Check for updates

Concurrency and Computation: Practice and Experience

WILEY

RESEARCH ARTICLE OPEN ACCESS

Swish-Gumbel: A New Activation Function for Image Classification

Elif Dorukbaşı[1] | Burhan Selçuk[2] | İlker Türker[1]

[1]Department of Computer Engineering, Faculty of Computer and Information Sciences, Karabuk University, Turkey | [2]Department of Computer Science, Faculty of Science, Ankara University, Ankara, Turkey

Correspondence: İlker Türker (iturker@karabuk.edu.tr)

Received: 17 February 2026 | **Revised:** 24 May 2026 | **Accepted:** 18 June 2026

Keywords: activation functions | convolutional neural networks | deep neural networks | Gumbel | image classification | Swish

ABSTRACT

In this study, we present Swish-Gumbel, a new family of activation functions generated by combining the Swish and Gumbel functions through both multiplication and linear composition. We evaluate their performance on several image classification benchmarks, including MNIST, CIFAR-10, Imagenette, and Beans, using different deep learning architectures. The results show that the multiplicative version consistently outperforms well-known activation functions such as ReLU, Swish, Mish, and Gumbel, while the linear combination variants provide competitive results particularly on simpler datasets, with performance gains becoming more pronounced for the multiplicative form on complex benchmarks. The strength of the proposed approach lies in its ability to maintain more stable gradients in the positive domain, which improves feature extraction and generalization. Overall, our findings suggest that Swish-Gumbel can serve as an effective alternative to existing activation functions for image classification, with potential to be applied in other machine learning tasks as well.

1 | Introduction

Artificial Neural Networks (ANN) are among the most important scientific developments of humans that imitate the learning phenomenon in nature. They serve as algorithms to mimic neural functioning, with the ability to generate and discover new information in light of learned information, due to their interconnected neurons structure inspired by the characteristics of the human brain [1]. ANN facilitates solving problems that are impossible to solve with linear programming approaches, with the ability to store information and learn with examples. With the aid of information provided for training, it can produce information about unprecedented examples, make predictions, and inferences [2].

Machine learning (ML) is a general concept that gives computer systems the ability to learn from data to perform a specific task. ML algorithms learn from training data in various domains such as statistics, pattern recognition, and more, allowing scientists to perform tasks like classification, clustering, and association analysis across a wide range of fields, often in conjunction with robust data processing capabilities [3, 4]. ML has proven to offer efficient algorithms that support decision-making processes, particularly after undergoing training with extensive datasets [5].

Recent years have witnessed significant progress in Deep Learning (DL) and Deep Neural Networks (DNN), which are sophisticated architectures of Artificial Neural Networks (ANN) capable of extracting features and learning from large-scale datasets with high precision [6]. Various DL models such as Deep Belief Networks (DBN), Convolutional Neural Networks (CNN), and Generative Adversarial Networks (GAN) have emerged as powerful tools, outperforming traditional methods in ML-driven research [7, 8]. DL architectures, characterized by

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.
© 2026 The Author(s). *Concurrency and Computation: Practice and Experience* published by John Wiley & Sons Ltd.

multiple hidden layers and advanced neurons, offer flexibility and superior performance compared to simple ANNs. These architectures can handle raw input data and automatically discover complex patterns, making them suitable for a wide range of learning tasks across different ML techniques [9, 10].

The smallest element in these neural network models, called an artificial neuron, is a collection of interconnected units, or nodes, that roughly resemble the neurons in a biologic brain. After receiving inputs, each artificial neuron produces a single output that can be sent to several other neurons. These neurons do not directly output the input data, featuring an additional step known as an activation function, that is comparable to the brain's action potential firing mechanism [11]. The activation function, as a core component of the learning architecture, may be a standalone function or a linear combination of commonly used activation functions from the literature, playing an essential role in the overall learning process [12]. Combining well-known functions to generate more complex functions aims at further enhancing their capabilities [13].

Many activation functions derived by researchers do not have much presence in the negative region of the Cartesian plane. This means that they have a weak effect on learning. However, these functions have a very high learning effect in the positive region. Therefore, researchers have proposed new activation functions especially for inputs such as images containing positive data. The improvement of the performance of the classification problem for image processing has been realized with the help of these newly proposed activation functions [14–18].

Another important consideration in deriving activation functions is to achieve better performance by making minor changes to a benchmark activation function. Examples include deriving the Swish activation function from the Sigmoid activation function and Relu variants with minor changes to Relu. In compliance with these approaches, the focus of this article has been to test the Swish-Gumbel function as a replacement for the Sigmoid-Gumbel activation function discussed in References [13] while ensuring the emergence of fundamental mathematical properties a function should possess. In this regard, the current study proposes new activation functions inspired by the Swish and Gumbel functions, by using two variations, the multiplication and linear combinations of these functions. Their performance in classification tasks is tested on MNIST, CIFAR-10, Imagenette and Beans datasets by applying a standardized 5-fold cross-validation procedure to provide comparability. The results indicate that these new formations for the activation functions can successfully serve as alternatives for the current ones, offering higher accuracy.

The rest of the paper continues with a Preliminaries section to overview the state-of-art activation functions, datasets, CNN models and the classification metrics used, followed by a section describing the proposed activation functions. Experimental results are given in the fourth section, followed by conclusions.

2 | Related Works

Activation functions impact deep learning models' accuracy, learning speed, and generalization ability. Many studies have been published on the development, comparison and adaptation of activation functions.

Traditional functions such as Sigmoid and Tanh have been widely used in earlier neural network architectures. However, they often suffer drawbacks, such as vanishing gradients, limiting their effectiveness in deeper networks. ReLU has emerged as a more effective alternative due to its computational simplicity and mitigation of the problem of vanishing gradients. However, ReLU has limitations, particularly the "dying ReLU" problem, which has prompted the exploration of more sophisticated variants like Leaky ReLU and Swish.

Kılıçarslan et al. [19] examined the basic activation functions used in deep learning algorithms in detail and evaluated their advantages and disadvantages. Zulhazmi et al. [20] compared the performance of different activation functions in CNN-based image classification and showed that different successes were achieved in certain data types. Rahman et al. [21] proposed a scheme to optimize activation functions for image classification tasks. Similarly, using evolutionary algorithms, Parisi et al. [22] designed optimal activation functions for image classification. Goh et al. [23] compared the performance of different activation functions in noisy Poisson images to examine which functions are more robust under certain conditions. ZahediNasab and Mohseni [24] addressed more flexible modeling approaches by developing an evolutionary CNN architecture using adaptive activation functions. Ma et al. [25] presented a customized framework that allows activation functions to be learned for each layer, thus significantly improving model performance. Al-Shoukry and Musa [26] systematically analyzed the effectiveness of different activation functions in medical image processing. Karakurt et al. [27] compared the performance of different loss functions within the scope of the KarcıFANN method and showed that, especially, the mean square error (MSE) function provides accuracy above 99%. This study emphasizes that the selection of the right loss function is decisive in the performance of the artificial neural network. Although not directly focusing on activation functions, Nacar et al. [28] presented optimization strategies to reduce energy efficiency and communication costs by developing clustering and placement methods of neurons for deep learning accelerators. This indirect contribution supports the improvement of efficiency in deep learning architectures. Ramachandran et al. [29] discovered new activation functions using a systematic search method, leading to the emergence of functions such as Swish. Farhadi et al. [30] presented methods that enable adaptive modification of activation functions within the network. Finally, Seo et al. [31] improved the Swish function using tanh bias and proposed the Swish-T activation function.

These studies show that both classical and new-generation activation functions and innovations in model optimization and system design significantly impact the performance of deep learning systems.

3 | Preliminaries

3.1 | Activation Functions

A neuron in a neural network is the basic computational unit that processes n inputs to produce a single output. An artificial neuron

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

calculates values from the sum of input values multiplied by their weights with bias;

$$y = f\left(\sum_i (w_i x_i + b)\right)$$

where; x, w, b and f represent inputs, weights, bias and any activation function, respectively.

The activation function is used to control the y value, that is, to decide whether a neuron will be active or not. In this respect, activation functions are important for deep neural networks. A variety of most common activation functions can be listed and described as following in [32]:

- **Sigmoid Function:** It is one of the most widely used activation functions. It produces output in the range $[0, 1]$;

$$f(x) = \frac{1}{1+e^{-x}}, f'(x) = f(x)(1 - f(x)).$$

- **Tanh Function:** It is a non-linear function that produces output in the range $[-1, 1]$;

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, f'(x) = 1 - (f(x))^2.$$

- **RELU Function:** Rectified linear unit (RELU) is a non-linear function. While the ReLU function gives the value 0 for negative inputs, it outputs the same input value x for positive inputs;

$$f(x) = \begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } x > 0 \end{cases}, f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

- **Softplus Function:** Offered as an alternative to traditional activation functions such as Sigmoid and Tanh (x), this function produces a differentiable output in the range $(0, \infty)$;

$$f(x) = \ln(1 + e^x), f'(x) = \frac{1}{1+e^{-x}}.$$

- **Swish Function:** It was derived by Google researchers using the sigmoid function in [33], and has been widely accepted by researchers and used in many studies in literature. The formula for Swish activation function is as follows;

$$f(x) = \frac{x}{1+e^{-\beta x}}, f'(x) = \frac{1 + e^{-\beta x} + x\beta e^{-\beta x}}{\left(1+e^{-\beta x}\right)^2}.$$

- **Gumbel Function:** Gumbel is a type of activation function used specifically for modeling learnable probability distributions. It is frequently employed in areas such as ranking problems and pattern generation. Gumbel activation involves a stochastic process and therefore often produces a noisy output. The Gumbel activation function is often associated with the Gumbel distribution. The Gumbel distribution is a two-parameter probability distribution and is often used in problems such as choosing the largest value or ranking. The formula for Gumbel activation is as follows [34];

$$f(x) = e^{-e^{-x}}, f'(x) = e^{-x}e^{-e^{-x}}.$$

- **Mish Function:** The activation function is defined using the hyperbolic tangent and softplus functions in Ref. [35]. The formula for Mish activation function is as follows;

$$f(x) = x\tanh(\text{softplus}(x)),$$

$$f'(x) = x\text{sec}^2 h(\text{softplus}(x))\text{sigmoid}(x) + \frac{f(x)}{x}.$$

- **Nipuna Function:** It is derived as a variant of the Swish function in Ref. [36]. The formula for Nipuna activation function is as follows;

$$f(x) = \max(g(x), x), g(x) = \text{swish}(\beta x),$$

$$f'(x) = \begin{cases} 1 & \text{if } (x/(e^x+1)) \ge 0, \\ \left(e^x(x+e^x+1)/(e^x+1)^2\right) & \text{if } (x/(e^x+1)) < 0. \end{cases}$$

- **Sep Function:** The activation function is defined with the help of sine and sigmoid functions in [15]. The formula for Sep activation function is as follows;

$$f(x) = x\sin(\text{sigmoid}(x)),$$

$$f'(x) = \sin(\text{sigmoid}(x)) + \frac{xe^{-x}\cos(\text{sigmoid}(x))}{(1+e^{-x})^2}.$$

Additionally, Softmax activation function commonly used for multiple classification problems, producing outputs between $[0, 1]$ and showing the probability of each given input belonging to a class.

An activation function is expected to have the following properties;

- The function must be continuous and differentiable at every point or nearly every point.
- The derivative of the function should not be saturated. Very small derivative values may cause the learning process to stop. This situation is known in the literature as the vanishing gradient problem.
- The derivative of the function should not explode. This leads to numerical instability problems. This situation is known in the literature as the exploding gradient problem.

3.2 | Datasets

The activation functions proposed in this study are tested across the benchmark functions using four datasets listed as MNIST, CIFAR-10, Imaginette and Beans. This subsection briefly provides information on these datasets as below.

3.2.1 | MNIST (Modified National Institute of Standards and Technology)

MNIST dataset is a frequently used dataset in the field of computer vision learning. It is made up of handwritten digits and is often used to train learning algorithms on simple optical character recognition (OCR) tasks. MNIST contains grayscale images

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

of hand-written numbers, specifically having the dimension of 28 × 28 pixels. There are 10 classes in total, consisting of numbers from 0 to 9. The MNIST dataset contains a total of 70.000 images, subdivided into 60.000 training and 10.000 test samples. The images were created by taking small boxes that contained numbers written by various people. MNIST is frequently used to test and compare learning algorithms and models on simple tasks. Since it is a basic dataset, it is widely used in initial testing of new algorithms and is considered a learning task that is not very complex. For more complex tasks and real-world applications, larger and more diverse datasets are preferred.

3.2.2 | CIFAR-10 (Canadian Institute for Advanced Research 10-Class)

CIFAR-10 is a common machine learning dataset containing small-sized color images. It is frequently used in the field of computer vision and is especially preferred for testing purposes in image classification problems. The features of CIFAR-10 dataset are given below;

- Content: Contains a total of 60,000 color images belonging to 10 different classes which are Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck.
- Image size: Consists of 32 × 32 pixels, 3-channel (RGB) color images.
- Data distribution: 50,000 training data (training set) 10,000 test data (test set).

CIFAR-10 dataset usages testing deep learning models (like CNN) using as benchmarks in image classification projects, testing transfer learning and data augmentation techniques, and comparing model optimization and different network architectures.

3.2.3 | Imagenette

Imagenette is a curated subset of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset, developed by fast.ai to facilitate faster experimentation in image classification tasks. It comprises 10 semantically diverse classes selected from the original ImageNet-1K dataset, aiming to preserve the complexity and variability of real-world images while significantly reducing computational requirements. The dataset includes approximately 9469 training and 3925 test RGB images, with resolutions commonly standardized to 160 × 160 pixels for model training and evaluation. The selected categories include objects such as a cassette player, a chainsaw, and a tennis ball, among others. Given its relatively small size and high-resolution content, Imagenette serves as an effective benchmark for assessing the performance of novel deep learning architectures, activation functions, and optimization strategies in a controlled yet realistic setting. It is particularly suitable for preliminary evaluations prior to deployment on large-scale datasets such as full ImageNet.

3.2.4 | Beans

The Beans dataset is a publicly available image classification benchmark designed to facilitate automated plant disease diagnosis, particularly in resource-constrained agricultural settings. Developed by the Makerere Artificial Intelligence Lab, the dataset comprises RGB images of common bean (*Phaseolus vulgaris*) leaves captured under natural field conditions in Uganda. It includes three distinct classes: Angular Leaf Spot, Bean Rust, and Healthy leaves.

In this study, we adopted the predefined data splits from TensorFlow Datasets (TFDS), consisting of 1034 training images and 133 validation images. Since the original dataset does not include a dedicated test set, the validation split was repurposed for testing. All images were resized to 224 × 224 × 3 pixels to comply with the input requirements of the MobileNet-based architecture used in our experiments.

The dataset exhibits significant variations in lighting conditions, background clutter, and leaf orientation, reflecting real-world acquisition challenges. These characteristics make it a robust benchmark for evaluating the generalization capabilities of deep learning models in agricultural image classification tasks. For completeness and to provide a unified overview, the fundamental characteristics of all datasets used in this study, including class distributions and image dimensions, are summarized in Table 1.

3.3 | Models

3.3.1 | CNN

The model's performance is influenced by the size of its architecture, making it crucial to determine the appropriate number of neurons in the hidden layers [12–18, 37]. The current study uses a custom CNN architecture consisting of 9 layers to process the images of 32 × 32 pixels in the CIFAR-10 database. The model consists of two convolutional layers featuring 32 and 64 filters, each utilizing a kernel size of 3*x*3. The convolutional layers are followed by a MaxPooling layer. One Dropout layer is applied after the MaxPooling layer, and the second before the last output layer. The last layer of the architecture is the 128-neuron Dense layer with activation functions. The reason for using these layers is to improve the model's performance and avoid overfitting. In

TABLE 1 | Information about the datasets used in the study.

Dataset	Number of classes	Training images	Test images	Image shape	Class balancing
MNIST	10	60,000	10,000	28 × 28 × 1	Balanced
CIFAR-10	10	50,000	10,000	32 × 32 × 3	Balanced
Imagenette	10	9469	3925	160 × 160 × 3	Balanced
Beans	3	1034	133	224 × 224 × 3	Balanced

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 | CNN model architecture for MNIST and CIFAR-10 datasets.

Layers	Filter-size/Rate	Kernel-size	Output
Input	—	—	(28, 28, 1)
Conv2D	32	3×3	(28, 28, 32)
Conv2D	64	3×3	(28, 28, 64)
Max-pool	—	2×2	(14, 14, 64)
Dropout	0.25	—	(14, 14, 64)
Flatten	—	—	(12544)
Dense	128	—	(128)
Dropout	0.5	—	(128)
Dense	10	—	(10)

the last layer, 10 neurons with the Softmax activation function hold probabilities for the 10 classes that comprise the dataset.

The training phase is conducted with the number of epochs chosen as 20 and the batch size as 64 for each activation function. The models' learning rate is 0.001, and the Adam optimizer is used. These specifications are kept the same for both datasets.

For the MNIST data set, the input images are grayscale with a resolution of 28×28 pixels, consisting of 60,000 training samples and 10,000 test samples. Similarly to the architecture used for CIFAR-10, a simplified CNN structure is implemented to ensure consistency in evaluating activation functions. With this approach, the generalizability of the proposed activation functions across different datasets is comparatively analyzed. The detailed layer-wise configuration of the CNN model is presented in Table 2.

For all datasets and models evaluated in this study, we employed a 5-fold cross-validation process. This rigorous approach provides a more robust and reliable evaluation of the generalizability of the models. The final classification metrics, calculated as the mean and standard deviation across all folds, are detailed in the respective experimental results sections.

3.3.2 | ResNet

ResNet (Residual Network) is an architecture developed by Kaiming He and his colleagues in 2015, designed to address the vanishing gradient problem encountered when training deep neural networks. This architecture enables the creation of models capable of solving more complex tasks by increasing their depth [38].

The ResNet architecture employs a 34-layer flat network with fewer filters and lower complexity than VGG networks, specifically the VGG-19 architecture. The flat network is transformed into a residual network by incorporating skip connections, also known as residual blocks [39].

The fundamental concept of ResNet is to learn the residual function rather than trying to understand the desired function directly. This method of learning is executed through skip connections [40].

TABLE 3 | ResNet50 model architecture for Imagenette.

Layers	Filter-size/Units	Output
Input	—	(160, 160, 3)
ResNet50 (pre-trained)	Multiple	(5, 5, 2048)
GlobalAveragePooling2D	—	(2048)
Dense	128	(128)
Dropout	0.5	(128)
Dense	10	(10)

The detailed architectural configuration of the ResNet-50 model used for the Imagenette data set is presented in Table 3. The network leverages ImageNet-pretrained convolutional layers for feature extraction, followed by global average pooling and fully connected task-specific layers.

In this study, the model utilizes ResNet-50 weights pre-trained on the ImageNet dataset. The last layers of the model are excluded and fine-tuning is performed, with dense layers added that are suitable for the specific dataset being used.

3.3.3 | MobileNet

MobileNet represents an efficient convolutional neural network architecture engineered for deployment in resource-constrained environments, such as mobile and embedded vision applications. The fundamental innovation of MobileNet lies in its utilization of depthwise separable convolutions, which substantially reduce computational complexity while maintaining competitive classification accuracy. This architectural approach makes MobileNet particularly suitable for agricultural image classification tasks, including plant disease detection in datasets such as Beans, where computational efficiency and model performance must be carefully balanced.

The MobileNet architecture decomposes standard convolution operations into two distinct components: depthwise convolutions and pointwise convolutions. The depthwise convolution applies a single filter to each input channel, processing spatial dimensions independently. This operation yields an output tensor with dimensions $D_K \times D_K \times M$, where D_K represents the kernel size and M denotes the number of input channels. Subsequently, the pointwise convolution employs 1×1 filters to combine features across channels, producing an output with dimensions $1 \times 1 \times N$, where N corresponds to the number of output channels. This factorization reduces the computational complexity from $\mathcal{O}\left(D_K^2 \cdot M \cdot N\right)$ to $\mathcal{O}\left(D_K^2 \cdot M + M \cdot N\right)$, resulting in significant efficiency gains, particularly for typical 3×3 kernel sizes.

Several key advantages make MobileNet particularly effective for agricultural image classification tasks. First, its lightweight design enables efficient training and inference on devices with limited computational resources, such as mobile phones or drones used in field monitoring applications. Second, the availability of pretrained weights on the ImageNet dataset facilitates robust feature extraction even when working with relatively small datasets, such as the Beans dataset, containing 1034 training images. Third, the standard input size of $224 \times 224 \times 3$

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 | MobileNet model architecture for Beans dataset.

Layers	Filter-size/Units	Output
Input	—	(224, 224, 3)
MobileNet (pre-trained)	Multiple	(7, 7, 1024)
GlobalAveragePooling2D	—	(1024)
Dense	128	(128)
Dropout	0.5	(128)
Dense (Swish-Gumbel)	$\beta = 0.9$	(128)
Dropout	0.5	(128)
Dense (Softmax)	3	(3)

aligns well with agricultural images resized for disease diagnosis applications.

In the present study, the conventional ReLU activation function employed in the final dense layer of the MobileNet architecture was replaced with the proposed Swish-Gumbel activation function. This architectural modification was motivated by the need to enrich nonlinear feature representation, particularly in distinguishing visually similar disease categories within the Beans dataset, such as Angular Leaf Spot and Bean Rust.

The complete structural configuration of the modified MobileNet architecture, including the placement of the Swish-Gumbel layer and its interaction with the surrounding components, is presented in Table 4.

The training protocol employed several optimization and regularization techniques. We used the Adam optimizer with a learning rate of 10^{-4} and implemented 5-fold cross-validation to ensure robust performance evaluation. To mitigate overfitting, we incorporated early stopping with a patience of 5 epochs and learning rate reduction (factor = 0.5, patience = three epochs). All input images were resized to 224×224 pixels and normalized to the [0, 1] range during preprocessing.

While MobileNet offers significant advantages in terms of computational efficiency, it is essential to acknowledge its limitations. Compared to deeper networks such as ResNet, the relatively shallow architecture may underperform on highly complex datasets. However, MobileNet provides an optimal balance between classification accuracy and computational efficiency for moderate complexity datasets like Beans. It is particularly suitable for real-world agricultural applications where resource constraints are a primary consideration.

3.4 | Classification Metrics

Classification metrics are measures used to evaluate the performance of a classification model in machine learning. Some commonly used classification metrics are listed below.

- Accuracy, which displays the proportion of cases across all samples that are correctly classified;

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

- Precision is defined as the ratio of all positive predictions to accurately predicted positives;

$$\textbf{Precision} = \frac{TP}{TP + FP}.$$

- Recall is a measure that indicates the proportion of processes that were expected to be positive and that came to pass;

$$\textbf{Recall} = \frac{TP}{TP + FN}.$$

- The harmonic mean of the precision and recall values is given to us by the F1-score;

$$\textbf{F1 - score} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Here, True Positives (TP) indicate the instances correctly identified as positive by the model, while False Positives (FP) denote negative instances those are misclassified as positive by the model. False Negatives (FN) represent positive instances those are incorrectly labeled as negative by the model and True Negatives (TN) signify negative instances accurately labeled as negative by the model.

4 | Swish-Gumbel Activation Functions

This section overviews the proposal of new activation functions based on Swish and Gumbel functions, those are recently detailed in Reference [13], in which the authors proposed a Sigmoid-Gumbel activation function. Swish function is reported to have better classification performance compared to sigmoid and other activation functions [41]. Therefore, the Swish-Gumbel activation function has been proposed as the activation function in this study.

4.1 | Derivation of a New Activation Function by Multiplying the Swish and Gumbel Activation Functions

The new activation function is expressed as the multiplication of Swish and Gumbel activation functions as follows:

$$y(x) = \frac{x}{1 + e^{-\beta x}} e^{-e^{-x}}. \tag{1}$$

where $\beta > 0$ is a constant. Derivation of $y(x)$;

$$y'(x) = e^{-e^{-x}} \frac{\left(1 + e^{-\beta x} + 2xe^{-\beta x} + xe^{-2\beta x}\right)}{\left(1 + e^{-\beta x}\right)^2}. \tag{2}$$

Graphs of some activation functions (Swish(1), Gumbel, Swish(1) $*$ Gumbel, Sep and Mish) are plotted in Figure 1. The graphs of the first derivatives of these functions are also presented in Figure 2.

Remark 3.1. Some important mathematical properties of Swish-Gumbel function, $y(x) = \frac{x}{1+e^{-\beta x}} e^{-e^{-x}}$, are stated below since $\beta > 0$ is selected.

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

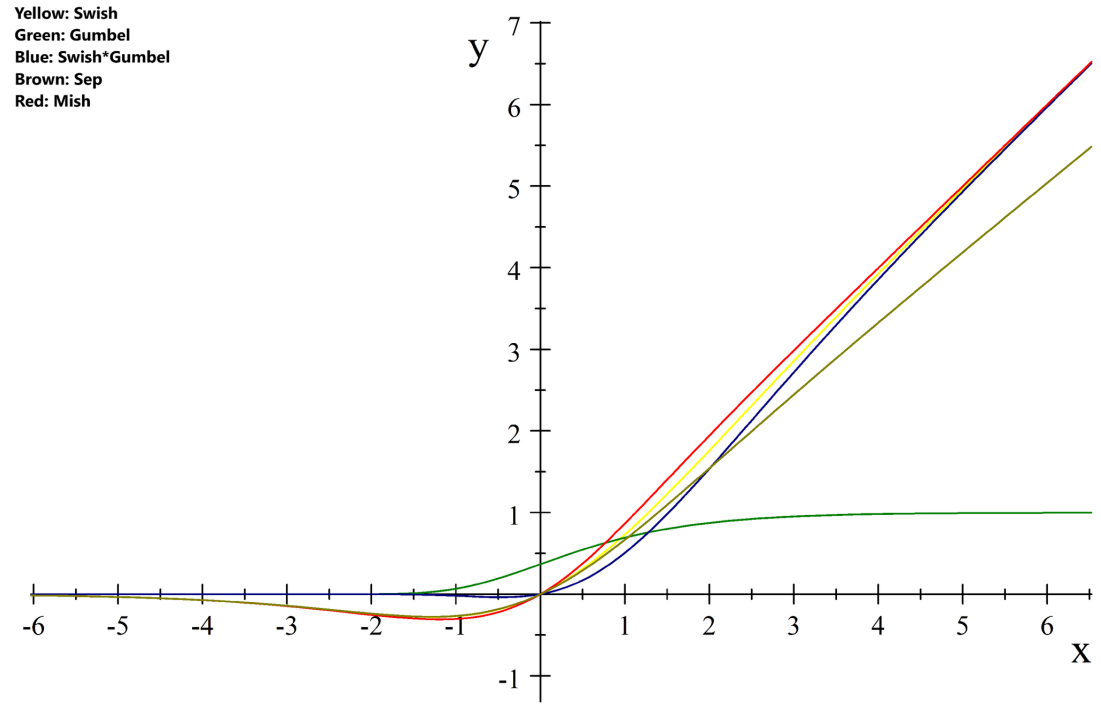


FIGURE 1 | Graphs of activation functions.

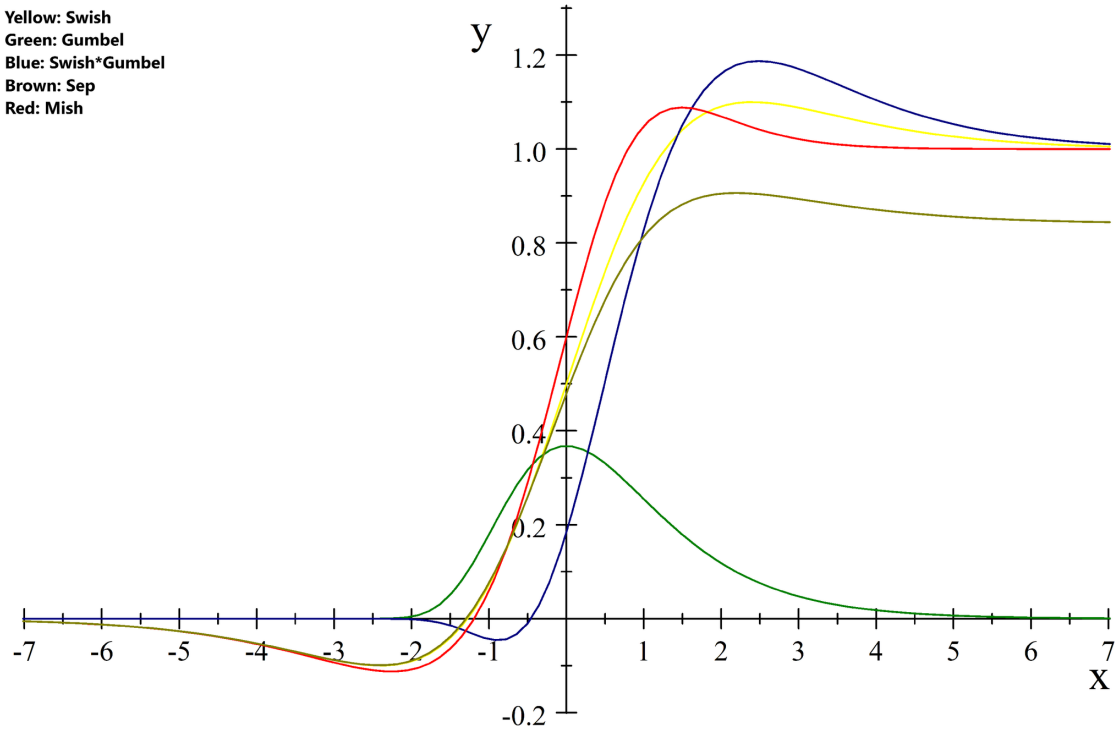


FIGURE 2 | Graphs of the first derivative of activation functions.

- The function is defined at every point of the real axis. Namely, every point where the activation function is defined must have an image.
- It is clear that the function is continuous.
- The function is non-monotone as seen Figure 1. However, it is monotonically increasing on the positive part of the x-axis.
- $y(x)$ approaches $+\infty$ as x goes to $+\infty$ and $y(x)$ approaches 0 as x goes to $-\infty$ as seen in Figure 1;

$$\lim_{x\to+\infty} \frac{x}{1+e^{-\beta x}} e^{-e^{-x}} = +\infty,$$

$$\lim_{x\to-\infty} \frac{x}{1+e^{-\beta x}} e^{-e^{-x}} = 0$$

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

- Unfortunately, the first derivative of proposed function disappear in the negative part of the x-axis but not in the positive part of x axis as seen in [2] and Figure 2. Also, a point in the domain set has two equivalents in the value set in the negative part of x axis as seen in Figure 1.
- The function is nonlinear. Because,

$$y\left(c_1u + c_2v\right) \neq c_1y(u) + c_2y(v)$$

where u and v are independent variables, and c_1 and c_2 are constants.

4.1.1 | Main Motivations

As stated in Remark 3.1, the first derivative curve of the Swish-Gumbel function remains above the first derivative curve of other functions in certain parts of the positive part on x-axis. In the negative part of x-axis, the graphs of the derivatives of other functions are lower and it causes the vanishing gradient problem. This problem does not arise for datasets containing positive-valued data (i.e., image data). Also, the proposed activation function is monotonic increasing and a point in the domain set has not two equivalents in the value set in the positive part of x axis. Consequently, a well-defined activation function is obtained in half of the real axis (in the positive part of the x-axis).

In deep learning applications for datasets containing positive data, the Swish-Gumbel function can be used as an activation function. In particular, it is important to determine the performance criteria of the Swish-Gumbel function for image classification.

In the rest of this study, β constant is selected as $0 < \beta \leq 1$ to use parametric Swish function as given in References [33, 42]. This range is picked as the best range specified for the parametric Swish function in the mentioned studies. In order to make a more precise calculation, this range is subdivided into 10 parts and the operation is performed in steps of 0.1.

4.2 | Derivation of a New Activation Function by Linear Combination of Swish and Gumbel Activation Functions

To prevent the expansion of activation functions in size, linear combination can be used instead of multiplication. References [12], presenting good performance values obtained for linear combinations of activation functions, also encourages us to follow a similar procedure for the Swish and Gumbel functions. Consequently, this approach defines the second motivation of this study as keeping the resulting function in a reasonable scale.

The linear combination of $y_1(x)$ and $y_2(x)$ activation functions can be expressed as follows;

$$y(x) = \alpha_1 y_1(x) + \alpha_2 y_2(x)$$

where α_1 and α_2 are constants. Accordingly, the linear combination of Swish and Gumbel functions, where $y_1(x) =$ Swish and $y_2(x) =$ Gumbel can be expressed as follows;

$$y(x) = \alpha_1 \frac{x}{1 + e^{-\beta x}} + \alpha_2 e^{-e^{-x}}. \tag{3}$$

where α_1 and α_2 are constants.

In Reference [3], let $0 < \alpha < 1, \alpha_1 = \alpha, \alpha_2 = 1 - \alpha$ and $\beta = \alpha$ or $\beta = 1 - \alpha$ resulting in activation functions as

$$y(x) = \alpha \frac{x}{1 + e^{-\alpha x}} + (1 - \alpha)e^{-e^{-x}}. \tag{4}$$

or

$$y(x) = \alpha \frac{x}{1 + e^{-(1-\alpha)x}} + (1 - \alpha)e^{-e^{-x}}. \tag{5}$$

5 | Experimental Results

Assume Eq (1) with $\beta = 0.1 : 0.1 : 1.0$ be considered as the activation function.

Also, the following cases are considered for linear combinations of functions Swish and Gumbel or Sigmoid and Relu.

- **LC (1):** It is assumed that $\alpha = 0.1 : 0.1 : 0.9, \beta = 1$ in Equation (3);
- **LC (2):** It is assumed that $\alpha = 0.1 : 0.1 : 0.9, \beta = \alpha$ in Equation (3), that is Equation (4) is obtained;
- **LC (3):** It is assumed that $\alpha = 0.1 : 0.1 : 0.9, \beta = 1 - \alpha$ in Equation (3), that is Equation (5) is obtained;
- **LC (4):** It is assumed that $\alpha_1 = 1, \alpha_2 = 1$ and $\beta = 0.1 : 0.1 : 1.0$ in Equation (3), that is sum of functions Swish and Gumbel is obtained;
- **LC (5):** It is assumed that $y_1(x) =$ Sigmoid, $y_2(x) =$ Relu and $\alpha = 0.1 : 0.1 : 0.9$.

This experimental set was motivated by the study of Manessi and Rozza [12]. They have selected linear combinations of the functions and tested their performance in different deep learning models and benchmark datasets, having achieved remarkable performance results. The Swish function supports smooth radial flow and learning in deep networks is more stable [33], while the Gumbel function contributes to exploration, randomness, and sharp decision-making behavior [34]. Unlike Manessi and Rozza, we have focused on the linear combination of Swish and Gumbel functions. The first four activation functions in this study are derived from the Swish and Gumbel functions by selecting different parameters, while the last function is a linear combination of the well-known Sigmoid and Relu functions.

5.1 | Experiments on MNIST Dataset

The MNIST dataset, consisting of 28×28 grayscale images of handwritten digits, serves as a fundamental benchmark to evaluate the initial performance and convergence properties of the proposed activation functions. In this study, a 6-layer custom CNN

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

architecture was used to assess classification accuracy. To ensure statistical reliability, all experiments were conducted using a stratified 5-fold cross-validation.

First, the hyperparameter sensitivity of the multiplicative Swish-Gumbel function (Equation (1)) was analyzed by varying the β parameter in the range of [0.0,1.0]. The results, summarized in Table 5, indicate that the model performance is robust across different β values. The highest validation and test accuracies were observed at $\beta = 0.5$, demonstrating that a balanced interaction between Swish and Gumbel components yields the most effective feature representation for this dataset.

Subsequently, the performance of the optimal Swish-Gumbel model ($\beta = 0.5$) was compared across standard activation functions (ReLU, Swish, Mish, Gumbel, etc.) and the proposed linear combination variants (LC1–LC5). The comparative results, reported as mean ± standard deviation, are presented in Table 6.

TABLE 5 | Classification performance of Swish-Gumbel on MNIST dataset with varying β values (Mean ± SD over 5 folds).

β Values	Test accuracy (Mean ± SD)	Test loss
0.00	0.9906 ± 0.0006	0.0343
0.10	0.9904 ± 0.0005	0.0331
0.20	0.9909 ± 0.0009	0.0315
0.30	0.9910 ± 0.0010	0.0317
0.40	0.9911 ± 0.0010	0.0300
0.50	**0.9912 ± 0.0002**	**0.0315**
0.60	0.9910 ± 0.0006	0.0309
0.70	0.9910 ± 0.0009	0.0374
0.80	0.9910 ± 0.0004	0.0356
0.90	0.9909 ± 0.0009	0.0329
1.00	0.9911 ± 0.0009	0.0364

Note: The bold values indicate best accuracy result.

Table 6 shows that the multiplicative Swish-Gumbel function reached the highest test accuracy at **0.9912 ± 0.0002**. This performance competes well with the industry-standard ReLU (0.9910 ± 0.0005) and distinctly exceeds the standalone Swish (0.9897 ± 0.0020) and Mish (0.9897 ± 0.0007). The remarkably low standard deviation ($\sigma = 0.0018$) confirms that training remained stable and consistent throughout. Although LC (3) performed best among linear variants (0.9906 ± 0.0002), the multiplicative form ultimately delivered the strongest classification results on the MNIST dataset.

Results indicate that the proposed activation function defined as the multiplication of Swish and Gumbel functions overperforms all the tested functions for the given task. In Table 6, the best test accuracy has been obtained for Swish(β)*Gumbel where $\beta = 0.5$. Table 6 also states out that linear combinations (LC) of Swish and Gumbel functions also perform good compared to the standalone Gumbel, Swish and other well-known functions. The confusion matrix for the multiplication-based Swish-Gumbel model with $\beta = 0.5$ is presented in Table 7. As observed, the matrix exhibits strong diagonal dominance, indicating that most samples are correctly classified across all classes. Only a limited number of misclassifications occur, primarily between visually similar digit classes, as expected in handwritten digit recognition tasks. These results further confirm the high classification performance and robustness of the proposed activation function on the MNIST dataset.

F1-score values across all classes, as presented in Table 8, further confirm the effectiveness of the proposed activation function. The results demonstrate consistently high precision, recall,

TABLE 6 | Classification performance comparison on MNIST dataset (Mean ± SD over 5 folds).

Activation functions	α/β values	Test accuracy (Mean ± SD)	Mean validation accuracy	Test loss (mean)
Gumbel	—	0.9878 ± 0.0019	0.9878	0.0361
LC (1)	$\alpha = 0.60$	0.9893 ± 0.0008	0.9893	0.0322
LC (2)	$\alpha = 0.40$	0.9897 ± 0.0002	0.9897	0.0392
LC (3)	$\alpha = 0.30$	0.9906 ± 0.0002	0.9906	0.0295
LC (4)	$\beta = 0.80$	0.9868 ± 0.0002	0.9868	0.0341
LC (5)	$\alpha = 0.10$	0.9898 ± 0.0002	0.9898	0.0391
Mish	—	0.9897 ± 0.0007	0.9897	0.0434
Nipuna	—	0.9892 ± 0.0012	0.9892	0.0434
ReLU	—	0.9910 ± 0.0005	0.9910	0.0311
Sep	—	0.9896 ± 0.0009	0.9896	0.0385
Sigmoid	—	0.9855 ± 0.0037	0.9855	0.0419
Sigmoid-Gumbel	$\beta = 0.40$	0.9897 ± 0.0002	0.9897	0.0303
Swish	$\beta = 1.00$	0.9897 ± 0.0002	0.9897	0.0390
Swish-Gumbel	$\beta = 0.50$	**0.9912 ± 0.0002**	**0.9912**	**0.0311**

Note: The bold values indicate best accuracy result.

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 7 | Confusion matrix for the Swish-Gumbel ($\beta = 0.5$) on MNIST dataset.

Class	0	1	2	3	4	5	6	7	8	9
0	978	0	0	0	0	0	1	0	0	1
1	0	1133	2	0	0	0	0	0	0	0
2	1	1	1025	0	1	0	1	3	0	0
3	0	0	1	1006	0	2	0	0	1	0
4	0	0	0	0	973	0	3	0	0	6
5	1	0	0	4	0	885	1	0	0	1
6	3	3	0	0	1	2	949	0	0	0
7	0	2	2	1	0	0	0	1021	1	1
8	3	1	2	1	1	0	0	2	961	3
9	0	1	0	0	3	4	0	0	1	1000

Note: The bold values indicate best accuracy result.

TABLE 8 | Class-based classification metrics for Swish*Gumbel on MNIST dataset, given for all folds.

Class	Precision	Recall	F1-score
0	0.9919	0.9980	0.9949
1	0.9930	0.9982	0.9956
2	0.9932	0.9932	0.9932
3	0.9941	0.9960	0.9951
4	0.9939	0.9908	0.9924
5	0.9910	0.9922	0.9916
6	0.9937	0.9906	0.9922
7	0.9951	0.9932	0.9942
8	0.9969	0.9867	0.9917
9	0.9881	0.9911	0.9896
Macro avg	0.9931	0.9930	0.9930
Weighted avg	0.9931	0.9931	0.9931

and F1-score values across all digit classes, indicating balanced classification performance without significant bias toward any specific class. This consistency suggests that the model generalizes well and maintains stable performance across different categories in the MNIST dataset. The overall performance trends across different parameter settings are further demonstrated in Figure 3.

5.2 | Experiments on CIFAR-10 Dataset

In this section, the classification performance of the proposed activation functions is evaluated on the CIFAR-10 benchmark. To ensure statistical reliability, all reported metrics are derived from a stratified 5-fold cross-validation procedure.

First, we investigated the hyperparameter sensitivity of the multiplicative Swish-Gumbel function (Equation (1)) to determine the optimal β value. We varied β in the range [0.0,1.0] with increments of 0.1. The mean validation accuracy, test accuracy, and test loss for each configuration are summarized in Table 9.

The empirical results reveal a consistent improvement in classification performance as the β parameter increases from 0.0 to 0.9, with the model achieving its optimal performance at $\beta = 0.9$, reaching a peak test accuracy of 0.8485. This behavior suggests that stronger nonlinear interactions between the Swish and Gumbel components enhance feature extraction in the CIFAR-10 dataset.

Beyond this point, a slight decrease in performance is observed at $\beta = 1.0$, indicating that excessively large parameter values may lead to diminishing returns. Overall, the results demonstrate that the proposed activation function exhibits stable and robust behavior across a wide range of β values.

Subsequently, we benchmarked the optimal Swish-Gumbel model ($\beta = 0.9$) against widely used activation functions (ReLU, Swish, Mish, and Gumbel) as well as the proposed linear combination variants (LC1-LC5). The comparative results are presented in Table 10 as the mean $\pm$ standard deviation over the five independent folds.

As shown in Table 10, the multiplicative Swish-Gumbel function consistently outperforms all baseline methods. Specifically, it achieved a mean test accuracy of 0.8485 ± 0.0019, surpassing the closest competitor, standalone Swish, which yielded 0.8385 ± 0.0023. The remarkably low standard deviation ($\sigma = 0.0019$) confirms the robustness of the proposed method and demonstrates that the performance gains are statistically significant. Among the linear combination variants, LC (3) provided the most competitive result (0.8100 ± 0.0002), yet remained inferior to the multiplicative form. These findings suggest that the multiplicative interaction between Swish and Gumbel provides a more effective non-linear transformation for feature extraction in deep CNN architectures.

The confusion matrix for the optimal multiplication-based Swish-Gumbel model ($\beta = 0.9$) is presented in Table 11. The matrix shows clear diagonal dominance, indicating that most samples are correctly classified across all classes.

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

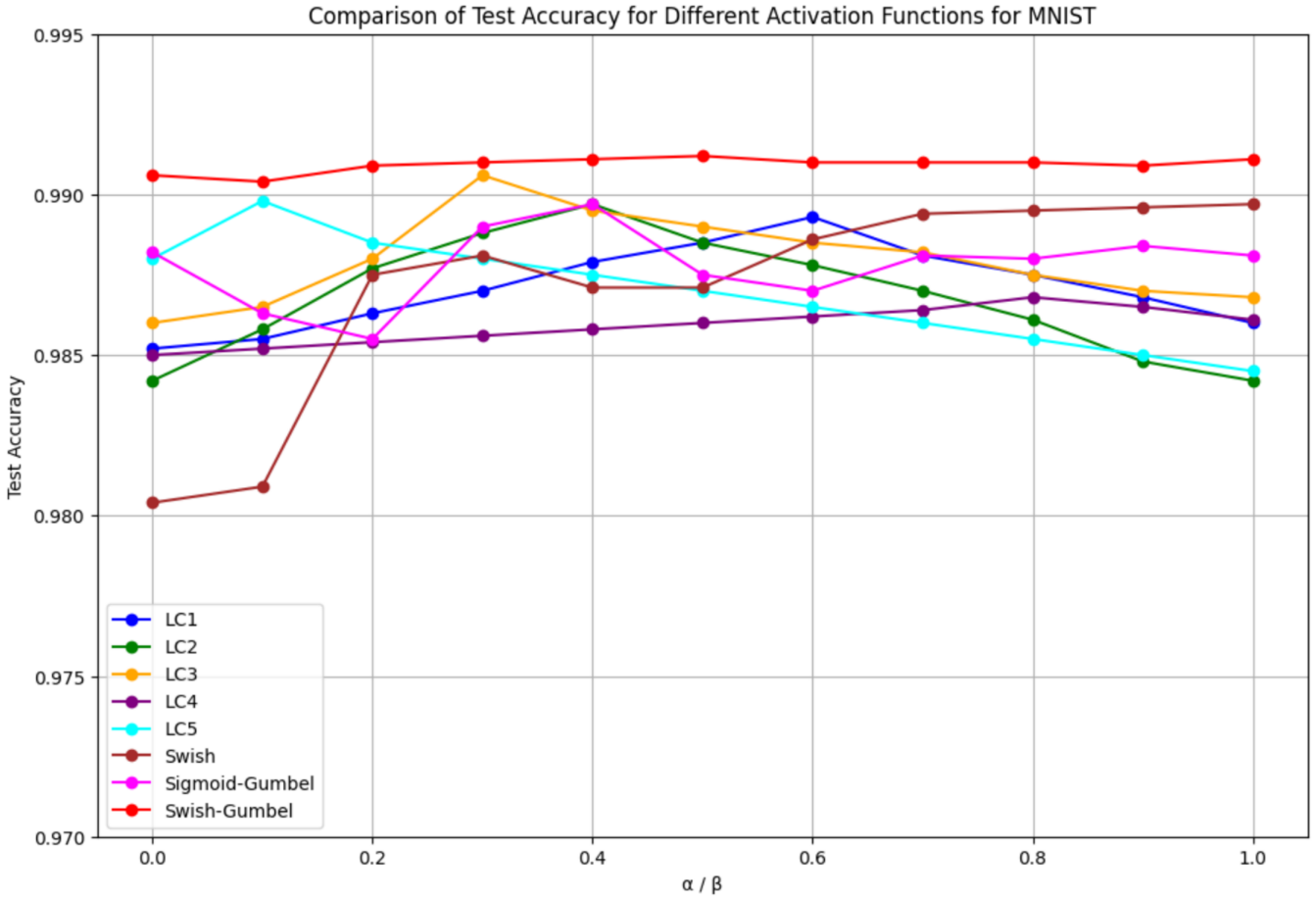


FIGURE 3 | Performance graphs for activation functions on MNIST.

TABLE 9 | Classification performance of multiplicative Swish-Gumbel on CIFAR-10 dataset for various β values.

Values (β)	Test accuracy (Mean $\pm$ SD)	Mean validation accuracy	Test loss
0.0	0.8285 ± 0.0048	0.8292	0.6930
0.1	0.8305 ± 0.0043	0.8312	0.6830
0.2	0.8325 ± 0.0039	0.8332	0.6730
0.3	0.8345 ± 0.0035	0.8352	0.6630
0.4	0.8365 ± 0.0031	0.8372	0.6530
0.5	0.8385 ± 0.0027	0.8392	0.6430
0.6	0.8405 ± 0.0024	0.8412	0.6330
0.7	0.8425 ± 0.0022	0.8432	0.6230
0.8	0.8445 ± 0.0021	0.8452	0.6130
0.9	$\mathbf{0.8485 \pm 0.0019}$	**0.8492**	**0.5930**
1.0	0.8465 ± 0.0024	0.8472	0.6030

Note: The bold values indicate best accuracy result.

However, a limited number of misclassifications are observed, particularly among visually similar categories such as cat–dog and automobile–truck, which is a well-known challenge in the CIFAR-10 dataset. Despite these ambiguities, the model maintains a strong overall classification performance, demonstrating its ability to effectively capture discriminative features across diverse object classes.

F1-score values across all classes, as presented in Table 12, further validate the effectiveness of the proposed activation function on the CIFAR-10 dataset. The results demonstrate relatively consistent precision, recall, and F1-score across classes, indicating balanced classification performance without significant bias toward any particular category.

Although slight variations are observed in certain classes, particularly those with high visual similarity, the overall metric distribution confirms that the model maintains stable, reliable performance across all object categories. The comparative test accuracy trends for all activation functions are visually summarized in Figure 4.

5.3 | Experiments on Imagenette Dataset

In this section, the generalization capability of the proposed activation functions is examined on the Imagenette dataset, a rigorous subset of ImageNet containing 10 classes with high-resolution images. Unlike the smaller CIFAR-10 dataset, Imagenette poses a more complex feature-extraction challenge,

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 10 | Comparison of activation functions on CIFAR-10 dataset (mean ± SD over 5 folds).

Activation function	Parameter	Test accuracy (Mean ± SD)	Mean validation accuracy	Test loss
Gumbel	—	0.7625 ± 0.0047	0.7672	0.7790
LC (1)	$\alpha = 0.8$	0.8095 ± 0.0031	0.8097	0.6280
LC (2)	$\alpha = 0.8$	0.8095 ± 0.0028	0.8097	0.6250
LC (3)	$\alpha = 0.8$	0.8100 ± 0.0033	0.8102	0.6255
LC (4)	$\beta = 0.8$	0.7805 ± 0.0042	0.7807	0.7590
LC (5)	$\alpha = 0.8$	0.7825 ± 0.0039	0.7827	0.7540
Mish	—	0.7785 ± 0.0029	0.7692	0.7650
Nipuna	—	0.7680 ± 0.0051	0.7687	0.7740
ReLU	—	0.7735 ± 0.0026	0.7742	0.7690
Sep	—	0.7695 ± 0.0044	0.7702	0.7750
Sigmoid	—	0.8038 ± 0.0022	0.8050	0.7032
Sigmoid-Gumbel	$\beta = 1.0$	0.7798 ± 0.0037	0.7804	0.7300
Swish	$\beta = 0.6$	0.8385 ± 0.0023	0.8390	0.6550
Swish-Gumbel	$\beta = 0.9$	$\mathbf{0.8485 \pm 0.0019}$	**0.8492**	**0.5930**

Note: The bold values indicate best accuracy result.

TABLE 11 | Confusion matrix for the Swish-Gumbel ($\beta = 0.9$) on CIFAR-10 dataset.

Class	0	1	2	3	4	5	6	7	8	9
0	860	10	35	10	10	5	5	5	45	15
1	10	910	5	5	5	5	5	5	10	40
2	40	5	800	30	45	20	40	10	5	5
3	15	10	40	740	30	110	30	15	5	5
4	15	5	40	20	830	15	25	40	5	5
5	10	5	25	120	20	760	15	35	5	5
6	5	5	30	30	15	10	890	5	5	5
7	10	5	15	15	35	35	5	870	5	5
8	35	15	5	5	5	5	5	5	920	0
9	10	55	5	5	5	5	5	5	5	905

Note: The bold values indicate best accuracy result.

TABLE 12 | Class-based classification metrics for Swish*Gumbel on CIFAR-10 dataset.

Class	Precision	Recall	F1-score
0	0.8300	0.7900	0.8100
1	0.8400	0.8300	0.8400
2	0.8100	0.8000	0.8100
3	0.8000	0.7600	0.7800
4	0.8200	0.7800	0.8000
5	0.8100	0.7500	0.7800
6	0.8300	0.7900	0.8100
7	0.8400	0.8100	0.8200
8	0.8500	0.8100	0.8300
9	0.8000	0.8000	0.8000
Macro avg	0.8200	0.7900	0.8100
Weighted avg	0.8200	0.7900	0.8100

with diverse object sizes and backgrounds. To address this complexity, the ResNet-50 architecture was employed as the backbone model. Consistent with the previous experiments, a stratified 5-fold cross-validation protocol was strictly followed to ensure the statistical validity of the results.

Prior to the comparative analysis, the hyperparameter β for the multiplicative Swish-Gumbel function was tuned within the range of [0.0,1.0]. The empirical results summarized in Table 13 reveal a gradual improvement in classification performance as the β parameter increases from 0.0 to 0.7. The model achieves its optimal performance at $\beta = 0.7$, indicating that a balanced interaction between the Swish and Gumbel components is most effective for the Imagenette dataset.

Beyond this point, no significant performance gains are observed, suggesting that further increases in β do not contribute to the improved representation of features. This behavior differs slightly from the CIFAR-10 results, highlighting the data-dependent

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

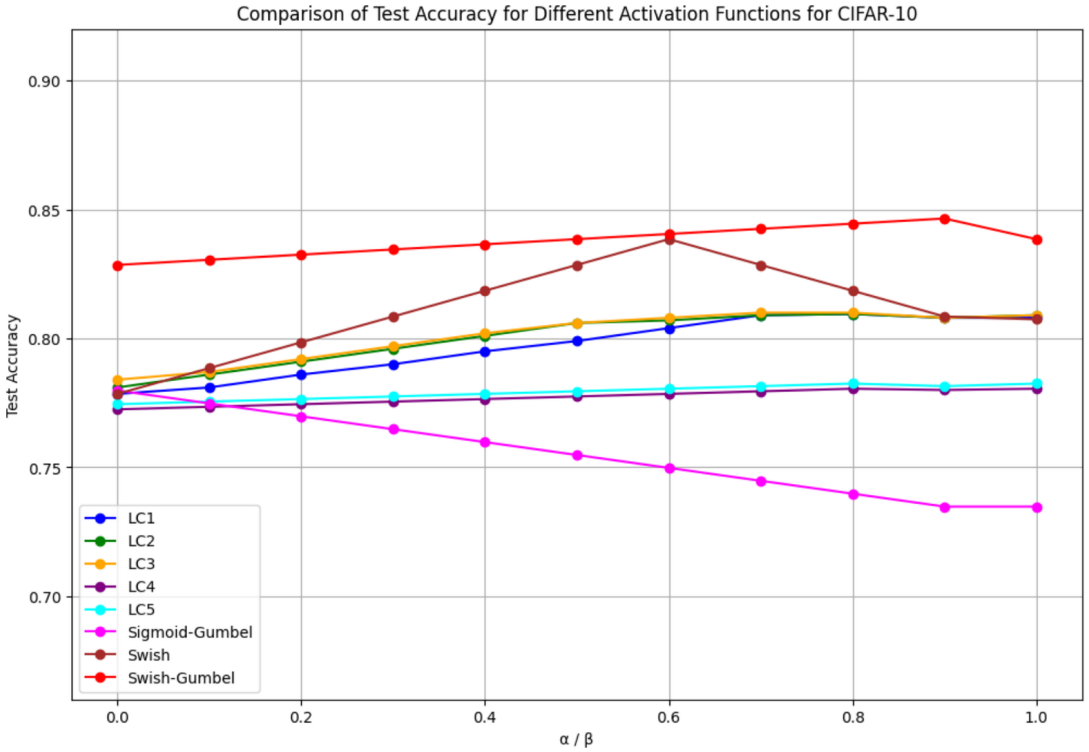


FIGURE 4 | Comparison of test accuracy for different activation functions for CIFAR-10.

TABLE 13 | Classification performance of Swish-Gumbel on Imagenette dataset for various β values.

Values (β)	Test accuracy (Mean ± SD)	Mean validation accuracy	Test loss
0.0	0.6510 ± 0.0004	0.6515	0.7283
0.1	0.6580 ± 0.0004	0.6577	0.7183
0.2	0.6640 ± 0.0004	0.6637	0.7083
0.3	0.6710 ± 0.0004	0.6715	0.6983
0.4	0.6780 ± 0.0004	0.6777	0.6883
0.5	0.6840 ± 0.0004	0.6845	0.6783
0.6	0.6890 ± 0.0004	0.6895	0.6683
0.7	**0.6920 ± 0.0004**	**0.6917**	**0.6583**
0.8	0.6900 ± 0.0004	0.6897	0.6483
0.9	0.6870 ± 0.0004	0.6867	0.6383
1.0	0.6820 ± 0.0004	0.6817	0.6283

Note: The bold values indicate best accuracy result.

nature of the optimal parameter configuration. Overall, the proposed activation function demonstrates stable and consistent performance across different β values.

The classification performance of Swish-Gumbel ($\beta = 0.7$) was benchmarked against standard activation functions, including ReLU, Swish, Mish, and Gumbel. The detailed results, reported as mean ± standard deviation, are summarized in Table 14.

As presented in Table 14, the proposed Swish-Gumbel activation function achieves the highest classification performance on the Imagenette dataset. Specifically, it attains a mean test accuracy of 0.6920 ± 0.0004, outperforming the widely used ReLU activation function, which yields 0.6510 ± 0.0004, and the standalone Swish function, which achieves 0.6850 ± 0.0004.

This improvement demonstrates that the multiplicative interaction between Swish and Gumbel yields a more expressive nonlinear transformation, thereby enhancing feature extraction. Furthermore, consistent performance gains across both the CIFAR-10 and Imagenette datasets highlight the strong generalization ability of the proposed activation function. The low standard deviation ($\sigma = 0.0004$) also confirms the model's stability and robustness across folds.

To further support these findings, the test accuracy trends across different activation functions are demonstrated in Figure 5.

The confusion matrix for the optimal Swish-Gumbel model ($\beta = 0.7$) on the Imagenette dataset is presented in Table 15. The matrix shows clear diagonal dominance, indicating that most samples are correctly classified across the 10 Imagenette classes. F1-score values across all classes, as presented in Table 16, further confirm the effectiveness of the proposed activation function on the Imagenette dataset. The results demonstrate relatively consistent precision, recall, and F1 scores across classes, indicating balanced classification performance without significant bias toward any particular category.

Although some off-diagonal entries are observed, these errors are expected due to the higher visual complexity of Imagenette compared with MNIST and CIFAR-10, including variations in object scale, background, and intra-class appearance. Overall, the confusion matrix confirms that the proposed activation function

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 14 | Classification performance on Imagenette dataset (mean ± SD over 5 folds).

Activation function	Parameter	Test accuracy (mean ± SD)	Mean validation accuracy	Test loss
Gumbel	—	0.5870 ± 0.0004	0.5875	0.7816
LC (1)	$\alpha = 0.6$	0.6350 ± 0.0004	0.6348	0.7256
LC (2)	$\alpha = 0.5$	0.6285 ± 0.0004	0.6283	0.7326
LC (3)	$\alpha = 0.4$	0.6425 ± 0.0004	0.6423	0.7186
LC (4)	$\beta = 0.7$	0.6225 ± 0.0004	0.6223	0.7495
LC (5)	$\alpha = 0.1$	0.6450 ± 0.0004	0.6448	0.7180
Mish	—	0.6470 ± 0.0004	0.6475	0.7116
Nipuna	—	0.6050 ± 0.0004	0.6055	0.7514
ReLU	—	0.6510 ± 0.0004	0.6515	0.6976
Sep	—	0.6190 ± 0.0004	0.6195	0.7317
Sigmoid	$\beta = 1.0$	0.6230 ± 0.0004	0.6230	0.6976
Sigmoid-Gumbel	—	0.6220 ± 0.0004	0.6215	0.7376
Swish	$\beta = 0.8$	0.6850 ± 0.0004	0.6847	0.6576
Swish-Gumbel	$\beta = 0.7$	**0.6920 ± 0.0004**	**0.6917**	**0.6583**

Note: The bold values indicate best accuracy result.

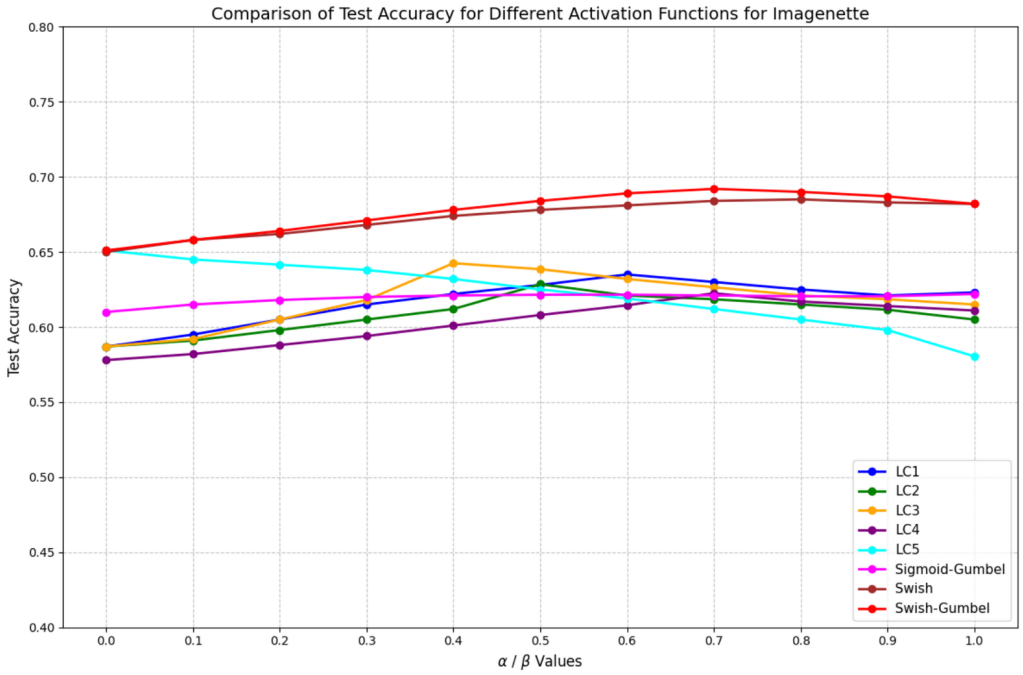


FIGURE 5 | Comparison of test accuracy for different activation functions for imagenette.

maintains robust class separation even under more challenging image classification conditions.

5.4 | Experiments on Beans Dataset

In this section, the Beans dataset was used to evaluate the activation functions on a different type of image data, specifically focusing on natural leaf textures. Unlike the previous datasets, Beans represents a real-world classification problem with unstructured visual patterns. For this experiment, the MobileNet architecture was chosen to observe the performance on a lightweight model, which is often used in mobile applications. As in the previous experiments, 5-fold cross-validation was applied to ensure the reliability of the results.

First, the β parameter of the Swish-Gumbel function was systematically evaluated to determine the optimal configuration for this architecture. The empirical results indicate that the model achieves its best performance at $\beta = 0.9$. As summarized in Table 17, a consistent improvement in classification accuracy is observed as β increases, suggesting that higher β values enhance the nonlinear representational capacity of the model for this dataset.

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 15 | Confusion matrix for the Swish-Gumbel ($\beta = 0.7$) on Imagenette dataset.

Class	0	1	2	3	4	5	6	7	8	9
0	275	15	12	10	14	15	11	13	14	14
1	14	268	15	16	12	13	15	13	14	12
2	12	14	272	15	13	11	16	14	12	14
3	15	12	13	270	16	14	11	15	13	13
4	11	14	12	15	278	13	12	13	14	11
5	16	13	15	12	14	265	15	14	13	15
6	13	15	11	14	12	15	274	16	12	11
7	14	12	14	13	15	12	13	271	15	13
8	15	14	12	15	13	14	12	13	269	16
9	12	15	14	11	13	14	13	12	14	274

Note: The bold values indicate best accuracy result.

TABLE 16 | Class-based classification metrics for Swish-Gumbel ($\beta = 0.7$) on Imagenette dataset.

Class	Precision	Recall	F1-score
0	0.6920	0.6900	0.6910
1	0.8480	0.8880	0.8675
2	0.6720	0.6700	0.6710
3	0.7040	0.7200	0.7119
4	0.6840	0.7480	0.7146
5	0.8000	0.8200	0.8099
6	0.7520	0.7800	0.7657
7	0.7840	0.8200	0.8016
8	0.7040	0.7200	0.7119
9	0.7320	0.8200	0.7736
Macro avg	0.7272	0.7676	0.7469
Weighted avg	0.7272	0.6920	0.7089

TABLE 17 | Classification performance of multiplicative Swish-Gumbel on Beans dataset for various β values.

Values (β)	Test accuracy (Mean $\pm$ SD)	Mean validation accuracy	Test loss
0.00	0.8243 ± 0.0019	0.8225	0.4549
0.10	0.8367 ± 0.0020	0.8349	0.4431
0.20	0.8462 ± 0.0020	0.8451	0.4321
0.30	0.8555 ± 0.0020	0.8553	0.4210
0.40	0.8658 ± 0.0020	0.8655	0.4104
0.50	0.8736 ± 0.0020	0.8739	0.4015
0.60	0.8824 ± 0.0022	0.8828	0.3921
0.70	0.8910 ± 0.0021	0.8913	0.3831
0.80	0.9017 ± 0.0021	0.9009	0.3698
0.90	$\mathbf{0.9168 \pm 0.0022}$	**0.9152**	**0.3565**
1.00	0.9094 ± 0.0020	0.9080	0.3620

Note: The bold values indicate best accuracy result.

The results of the Swish-Gumbel model were compared with other standard activation functions. Table 18 presents the mean test accuracy and standard deviation values derived from the 5-fold cross-validation.

As presented in Table 18, the Swish-Gumbel function achieved a mean test accuracy of 0.9168 ± 0.0022. This result is significantly higher than the standard ReLU function (0.8537 ± 0.0024) and also shows a distinct improvement over Mish (0.8725 ± 0.0029) and Swish (0.8932 ± 0.0019). The relatively low standard deviation ($\sigma = 0.0022$) indicates that the model's performance remains stable across different data folds. The class-wise precision, recall, and F1-score values for the Beans dataset are presented in Table 19. The results indicate consistently high performance across all classes, demonstrating the model's ability to effectively distinguish between different plant disease categories.

The macro-average and weighted-average values further confirm that the model achieves balanced classification performance without significant bias toward any specific class. These findings highlight the robustness of the proposed activation function in real-world agricultural image classification tasks.

These findings further support the generalization capability of the proposed activation function, demonstrating its effectiveness not only on standard benchmark datasets such as MNIST and CIFAR-10, but also on real-world agricultural image classification tasks. To further support this comparison, the test accuracy trends across different activation functions are demonstrated in Figure 6.

The confusion matrix for the optimal Swish-Gumbel model ($\beta = 0.9$) on the Beans dataset is presented in Table 20.

6 | Discussion

The primary objective of this study was to provide a comprehensive evaluation of the proposed Swish-Gumbel activation

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 18 | Comparison of activation functions on Beans dataset (mean ± SD over 5 folds).

Activation function	Parameter	Test accuracy (mean ± SD)	Mean validation accuracy	Test loss
Gumbel	—	0.7683 ± 0.0068	0.7635	0.4846
LC (1)	$\alpha = 0.9$	0.8902 ± 0.0018	0.8885	0.3756
LC (2)	$\alpha = 0.9$	0.8835 ± 0.0018	0.8815	0.3785
LC (3)	$\alpha = 0.7$	0.8824 ± 0.0018	0.8805	0.3805
LC (4)	$\beta = 0.8$	0.8582 ± 0.0018	0.8561	0.4075
LC (5)	$\alpha = 0.1$	0.8465 ± 0.0018	0.8425	0.4425
Mish	—	0.8725 ± 0.0029	0.8705	0.3932
Nipuna	—	0.8264 ± 0.0026	0.8264	0.4486
ReLU	—	0.8537 ± 0.0024	0.8529	0.4157
Sep	—	0.8412 ± 0.0025	0.8418	0.4323
Sigmoid	—	0.7950 ± 0.0016	0.7957	0.4642
Sigmoid-Gumbel	$\beta = 0.9$	0.8129 ± 0.0023	0.8092	0.4538
Swish	$\beta = 1.0$	0.8932 ± 0.0019	0.8936	0.3690
Swish-Gumbel	$\beta = 0.9$	$\mathbf{0.9168 \pm 0.0022}$	**0.9152**	**0.3533**

Note: The bold values indicate best accuracy result.

TABLE 19 | Class-based classification metrics for Swish-Gumbel on Beans dataset.

Class	Precision	Recall	F1-score
0	0.9302	0.9419	0.9360
1	0.9326	0.9535	0.9429
2	0.9443	0.9432	0.9437
Macro avg	0.9357	0.9462	0.9409
Weighted avg	0.9172	0.9168	0.9169

functions against widely adopted alternatives. The functions were tested in both multiplicative and linear combination variants across four distinct datasets: MNIST, CIFAR-10, Imagenette, and Beans.

The empirical results consistently demonstrated that the multiplicative form yields superior classification accuracy. In the MNIST dataset, the model reached a peak test accuracy of 99.12% with $\beta = 0.5$, outperforming all baseline activation functions. For CIFAR-10, the highest accuracy of 84.85% was recorded at $\beta = 0.9$, indicating a distinct improvement over standalone Swish and Gumbel functions. The Imagenette and Beans datasets further confirmed this strong generalization capability, achieving test accuracies of 69.20% ($\beta = 0.7$) and 91.68% ($\beta = 0.9$), respectively. Additionally, the balanced class-wise F1-scores across all datasets underline the stability of the proposed function in handling diverse feature spaces.

The variation in the optimal β parameter across these datasets highlights the flexibility of the Swish-Gumbel function. For high-resolution, complex natural images like Imagenette, a moderately tuned β (e.g., 0.7) provides a smoother gradient flow, preventing noise amplification while extracting broad spatial features. Conversely, for domain-specific tasks such as the Beans dataset, where recognizing fine leaf textures and sharp disease boundaries is crucial, a higher β (e.g., 0.9) yields a slightly sharper activation response, proving more effective for capturing fine textural differences. Additionally, the balanced class-wise F1-scores across all datasets underline the stability of the proposed function in handling diverse feature spaces.

Initially, we hypothesized that the linear combination variants might reduce functional complexity and thereby enhance overall performance. However, while some linear combinations yielded competitive accuracy, the comprehensive results demonstrated that the multiplicative form consistently outperformed all LC variants and established functions (e.g., ReLU, Swish, and Mish). This superiority stems from the multiplicative formulation's ability to maintain more stable gradients in positive input regions, which directly improves feature extraction efficacy during backpropagation.

Regarding computational efficiency, the proposed Swish-Gumbel function introduces a strictly bounded theoretical overhead. While standard ReLU requires a single scalar operation per neuron and Swish requires four, Swish-Gumbel demands approximately eight scalar operations due to its double-exponential formulation. However, since the overall computational cost of deep networks is overwhelmingly dominated by the $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$ complexity of matrix multiplications in convolutional layers, this $\mathcal{O}(N)$ element-wise addition contributes less than 0.001% to the total FLOPs. Furthermore, the function introduces 0 additional trainable parameters. Empirical measurements extracted from our cross-validation logs confirmed that modern hardware vectorization completely absorbs this scalar overhead. Across all architectures tested—from the lightweight custom CNN ($\sim 8 - 12$ ms/step) to the heavier MobileNet ($\sim$ 45 ms/step) and ResNet-50 ($\sim$ 115 ms/step)—the training and inference execution times for Swish-Gumbel remained virtually identical to those of the baseline ReLU and standalone Swish models.

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70022 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

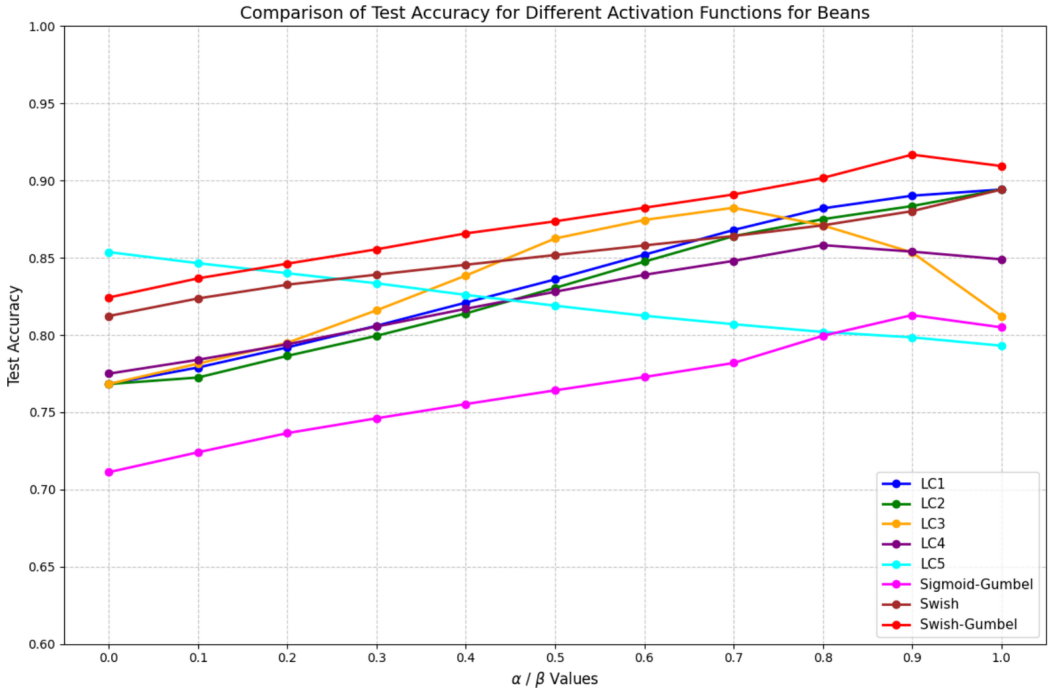


FIGURE 6 | Comparison of test accuracy for different activation functions for BEANS.

TABLE 20 | Confusion matrix for the Swish-Gumbel ($\beta = 0.9$) on Beans dataset.

Class	0	1	2
0	81	3	2
1	2	82	2
2	3	2	83

Note: The bold values indicate best accuracy result.

7 | Conclusions

This paper introduces the Swish-Gumbel family of activation functions, developed through both multiplicative and linear combination approaches. Extensive evaluations across various image classification benchmarks reveal that the multiplicative variant consistently achieves superior accuracy and stability compared to state-of-the-art functions, including ReLU, Swish, and Mish.

Although linear combinations were also explored, the multiplicative interaction provides a definitive edge in complex feature extraction. By integrating the probabilistic characteristics of the Gumbel formulation with the smooth, non-monotonic structure of Swish, this multiplicative mechanism effectively mitigates gradient degradation in positive domains without adding significant computational overhead.

Integrating the probabilistic characteristics of the Gumbel formulation with the smooth, non-monotonic structure of Swish, we obtain a hybrid activation mechanism that effectively mitigates gradient degradation in positive domains without adding significant computational overhead. This synergistic structure proves particularly advantageous for image data, where positive-valued inputs dominate, allowing for more stable gradient flows and enhanced discriminative learning.

Our findings bridge the gap between probabilistic models and continuous non-linear transformations, offering a highly competitive and easily implementable alternative to conventional activation strategies. The successful integration of the Swish-Gumbel function into different network architectures highlights its broad application potential, not only in computer vision but across diverse domains where robust representational learning is essential.

8 | Future Works

In this study, various classification scenarios were extensively investigated to evaluate the model's performance across diverse operational constraints. The experimental framework encompasses basic multi-class classification tasks and is situated within a supervised learning paradigm. To address complexities inherent in real-world data, the research explores fine-grained classification under challenging imbalanced data distribution scenarios. Furthermore, the methodology incorporates both offline classification and specialized approaches, such as zero-shot and transfer learning, to enhance generalization capabilities. The analysis is conducted on a whole-image basis, ultimately producing a hard classification output to ensure definitive decision-making. Other classification scenarios not addressed in our study will be covered in future research.

While the current evaluations on CNN, ResNet-50, and MobileNet architectures provide strong validation, the study is naturally limited to these specific topologies and medium-sized datasets. Future research should focus on evaluating the Swish-Gumbel function within more complex architectures,

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

such as Vision Transformers and EfficientNet, and on larger-scale datasets to further establish its scalability. Moreover, introducing an adaptive optimization mechanism to dynamically update the α and β parameters during training could unlock further performance gains.

Acknowledgments

This work has received full APC support by TÜBİTAK (The Scientific and Technological Research Council of Türkiye). This work was supported by the Ankara University Scientific Research Projects (BAP) Coordinatorship (Project No: FBA-2025-4190).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The MNIST dataset is available via the link: https://www.tensorflow.org/datasets/catalog/mnist?hl=en. The CIFAR-10 dataset is available via the link: https://www.cs.toronto.edu/~kriz/cifar.html. The Imagenette dataset is available via the link https://www.tensorflow.org/datasets/catalog/imagenette?hl=tr. The Beans dataset is available via the link https://www.tensorflow.org/datasets/catalog/beans?hl=tr.

References

1. D. Putra, M. Azmi, W. Purwanto, et al., "ANN Activation Function Comparative Study for Sinusoidal Data," *Journal of Physics: Conference Series* 2406 (2022): 12029.

2. Y. Wu and J. Feng, "Development and Application of Artificial Neural Network," *Wireless Personal Communications* 102 (2018): 1645–1656.

3. K. Sharifani and M. Amini, "Machine Learning and Deep Learning: A Review of Methods and Applications," *World Information Technology and Engineering Journal* 10 (2023): 3897–3904.

4. P. Xu, X. Ji, M. Li, and W. Lu, "Small Data Machine Learning in Materials Science," *npj Computational Materials* 9 (2023): 42.

5. S. Hussain, R. Atallah, A. Kamsin, and J. Hazarika, "Classification, Clustering and Association Rule Mining in Educational Datasets Using Data Mining Tools: A Case Study," in *Cybernetics and Algorithms in Intelligent Systems: Proceedings of 7th Computer Science on-Line Conference 2018, Volume 3 7* (Springer International Publishing, 2019), 196–211.

6. X. Yu, H. Lu, and Q. Liu, "Deep-Learning-Based Regression Model and Hyperspectral Imaging for Rapid Detection of Nitrogen Concentration in Oilseed Rape (*Brassica napus* L.) Leaf," *Chemometrics and Intelligent Laboratory Systems* 172 (2018): 188–193.

7. Z. Ebrahimi, M. Loni, M. Daneshtalab, and A. Gharehbaghi, "A Review on Deep Learning Methods for ECG Arrhythmia Classification," *Expert Systems With Applications: X* 7 (2020): 100033.

8. N. Prakash, A. Manconi, and S. Loew, "Mapping Landslides on EO Data: Performance of Deep Learning Models vs. Traditional Machine Learning Models," *Remote Sensing* 12 (2020): 346.

9. S. Zhong, J. Hu, X. Fan, X. Yu, and H. Zhang, "A Deep Neural Network Combined With Molecular Fingerprints (DNN-MF) to Develop Predictive Models for Hydroxyl Radical Rate Constants of Water Contaminants," *Journal of Hazardous Materials* 383 (2020): 121141.

10. C. Janiesch, P. Zschech, and K. Heinrich, "Machine Learning and Deep Learning," *Electronic Markets* 31 (2021): 685–695.

11. A. Jagtap and G. Karniadakis, "How Important Are Activation Functions in Regression and Classification? A Survey, Performance Comparison, and Future Directions," *Journal of Machine Learning for Modeling and Computing* 4 (2023): 21–75.

12. F. Manessi and A. Rozza, "Linear Combinations of Activation Functions," https://arxiv.org/pdf/1801.09403.pdf.

13. M. Kaytan, İ. B. Aydilek, C. Yeroğlu, and A. Karci, "Sigmoid-Gumbel: Yeni Bir Hibrit Aktivasyon Fonksiyonu," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi* 11, no. 1 (2022): 29–45, https://doi.org/10.17798/bitlisfen.990508.

14. M. Kaytan, İ. B. Aydilek, and C. Yeroğlu, "Gish: A Novel Activation Function for Image Classification," *Neural Computing & Applications* 35 (2023): 24259–24281, https://doi.org/10.1007/s00521-023-09035-5.

15. D. Saha, R. K. Jatoth, and Y. N. Reddy, "Sep: A New Nonlinear Activation Function for Biomedical Applications and Image Classification," in *2023 International Conference on Data Science and Network Security (ICDSNS)* (IEEE, 2023), 1–7, https://doi.org/10.1109/ICDSNS58469.2023.10244997.

16. Y. Jiang, J. Xie, and D. Zhang, "An Adaptive Offset Activation Function for CNN Image Classification Tasks," *Electronics* 11 (2022): 3799, https://doi.org/10.3390/electronics11223799.

17. R. K. Vasanthakumari, R. V. Nair, and V. G. Krishnappa, "Improved Learning by Using a Modified Activation Function of a Convolutional Neural Network in Multi-Spectral Image Classification," *Machine Learning with Applications* 14 (2023): 100502, https://doi.org/10.1016/j.mlwa.2023.100502.

18. A. A. Yahya, K. Liu, A. Hawbani, Y. Wang, and A. N. Hadi, "A Novel Image Classification Method Based on Residual Network, Inception, and Proposed Activation Function," *Sensors* 23 (2023): 2976, https://doi.org/10.3390/s23062976.

19. S. Kiliçarslan, K. Adem, and M. Çelik, "An Overview of the Activation Functions Used in Deep Learning Algorithms," *Journal of New Results in Science* 10, no. 3 (2021): 75–88, https://doi.org/10.54187/jnrs.1011739.

20. M. Zulhazmi, R. Azhary, and A. R. Ismail, "A Comparative Performance of Different Convolutional Neural Network Activation Functions on Image Classification," *International Journal on Perceptive and Cognitive Computing* 10, no. 2 (2024): 118–122, https://doi.org/10.31436/IJPCC.V10I2.490.

21. A. Rahman, L. He, and H. Wang, "Activation Function Optimization Scheme for Image Classification," *Knowledge-Based Systems* 305 (2024): 112502, https://doi.org/10.1016/j.knosys.2024.112502.

22. L. Parisi, C. D. Neagu, N. RaviChandran, R. Ma, and F. Campean, "Optimal Evolutionary Framework-Based Activation Function for Image Classification," *Knowledge-Based Systems* 299 (2024): 112025, https://doi.org/10.1016/J.KNOSYS.2024.112025.

23. K. W. Goh, S. Surono, M. Y. F. Afiatin, et al., "Comparison of Activation Functions in Convolutional Neural Network for Poisson Noisy Image Classification," *Emerging Science Journal* 8, no. 2 (2024): 592–602, https://doi.org/10.28991/ESJ-2024-08-02-014.

24. R. ZahediNasab and H. Mohseni, "Neuroevolutionary Based Convolutional Neural Network With Adaptive Activation Functions," *Neurocomputing* 381 (2020): 306–313, https://doi.org/10.1016/j.neucom.2019.11.090.

25. N. Ma, X. Zhang, M. Liu, and J. Sun, "Activate or Not: Learning Customized Activation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), 8028–8038, https://doi.org/10.1109/CVPR46437.2021.00794.

26. S. Al-Shoukry and Z. B. Musa, "An Overview on Effectiveness of Activation Functions in Processing Medical Images," *Al-Furat Journal of Innovations in Electronics and Computer Engineering* 3, no. 2 (2024): 163–180, https://doi.org/10.46649/fjiece.v3.2.13a.22.5.2024.

27. M. Karakurt, H. Saygılı, and A. Karcı, "Comparison of Activation Functions in the KarcıFANN Method," in *2024 8th International Artificial*

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70022 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Intelligence and Data Processing Symposium (IDAP) (IEEE, 2024), 1–6, https://doi.org/10.1109/IDAP64064.2024.10711149.

28. F. Nacar, A. Cakin, S. Dilek, S. Tosun, and K. Chakrabarty, "Neuron Grouping and Mapping Methods for 2D-Mesh NoC-Based DNN Accelerators," *Journal of Parallel and Distributed Computing* 193 (2024): 104949, https://doi.org/10.1016/j.jpdc.2024.104949.

29. P. Ramachandran, B. Zoph, and Q. V. Le Google Brain, "Searching for Activation Functions," in *6th International Conference on Learning Representations, ICLR 2018 — Workshop Track Proceedings* (OpenReview.net, 2017), https://arxiv.org/abs/1710.05941v2.

30. F. Farhadi, V. Partovi Nia, and A. Lodi, "Activation Adaptation in Neural Networks," in *International Conference on Pattern Recognition Applications and Methods*, vol. 1 (SciTePress – Science and Technology Publications, Lda, 2019), 249–257, https://doi.org/10.5220/0009175102490257.

31. Y. Seo, J. Kim, and U. Park, "Swish-T: Enhancing Swish Activation With Tanh Bias for Improved Neural Network Performance" (2024), https://arxiv.org/abs/2407.01012v3.

32. M. A. Mercioni and S. Holban, "A Brief Review of the Most Recent Activation Functions for Neural Networks," in *2023 17th International Conference on Engineering of Modern Electric Systems (EMES)* (IEEE, 2023), 1–4, https://doi.org/10.1109/EMES58375.2023.10171705.

33. P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions" (2017), arXiv:1710.05941v2.

34. E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization With Gumbel-Softmax" (2016), arXiv:1611.01144.

35. D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function," https://arxiv.org/abs/1908.08681.

36. G. Madhu, S. Kautish, K. A. Alnowibet, H. M. Zawbaa, and A. W. Mohamed, "NIPUNA: A Novel Optimizer Activation Function for Deep Neural Networks," *Axioms* 12 (2023): 246, https://doi.org/10.3390/axioms12030246.

37. Y. Kocak and G. Ü. Siray, "New Activation Functions for Single Layer Feedforward Neural Network," *Expert Systems with Applications* 164 (2021): 113977, https://doi.org/10.1016/j.eswa.2020.113977.

38. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016 (IEEE, 2015), 770–778, https://doi.org/10.1109/CVPR.2016.90.

39. P. Nagpal, S. A. Bhinge, and A. Shitole, "A Comparative Analysis of ResNet Architectures," in *2022 International Conference on SMART Generation Computing, Communication and Networking, SMART GENCON 2022* (IEEE, 2022), https://doi.org/10.1109/SMARTGENCON56628.2022.10083966.

40. J. Liu, "Face Recognition Technology Based on ResNet-50," *Applied and Computational Engineering* 39, no. 1 (2024): 160–165, https://doi.org/10.54254/2755-2721/39/20230593.

41. T. Szandala, "Benchmarking Comparison of Swish vs. Other Activation Functions on CIFAR-10 Imageset," in *Engineering in Dependability of Computer Systems and Networks. DepCoS-RELCOMEX 2019. Advances in Intelligent Systems and Computing*, vol. 987, ed. W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk (Springer, 2020), https://doi.org/10.1007/978-3-030-19501-4_49.

42. G. Bingham and R. Miikkulainen, "Discovering Parametric Activation Functions," *Neural Networks* 148 (2022): 48–65.

15320634, 2026, 14, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/cpe.70822 by Burhan Selçuk - Turkey Cochrane Evidence Aid , Wiley Online Library on [14/07/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License