank equal to 9.25. This fact is supported by the Wilcoxon results in Table 12. To show that HcLSH is performing better than the other functions, the markers R+ and R- are used. Note that the higher the difference in these scores, the more consistent in performance. As demonstrated in Tables 11 and 12, the performance differences were strong enough that Friedman and Wilcoxon tests could detect the significance and rejected their null hypothesis, which claimed that there is no difference between our proposed HcLSH activation function and the compared one by obtaining p-values less than the significant level (0.05). Thus, the reported enhancements and improvements of HcLSH are statistically significant. The top activation functions with respect to loss metric, according to Table 11, are HcLSH, RSigELU, penalized tanh, ELiSH, and L-ReLU.

The concluding results and analysis in Table 3 to Table 12 demonstrate that the proposed HcLSH performance is stable, consistent, robust, and effective on various datasets and models. The proposed HcLSH generally guarantees performance improvements, i.e., an increase in accuracy and a decrease in loss, in the tested models for image classification. Moreover, the results obtained from the various architectures with different depths and capacities indicate the good modeling ability and adaptation of the proposed HcLSH to the scalability requirement.

3) THE CONVERGENCE CURVE

The convergence characteristics of the two investigated metrics in the experiments are shown in Figure 6 and Figure 7, in which the learning behaviors of four top-performing activation functions: Mish, ELiSH, TanhExp, and RSigELU, in addition to HcLSH, are reported for the InceptionNetV4 [41] and MobileNetV1 [38] models trained on the CIFAR10 dataset. Regarding InceptionNetV4 model Mish, TanhExp, and ELiSH achieved better validation accuracy than HcLSH in the first half of training epochs, as observed in Figure 6(a). Then, the performance of Mish and ELiSH started to

TABLE 5. The mean (± standard deviation) of the test loss and test accuracy (%) for the KerasNet [39] model obtained using different activation functions tested on the datasets: Fashion-MNIST, CIFAR 10, CIFAR 100, and SVHN. The red values denote the best result in the normalized accuracy, the green denotes the second-best result, and the blue denotes the third-best result. The underlined values denote the activation function with the poorest performance. Best viewed in color.

AF	Fashion MNIST			SVHN			CIFAR 10			CIFAR 100		
	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy
L-ReLU [16]	0.2401± 3.81E-03	90.98%± 2.86E-03	100.0%	0.2671± 6.65E-03	92.94%± 1.99E-04	99.6%	0.6592± 5.40E-03	77.24%± 1.82E-03	99.1%	2.0574± 7.33E-03	47.97%± 2.41E-03	**103.7%**
PenTanh [17]	0.2367± 1.26E-03	91.14%± 4.04E-04	100.2%	0.3008± 3.41E-03	92.92%± 1.33E-04	99.6%	0.6551± 4.33E-03	77.96%± 2.26E-03	**100.0%**	2.2127± 4.82E-02	46.73%± 5.04E-03	101.1%
Swish [20]	0.2755± 7.94E-03	89.78%± 2.53E-03	98.7%	0.3074± 8.654E-03	92.49%± 2.84E-03	99.1%	0.7647± 4.16E-03	74.16%± 1.92E-03	95.2%	2.3110± 2.53E-02	44.36%± 4.05E-03	95.9%
ELiSH [21]	0.2271± 4.67E-04	91.65%± 1.26E-03	**100.8%**	0.2748± 4.62E-03	93.26%± 9.89E-04	**100.0%**	0.6524± 1.10E-02	78.35%± 2.70E-03	**100.5%**	2.1020± 4.59E-02	47.40%± 5.97E-03	102.5%
Mish [23]	0.2535± 1.50E-03	90.69%± 7.94E-04	99.7%	0.2931± 7.43E-03	92.88%± 4.11E-04	99.5%	0.7079± 2.74E-02	76.45%± 7.65E-03	98.1%	2.2221± 5.11E-03	45.74%± 3.20E-03	98.9%
RSigELU [26]	0.2389± 3.78E-03	91.25%± 1.00E-03	100.3%	0.2977± 1.66E-03	93.00%± 1.04E-03	**99.7%**	0.6656± 5.96E-03	77.59%± 6.51E-04	99.6%	2.1507± 1.78E-02	47.96%± 6.77E-03	**103.7%**
TanhExp [25]	0.2356± 1.09E-03	91.32%± 4.58E-04	**100.4%**	0.3093± 4.899E-02	92.94%± 7.88E-04	99.6%	0.6619± 4.02E-03	77.85%± 2.30E-03	99.9%	2.0659± 2.72E-02	48.92%± 3.17E-03	**105.8%**
PFLU [22]	0.2592± 5.89E-03	90.40%± 3.00E-03	99.4%	0.3040± 3.26E-03	92.88%± 8.27E-04	99.5%	0.7688± 1.47E-02	74.55%± 4.23E-03	95.7%	2.3674± 1.82E-02	43.02%± 2.30E-03	93.0%
GCU [28]	0.2375± 2.70E-03	91.47%± 6.08E-04	**100.6%**	0.2870± 4.69E-03	92.68%± 1.46E-03	99.3%	0.6978± 9.88E-03	77.12%± 2.94E-03	99.0%	2.1472± 1.52E-02	46.56%± 2.87E-03	100.7%
ReLU [18]	0.2416± 6.24E-03	90.97%± 1.70E-03	100.0%	0.2681± 3.29E-03	93.30%± 2.22E-05	**100.0%**	0.6480± 7.56E-03	77.94%± 4.65E-03	**100.0%**	2.1397± 1.88E-02	46.23%± 6.53E-03	100.0%
HcLSH (ours)	0.2428± 2.72E-03	91.10%± 1.00E-03	100.2%	0.2899± 1.01E-03	93.10%± 1.33E-04	**99.8%**	0.6430 ± 1.48E-02	78.51%± 3.99E-03	**100.7%**	2.0248± 7.79E-03	50.76%± 2.91E-03	**109.8%**

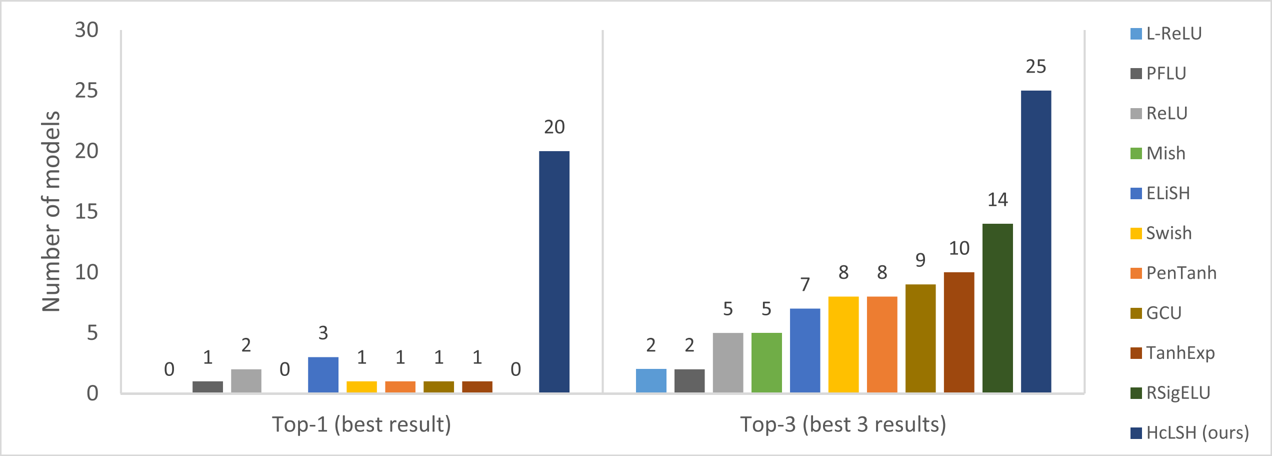


FIGURE 5. The number of models in which each activation function achieved the best result (Top-1) and the three best results (Top-3).

alternate. On the other hand, HcLSH reached the best final accuracy at the end of the training process in a stable gradual growth trend. Also, HcLSH was faster than RSigELU. Regarding the validation loss convergence curve, Figure 6(b), Mish was the fastest activation function that obtained the best loss values in the initial training epochs. However, after epoch 42, it seems that it encountered an overfitting issue because of the sharp increase in loss values in its following epochs. Meanwhile, HcLSH decreased the loss in steady steps with the increase in training epochs without any fluctuations or sudden changes in the loss values and obtained the lowest loss at the end of the training epochs.

On the other hand, the superiority of the performance of the MobileNetV1 model equipped with the proposed HcLSH function is more obvious, as noted in Figure 7, in terms of validation loss and accuracy, followed with a decent gap by the models that utilized ELiSH, TanhExp, RSigELU, and Mish.

TABLE 6. The mean (± standard deviation) of the test loss and test accuracy (%) for the SqueezeNet [40] model obtained using different activation functions tested on the datasets: Fashion-MNIST, CIFAR 10, CIFAR 100, and SVHN. The red values denote the best result in the normalized accuracy, the green denotes the second-best result, and the blue denotes the third-best result. The underlined values denote the activation function with the poorest performance. Best viewed in color.

	Fashion MNIST			SVHN			CIFAR 10			CIFAR 100		
AF	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy
L-ReLU [16]	0.3870± 1.57E-02	86.15%± 4.33E-03	99.3%	0.4552± 1.08E-02	87.30%± 4.99E-04	102.3%	1.1311± 5.882E-02	61.42%± 1.86E-02	104.7%	3.3245± 3.53E-02	21.84%± 9.77E-03	136.4%
PenTanh [17]	0.3961± 1.43E-02	85.76%± 4.33E-03	98.9%	0.4483± 1.20E-02	87.81%± 1.13E-03	102.9%	1.1192± 3.770E-02	62.22%± 1.05E-02	106.1%	3.1142± 5.09E-02	25.33%± 4.88E-03	158.2%
Swish [20]	0.5063± 3.74E-03	81.21%± 1.70E-03	93.6%	0.6135± 1.76E-02	81.69%± 4.86E-03	95.7%	1.3686± 4.603E-02	49.88%± 2.39E-02	85.0%	3.1265± 2.80E-02	24.35%± 1.22E-02	152.1%
ELiSH [21]	0.3638± 8.48E-03	86.74%± 3.13E-03	**100.0%**	0.4645± 2.99E-03	87.68%± 2.12E-03	102.7%	1.1756± 5.564E-02	62.52%± 8.25E-03	106.6%	3.0459± 7.70E-02	26.96%± 9.38E-03	168.4%
Mish [23]	0.4455± 8.56E-03	83.91%± 3.21E-03	96.7%	0.5925± 6.50E-02	83.89%± 1.24E-02	98.3%	1.1897± 2.339E-02	57.98%± 5.71E-03	98.8%	2.9348± 3.99E-02	27.18%± 1.11E-02	169.8%
RSigELU [26]	0.3785± 2.32E-02	86.26%± 8.81E-03	99.4%	0.4632± 2.61E-02	87.66%± 6.74E-03	102.7%	1.1218± 2.718E-02	64.01%± 5.71E-03	**109.1%**	3.0257± 2.96E-02	27.32%± 2.26E-03	**170.6%**
TanhExp [25]	0.3745± 2.94E-02	86.52%± 8.16E-03	**99.7%**	0.4327± 1.38E-02	90.51%± 5.32E-0	**106.0%**	1.1520± 3.422E-02	62.49%± 1.05E-02	106.5%	2.9450± 3.16E-02	29.18%± 4.97E-03	**182.3%**
PFLU [22]	0.4628± 1.34E-02	83.17%± 4.52E-03	95.9%	0.6765± 9.20E-02	80.14%± 1.99E-02	93.9%	1.2696± 2.238E-02	55.13%± 1.22E-02	94.0%	3.0881± 3.65E-02	24.53%± 5.46E-03	153.2%
GCU [28]	1.4061± 4.82E-02	85.14%± 8.22E-03	98.1%	1.4862± 3.57E-03	87.98%± 4.48E-03	**103.1%**	1.6988± 2.551E-03	63.11%± 1.18E-02	**107.6%**	4.0182± 6.47E-03	16.87%± 2.61E-03	105.4%
ReLU [18]	0.3730± 3.92E-03	86.75%± 1.30E-03	**100.0%**	0.5023± 5.61E-02	85.36%± 2.15E-02	100.0%	1.1814± 6.137E-02	58.67%± 2.45E-02	100.0%	3.9443± 2.41E-02	16.01%± 1.16E-02	100.0%
HcLSH (ours)	0.3566± 4.58E-03	87.28%± 1.73E-03	**100.6%**	0.5205± 8.04E-03	88.23%± 1.86E-03	**103.4%**	1.2751± 5.363E-02	64.87%± 1.63E-03	**110.6%**	3.0313± 2.50E-02	31.44%± 4.51E-03	**196.4%**

4) TIME COMPLEXITY

The computational impact of the different activation functions on the integrated model is tested by recording the time (in seconds) required to complete one training epoch. This subsection investigates the epoch time of the InceptionNetV4 [41] model tested on the CIFAR10 image classification dataset under different AFs. The average training time of the first three epochs of each experiment is reported in Table 13. The activation function's training time depends on the implementation and hardware optimization for speed enhancement. Thus, all the AFs were implemented following the same procedure, except for ReLU and L-ReLU functions, to provide a fair comparison.

As can be observed, the models with ReLU and L-ReLU were the fastest, with the least runtime to complete an epoch due to their simplicity. However, the challenges mentioned earlier that face the rectified functions family cause degradation in the obtained performance, as demonstrated in the previous subsections. On the other hand, despite the impressive and noteworthy performance of HcLSH on most of the conducted experiments, it is more computationally intensive than Swish, TanhExp, GCU, and RSigELU, while utilizing similar computational costs as PenTanh and Mish. HcLSH is less demanding than ELiSH and PFLU. In summary, the computational complexity of these AFs is roughly sorted in ascending order as in Table 13, according to the conducted experiments based on per epoch training time.

D. ABLATION STUDY

This section focuses on the performance of the proposed HcLSH activation function by adjusting several hyperparameters. These parameters include the depth of the network represented by the number of layers, the weight initialization methods, the use of value normalization, and the optimizers. To better reflect the performance of the activation function, the InceptionNetV4 model [41] is employed as our network model for the experiments in this section trained and evaluated on the CIFAR10 dataset. Different choices of these parameters were manipulated one at a time to observe their effects and roles on the performance. Except for the examined parameter in each subsection, all other parameters' values were kept constant, the same as in section IV-B. To better visualize the results, we chose four activation functions from the top-performing ones: RSigELU, TanhExp, Mish, and ELiSH, to conduct the experiments in the following subsections. GCU was excluded in evaluations of this section since it encountered bad performance under this particular choice of network architecture.

1) ANALYSIS OF THE NUMBER OF LAYERS

Layer-wise test accuracy of various layers is carried out to confirm that HcLSH remains stable as the number of model layers increases. The experiments were carried out on a model consisting of a 2D convolution layer with 20 filters with a size of 5 × 5, then another 2D convolution layer with 50 filters with

TABLE 7. The mean (± standard deviation) of the test loss and test accuracy (%) for the InceptionNetV4 [41] model obtained using different activation functions tested on the datasets: Fashion-MNIST, CIFAR 10, CIFAR 100, and SVHN. The red values denote the best result in the normalized accuracy, the green denotes the second-best result, and the blue denotes the third-best result. The underlined values denote the activation function with the poorest performance. Best viewed in color.

AF	Fashion MNIST			SVHN			CIFAR 10			CIFAR 100		
	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy
L-ReLU [16]	0.2630± 1.55E-02	90.63%± 5.89E-03	99.6%	0.1935± 3.07E-03	95.15%± 5.06E-04	100.0%	0.5424± 2.40E-02	82.57%± 7.30E-03	102.0%	1.6994± 4.33E-02	54.72%± 4.54E-03	106.8%
PenTanh [17]	0.2324± 3.82E-03	91.82%± 1.14E-03	**100.9%**	0.1914± 5.83E-04	95.18%± 5.32E-04	**100.1%**	0.5210± 1.29E-02	83.43%± 1.63E-03	**103.1%**	1.6360± 3.87E-02	56.04%± 6.03E-03	109.4%
Swish [20]	0.2391± 1.59E-02	91.70%± 4.23E-03	100.8%	0.1985± 3.79E-04	95.24%± 4.66E-04	**100.1%**	0.5210± 4.64E-02	84.11%± 1.23E-02	**103.9%**	1.5771± 5.11E-03	58.21%± 3.95E-03	**113.6%**
ELiSH [21]	0.2373± 2.95E-02	91.67%± 1.08E-02	100.8%	0.1861± 6.94E-03	95.50%± 1.73E-03	**100.4%**	0.5676± 4.69E-02	82.58%± 1.34E-02	102.0%	1.6383± 1.20E-02	55.82%± 4.23E-03	109.0%
Mish [23]	0.2359± 1.39E-02	91.93%± 1.05E-03	**101.1%**	0.2024± 1.16E-02	95.13%± 2.61E-04	100.0%	0.5884± 5.19E-02	82.26%± 1.26E-02	101.6%	1.5880± 4.45E-03	57.96%± 2.38E-03	**113.1%**
RSigELU [26]	0.2445± 1.74E-02	91.50%± 5.16E-03	100.6%	0.2020± 1.11E-02	95.01%± 2.14E-04	99.9%	0.5315± 1.69E-02	82.70%± 4.23E-03	102.2%	1.5379± 1.30E-02	57.81%± 4.85E-03	112.8%
TanhExp [25]	0.2426± 8.56E-03	91.64%± 8.00E-04	100.7%	0.2093± 1.56E-02	94.99%± 1.71E-03	99.9%	0.5034± 6.10E-03	84.03%± 1.27E-03	**103.8%**	1.5793± 7.64E-02	57.81%± 1.59E-02	112.8%
PFLU [22]	0.2213± 5.74E-03	92.34%± 1.15E-03	**101.5%**	0.1991± 1.97E-02	95.17%± 1.49E-03	100.0%	0.5641± 4.30E-02	82.77%± 9.19E-03	102.3%	1.6397± 1.30E-02	56.38%± 7.37E-04	110.1%
GCU [28]	1.5482± 7.08E-02	37.79%± 3.20E-02	41.5%	2.1745± 3.49E-02	22.17%± 2.10E-02	23.3%	2.0875± 1.71E-02	23.56%± 5.84E-03	29.1%	4.2986± 6.99E-02	4.59%± 8.22E-03	9.0%
ReLU [18]	0.2622± 5.34E-02	90.96%± 1.53E-02	100.0%	0.1937± 4.94E-03	95.13%± 8.81E-04	100.0%	0.6012± 7.96E-02	80.93%± 2.26E-02	100.0%	1.8550± 1.19E-01	51.23%± 2.12E-02	100.0%
HcLSH (ours)	0.2423± 1.01E-03	91.54%± 1.45E-03	100.6%	0.1796± 1.18E-02	95.36%± 5.51E-04	**100.2%**	0.4835± 1.43E-02	84.08%± 4.32E-03	**103.9%**	1.4817± 2.97E-02	58.76%± 8.75E-03	**114.7%**

a size of 5 × 5, followed by MaxPooling2D and Flatten layers, then a loop with fully connected layers with 500 neurons with the number of layers gradually increased from 8 to 30. Each layer in the model is followed by batch normalization and the tested activation function. The models were trained for ten epochs with a batch size of 128 and a learning rate of 10^{-3} with categorical cross-entropy loss optimized using ADAM optimizer. The results are visualized in Figure 8.

The superiority of HcLSH is clearly shown by obtaining an accuracy curve that outperformed the other tested activation functions with a good margin. Then RSigELU followed, while Mish and TanhExp obtained mixed performance. The worst accuracy belonged to the ELiSH activation function. In summary, HcLSH obtained an accuracy of ∼42.5%, RSigELU obtained ∼27%, Mish and TanhExp obtained ∼21%, and ELiSH obtained ∼11%. As seen in Figure 8, the testing accuracies tend to generally decrease as the number of layers increases, which is expected since the complexity of the model and the number of trainable parameters increased, and the optimization process became harder. However, the significant decline in test accuracy in the model trained with the ELiSH activation function suggested that it suffered from an overfitting issue as the model became deeper; on the other hand, HcLSH maintained more stability in the decrease of accuracy as the number of layers increases.

2) ANALYSIS OF WEIGHT INITIALIZATION METHODS

The initialization of the parameter weights at the beginning of the training process is a contributing factor in model performance. The effects of applying different parameter initialization in the examined model with HcLSH are investigated. Four different initialization methods are tested: Glorot-normal initialization (AKA Xavier initialization) [45], Glorot-uniform initialization [45], He-uniform initialization [46], and Lecun-normal initialization [47].

As can be observed from Figure 9, the performance of different weight initialization methods varies among different activation functions but with insignificant margins. The Lecun-normal method generally produces better results in all five tested activation functions. In contrast, the He-uniform method has the poorest performance among the different initialization methods. HcLSH produced the best results with whatever the initial method used. For instance, it obtained an average test accuracy in the Glorot-uniform method of 84.08%, in the Glorot-normal method of 83.87%, in the He-uniform method of 82.81%, and in the Lecun-normal method of 84.43%. This illustrates that HcLSH is adaptive to different initial values and thus reduces the sensitivity to initialization. As shown in Figure 9, the performance of TanhExp followed the performance achieved by HcLSH in three initialization methods (Glorot-uniform, He-uniform, and Lecun-normal methods), whereas Mish was the second-ranking activation

TABLE 8. The mean (± standard deviation) of the test loss and test accuracy (%) for the ResNetV2-20 [42] model obtained using different activation functions tested on the datasets: Fashion-MNIST, CIFAR 10, CIFAR 100, and SVHN. The red values denote the best result in the normalized accuracy, the green denotes the second-best result, and the blue denotes the third-best result. The underlined values denote the activation function with the poorest performance. Best viewed in color.

	Fashion MNIST			SVHN			CIFAR 10			CIFAR 100		
AF	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy
L-ReLU [16]	0.5450± 2.06E-02	90.22%± 5.47E-03	101.3%	0.5383± 1.15E-01	90.82%± 1.77E-02	100.5%	2.4062± 7.18E-02	63.06%± 6.47E-03	105.6%	5.0488± 1.66E-01	32.89%± 1.10E-02	108.8%
PenTanh [17]	0.5738± 1.16E-02	90.29%± 3.15E-03	101.4%	0.4351± 6.61E-02	92.44%± 1.19E-02	**102.3%**	2.2832± 2.21E-01	63.42%± 1.42E-02	106.2%	4.7684± 1.57E-01	33.76%± 1.35E-02	111.6%
Swish [20]	0.5963± 2.42E-02	90.26%± 2.68E-03	101.3%	0.4631± 5.36E-02	92.01%± 1.57E-02	**101.8%**	2.0718± 4.91E-02	67.35%± 3.07E-03	**112.8%**	4.9935± 7.03E-02	33.66%± 4.31E-03	111.3%
ELiSH [21]	0.5499± 9.15E-03	90.23%± 1.80E-03	101.3%	0.5207± 7.68E-02	91.62%± 1.20E-02	101.4%	2.3585± 1.11E-01	63.07%± 2.66E-03	105.6%	5.0004± 1.63E-01	32.52%± 1.19E-02	107.5%
Mish [23]	0.5652± 1.04E-02	90.53%± 8.50E-04	**101.6%**	0.4749± 2.79E-02	91.82%± 8.84E-03	101.6%	2.3372± 2.10E-01	64.44%± 1.16E-02	107.9%	4.9360± 8.25E-02	34.18%± 5.17E-03	**113.0%**
RSigELU [26]	0.5624± 7.37E-03	90.66%± 5.03E-04	**101.8%**	0.5276± 1.39E-01	90.50%± 2.01E-02	100.2%	2.2105± 1.37E-01	64.53%± 1.70E-02	**108.1%**	4.4766± 7.77E-02	35.71%± 2.72E-03	**118.1%**
TanhExp [25]	0.5744± 3.80E-03	90.31%± 1.31E-03	101.4%	0.5700± 1.18E-01	91.13%± 1.54E-02	100.9%	2.2714± 6.04E-02	64.50%± 5.96E-03	108.0%	4.7962± 4.11E-02	34.00%± 4.41E-03	112.4%
PFLU [22]	0.5639± 7.99E-03	90.23%± 1.97E-03	101.3%	0.4465± 1.70E-02	92.32%± 4.44E-05	**102.2%**	2.5088± 1.56E-01	61.92%± 1.11E-02	103.7%	5.3909± 2.76E-01	31.30%± 1.91E-02	103.5%
GCU [28]	0.7079± 1.23E-02	80.73%± 5.89E-03	90.6%	2.9055± 1.67E-02	42.14%± 9.91E-03	46.6%	1.8872± 2.29E-02	37.93%± 9.75E-03	63.5%	3.6678± 4.60E-02	17.31%± 8.90E-03	57.2%
ReLU [18]	0.5961± 1.15E-02	89.07%± 8.18E-03	100.0%	0.5570± 1.21E-01	90.35%± 1.76E-02	100.0%	2.4256± 6.39E-02	59.72%± 1.08E-02	100.0%	5.1376± 2.80E-01	30.24%± 1.28E-02	100.0%
HcLSH (ours)	0.5662± 2.32E-02	91.01%± 1.71E-03	**102.2%**	0.4441± 4.49E-03	92.36%± 9.88E-04	**102.2%**	1.8490± 8.74E-02	69.49%± 1.07E-02	**116.4%**	3.9240± 1.65E-01	39.36%± 6.09E-03	**130.2%**

function in the Glorot-normal method. The initialization method that produced more stable results (indicated with a small standard deviation of three runs) belongs to He-uniform and Lecun-normal methods. On the other hand, high variance is noticed in the Glorot-normal method.

3) ANALYSIS OF NORMALIZATIONS

Four potential cases were tested for the five activation functions to analyze the effects of normalization on the training and the evaluation of deep neural networks.

Case 1: no normalization is performed for the model input and without using batch normalization [48] between the model layers.

Case 2: normalization is performed for the model input without batch normalization.

Case 3: no normalization is performed for the model input with batch normalization.

Case 4: normalization is performed for the model input with batch normalization.

All the tested activation functions: HcLSH, RSigELU, Mish, TanhExp, and ELiSH, failed to converge and produced (NAN) results when no normalization is done to the input in the preprocessing and no batch normalization (BN) is used in the model (case 1). This indicates the importance of value normalization to any deep model. The results in Figure 10 prove the effectiveness of BN in enhancing the quality of the evaluation of the test results since the two cases using batch normalization (case 3 and 4) are better than those without. BN depends on the statistics of a specific portion of the input contained in the minibatch to normalize the activation values and reduce their reliance on initialization. The network with normalized inputs and no BN (case 2) succeeded in obtaining relatively good results without failing to converge, unlike the case with no value normalizations at all (case 1). However, the results were worse than with including batch normalization in the network. As expected, the case of normalizing the input and using batch normalization (case 4) obtained the best results when testing HcLSH. However, without performing input normalization but with batch normalization (case 3), the results were better for the rest of the activation functions; nevertheless, HcLSH also obtained the best among them.

4) ANALYSIS OF DIFFERENT OPTIMIZERS

In deep learning, many possible optimizer choices exist that have different characteristics and methods of parameter adjustments during the optimization process. Thus, choosing the appropriate optimizer is vital in network training. In the current study, ADAM [35], RMSprop [49], and SGD [50] optimizers are selected to test their effects. As shown in Figure 11, all five tested activation functions exhibited the best accuracy with ADAM optimizer, except for Mish, which performed better in RMSprop optimizer.

TABLE 9. The mean (± standard deviation) of the test loss and test accuracy (%) for the ShuffleNetV2 [43] model obtained using different activation functions tested on the datasets: Fashion-MNIST, CIFAR 10, CIFAR 100, and SVHN. The red values denote the best result in the normalized accuracy, the green denotes the second-best result, and the blue denotes the third-best result. The underlined values denote the activation function with the poorest performance. Best viewed in color.

AF	Fashion MNIST			SVHN			CIFAR 10			CIFAR 100		
	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy	Test Loss	Test Accuracy (%)	Normalized Accuracy
L-ReLU [16]	0.7017± 2.40E-02	86.28%± 2.79E-03	101.5%	1.2633± 6.79E-02	78.79%± 8.80E-03	104.3%	3.1699± 1.05E-02	50.24%± 3.64E-03	105.4%	7.4525± 1.11E-01	21.57%± 3.19E-03	109.7%
PenTanh [17]	0.7079± 3.47E-02	86.50%± 4.93E-03	101.8%	1.2146± 4.22E-02	79.95%± 1.51E-02	105.8%	3.3162± 1.07E-01	50.72%± 6.37E-03	106.4%	7.5481± 1.12E-01	22.46%± 8.05E-03	114.2%
Swish [20]	0.7219± 1.57E-02	86.94%± 4.08E-03	102.3%	1.2382± 2.18E-02	81.01%± 3.60E-03	**107.2%**	3.4164± 9.15E-02	52.78%± 1.31E-02	**110.7%**	7.8424± 1.20E-01	22.64%± 4.48E-03	**115.2%**
ELiSH [21]	0.7212± 4.01E-02	86.13%± 6.15E-03	101.3%	1.3100± 5.49E-02	77.55%± 9.82E-03	102.6%	3.4198± 8.68E-02	48.59%± 3.53E-03	101.9%	7.8519± 1.90E-01	20.02%± 6.38E-03	101.8%
Mish [23]	0.7171± 2.57E-02	87.22%± 3.79E-04	**102.6%**	1.2743± 7.72E-02	79.85%± 1.12E-02	105.7%	3.4617± 9.60E-02	51.34%± 6.41E-03	107.7%	7.9366± 4.98E-02	21.66%± 5.28E-03	110.2%
RSigELU [26]	0.6693± 4.05E-02	87.55%± 4.47E-03	**103.0%**	1.1178± 8.34E-02	82.33%± 8.85E-03	**108.9%**	3.1793± 3.98E-02	54.34%± 1.10E-02	**114.0%**	7.4178± 7.32E-02	24.48%± 1.06E-03	**124.5%**
TanhExp [25]	0.6930± 1.29E-02	87.23%± 1.86E-03	**102.6%**	1.3759± 9.22E-02	78.42%± 1.06E-02	103.8%	3.4421± 9.12E-02	50.87%± 6.85E-03	106.7%	7.9647± 1.49E-01	21.71%± 5.33E-03	110.4%
PFLU [22]	0.7723± 2.21E-02	85.64%± 5.76E-03	100.8%	1.3269± 4.79E-02	78.13%± 8.35E-03	103.4%	3.5207± 5.89E-02	47.77%± 1.06E-02	100.2%	7.8815± 1.08E-01	20.42%± 8.72E-03	103.8%
GCU [28]	1.9364± 7.89E-02	31.63%± 2.84E-02	37.2%	2.2221± 1.05E-02	19.28%± 2.46E-03	25.5%	2.2417± 5.81E-03	16.87%± 3.95E-03	35.4%	4.5166± 1.11E-02	2.68%± 2.28E-03	13.6%
ReLU [18]	0.6825± 1.47E-02	84.98%± 2.14E-03	100.0%	1.4242± 1.31E-01	75.57%± 1.68E-02	100.0%	3.2017± 1.25E-02	47.68%± 7.64E-04	100.0%	7.4691± 1.47E-01	19.66%± 6.41E-03	100.0%
HcLSH (ours)	0.6580± 3.87E-02	88.28%± 4.95E-03	**103.9%**	1.0149± 1.78E-02	84.38%± 4.58E-03	**111.7%**	2.9582± 1.44E-02	58.88%± 5.10E-03	**123.5%**	7.4330± 6.90E-02	27.48%± 3.50E-03	**139.8%**

TABLE 10. The number of models on which HcLSH outperforms, equally performs, or underperforms each baseline activation function during the conducted experiments with respect to the normalized test accuracy value.

Baselines	HcLSH > baseline	HcLSH= baseline	HcLSH < baseline
ReLU [18]	26	0	2
L-ReLU [16]	27	1	0
PenTanh [17]	24	1	3
Swish [20]	26	1	1
ELiSH [21]	24	0	4
Mish [23]	27	0	1
RSigELU [26]	26	1	1
TanhExp [25]	24	0	4
PFLU [22]	26	1	1
GCU [28]	25	0	3

TABLE 11. Ranking of different competitive activation functions on the different architectures, achieved by the Friedman test at 0.05 significance value based on average test loss values. The higher the value of the rank, the better performing the algorithm.

Algorithm	Ranking
ReLU [18]	5.607
L-ReLU [16]	6.536
PenTanh [17]	7.250
Swish [20]	3.821
ELiSH [21]	7.107
Mish [23]	4.893
RSigELU [26]	7.179
TanhExp [25]	5.893
PFLU [22]	3.357
GCU [28]	5.107
HcLSH (ours)	9.250
Statistic	73.681818
***p*-value**	5.4006022E-11

However, RMSprop optimizer seems to produce good results with slight differences compared with ADAM optimizer, -0.28%, -0.09%, -0.35%, and -0.35% for HcLSH, ELiSH, RSigELU, and TanhExp, and it also outperformed Mish with ADAM optimizer with 1.09% in terms of testing accuracy.

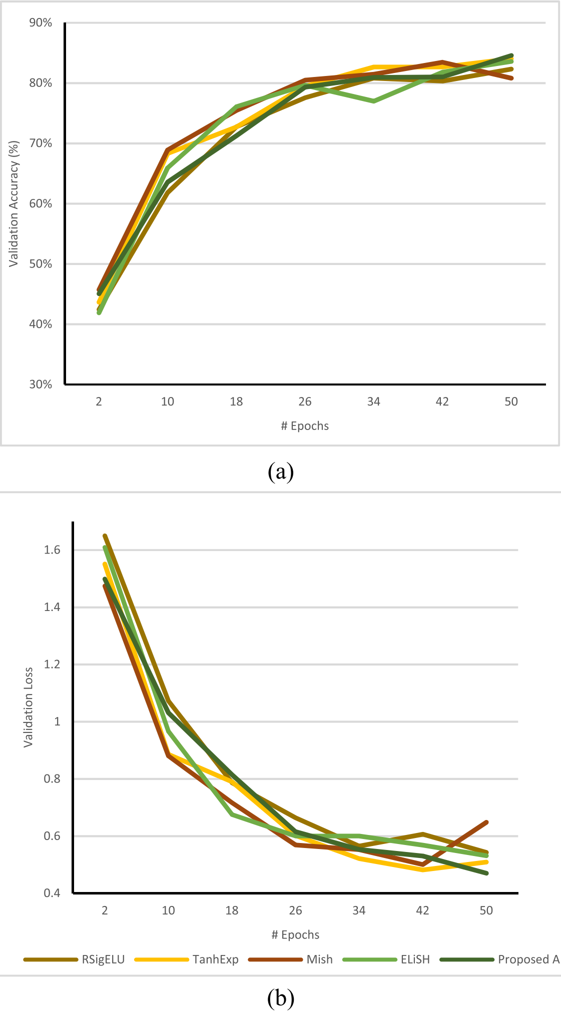
On the other hand, SGD optimizer was the worst-performing optimizer among the three. The possible reason behind this performance is the low learning rate used (10^{-4}) compared with the recommended value for that optimizer in the Keras library (10^{-2}). The performance of HcLSH is better in all three optimizers.

V. DISCUSSION

Because of their different characteristics, no activation function works best on every architecture and dataset. Under the same experimental setting, it can be depicted that the test accuracy of the Fashion MNIST dataset is the highest

TABLE 12. The results of the Wilcoxon signed-rank test for the different competitive activation functions against the proposed HcLSH activation function based on average test loss values. The significance value is 0.05.

Proposed HcLSH vs.	R+	R-	P-value	L-value
ReLU [18]	359	47	8.202E-05	+
L-ReLU [16]	346	60	7.921E-04	+
PenTanh [17]	352	54	3.418E-04	+
Swish [20]	404	2	1.490E-08	+
ELiSH [21]	349	57	1.867E-03	+
Mish [23]	356	50	3.663E-05	+
RSigELU [26]	314	92	6.468E-04	+
TanhExp [25]	340	66	6.468E-04	+
PFLU [22]	379	27	4.098E-07	+
GCU [28]	326	80	2.440E-03	+

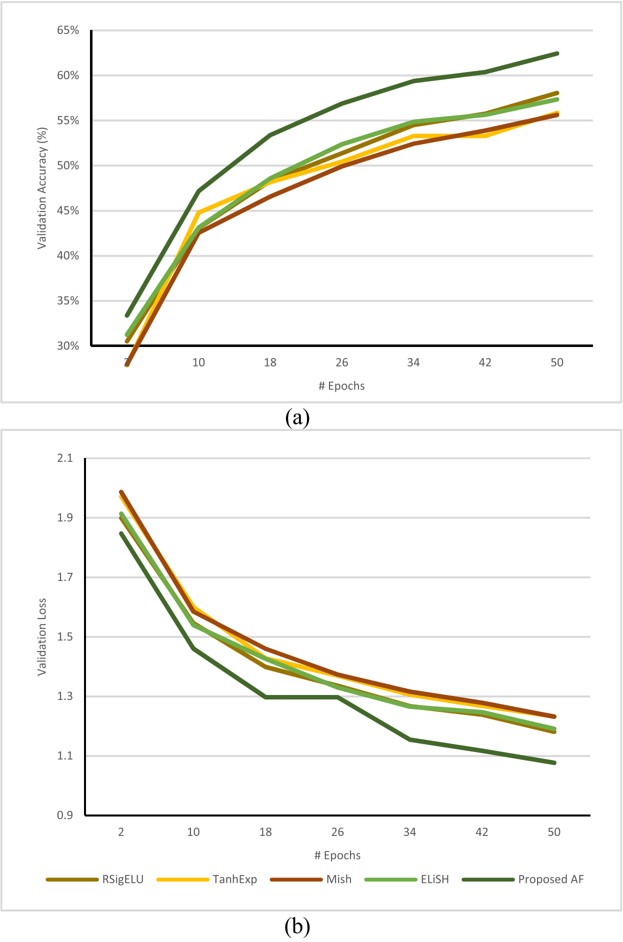


(a)

(b)

FIGURE 6. Plot of validation Accuracy (top part) and validation loss (bottom part) of different activation functions against HcLSH in the InceptionNetV4 on the CIFAR10 dataset. Best viewed in color.

among the four tested and used benchmark datasets across different models and architectures, followed by the SVHN and CIFAR10 datasets, while the CIFAR100 dataset was the most complex and challenging. This is observed by the poor accuracy obtained by the different models when operated under the same conditions in the CIFAR100 dataset.



(a)

(b)

FIGURE 7. Plot of validation Accuracy (top part) and validation loss (bottom part) of different activation functions against HcLSH in the MobileNetV1 on the CIFAR10 dataset. Best viewed in color.

TABLE 13. The comparison between computational training epoch time (in seconds) of different activation function measured using InceptionNetV4 [41] model.

	Epoch training time (seconds)
ReLU [18]	36.363
L-ReLU [16]	46.138
Swish [20]	48.093
TanhExp [25]	54.740
GCU [28]	55.913
RSigELU [26]	58.650
HcLSH (ours)	62.169
Mish [23]	63.733
PenTanh [17]	64.124
ELiSH [21]	65.688
PFLU [22]	76.636

As can be observed from the past reported results, the general performance of some AFs, that are injected into the different classification models varies across the datasets. For instance, ELiSH, RSigELU, TanhExp, and Mish performed well in the FASHION MNIST dataset; PenTanh, ELiSH, PFLU, RSigELU, ReLU, and HcLSH performed well in the SVHN dataset; HcLSH, RSigELU, Swish, and TanhExp performed well in CIFAR10 dataset; and finally, HcLSH,

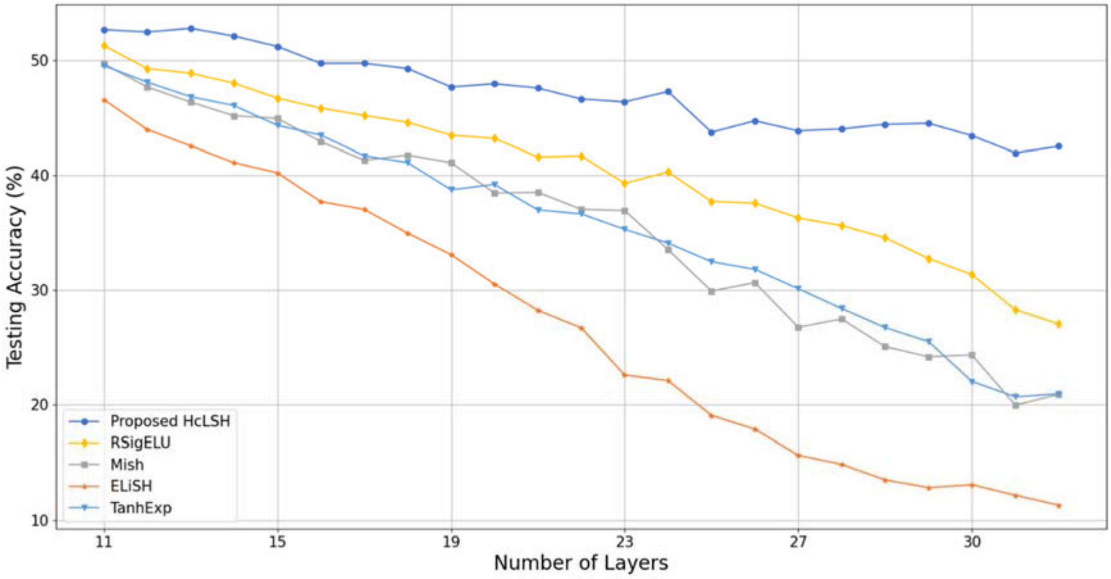


FIGURE 8. The testing accuracy vs. the number of layers on the CIFAR10 dataset using HcLSH, RSigELU, Mish, ELiSH, and TanhExp activation functions.

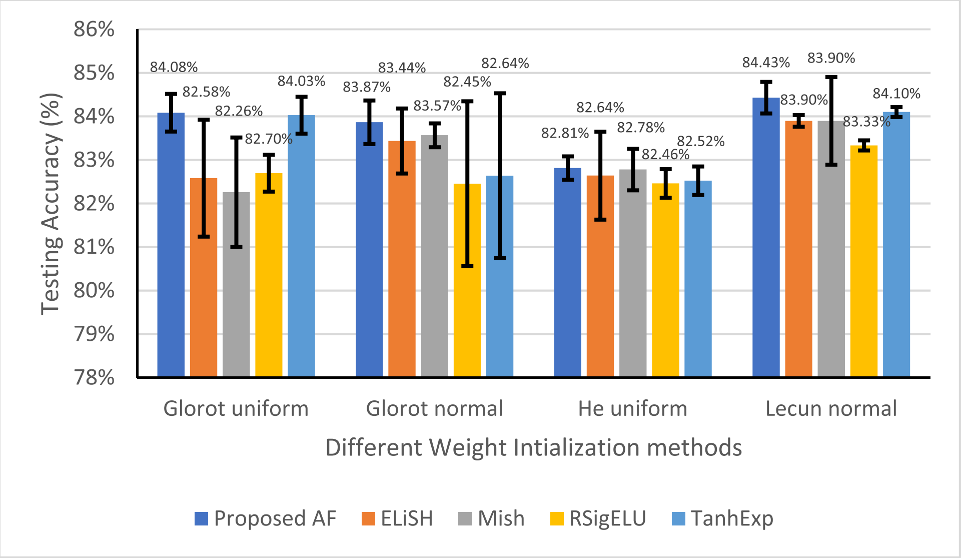


FIGURE 9. The effects of different weight initialization methods on average testing accuracy using InceptionNetV4 on the CIFAR10 dataset using HcLSH, RSigELU, Mish, ELiSH, and TanhExp activation functions. The error bars indicate the standard deviation.

RSigELU, Mish, and TanhExp performed well in CIFAR100 dataset. On the other hand, Swish's general performance is bad in FASHION MNIST, SVHN, and CIFAR10 datasets. PFLU and ReLU usually achieved relatively low results in the CIFAR100 dataset. On the contrary, GCU performed exceptionally poorly with specific deep architectures in all four datasets.

Regarding the effects with respect to the used classification model, ReLU, PenTanh, TanhExp, and GCU usually obtained better results in the LeNet model. At the same time, PFLU and Swish were poor performers. For MobileNetV1, SqueezeNet, and KerasNet models, HcLSH, GCU, RSigELU, ELiSH, and TanhExp achieved the highest results, while Swish and PFLU achieved the lowest. HcLSH, RSigELU, and Swish were the top performers in the ShuffleNetV2 model, while GCU was the poorest. Regarding InceptionNetV4 and ResNetV2-20 models, HcLSH, PenTanh, Swish, and RSigELU obtained the best results. In contrast, GCU fails in both models with a big gap, possibly due to its unsaturated oscillated nature that is easily trapped in local optima.

The proposed activation function (HcLSH) accomplished good performance in various datasets and models due to its characteristics and properties, discussed in Section III, introduced in the development of HcLSH to address the

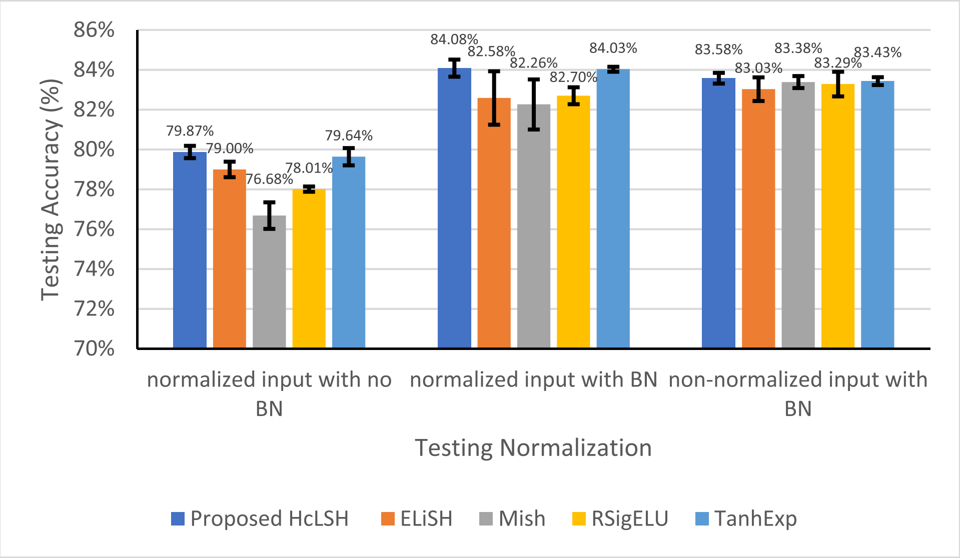


FIGURE 10. The effects of values normalization on average testing accuracy using InceptionNetV4 on the CIFAR10 dataset using HcLSH, RSigELU, Mish, ELiSH, and TanhExp activation functions. The error bars indicate the standard deviation.

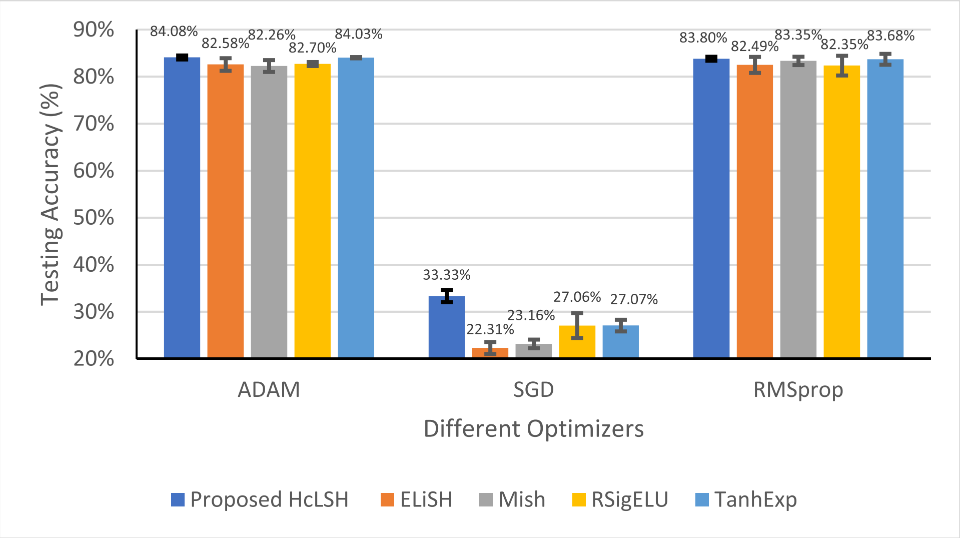


FIGURE 11. The effects of different optimizers on average testing accuracy using InceptionNetV4 on the CIFAR10 dataset using HcLSH, RSigELU, Mish, ELiSH, and TanhExp activation functions. The error bars indicate the standard deviation.

different issues and mimic the appealing advantages of the currently existing activation functions. The result of the proposed HcLSH seems less promising in the simple Fashion MNIST and SVHN datasets because they are considered easy to classify with low complexity, which enabled many models and activation functions to perform relatively well on them. Thus, the improvements achieved by HcLSH in these datasets were less impressive than that of CIFAR10 and CIFAR100. In general, the performance of the proposed activation function in the CIFAR100 dataset was the best. It can be concluded that HcLSH consistently outperformed the other ten activation functions in any investigated model on that dataset.

VI. LIMITATIONS OF HcLSH

Due to the existence of multiple mathematical functions in the definition of the proposed activation function, the learned complex nonlinearities might not be suited for small networks and datasets. Therefore, the correlation between the dataset complexity and the efficiency of the performance of HcLSH can be observed from the experiments conducted in Section IV-C, such that the more complex the dataset is and the heavier the deep architecture is, the more significant the improvement we can attain by using the proposed HcLSH activation function. This was proved by the achieved results in CIFAR 100 dataset and ResNet and InceptionNet models.

However, the proposed HcLSH's good performance comes at the cost of an increased computational overhead compared with ReLU, which introduced a tradeoff between the performance and the execution speed, which limits the applicability of the proposed HcLSH, in its current implementation, in light real-time applications.

VII. CONCLUSION

This paper proposed a new monotonic piecewise semi-saturated activation function called Hyperbolic cosine Linearized SquasHing function (HcLSH) for training deep neural networks. Its performance has been extensively evaluated using four image classification benchmark datasets: Fashion MNIST, SVHN, CIFAR10, and CIFAR100. Seven deep architectures with varying characteristics were selected as the experiments' testbeds. Two metrics were tracked and measured during performance evaluation: average test accuracy and average test loss. Ten popular state-of-the-art activation functions participated in the comparison experiments. HcLSH showed superior performance and noteworthy improvements and outperformed other activation functions with regard to different datasets and models, especially with more complex architectures and datasets. HcLSH achieved the Top-1 and Top-3 results in 20 and 25 conducted experiments, respectively. In addition, the results of the proposed HcLSH were statistically significant, as proved by the applied non-parametric tests. Furthermore, the ablation study emphasized the robustness and stability of HcLSH performance against possible hyperparameter choices.

Further enhancements will be conducted in a future work from four perspectives to provide an overall assessment of the proposed activation function.

1) Different tasks in different fields, such as language processing, object detection, or image segmentation.

2) Different deep architectures, such as AutoEncoders, EfficientNets, or generative-based models.

3) Different datasets, such as COCO [51] and ImageNet [52] datasets.

4) Propose an efficient hardware-optimized implementation of HcLSH to speed up the training time.

REFERENCES

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[2] F. Shao, L. Chen, J. Shao, W. Ji, S. Xiao, L. Ye, Y. Zhuang, and J. Xiao, "Deep learning for weakly-supervised object detection and localization: A survey," *Neurocomputing*, vol. 496, pp. 192–207, Jul. 2022, doi: 10.1016/j.neucom.2022.01.095.

[3] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022, doi: 10.1016/j.neucom.2022.01.005.

[4] S. R. Dubey, "A decade survey of content based image retrieval using deep learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2687–2704, May 2022, doi: 10.1109/TCSVT.2021.3080920.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[6] W. Duch and N. Jankowski, "Survey of neural transfer functions," *Neural Comput. Surveys*, vol. 2, no. 1, pp. 163–212, 1999.

[7] L. Datta, "A survey on activation functions and their relation with xavier and he normal initialization," 2020, *arXiv:2004.06632*.

[8] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Jun. 2011, pp. 315–323. [Online]. Available: https://proceedings.mlr.press/v15/glorot11a.html

[9] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.

[10] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," 2014, *arXiv:1412.6830*.

[11] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Netw.*, vol. 138, pp. 14–32, Jun. 2021, doi: 10.1016/j.neunet.2021.01.026.

[12] E. Chai, W. Yu, T. Cui, J. Ren, and S. Ding, "An efficient asymmetric nonlinear activation function for deep neural networks," *Symmetry*, vol. 14, no. 5, p. 1027, May 2022, doi: 10.3390/sym14051027.

[13] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022, doi: 10.1016/j.neucom.2022.06.111.

[14] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 9–48, doi: 10.1007/978-3-642-35289-8_3.

[15] X. Wang, Y. Qin, Y. Wang, S. Xiang, and H. Chen, "ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis," *Neurocomputing*, vol. 363, pp. 88–98, Oct. 2019, doi: 10.1016/j.neucom.2019.07.017.

[16] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3.

[17] B. Xu, R. Huang, and M. Li, "Revise saturated activation functions," 2016, *arXiv:1602.05980*.

[18] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[20] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.

[21] M. Basirat and P. M. Roth, "The quest for the golden activation function," 2018, *arXiv:1808.00783*.

[22] M. Zhu, W. Min, Q. Wang, S. Zou, and X. Chen, "PFLU and FPFLU: Two novel non-monotonic activation functions in convolutional neural networks," *Neurocomputing*, vol. 429, pp. 110–117, Mar. 2021, doi: 10.1016/j.neucom.2020.11.068.

[23] D. Misra, "Mish: A self regularized non-monotonic activation function," 2019, *arXiv:1908.08681*.

[24] X. Wang, H. Ren, and A. Wang, "Smish: A novel activation function for deep learning methods," *Electronics*, vol. 11, no. 4, p. 540, Feb. 2022, doi: 10.3390/electronics11040540.

[25] X. Liu and X. Di, "TanhExp: A smooth activation function with high convergence speed for lightweight neural networks," *IET Comput. Vis.*, vol. 15, no. 2, pp. 136–150, Mar. 2021, doi: 10.1049/cvi2.12020.

[26] S. Kiliçarslan and M. Celik, "RSigELU: A nonlinear activation function for deep neural networks," *Exp. Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114805, doi: 10.1016/j.eswa.2021.114805.

[27] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*.

[28] M. Mithra Noel, A. L, A. Trivedi, and P. Dutta, "Growing cosine unit: A novel oscillatory activation function that can speedup training and reduce parameters in convolutional neural networks," 2021, *arXiv:2108.12943*.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] Q. Duan, X. Li, Q. Yin, L. Feng, J. Zhao, Y. Teng, X. Duan, Y. Zhao, M. Gao, J. Wang, W. Cai, and R. Li, ''A study on the generalized normalization transformation activation function in deep learning based image compression,'' in *Proc. 6th Int. Congr. Inf. Commun. Technol.*, 2022, pp. 351–359, doi: 10.1007/978-981-16-2377-6_33.

[31] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, ''Visualizing the loss landscape of neural nets,'' in *Proc. Adv Neural Inf Process Syst*, vol. 31, 2018, pp. 1–11.

[32] H. Xiao, K. Rasul, and R. Vollgraf, ''Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,'' 2017, *arXiv:1708.07747*.

[33] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, ''Reading digits in natural images with unsupervised feature learning,'' in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, pp. 1–9.

[34] A. Krizhevsky and G. Hinton, ''Learning multiple layers of features from tiny images,'' Univ. Toronto, Toronto, ON, Canada, Tech. Rep. Learning-features-2009-TR, 2009.

[35] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*.

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, and G. Irving, ''TensorFlow: A system for large-scale machine learning,'' in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[37] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*.

[39] P. Velickovic, D. Wang, N. D. Lane, and P. Lio, ''X-CNN: Cross-modal convolutional neural networks for sparse datasets,'' in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8, doi: 10.1109/SSCI.2016.7849978.

[40] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, ''SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,'' 2016, *arXiv:1602.07360*.

[41] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, ''Inception-v4, inception-resnet and the impact of residual connections on learning,'' in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[42] K. He, X. Zhang, S. Ren, and J. Sun, ''Identity mappings in deep residual networks,'' in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645, doi: 10.1007/978-3-319-46493-0_38.

[43] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, ''ShuffleNet V2: Practical guidelines for efficient CNN architecture design,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[44] S. García, A. Fernández, J. Luengo, and F. Herrera, ''Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power,'' *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010, doi: 10.1016/j.ins.2009.12.010.

[45] X. Glorot and Y. Bengio, ''Understanding the difficulty of training deep feedforward neural networks,'' in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, vol. 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[46] K. He, X. Zhang, S. Ren, and J. Sun, ''Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[47] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, ''Self-normalizing neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/5d44ee6f2c3f71b73125876103c8f6c4-Paper.pdf

[48] B. Xu, N. Wang, T. Chen, and M. Li, ''Empirical evaluation of rectified activations in convolutional network,'' 2015, *arXiv:1505.00853*.

[49] G. Bingham and R. Miikkulainen, ''Discovering parametric activation functions,'' *Neural Netw.*, vol. 148, pp. 48–65, Apr. 2022, doi: 10.1016/j.neunet.2022.01.001.

[50] J. Park, M. J. Kim, W. Jung, and J. H. Ahn, ''AESPA: Accuracy preserving low-degree polynomial activation for fast private inference,'' 2022, *arXiv:1505.00853*.

[51] T.-Y. Lin, ''Microsoft COCO: Common objects in context,'' in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, 2014, pp. 740–755.

[52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ''ImageNet: A large-scale hierarchical image database,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, doi: 10.1109/CVPR.2009.5206848.

HEBA ABDEL-NABI received the bachelor's degree in computer engineering and the master's degree in electrical engineering from Princess Sumaya University for Technology (PSUT), in 2010 and 2015, respectively, where she is currently pursuing the Ph.D. degree in computer science. Her research interests include digital image processing and information security, deep learning, artificial intelligence, and evolutionary algorithms.

GHAZI AL-NAYMAT received the Ph.D. degree from the School of Information Technologies, The University of Sydney, Australia, in May 2009. He is currently an Associate Professor and the Data Analytics Program Coordinator with the College of Engineering and Information Technology, Ajman University, United Arab Emirates. He has wide-ranging academic, industrial, and organizational experience. He has taught many computer science, information systems, and data science courses at all levels (doctoral, master's, and bachelor's) in more than six universities worldwide. His research interests include data mining and machine learning, big data, and data science.

MOSTAFA Z. ALI (Senior Member, IEEE) received the bachelor's degree in applied mathematics from the Jordan University of Science & Technology (JUST), Irbid, Jordan, in 2000, the master's degree in computer science from the University of Michigan-Dearborn, Michigan, USA, in 2003, and the Ph.D. degree in computer science/artificial intelligence from Wayne State University, Michigan, USA, in 2008. He is currently a Professor with the Department of Computer Information Systems, JUST. His research interests include artificial intelligence applications, evolutionary computation, machine learning, deep learning, virtual/augmented reality, and gaming. He is a member of the IEEE Computer Society, the American Association of Artificial Intelligence (AAAI), and the ACM. He is also an Associate Editor of the *Swarm and Evolutionary Computation* (SWEVO) (Elsevier) and *Information Sciences* (INS) (Elsevier).

ARAFAT AWAJAN received the Ph.D. degree in computer science from the University of Franche-Comte, France, in 1987. He is currently a full professor in computer science. He held different academic positions with the Royal Scientific Society and Princess Sumaya University for Technology and Mutah University. He is also the President of Mutah University, Jordan. His research interests include natural language processing, Arabic text mining, and digital image processing.