Asia Pacific Academy of Science Pte. Ltd.

Mathematics and Systems Science 2024, 2(2), 2870.
https://doi.org/10.54517/mss.v2i2.2870

Article

Modified scaled exponential linear unit

Nimra[1], Jamshaid Ul Rahman[1,2,*], Dianchen Lu[2]

[1] Abdus Salam School of Mathematical Sciences, Govt College University, Lahore 54600, Pakistan
[2] School of Mathematical Sciences, Jiangsu University, Zhenjiang 212013, China
*** Corresponding author:** Jamshaid Ul Rahman, jamshaid@sms.edu.pk

CITATION

Nimra, Rahman JU, Lu D. Modified scaled exponential linear unit. Mathematics and Systems Science. 2024; 2(2): 2870. https://doi.org/10.54517/mss.v2i2.2870

ARTICLE INFO

Received: 5 August 2024
Accepted: 24 September 2024
Available online: 8 October 2024

Abstract: Activation functions assume a crucial role in elucidating the intricacies of training dynamics and the overall performance of neural networks. Despite its simplicity and effectiveness, the ubiquitously embraced ReLU activation function harbors certain drawbacks, notably the predicament recognized as the "Dying ReLU" issue. To address such challenges, we propose the introduction of a pioneering activation function, the modified scaled exponential linear unit (M-SELU). Drawing from an array of experiments conducted across diverse computer vision tasks employing cutting-edge architectures, it becomes apparent that M-SELU exhibits superior performance compared to ReLU (used as the baseline) and various other activation functions. The simplicity of the proposed activation function (M-SELU) makes this solution particularly suitable for multi-layered deep neural architecture, including applications in CNN, CIFAR-10, and the broader field of deep learning.

Keywords: activation functions; CNN; CIFAR-10; deep learning; modified scaled exponential linear unit (M-SELU)

1. Introduction

Deep learning [1], which falls under the umbrella of machine learning [2], has proven to be highly successful across diverse applications [3]. These tasks encompass recognizing images and speech, handling natural language processing [4], conducting medical diagnoses [5], and engaging in strategic game playing [6]. The power of deep neural networks [7], a fundamental element of deep learning, lies in their ability to independently grasp intricate structures and patterns from data. This capability has established deep learning as a formidable tool within the realm of artificial intelligence. Key components in deep learning include neural networks, activation functions [8], back propagation, forward propagation, convolution neural networks (CNNs) [9–12] recurrent neural networks (RNNs) [13,14].

In the sphere of neural networks and deep learning, an activation function constitutes a mathematical operation systematically imposed upon the output of each individual neuron within a given layer. The infusion of non-linearity [15–17] serves the pivotal purpose of endowing the network with the capacity to assimilate and approximate intricate mappings derived from inputs to outputs. At its core, the operational sequence involves neurons receiving a multitude of inputs, undergoing a process of weighted summation [18], and subsequently subjecting this aggregate to an activation function to yield the neuron's output. The crux of the activation function [19,20] resides in its discerning role—deciding whether a neuron merits activation, signifying the consequentiality of its output for subsequent layers, or whether it should remain in a quiescent state [21–23].

In the context of linear models within neural networks, the process involves mapping input functions to output through an affine transformation in the hidden layers, typically before making final predictions for class scores. The neural networks generate linear results from this mapping, and the need for an activation function arises. Activation functions are crucial to convert these linear outputs into non-linear ones, facilitating the learning of intricate patterns in the data. The non-linear output, following the application of the activation function, is expressed as:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

here, σ in Equation (1) represents the activation function used w_i are the weights, and x_i are the input features. Thus, the activation function emerges as an indispensable constituent [24], conferring upon the neural network the agility to navigate and comprehend intricate patterns and relationships inherent in the data. This renders the network proficient in addressing multifaceted tasks such as image recognition [25,26], language comprehension [27], or predictive analysis based on intricate datasets. In summary, the selection of an appropriate activation function emerges as a decisive factor [28], wielding substantial influence over the neural network's performance across a spectrum of tasks [29,30].

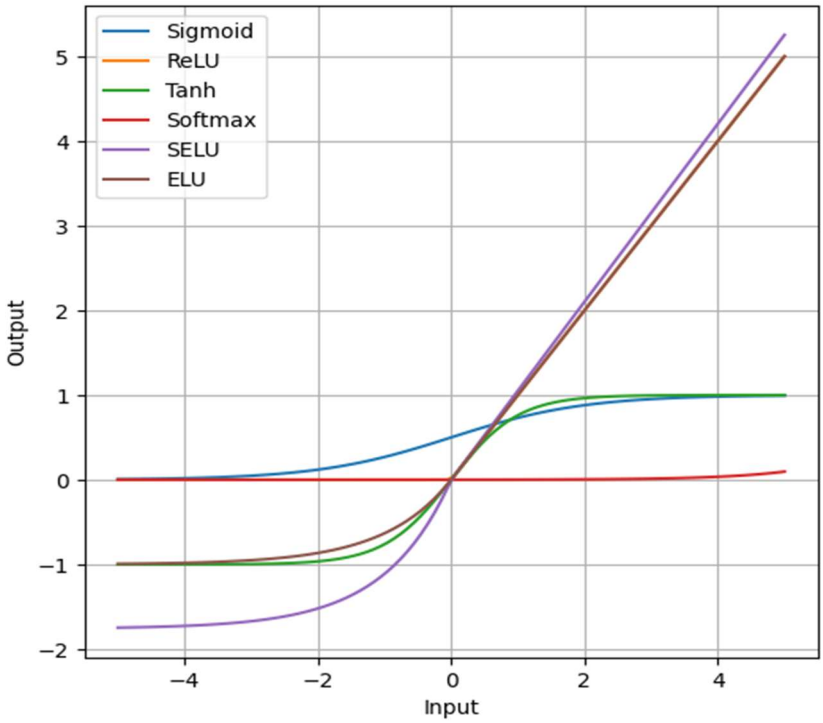


Figure 1. Graph of different activation functions.

Although the pre-existing activation functions represented in **Figure 1** play a crucial role in improving the accuracy of computer vision tasks [31–34], there were certain challenges and limitations with them as well. The vanishing gradient problem [35–37] is closely related to the choice of activation functions in neural networks. This problem arises during the training process when the gradients of the loss function with

respect to the parameters (weights) become very small, approaching zero [38,39]. As a result, the network has difficulty learning, and the weights may not be updated effectively during backpropagation. The choice of activation functions plays a significant role [40] in the occurrence of the vanishing gradient problem [41,42]. ReLU [43] does not suffer from the vanishing gradient problem to the same extent as sigmoid or Tanh. However, ReLU has its own challenges, such as the "Dying ReLU" problem [43,44] where neurons can become inactive during training. Sigmoid and hyperbolic tangent activation functions compress input values within specific ranges (0 to 1 for sigmoid and −1 to 1 for hyperbolic tangent) [45]. When dealing with notably positive or negative inputs, the gradients of these functions tend to diminish, approaching zero. In the case of ReLU, when the input to a neuron becomes negative, the output is zero, and the neuron's weights may no longer receive updates during backpropagation if the gradient is consistently zero. This can happen in scenarios where a large gradient flows through a ReLU unit, causing the weights to be updated in such a way that the neuron always produces a negative output [46]. In such cases, M-SELU is very helpful and stable to use.

The Modified Scaled Exponential Linear Unit, or M-SELU, is a function that was created to overcome the drawbacks of more conventional activation functions such as ReLU, particularly with respect to negative input values. For negative inputs in ReLU, the gradient during backpropagation becomes 0, halting weight updates and resulting in neuronal death. This problem is solved by M-SELU. M-SELU produces tiny, non-zero values in contrast to ReLU, which outputs zero for negative inputs. This guarantees that neurons stay active and keep updating their weights throughout backpropagation. This prevents neurons from being dormant, which is a significant flaw in ReLU.

In addition, M-SELU modifies both positive and negative inputs to keep the gradient flow in the network constant. Particularly in deep networks, this enhanced stability greatly lowers the probability of vanishing gradients. The non-linear transformations of M-SELU assist in maintaining the dynamic range of activations in situations when ReLU frequently fails, leading to more efficient training and quicker convergence.

M-SELU provides superior performance and versatility over SELU. Although SELU works well for vanishing gradients, M-SELU adds a scaling parameter that allows for more precise control over the handling of negative inputs, improving adaptability and hastening convergence in intricate designs. Because of this, M-SELU is a more useful activation function in some deep learning applications.

In order to have a better understanding of the behavior of the M-SELU activation function, we examine the feature maps produced by various beta (β) values in this work. Previous research has demonstrated that feature map visualization, especially when utilizing modified activation functions, can offer important insights into how neural networks represent and process information. For example, Zeiler and Fergus [47] showed that one can gain a better grasp of the hierarchical structure of learned features in convolutional neural networks by viewing feature maps. Furthermore, the influence of parameters like β values on feature extraction and network performance has been investigated in studies such as Springenberg et al. [48]. Specifically with negative inputs, the β values in M-SELU refine the behavior of the activation function,

resulting in discernible modifications in feature maps [49]. The goal of this research is to determine if changing β values improves network performance and feature map variety.

2. The modified scaled exponential linear unit

Our Proposed Activation function which is named as Modified Scaled Exponential Linear Unit is defined as:

$$f(x) = \begin{cases} \alpha\left(e^{\beta x} - 1\right) & x < 0 \\ \alpha x & x \geq 0 \end{cases} \tag{2}$$

With two parameters α and β where $\alpha = 1$ and $\beta = 0.25, 0.5, 0.75, 1$ etc.

Activation functions in deep learning architectures are expected to have features such as being non-linear, reaching the global optimum without being stuck in the local optimum. Clearly, the proposed activation function satisfies these properties. The visual depiction of the M-SELU can be observed in **Figure 2**.

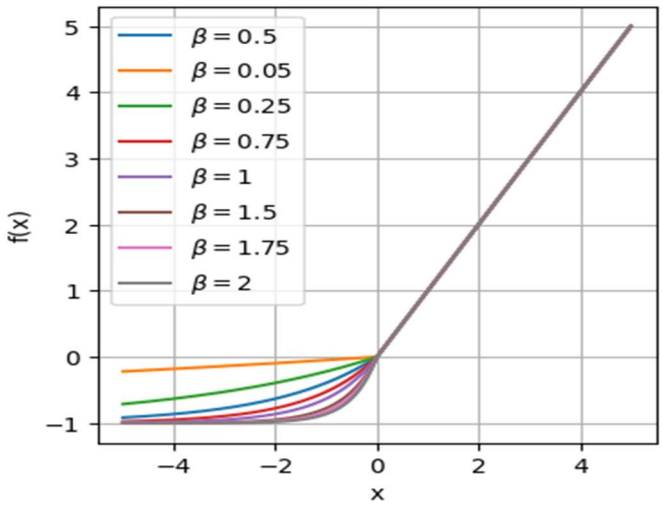


Figure 2. Graph of piecewise function for different values of β.

3. Delving into the architectural complexity of CNNs

CNNs stand out as a specialized category within deep neural networks crafted to interpret and process visual information, including images and videos. They wield remarkable prowess in endeavors such as recognizing images, detecting objects, and classifying images. This makes CNNs especially well-suited for navigating the intricate details of visual data, unraveling patterns, and making sense of the content within images or video frames.

The CIFAR-10 dataset serves as a widely recognized benchmark in the realm of computer vision. Its name is derived from the Canadian Institute for Advanced Research (CIFAR), the organization responsible for its curation. Comprising 60,000 color images, each measuring 32 × 32 pixels, the dataset encompasses a diverse array of visuals distributed across 10 distinct classes. Within each category, there are 6000 images, creating a comprehensive and balanced collection. This dataset is extensively

used to evaluate and compare the performance of various machine learning models, particularly in the domain of image classification.

The architecture is delineated as follows:

To extract and improve features from input photos, we use a sequence of convolutional layers in our Convolutional Neural Network (CNN) architecture, followed by max pooling layers.

Conv2d_1 uses a stride of (1, 1) and a filter size of (3, 3) to collect mid-level characteristics like limb structures and fur texture. This layer serves as the basis for feature extraction by emphasizing the important structures and patterns present in the image. Then, with the same filter size of (3, 3) and stride of (1, 1), the conv2d_2, conv2d_3, and conv2d_4 layers go deeper into the picture. These layers are intended to capture more minute details and complex patterns, which are essential for identifying small differences such as variances in facial features, stance, and fur patterns.

Max pooling layers are positioned after each convolutional layer to improve feature extraction by down sampling feature maps and highlighting important features while lowering spatial dimensions. Accuracy and feature recognition are enhanced as a result.

In the end, the network consists of dense layers that include the features that have been learned from earlier levels. This allows the network to make precise predictions, like categorizing dog photos according to intricate patterns. With the help of this architectural strategy, the CNN can efficiently examine intricate patterns in photos, producing precise classifications and high-performance forecasts.

4. Experimental configuration

This study focuses on the CIFAR-10 dataset, characterized by precise input dimensions of 32 × 32 × 3. The key investigation revolves around the intentional adoption of the M-SELU activation function as the primary activation paradigm. The architectural framework employed is comprehensive, consisting of a total of 10 layers that incorporate convolutional, max-pooling, flattening, and dense layers. Within the dense layers, soft max is elegantly chosen as the activation function. Optimization is executed through the adaptive moment estimation (Adam) algorithm [50], with stride values meticulously configured at a rate of 1 × 1. The selected loss function for this context is Sparse Categorical Cross-Entropy. To maintain uniformity throughout the tests, we employed a batch size of 32 for all training sessions in our experimental setup. Using the Adam optimizer, which dynamically adjusts to the dataset and offers consistent training results, the learning rate was set at 0.001. We used the Glorot Uniform initialization method for weights, which helps maintain gradient variation across network layers and facilitates effective training. Every experiment was run once to compare performance, using various activation functions. Even though the outcomes shown here are for a single run, conducting many runs can improve statistical reliability.

The algorithm's training regimen spans a carefully chosen interval of 20 epochs. Experimental results are thoughtfully presented through a dual approach, featuring both tabular formats and graphical visualizations.

5. Discussion and results

We conducted an experiment utilizing the CIFAR-10 dataset. The focal point of our study was the activation function we introduced, known as M-SELU. In the forthcoming discussion, we will dig into the details of the experiment and analyze the results. The configuration contributed to the refinement of the model's parameters and its overall performance during the training phase. It is worth emphasizing that the network consistently demonstrated improvements in accuracy during the entire training process, and this enhancement can be attributed to the utilization of the proposed activation function, M-SELU.

5.1. Feature map visualization

In this implementation, the initial stages of the network employ a max pooling operation with a size of (2, 2) and a stride of 2 for the first two convolutional layers. The purpose of this approach is to conduct feature extraction in a hierarchical manner within the convolutional layers of the network. The next figures illustrate the resulting feature maps from various layers within the constructed CNN architecture.

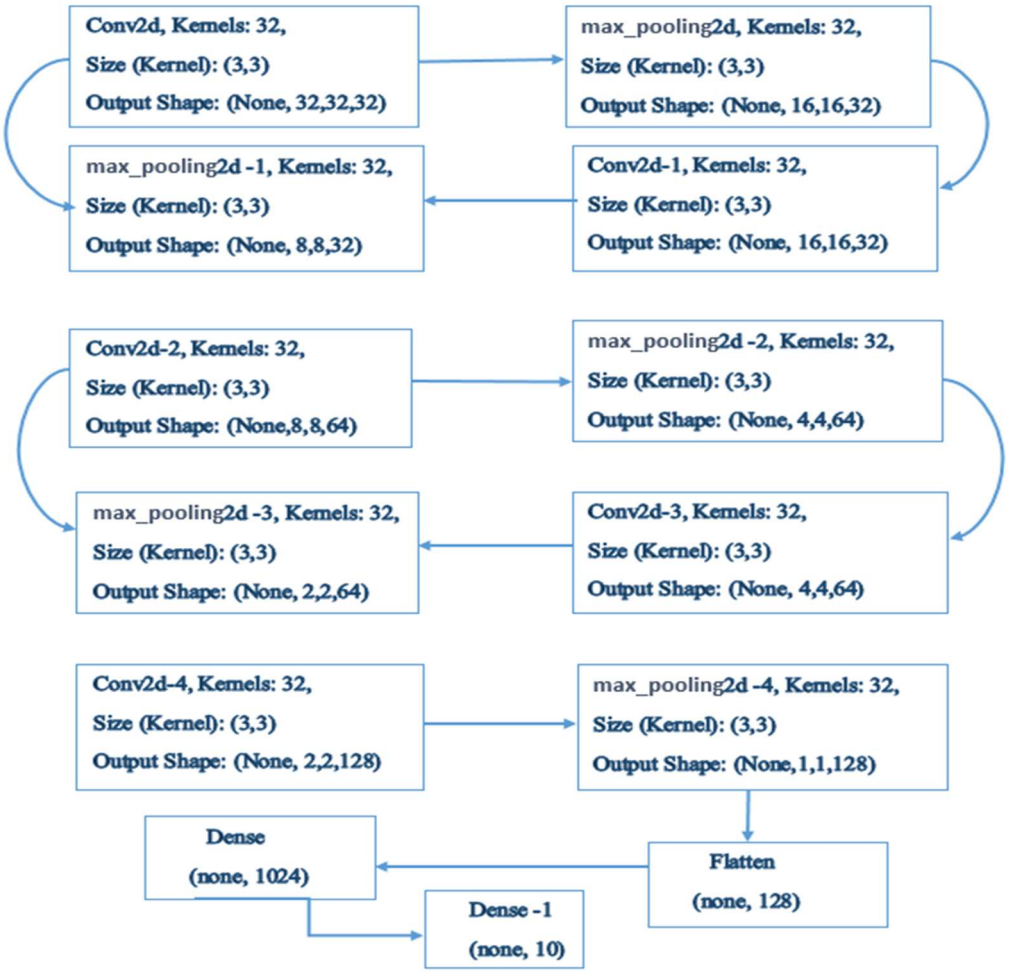


Figure 3. Flowchart diagram on CNN architecture.

Hence, the above **Figure 3** shows that creating a convolutional neural network

(CNN) experimentally involves defining the layer names, specifying the number of filters in each convolutional layer, and determining the output dimensions of the feature maps produced by each convolutional layer. **Figure 4** shows that in the neural network's initial convolutional layer, conv2d, 32 filters with a specified size and stride detect fundamental dog features, emphasizing the overall shape, body contour, and tail structure. Leveraging M-SELU as the activation function, this layer captures essential visual cues indicative of basic dog features. In **Figure 5**, a consistent behavior is observed across layers for β = 0.5, reminiscent of the described neural network architecture. The initial convolutional layer focuses on foundational dog features, with subsequent layers discerning increasingly complex patterns. Max pooling layers strategically placed contribute to a hierarchical representation, akin to the designed network structure. From **Figures 6** and **7**, we can see a similar pattern to the described neural network setup, especially when β is set to 0.75 and 1.0, respectively. The layers consistently pick up different levels of details in the images, following a comparable structure to the network's design.

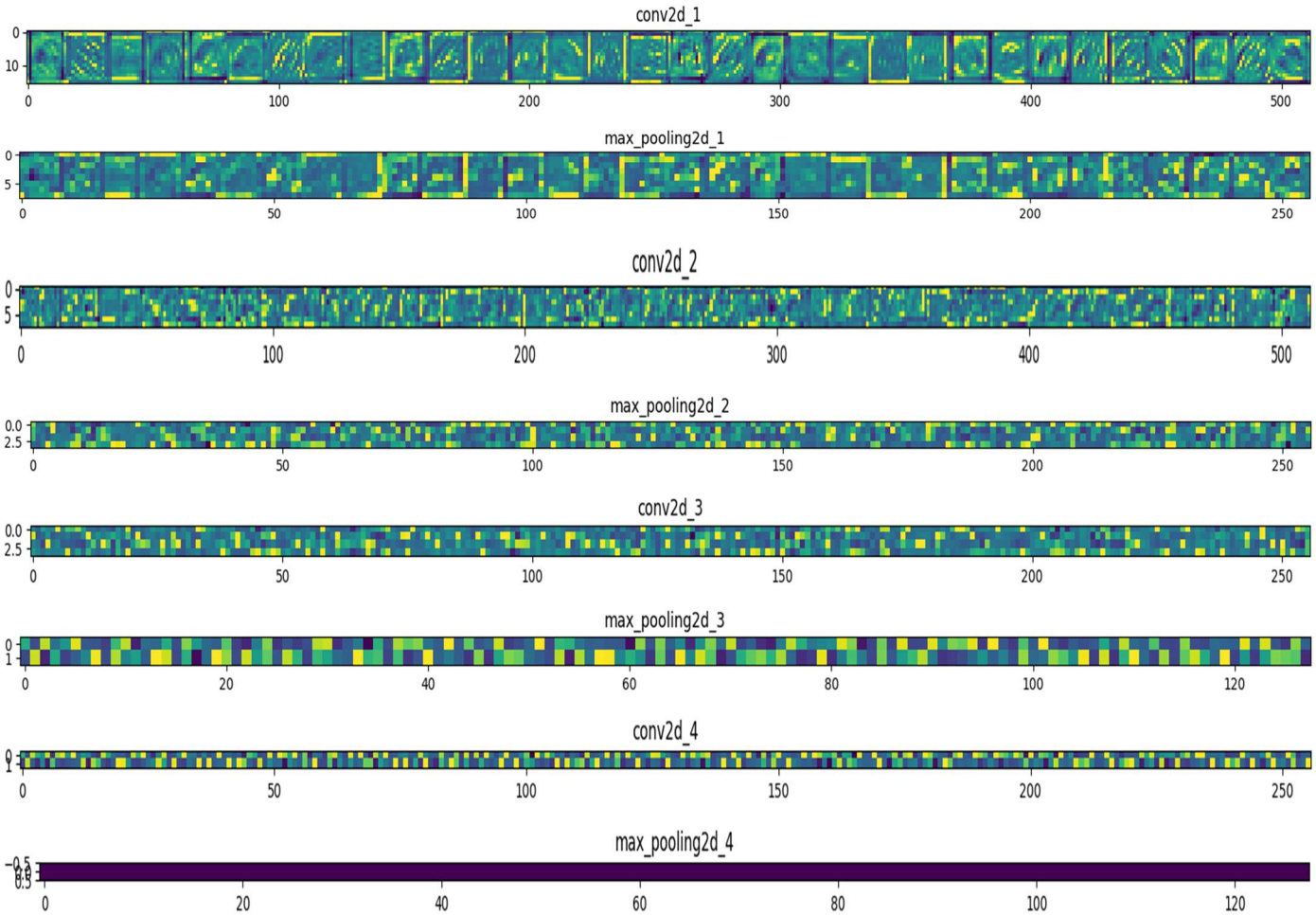


Figure 4. When a dog image is input into the visualization model, the subsequent figures depict the visualization of feature maps from diverse layers of the CNN (β = 0.25).

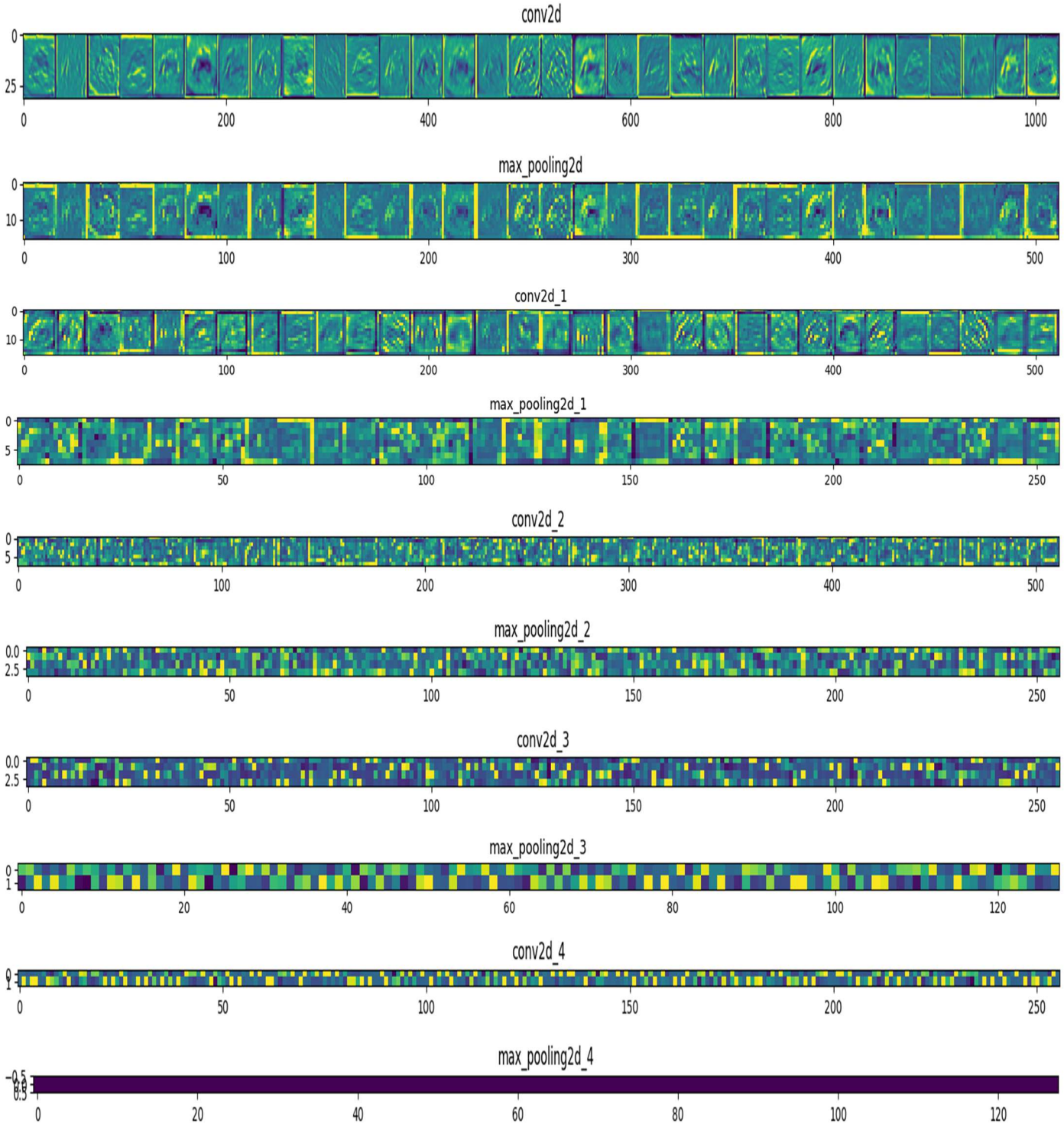


Figure 5. When a dog image is input into the visualization model, the subsequent figures depict the visualization of feature maps from diverse layers of the CNN ($\beta = 0.5$).

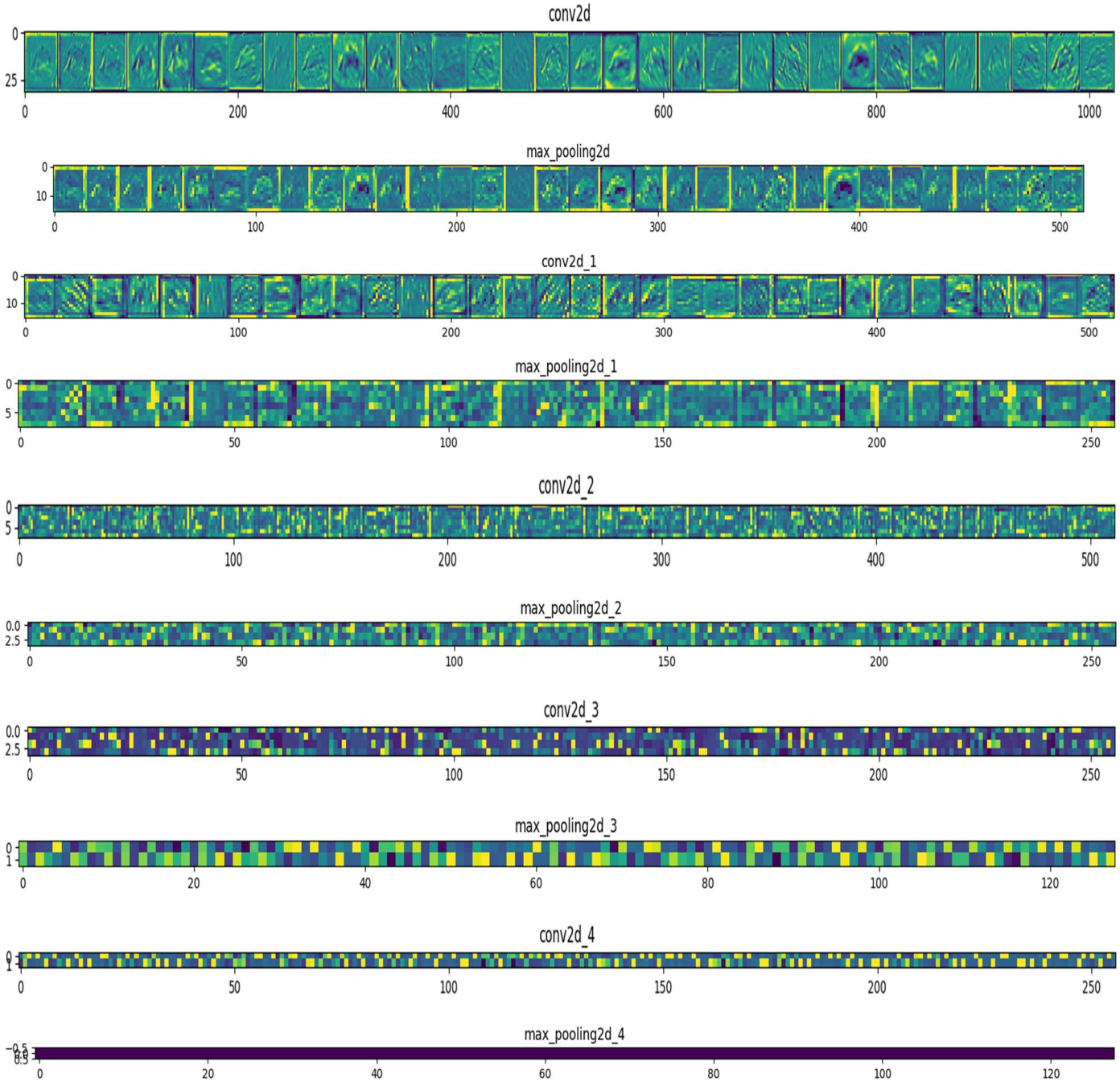


Figure 6. When a dog image is input into the visualization model, the subsequent figures depict the visualization of feature maps from diverse layers of the CNN ($\beta = 0.75$).

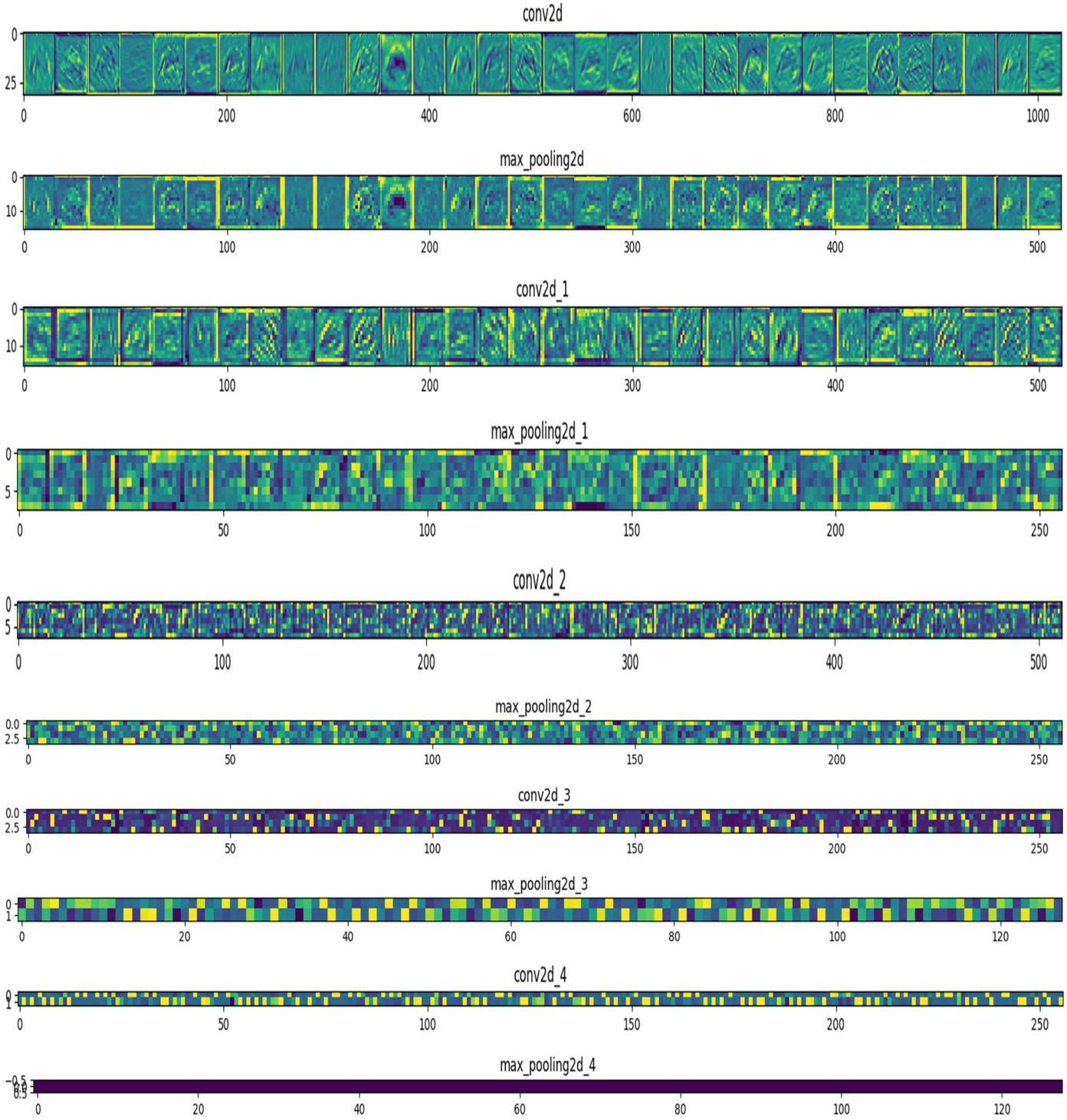


Figure 7. When a dog image is input into the visualization model, the subsequent figures depict the visualization of feature maps from diverse layers of the CNN ($\beta = 1.0$).

Now, we will delve into the discussion of results obtained with varying values of β.

The tables presented above offer a comprehensive overview of the results achieved with different values of β. This organized presentation facilitates a clear and

concise analysis, allowing us to discern the impact of varying β values on the observed outcomes. We will now proceed to illustrate the findings graphically, providing a visual representation of the trends observed with different β values. These graphs will offer a more intuitive understanding of the model's performance dynamics.

The tables unequivocally demonstrate that the accuracy of M-SELU is markedly superior to that of SELU, showcasing not only its robust performance but also establishing it as a standout choice among the mentioned activation functions.

Table 1 and **Figure 8a** reveal that, with a β value of 0.25, the accuracy of the M-SELU activation function reaches 94.25%, surpassing SELU, ReLU, ELU, and Tanh. Furthermore, **Figure 4** presents the visualization of feature maps. Likewise, **Table 2** and **Figure 9a** demonstrate that when utilizing a β value of 0.5, the M-SELU activation function achieves an accuracy of 94.31%, outperforming SELU, ReLU, ELU, and Tanh. Additionally, **Figure 5** showcases the visualization of feature maps. In a similar fashion, as depicted in **Table 3** and **Figure 10a**, the M-SELU activation function achieves an accuracy of 93.74% when employing a β value of 0.75, outperforming SELU, ReLU, ELU, and Tanh. Additionally, **Figure 6** provides a visual representation of the feature maps. As indicated by the data in **Table 4** and the visual depiction in **Figure 11a**, the M-SELU activation function achieves an accuracy of 93.54% when employing a β value of 1.0. This performance surpasses that of other activation functions. Additionally, **Figure 7** provides an insight into the visual representation of feature maps. This notable superiority underscores the effectiveness of M-SELU in the context of the experiment, highlighting its potential as a preferred activation function for similar applications. Among the various β values considered, the highest accuracy was achieved when utilizing a β value of 0.5, and the training loss in this case is 0.1750, as it can be seen in **Figure 9b**.

Table 1. It illustrates the training and validation accuracy of M-SELU and alternative activation functions, specifically when β is configured to 0.25 within the M-SELU framework.

Activation Function	Training Accuracy (%)	Validation Accuracy (%)
M-SELU	94.31	72.54
Tanh	91.66	70.15
SELU	89.14	68.46
ELU	93.32	69.43
RELU	90.03	71.80

Table 2. It illustrates the training and validation accuracy of M-SELU and alternative activation functions, specifically when β is configured to 0.5 within the M-SELU framework.

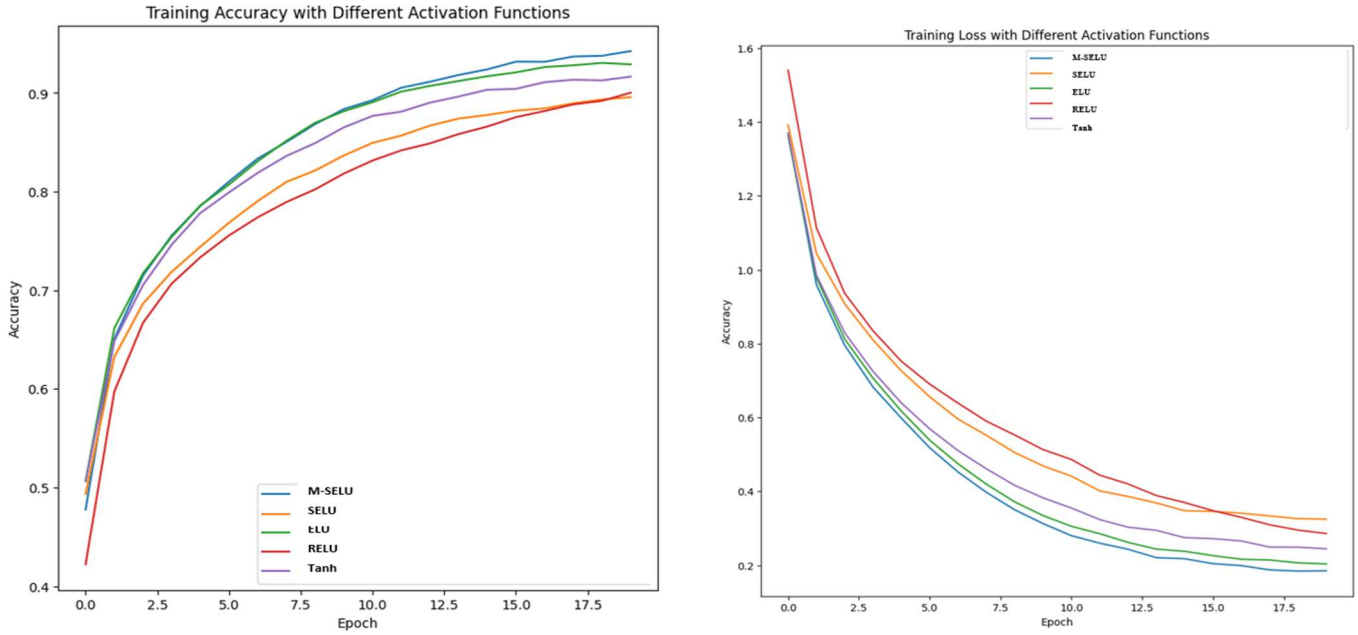
Activation Function	Training Accuracy (%)	Validation Accuracy (%)
M-SELU	94.25	71.79
Tanh	91.66	70.15
SELU	89.58	67.47
ELU	92.91	69.73
RELU	90.85	71.53

Table 3. It illustrates the training and validation accuracy of M-SELU and alternative activation functions, specifically when β is configured to 0.75 within the M-SELU framework.

Activation Function	Training Accuracy (%)	Validation Accuracy (%)
M-SELU	93.74	70.17
Tanh	91.30	69.42
SELU	89.50	68.84
ELU	93.52	70.16
RELU	90.24	71.42

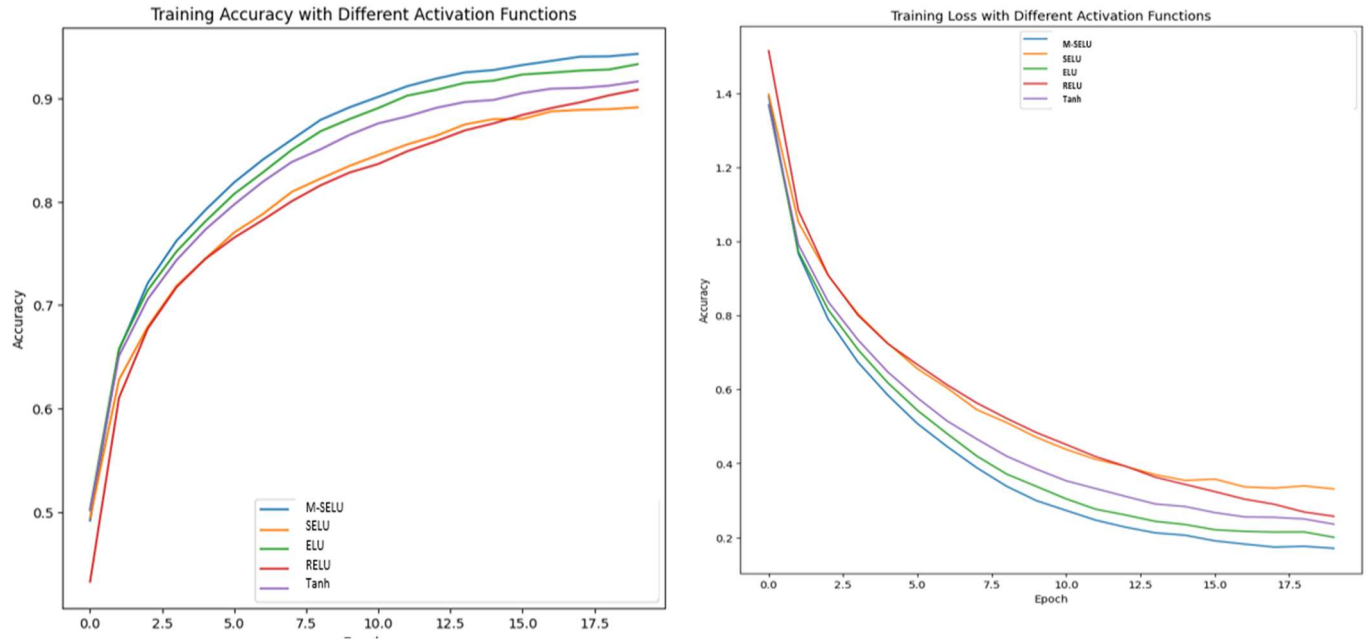
Table 4. It illustrates the training and validation accuracy of M-SELU and alternative activation functions, specifically when β is configured to 1.0 within the M-SELU framework.

Activation Function	Training Accuracy (%)	Validation Accuracy (%)
M-SELU	93.54	70.17
Tanh	91.29	69.42
SELU	89.21	68.84
ELU	93.52	70.16
RELU	90.24	71.42



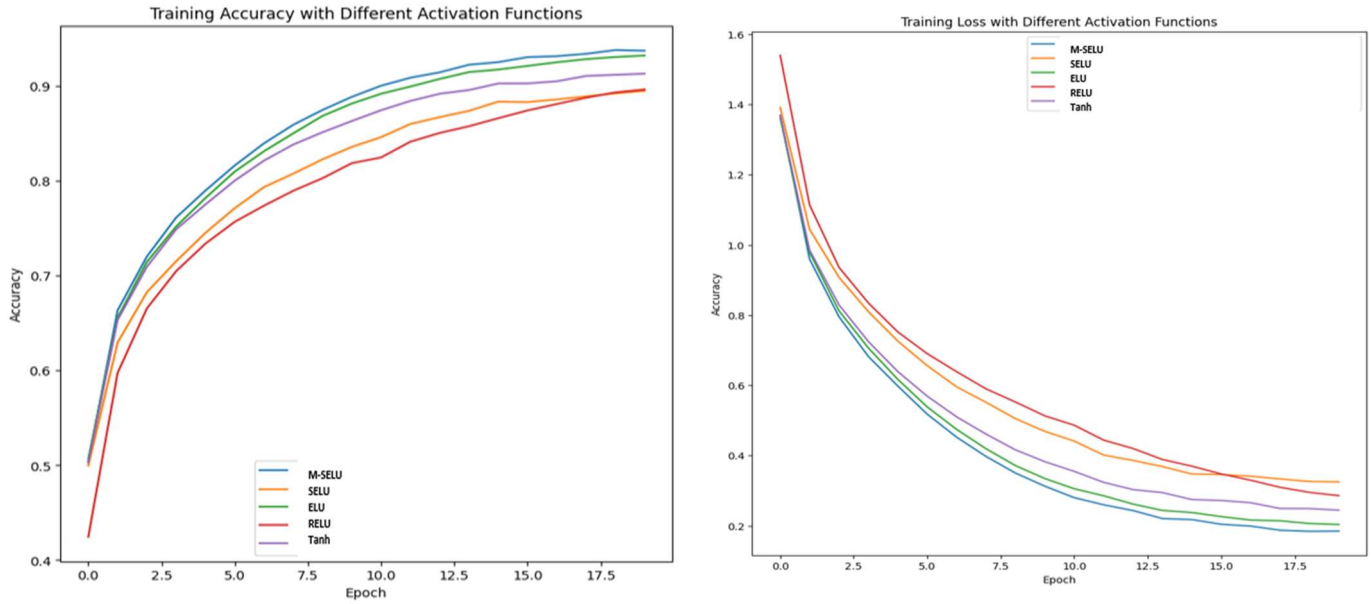
(a) **(b)**

Figure 8. Training accuracy and loss at β is 0.25 **(a)** Depicting the comparison of training accuracy when the value of β is 0.25; **(b)** depicting the comparison of training loss when the value assigned to β is 0.25.



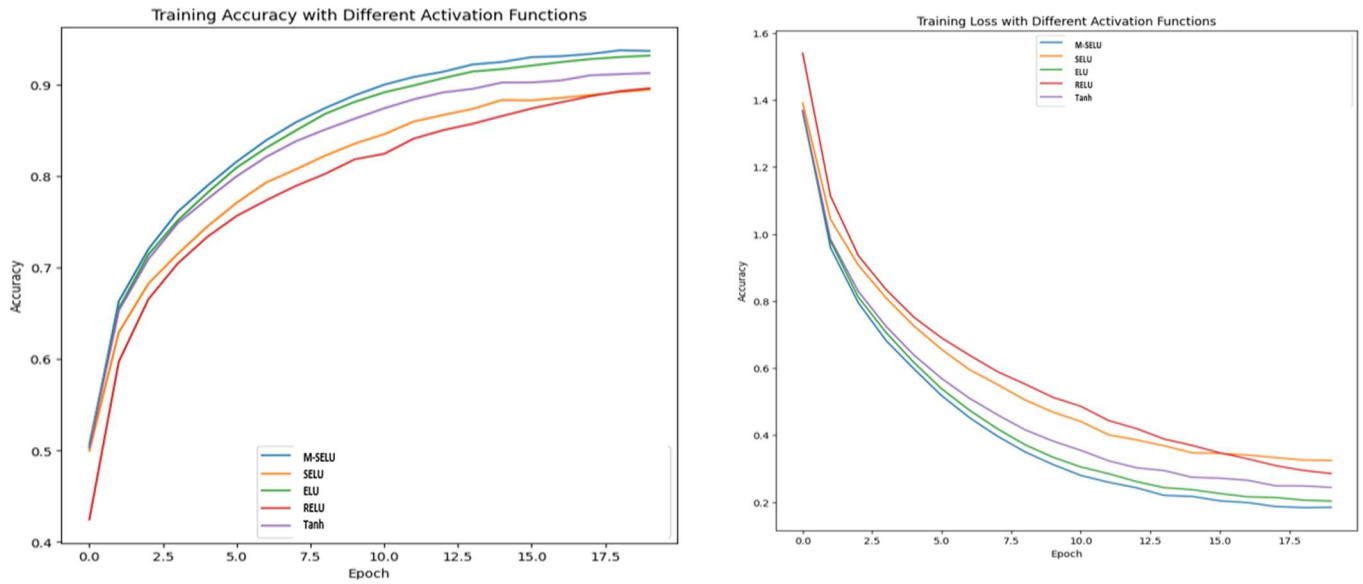
(a) **(b)**

Figure 9. Training accuracy and loss at β is 0.5 **(a)** Depicting the comparison of training accuracy when the value of β is 0.5; **(b)** depicting the comparison of training loss when the value assigned to β is 0.5.



(a) **(b)**

Figure 10. Training accuracy and loss at β is 0.75 (**a)** Depicting the comparison of training accuracy when the value of β is 0.75; **(b)** depicting the comparison of training loss when the value assigned to β is 0.75.



(a) **(b)**

Figure 11. Training accuracy and loss at β is 1 **(a)** Depicting the comparison of training accuracy when the value of β is 1; (b) depicting the comparison of training loss when the value assigned to β is 1.

The visual representation below supplements the outcomes elucidated in the previously mentioned tables.

Figure 12 illustrates the comprehensive performance of M-SELU using a graphical representation. Moreover, the presented graph in **Figure 13** illustrates the relationship between training loss and β values for the M-SELU activation function. The x-axis represents different β values, while the y-axis depicts the corresponding training loss values.

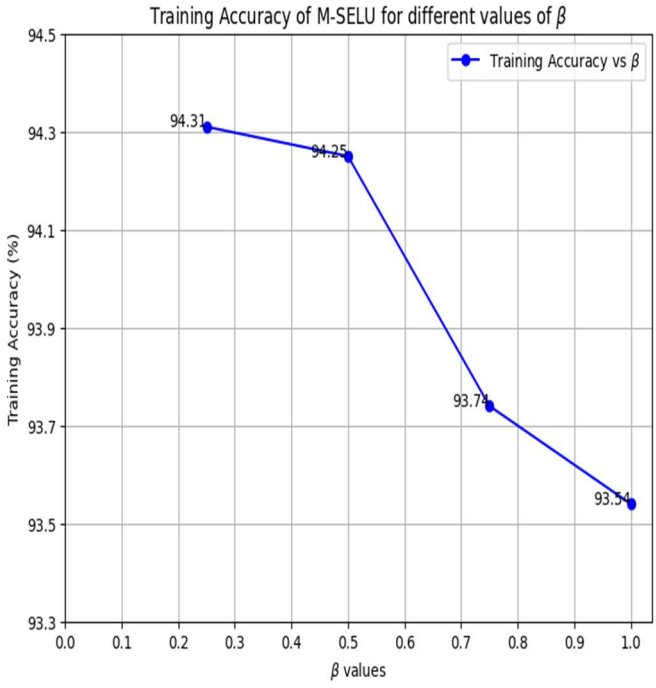


Figure 12. Composite performance of M-SELU through a graphical representation.

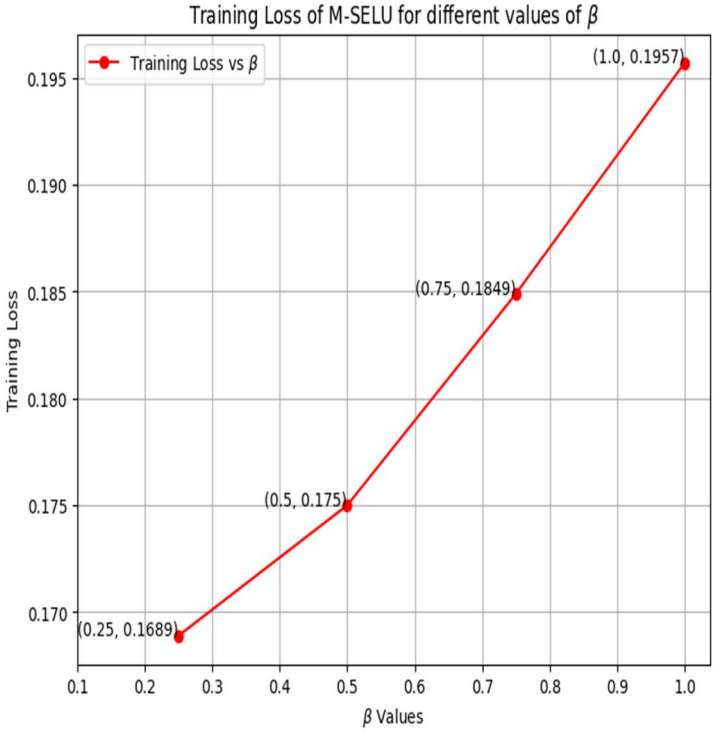


Figure 13. Illustration of training loss of M-SELU for different values of β.

5.2. Conclusion

In this paper, we introduce a novel activation function termed M-SELU, formally presented in Section 2, Equation (2). The focus of our study centers on evaluating the efficacy of M-SELU in image classification tasks. Utilizing the CIFAR-10 dataset, our CNN incorporating M-SELU across its layers demonstrates promising results when compared to SELU, ReLU, and other activation functions. Notably, with various values of β, our approach achieves state-of-the-art performance, particularly excelling with $\beta = 0.5$. The training accuracy exhibits a significant improvement of 3.46% over ReLU and 4.9% over SELU. Furthermore, validation accuracy also experiences enhancements. Additionally, the introduced M-SELU activation function addresses the common issue of the "dying ReLU" problem, further enhancing its potential impact on model performance in image classification tasks. Future research efforts might look at the implementation of this new activation function in more complex models, which could potentially produce novel state-of-the-art outcomes spanning various datasets. To assess the activation function's performance on a broader variety of tasks and datasets, such as various image recognition challenges, tasks involving natural language processing, and other domains in which activation functions are crucial, it could be specifically applied to more intricate neural network architectures. Furthermore, this work can be effectively applied to other areas, such as [50].

Author contributions: Conceptualization, N and JUR; methodology, N; software, JUR; validation, N, JUR and DL; formal analysis, N; investigation, N and JUR; data curation, N; writing—original draft preparation, JUR; writing—review and editing, JUR; visualization, N and JUR; supervision, JUR; project administration, DL; funding

acquisition, DL. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References

1. Janiesch C, Zschech P, Heinrich K. Machine learning and deep learning. Cornell University; 2021.
2. Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. ScienceMag.org; 2015.
3. Langley P, Herbert A, Simon. Applications of Machine Learning and Rule Induction. Advances in Traditional AI. 1995.
4. Deng L, Liu Y, eds. Deep Learning in Natural Language Processing. Springer Singapore; 2018. doi: 10.1007/978-981-10-5209-5
5. Kononenko I. Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in Medicine. 2001; 23(1): 89-109.
6. Wang J, Hong Y, Wang J, et al. Cooperative and Competitive Multi-Agent Systems: From Optimization to Games. IEEE/CAA Journal of Automatica Sinica. 2022; 9(5): 763-783. doi: 10.1109/jas.2022.105506
7. Rahman JU, Danish S, Lu D. Oscillator Simulation with Deep Neural Networks. Mathematics. 2024; 12(7): 959. doi: 10.3390/math12070959
8. Sharma S, Sharma S, Athaiya A. Activation functions in neural networks. International Journal of Engineering Applied Sciences and Technology. 2020; 04(12): 310-316. doi: 10.33564/ijeast.2020.v04i12.054
9. Abien Fred M, Agarap. Deep Learning using Rectified Linear Units (ReLU). Cornell University; 2019.
10. Alzubaidi L, Zhang J, Humaidi AJ, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data. 2021; 8(1). doi: 10.1186/s40537-021-00444-8
11. Puja B, Ankita P. Deep learning techniques—R-CNN to mask R-CNN: a survey. In: Proceeding of the Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019; 2020.
12. Mahbub H, Jordan J. Bird, Diego R. Faria. A study on CNN transfer learning for image classification. In: Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence (CVC). Springer International Publishing; 2019.
13. Sharma N, Jain V, Mishra A. An Analysis of Convolutional Neural Networks for Image Classification. Procedia Computer Science. 2018; 132: 377-384. doi: 10.1016/j.procs.2018.05.198
14. Ul Rahman J, Chen Q, Yang Z. Additive Parameter for Deep Face Recognition. Communications in Mathematics and Statistics. 2019; 8(2): 203-217. doi: 10.1007/s40304-019-00198-z
15. Shiv RD, Satish KS, Bidyut BC. Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing. 2022.
16. Andrinandrasana DR, Fouzia A, Peter S. A review of activation function for artificial neural network. In: Proceeding of the 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI).
17. Ul Rahman J, Rubiqa Z, Asad K. SwishReLU: A Unified Approach to Activation Functions for Enhanced Deep Neural Networks Performance. Cornell University; 2024.
18. Ul Rahman J, Makhdoom F, Lu D. Amplifying Sine Unit: An Oscillatory Activation Function for Deep Neural Networks to Recover Nonlinear Oscillations Efficiently. Cornell University; 2023.
19. Ul Rahman J, Makhdoom F, Ali A, et al. Mathematical modeling and simulation of biophysics systems using neural network. International Journal of Modern Physics B. 2023; 38(05). doi: 10.1142/s0217979224500668
20. Tian Y. Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm. IEEE Access. 2020; 8: 125731-125744. doi: 10.1109/access.2020.3006097
21. Meiyin Wu, Li Chen. Image recognition based on deep learning. 2015 Chinese Automation Congress (CAC). 2015; 32: 542-546. doi: 10.1109/cac.2015.7382560
22. Olsson F. A literature survey of active machine learning in the context of natural language processing. ResearchGate; 2009.
23. Larabi Marie-Sainte S, Alalyani N, Alotaibi S, et al. Arabic Natural Language Processing and Machine Learning-Based Systems. IEEE Access. 2019; 7: 7011-7020. doi: 10.1109/access.2018.2890076
24. Brody H. Deep learning for highway driving. Cornell University; 2015.
25. Kallenberg M, Petersen K, Nielsen M, et al. Unsupervised Deep Learning Applied to Breast Density Segmentation and

Mammographic Risk Scoring. IEEE Transactions on Medical Imaging. 2016; 35(5): 1322-1331. doi: 10.1109/tmi.2016.2532122
26. Awni H. Deep speech: Scaling up end-to-end speech recognition. Cornell University; 2014.
27. Wang Z, She Q, Ward TE. Generative Adversarial Networks in Computer Vision. ACM Computing Surveys. 2021; 54(2): 1-38. doi: 10.1145/3439723
28. Tom Michael Mitchell. The discipline of machine learning. Carnegie Mellon University; 2006.
29. Voulodimos A, Doulamis N, Doulamis A, et al. Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience. 2018; 2018: 1-13. doi: 10.1155/2018/7068349
30. O'Mahony N. Deep learning vs. traditional computer vision. In: Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference. Springer International Publishing; 2020.
31. Tan H, Lim HK. Vanishing gradient mitigation with deep learning neural network optimization. In: Proceeding of the 2019 7th International Conference on Smart Computing & Communications (ICSCC).
32. Hanin B. Which neural net architectures give rise to exploding and vanishing gradients? Advances in Neural Information Processing Systems. 2018.
33. Roodschild M, Gotay Sardiñas J, Will A. A new approach for the vanishing gradient problem on sigmoid activation. Progress in Artificial Intelligence. 2020; 9(4): 351-360. doi: 10.1007/s13748-020-00218-y
34. Hu Y. Overcoming the vanishing gradient problem in plain recurrent networks. Coenell University; 2018.
35. Le P, Zuidema W. Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. Coenell University; 2016.
36. Lu L. Dying ReLU and initialization: Theory and numerical examples. Coenell University; 2019.
37. Shin Y, Karniadakis GE. Trainability of relu networks and data-dependent initialization. Journal of Machine Learning for Modeling and Computing. 2020; 1(1): 39-74. doi: 10.1615/jmachlearnmodelcomput.2020034126
38. Ul Rahman J, Danish S, Lu D. Deep Neural Network-Based Simulation of Sel'kov Model in Glycolysis: A Comprehensive Analysis. Mathematics. 2023; 11(14): 3216. doi: 10.3390/math11143216
39. Wang X, Qin Y, Wang Y, et al. ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis. Neurocomputing. 2019; 363: 88-98. doi: 10.1016/j.neucom.2019.07.017
40. Lee H, Kim Y, Yang SY, et al. Improved weight initialization for deep and narrow feedforward neural network. Neural Networks. 2024; 176: 106362. doi: 10.1016/j.neunet.2024.106362
41. Hochreiter S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 1998; 06(02): 107-116. doi: 10.1142/s0218488598000094
42. Hu Z, Zhang J, Ge Y. Handling Vanishing Gradient Problem Using Artificial Derivative. IEEE Access. 2021; 9: 22371-22377. doi: 10.1109/access.2021.3054915
43. Daniel AL, Jennifer MN. Comparing activation functions for deep neural networks. IEEE Transactions on Neural Networks and Learning Systems. 2016; 27(11): 2207-2216.
44. Mycroft GK. Activation functions in machine learning algorithms. Journal of Machine Learning Research. 2020; 21: 1-24.
45. Schulman J. Proximal Policy Optimization Algorithms. Cornell University; 2017.
46. Schaul T. Noisy Networks for Exploration. Cornell University; 2017.
47. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014: 13th European Conference on Computer Vision. Springer International Publishing; 2014.
48. Springenberg JT, Dosovitskiy A, Brox T, et al. Striving for simplicity: The all convolutional net. Cornell University; 2014.
49. Huff DT, Weisman AJ, Jeraj R. Interpretation and visualization techniques for deep learning models in medical imaging. Physics in Medicine & Biology. 2021; 66(4): 04TR01. doi: 10.1088/1361-6560/abcd17
50. Ul Rahman J. DiffGrad for Physics-Informed Neural Networks. Cornell University; 2024.