Journal of Applied Engineering and Technological Science
Vol 6(2) 2025: 1254-1267

COMPARISON OF ACTIVATION FUNCTIONS IN FEATURE EXTRACTION LAYER USING SHARPENING FILTERS

Oktavia Citra Resmi Rachmawati[1*], Ali Ridho Barakbah[2], Tita Karlita[3]
Department of Information and Computer Engineering, Electronic Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia[123]
Email: citra.research@gmail.com

Received: 13 August 2024, Revised: 12 December 2024, Accepted: 18 January 2025
**Corresponding Author*

ABSTRACT
Activation functions are a critical component in the feature extraction layer of deep learning models, influencing their ability to identify patterns and extract meaningful features from input data. This study investigates the impact of five widely used activation functions—ReLU, SELU, ELU, sigmoid, and tanh—on convolutional neural network (CNN) performance when combined with sharpening filters for feature extraction. Using a custom-built CNN program module within the researchers' machine learning library, Analytical Libraries for Intelligent-computing (ALI), the performance of each activation function was evaluated by analyzing mean squared error (MSE) values obtained during the training process. The findings revealed that ReLU consistently outperformed other activation functions by achieving the lowest MSE values, making it the most effective choice for feature extraction tasks using sharpening filters. This study provides practical and theoretical insights, highlighting the significance of selecting suitable activation functions to enhance CNN performance. These findings contribute to optimizing CNN architectures, offering a valuable reference for future work in image processing and other machine-learning applications that rely on feature extraction layers. Additionally, this research underscores the importance of activation function selection as a fundamental consideration in deep learning model design.
Keywords: *Convolutional Neural Network, Activation Function, Feature Extraction, Sharpening Filter, Image Processing, Deep Learning*

1. Introduction

In recent years, deep learning as an essential modeling technique in artificial intelligence has achieved cutting-edge performance in solving numerous machine learning tasks (Liu et al., 2019), such as voice analysis (Khaskhoussy & Ayed, 2023), pattern recognition (K. Jermsittiparsert et al., 2020), and object classification (Vaidya & Paunwala, 2019). Researchers have introduced various strategies to enhance the efficiency of neural network training, focusing on selecting optimal activation functions. As inputs pass through neurons, they are multiplied by weights, and the weighted sum—along with a bias term—is processed by an activation function. This function determines each neuron's output, ultimately shaping the network's predictions. Activation functions, either linear or non-linear, influence the network's ability to model complex relationships (Wong et al., 2022). Rectified linear units (ReLU) are standard activation functions used to develop modern intelligent applications today (Eckle & Schmidt-Hieber, 2019). Neural networks place the activation function as the central decision-making unit (Szandała, 2020), and therefore the activation function is a crucial component to be considered (Xiangyang et al., 2023).

Several studies compared the use of activation functions for case studies such as the production and consumption prediction system for electricity power usage using a multilayer neural network (Salam et al., 2021). It was identified that different activation functions have varied impacts on the performance of neural networks, especially in tasks involving image data processing. This paper investigates how activation functions perform within the feature extraction layer of convolutional neural networks (CNNs), focusing on their role in image dataset processing. CNNs rely on activation functions to maintain essential features, filter out redundant data, and effectively map extracted features (Qiumei et al., 2019). This study provides a comparative analysis of activation functions in CNN's feature extraction layer, contributing to better activation function choices and, ultimately, more efficient deep learning models for image analysis tasks.

Feature extraction in CNNs benefits significantly from sharpening filters, which enhance essential details such as edges and contours within images. By calculating the gradient magnitude, these filters highlight edges where the pixel intensity changes significantly, aiding CNNs in identifying informative features for numerous tasks, including object recognition. However, sharpening filters must be applied carefully as they can also amplify noise or create unwanted edge effects. When fine-tuned for specific image types, sharpening filters can significantly enhance CNN performance by clarifying crucial structural details (Pham, 2022). The capability of sharpening filters to enhance critical structural details forms the basis for this paper's focus on comparing activation functions in the feature extraction layer of CNNs using sharpening filters (Lin et al., 2021).

This paper offers a unique contribution through the development of Analytical Libraries for Intelligent-computing (ALI), a Java-based software library designed to streamline research in machine learning. ALI provides an integrated module that simplifies the process of running deep learning experiments without requiring researchers to build programs from scratch. It supports the comparative analysis of activation functions within feature extraction layers, a key focus of this study. Additionally, ALI generates models that can be seamlessly embedded into desktop, mobile, and web applications developed in Java. ALI also includes several machine learning algorithms to address diverse problem-solving needs such as automatic clustering (Shodiq et al., 2019), hierarchical K-Means (Ramadijanti et al., 2018), independent clustering, k-nearest neighbors (Subhan et al., 2017), and neural networks (Shodiq et al., 2017). By integrating these features, ALI is a valuable tool for advancing research and development in machine learning.

Consequently, this paper provides several contributions, including: (1) a qualitative analysis of the effect of activation functions on CNN performance in image-processing tasks, and (2) a quantitative analysis involving evaluating the performance of activation functions against computational loads that impact execution speed.

This paper presents the experimental study using similar CNN architecture to try different activation functions. The tested activation functions feature certain limitations as follows: (1) The examined activation functions are used in the same feature extraction layer, (2) The settings in the feature extraction layer are not replaced when a different activation function is utilized, and (3) This study aims to show that one of all these functions works with a low MSE value with images in the input vector and convolution layers.

2. Literature Review

The researchers reviewed several previous studies related to activation function comparison on artificial neural networks with the aim of establishing research position and demonstrating the differences.

Abdulwahed et al. (2021) investigated the performance of various activation functions in multilayer neural network models for electricity power prediction systems. The study evaluated activation functions such as sigmoid, hyperbolic tangent, ReLU, leaky ReLU, ELU, and swish using the root mean squared error (RMSE) metric. However, the study lacks a comprehensive statistical descriptive analysis of the models' performance, such as minimum, maximum, and mean values, which are crucial for a deeper understanding of activation function behavior during model training (Salam et al., 2020).

Similarly, Lee (2020) compared activation functions in a reinforcement learning-based neural network model for a 2D racing game. While this study observed activation functions such as ReLU6, leaky ReLU, and CReLU using total reward as a performance metric, it did not include a comparison of computational efficiency. Understanding the computational impact of activation functions is essential in scenarios where processing time is a critical factor.

The existing literature on activation functions in neural networks highlights their critical role in model performance, particularly in feature extraction and optimization tasks. Several studies examined the effects of popular activation functions such as ReLU, sigmoid, tanh and their variants on various neural network architectures. In evaluating performance, these studies typically focused on metrics such as accuracy, convergence rates, and error minimization. However, their limitation necessitates a detailed comparative analysis of computational time and descriptive statistics during the training process, such as minimum, maximum, and mean error values. Addressing these gaps would provide a more holistic understanding of how activation functions impact efficiency and performance.

In contrast, this study focuses on convolutional layers in a CNN model. It compares activation functions based on mean squared error (MSE) values and visualizes the learning process through line charts and observations of statistical descriptions such as minimum, maximum, and mean values. Furthermore, this study incorporates an analysis of computational time, providing a more holistic evaluation of activation function performance. By addressing the gaps in prior studies, such as the need for more statistical descriptive analysis and computational comparisons, this research offers a novel contribution to the field. It enhances the understanding of activation function performance in convolutional layers. In this paper, the researchers compare the performance of convolutional layers for feature extraction by measuring the MSE values of the examined activation functions. This study examines five activation functions: ReLu, SeLu, ELU, sigmoid, and tanh. The researchers visualize the learning process through multiple line charts and observe statistical descriptions to support the comparative analysis of activation functions. In addition, the researchers also pay attention to the difference in computation time between the activation functions in the convolution layer of the learning process.

3. Research Methods

This research employed a systematic three-stage methodology to explore the performance of different activation functions in CNNs with the aim of providing a comprehensive analysis that compares the error of activation functions and evaluates their computational efficiency and statistical properties. This structured approach ensured the reliability of the findings and highlighted the practical implications of activation function selection for feature extraction tasks in deep learning.

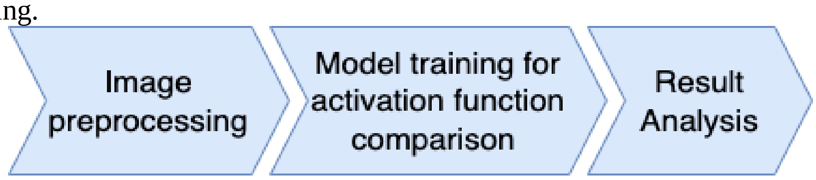


Fig. 1. Main Stages of Research

The first stage involved image preprocessing, in which input data were prepared to enhance their quality and compatibility with the CNN model. This process included applying a sharpening filter to emphasize critical image features, resizing the images to ensure uniformity, and normalizing pixel values to stabilize the training process. These steps were essential for reducing noise and highlighting relevant patterns in the data, thus providing a strong foundation for model training (Rachmawati & Darmawan, 2024).

The study focused on model training for activation function comparison in the second stage. The five activation functions—ReLU, SeLU, ELU, sigmoid, and tanh—were implemented within the feature extraction layers of a CNN model. Each activation function was tested to observe its impact on model performance, measured using the MSE metric. Additionally, the computational time required for training with each activation function was recorded to evaluate their efficiency. This stage provided critical insights into how activation functions influence the learning process regarding accuracy and computational resources.

The final stage, result analysis, was dedicated to interpreting the findings. The performance of each activation function was visualized using line charts, and detailed statistical descriptions—such as the minimum, maximum, and mean MSE values—were presented. In addition, the differences in computational time were analyzed to highlight the trade-offs between accuracy and efficiency for each activation function. This thorough analysis identified the strengths and weaknesses of the activation functions and offered practical guidance for selecting the most suitable function for similar tasks.

This study adopted a methodical approach to uncover the complexities of activation function performance in CNNs. Integrating image preprocessing, comparative model training, and detailed result analysis provided a holistic evaluation of the activation functions in question. The findings contribute to the growing body of knowledge in deep learning, offering valuable insights for researchers and practitioners aiming to optimize feature extraction in convolutional neural networks.

Activation Functions

CNNs are network structures with multi-layers that can be trained and feature three primary stages: feature extraction, nonlinear activation, and downsampling (Qiumei et al., 2019). Convolution layers equipped with activation functions have a particular threshold value to reduce the contribution value of less relevant features or increase the contribution value of significant features, making pattern recognition easier (Chegeni et al., 2022).

Neural networks of any type are composed of layers of neurons containing mathematical functions that combine inputs with weights that amplify or dampen the inputs to give importance to the corresponding inputs (Lee, 2020). The output of a neuron is a weighted sum of its inputs plus a bias. It passes through the activation function to establish whether and to what extent the input should progress further through the network for classification (Lee, 2020). The researchers examined several commonly used activation functions, including ReLu, SeLu, ELU, sigmoid, and tanh.

Table 1 - Range of Activation Function

Activation Function	Range
ReLU	0, +∞
SELU	-5, 10
ELU	-α, +∞
Sigmoid	-1, 1
Tanh	0, 1

ReLU is an activation function that preserves the value sent from the feedforward neuron if it is a positive number and replaces 0 if it is a negative number (Liu et al., 2019). The derived function changes the value sent from the backpropagation neuron to 1 if the value is positive and vice versa (Job et al., 2022). Although ReLU offers advantages such as easier alleviation of vanishing gradients and fast execution processes due to formula simplicity, the network experiences the problem of excess neuron death, causing many features to be lost due to values sent from feedforward/backpropagation neurons dominating existing features as many values are mapped to 0 (Hu et al., 2019).

$$f(x) = max(0, x) \tag{1}$$
$$f'(x) = \{0,\ x < 0\ 1,\ x > 0 \tag{2}$$

Scaled exponential linear unit (SELU) is an activation function that preserves the value sent from the feedforward neuron if the value is positive and replaces the exponential linear formulation with additional scaling parameters (Liu et al., 2019). SELU includes the development of the exponential linear unit (ELU) activation function so that SELU can be identical to ELU (Liu et al., 2019). SELU relies on weight initialization if insufficient initialization results in unstable network performance. However, if the initialization of weights is appropriate, it facilitates weight regulation and strengthen learning because of its self-normalizing nature (Sakketou & Ampazis, 2019).

$$f(x) = \lambda\{\alpha(e^x - 1),\ x < 0\ x,\ x \geq 0 \tag{3}$$
$$f'(x) = \{\lambda,\ x > 0\ \lambda\alpha e^x,\ x \leq 0 \tag{4}$$

Exponential linear unit (ELU) is an activation type with an alpha variable that controls the output value, especially for negative values from the input of feedforward neurons. ELU helps alleviate the problem of vanishing gradients by capturing both linear and exponential values optimized during the backpropagation process (Narmadha & Vijayakumar, 2019). By specifying values in the alpha variable ranging from 0 to 1, ELU applies normalization across all network layers without the need for additional normalization layers (Cococcioni et al., 2020).

$$f(x) = \{\alpha(e^x - 1),\ x < 0\ x,\ x \geq 0 \tag{5}$$
$$f'(x) = \{f(x) + \alpha,\ x < 0\ 1,\ x \geq 0 \tag{6}$$

Sigmoid is an activation function that ranges from 0 to 1 and is commonly used for probability prediction models as output values. Thus, sigmoid is extremely popular in solving binary classification problems because it is an essential example of complex operations using division and exponential (Shatravin et al., 2022). In addition, sigmoid has a distributed implementation that effectively approximates other functions.

$$f(x) = \frac{1}{1+e^{-z}} \tag{7}$$
$$f'(x) = f(x)\big(1 - f(x)\big) \tag{8}$$

Tanh is a type of activation that produces an output value with a range of -1 to 1, where a negative input value gives an output close to -1, a positive input value gives an output close to 1, and a zero input value produces a zero output (Kalyanam & Katkoori, 2023). Because tanh produces a value of 0, it produces dead neurons such as ReLU during computational runs. Tanh is often used in recurrent neural networks (RNNs) as it produces periodic responses at time sequences (Nguyen et al., 2021).

In this study, the researchers compared the activation functions ReLU, SeLU, ELU, sigmoid, and tanh within the feature extraction layers of a CNN model to address specific gaps and challenges in deep learning research. These activation functions were selected since they represent diverse behaviors and characteristics, making them suitable for evaluating the trade-offs between accuracy, computational efficiency, and stability in the learning process.

ReLU (rectified linear unit) and its variants, SeLU (scaled exponential linear unit) and ELU (exponential linear unit), are prevalent in deep learning due to their ability to mitigate the vanishing gradient problem, which is critical for training deep networks. ReLU is computationally efficient and promotes sparse activation (Ismail et al., 2023), while SeLU and ELU offer additional advantages in stabilizing the learning process (Pappas et al., 2023), especially in cases where negative outputs are significant. However, their relative performance varies depending on the dataset and architecture, necessitating a comparative analysis.

On the other hand, sigmoid and tanh are classical activation functions widely used in earlier neural network models. Sigmoid maps input values to a range between 0 and 1, which is helpful for probabilistic interpretations but suffers from vanishing gradient issues (Chattopadhyay & Gayen, 2023). Tanh, an extension of sigmoid, maps values to a range between -1 and 1, providing better convergence in certain cases due to its symmetric nature (Liu et al., 2023). Despite their limitations, these functions remain relevant for understanding the trade-offs in model performance.

Dataset

The researchers used image datasets published on the Kaggle site for easier binary or multiple image classification[1]. The image datasets featured two main folders, namely training and test, making it easier to separate data. Subsequently, there were three sub-folders of each main folder with similar titles: empty, good, and crack. The researchers set the balance from the original dataset number by making a 90:10 comparison for the training and test data. In addition, the researchers performed an undersampling technique for each class's data to ensure that no class has the most members because the researchers were still testing the reliability of the deep learning library developed under ideal data conditions.

[1] https://www.kaggle.com/datasets/frankpereny/broken-eggs

Table 2 - Image Dataset Distribution

Subset	Label	Amount	Total
Training	Crack	90	270
	Good	90	
	Empty	90	
Test	Crack	10	30
	Good	10	
	Empty	10	

Before processing the image dataset into a convolutional neural network's feature extraction layer, the researchers preprocessed the dataset by cropping the center-focused image and resizing it to 150x150 pixels. Grayscale was used as the color mode for numerical values for further processing. The researchers only paid attention to the contour of the egg crack line based on the visible contrast without requiring color review, thus requiring less computational power while processing the dataset as fewer mathematical operations were performed (Zeger et al., 2021).

(a)

(b)

Fig. 2. Difference Between Original And Grayscale Image Before Entering Extraction Feature Layer In CNN Model

The researchers normalized the data against grayscale values that had been converted into numerical arrays to keep the weight values in the network within a linear range. This reduced numerical differences in the pixel range (e.g., 0-255), aiding training stability and convergence and simplifying the process of training algorithms that are sensitive to scale differences (Herwanto et al., 2021).

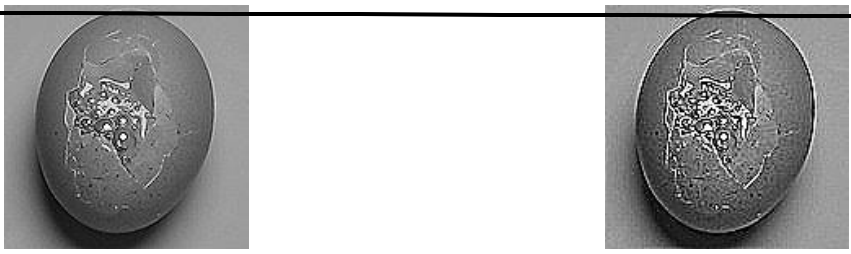
$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{11}$$

The researchers processed the image dataset using sharpening filters with two levels: light and heavy. Sharpening filters, commonly known as high-pass filters, are one of the techniques in image processing used to increase the clarity of image details by strengthening the contrast around the edges of objects in the image to make them appear sharper (Bogdan et al., 2024).

Table 3 - Matrix of Sharpening Filters

Sharpening Filter	Matrix
Light	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Heavy	$\begin{bmatrix} -1 & -2 & -1 \\ -2 & 13 & -2 \\ -1 & -2 & -1 \end{bmatrix}$

Light sharpening refers to the contrast around the edges of objects in the image, which does not appear sharper than heavy sharpening. The sharpness of the filter can be adjusted to the needs of processing images with a specific purpose as excessive use of sharpening filters causes increased noise when the model learns the pattern (Demir & Kaplan, 2023). The use of sharpening filters in the convolution layer intends to improve the network's ability to identify and retrieve essential features of the image, which can then be used for tasks such as classification, object detection, or segmentation (Choi et al., 2023).



(a) (b)

Fig. 3. Difference Between Light And Heavy Sharpening Filter On An Image Sample

CNN Module

In this activation function comparison, the researchers developed and utilized their deep learning library. In the previous development iteration, the researchers developed and published a deep neural network module on a site with an affiliate's domain. In the current development iteration, the researchers were developing a CNN module that was part of their machine-learning library. The researchers developed the CNN module to make it easier for other researchers and practitioners to run experiments to find the best model architecture to solve their problems.

The researchers developed the CNN module with several concepts, including the exploitation of design patterns for the code interface for users' ease of understanding. To guarantee correctness of functionalities, the researchers performed unit testing before publishing. This study also showed that the researchers' deep learning library supports the execution of experiments related to selecting hyperparameters suitable for the case study. Table 4 below outlines the functionalities that the researchers provided in the CNN module that the researchers were developing.

Table 4 - Functionalities of Researchers' CNN Module

Layer	Class
Feature Extraction	• Convolution1D • Convolution2D
Downsampling	• Subsample1D • Subsample2D • Globalsample1D • Globalsample2D • Flatten
Support	• Input1D • Input2D • Padding1D • Padding2D

The researchers provided a CNN module for processing one-dimensional data, such as text, and two-dimensional data, such as images. However, CNN modules for processing three-dimensional data, such as 3D images, have not been developed. To meet research needs, the development of the researchers' machine learning library will continue as software products require updates and improvements over time to correct errors and enhance features.

4. Results and Discussion

Based on the limitations stated in the introduction, the researchers compared the activation functions of the convolutional layers with hyperparameters. Table 5 details the convolutional layers tested in the study.

Table 5. Parameters in Feature Extraction Layer

Parameter	Value
Class	Convolution2D
Filter	Light and Heavy Sharpening (3x3)
Stride	1
Dilation	1
Learning rate	1.0
Metric	Mean Squared Error
Epoch	20

Gradient descent is a concept used in training neural networks to find the slightest error when updating the weights of each feature to achieve better performance (Goodfellow et al., 2016). The gradient descent process starts with a random initialization of weights applied to the training data, and the resulting prediction is compared with the actual label to calculate the error (Tian et al., 2023). The influence of each weight on the error is subsequently calculated and adjusted in proportion to the gradient using the learning rate to reduce the error gradually (Suganya & Sasipraba, 2023). This process is repeated in various iterations (epochs) until the model performs sufficiently.

The researchers balanced the distribution of the number of classes and normalized the numerical values of the data to ensure that the training of convolution models using either heavy or light sharpening filters with activation functions, either SELU, ReLU, ELU, sigmoid or tanh, converges the training and test data. In addition, the convolution layer for sharpness feature extraction can recognize patterns from objects in the image dataset.

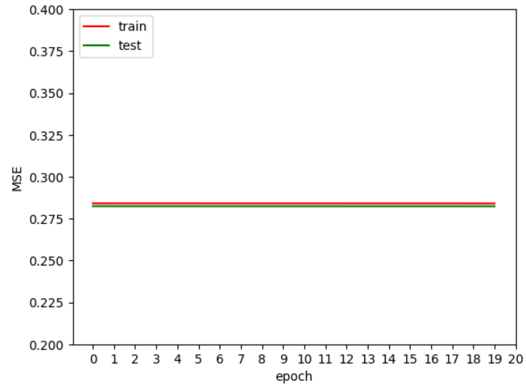
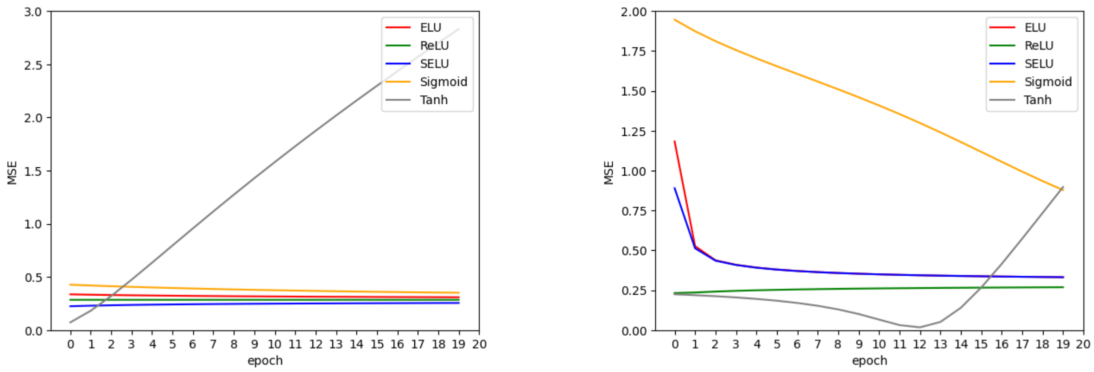


Fig. 4. Training History Of Convolution Layer With Sharpening Filter Using Relu Activation Function

The researchers selected MSE as the evaluation method of the model learning process in this case study because MSE provides a continuous measurement of the difference between the predicted and actual values, which helps the model to optimize for better convergence to produce a feature representation that matches the original image as much as possible. Thus, MSE is a metric sensitive to major or minor differences in the model's error in capturing essential features in the image (Ye & He, 2024).

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \tilde{y}_i)^2 \quad (12)$$

The researchers utilized multiple line charts to enable easier comparison of the movement patterns of the activation functions during the learning period represented in the same epoch sequence. The convolution layer using the sharpening filter showed different fluctuations among the five tested activation functions. Subsequently, the researchers gained knowledge regarding the stable activation function in minimizing the small error value in the feature extraction process to sharpen the contours of the objects in the image.



(a) Light Sharpening Filter (b) Heavy Sharpening Filter

Fig. 5. Moving MSE Of Activation Functions In Feature Extraction Layer Using Sharpening Filter During Training Process

Based on Fig. 5 (a), tanh is the most prominent activation function compared to other activation functions because there is a sharp increase in MSE value from the beginning to the end of learning. Meanwhile, other activation functions demonstrate MSE values that tend to decrease over time. Tanh is widely used in RNN models, various statistical analyses, and mathematical models that assume asymmetric and sharp distributions. Thus, using tanh in the feature extraction layer using the light sharpening filter is unsuitable because it cannot create sharp contours from a matrix containing values not too far from 0. Meanwhile, the convolution layer for feature extraction using the light sharpening filter equipped with ReLU, ELU, or SELU activation functions has close MSE values because the three activation functions contain similar basis, namely conditional mechanism of less than 0 and more than 0 being processed differently.

Based on Fig. 5 (b), the movement of convolutional layer learning for feature extraction using the heavy sharpening filter with the tanh activation function appears unstable because a decrease in MSE value continues to increase MSE value drastically. However, the tanh function obtains the lowest MSE value compared to other activation functions. Meanwhile, the convolutional layer for feature extraction using a heavy sharpening filter with the sigmoid activation function has the highest MSE value but gradually decreases as the learning process progresses. On the other hand, the convolutional layer for feature extraction using the heavy sharpening filter with the ReLU activation function has a low MSE value but increases as the learning process progresses. The MSE movement of the convolutional layer for feature extraction using the heavy sharpening filter with the ELU and SELU activation functions has a different MSE value at the beginning. However, as the learning process progresses, the MSE values are very close because the formulas are almost the same due to the SELU function being a development of the ELU function.

Using a sharpening filter, the researchers analyzed the performance comparison between utilizing activation functions on the convolutional layer by understanding the descriptive statistics of the learning processes (Yan et al., 2022). Descriptive statistics enables easy observation of the difference between the performance of activation functions. The researchers used three parameters in the descriptive statistics in this observation, including min, max, and mean, enabling the determination of the smallest MSE value, the highest MSE value, and the average MSE value calculated based on the learning period performed by the convolution layer to extract features using the sharpening filter.

Table 6 - Statistic Descriptive for MSE Comparison of Activation Function Impact in Feature Extraction Layer

Sharpening Filter	Activation Function	Statistic Descriptive		
		Min	Max	Mean
Light	ELU	0.308251	0.335505	0.318057
	ReLU	0.284030	0.284082	0.284056
	SELU	0.223798	0.254559	0.244833
	Sigmoid	0.350463	0.426229	0.380470
	Tanh	0.071863	2.830942	1.471629
Heavy	ELU	0.331237	1.182580	0.407933
	ReLU	0.231934	0.268378	0.256704
	SELU	0.331171	0.889216	0.392233
	Sigmoid	0.877779	1.945771	1.416613
	Tanh	0.016775	0.896488	0.248847

Table 6 demonstrates that tanh has a significant difference in MSE value between the lower and upper bounds compared to other activation functions to be applied to the convolution layer using light and heavy sharpening filters. Tanh could be an activation function that optimizes learning in convolutional layers requiring contour sharpness to object lines in the image. However, tanh can also reduce learning performance because it fails to bring the feature prediction value closer to the target value.

Meanwhile, ReLU has the closest difference in MSE value between the lower and upper bounds compared to other activation functions to be applied to the convolutional layer using light and heavy sharpening filters. The use of ReLU in CNN models stabilizes the learning process because it is elementary in its calculation. By replacing negative values with zero and maintaining positive values unchanged, model convergence can be achieved in the early stages of training, and higher accuracy can be obtained with fewer iterations (epochs).

Hyperparameter is one factor that affects the length of computation time in the learning process of deep learning models, in addition to dataset size and resources. The activation function in the learning process affects the computation time since it is used to calculate the activation of each node in the artificial neural network, which must be evaluated repeatedly. Activation functions with more complex mathematical formulas in their calculations will significantly impact computation time as they have more effort in mathematical operations. Therefore, the researchers also observed the computation time taken by the convolution layer using the predefined activation functions. During this comparison experiment, the researchers utilized a MacOS operating system device, and a Java programming language IDE called IntelliJ Idea.

Table 7 - Comparison of Computation Time

Activation Function	Sharpening Filter	
	Light (seconds)	Heavy (seconds)
ELU	197.974	179.694
ReLU	182.912	262.877
SELU	205.612	264.940
Sigmoid	229.402	178.534
Tanh	205.883	204.843

Table 7 shows that the pruning of the image dataset in the convolutional layer using the light sharpening filter with the ReLU activation function takes the least time, and the sigmoid activation function takes the longest time. The image dataset processing on the convolutional layer using a heavy sharpening filter with the sigmoid activation function takes the least time, and the SELU activation function takes the longest. Computation time cannot be used as a complete benchmark to determine the activation function that is the fastest in processing as the resources used are of different types and inequality in usage time (for instance, running a deep learning model at the same time as running other programs so that it divides memory).

Three findings were identified through the comparison experiment conducted as follows: (1) Improved performance with increasing epochs does not always lead to a reduced MSE value, as demonstrated in Fig. 5, (2) Based on Table 5 and Fig. 5 (b), tanh is not only utilized for RNN but also feature extraction in CNN, and can even produce a much smaller MSE value than ReLU, but the fluctuation of the learning process could be more stable, (3) SELU is an activation function that works optimally for feature extraction to sharpen the contours finely, as evidenced by Fig. 5 (b).

Discussion

The findings in this study reveal important insights into the performance of various activation functions (ReLU, SELU, ELU, sigmoid, and tanh) within convolutional layers for feature extraction using sharpening filters. These insights build on and expand existing knowledge in the neural networks field by highlighting the strengths and limitations of each activation function in different scenarios. To provide a deeper understanding of these findings, this discussion compares the results with those from relevant prior research and explores their broader implications for feature extraction tasks in CNNs.

Based on Table 6, tanh demonstrates a significantly more extensive range of MSE values than other activation functions. This fluctuation indicates that while tanh achieves a lower MSE in some instances, its instability may hinder its practical application in tasks requiring consistent performance. The sharp increase and subsequent fluctuations in MSE values observed in Fig. 5 further support this, suggesting that tanh's saturation effects contribute to its sensitivity to small changes in input values. These results align with existing studies that indicate tanh's suitability for RNNs due to its ability to handle sequential data but challenge the notion of its optimal use in CNNs for feature extraction. However, this study’s findings also indicate that tanh can be leveraged for specific feature extraction tasks where achieving delicate contours is more critical than ensuring stability.

Conversely, ReLU consistently exhibited stable MSE performance across the training period, as shown in Figures 4 and 5. This stability stems from ReLU's simplicity, where negative values are replaced by zero, and positive values remain unchanged, leading to efficient gradient flow and faster convergence. These results align with prior research highlighting ReLU's effectiveness in deep learning models, particularly its computational efficiency and ability to avoid the vanishing gradient problem. The minimal difference in MSE values between the lower and upper bounds also underscores ReLU's reliability, making it a strong candidate for tasks requiring consistent feature representation.

SELU and ELU, as shown in Table 6 and Figure 4, demonstrated comparable performance in terms of MSE, especially during the later stages of training. SELU's slightly improved performance over ELU can be attributed to its self-normalizing property, which ensures stable learning by maintaining mean activations close to zero and unit variance. This property becomes particularly beneficial in feature extraction tasks involving sharpening filters, where maintaining consistent activation is critical for capturing intricate details in image data. These findings align with theoretical expectations but extend existing knowledge by demonstrating SELU's specific advantage in enhancing contour sharpness in CNNs.

While historically significant, sigmoid exhibited the highest MSE values and the slowest computational time, particularly when paired with light sharpened filters, as indicated in Table 7. This underperformance can be attributed to sigmoid's tendency to saturate, leading to vanishing gradients and slower convergence. Despite its limitations, sigmoid may still find niche applications where its probabilistic output range between 0 and 1 is necessary, though other activation functions in feature extraction tasks generally outperform it.

The computational time analysis further highlights the trade-offs between activation function complexity and performance. For instance, ReLU's straightforward mathematical formulation allows for faster computation, while SELU's additional operations result in longer processing times. These findings emphasize the importance of considering computational efficiency alongside accuracy when selecting activation functions, particularly for large-scale real-world applications.

This study provides novel insights compared to prior research by juxtaposing performance metrics such as MSE and computational time with theoretical properties of activation functions. For example, while Lee (2020) demonstrated the effectiveness of ReLU6 in reinforcement learning tasks, this study's findings show that ReLU and its variants also excel in feature extraction tasks within CNNs. Similarly, while Abdulwahed et al. (2021) highlighted the superiority of ReLU and Leaky ReLU in general neural network models, this study nuances these findings by demonstrating SELU's potential for specialized tasks such as sharpening object contours.

This study comprehensively evaluated five activation functions in CNNs for feature extraction using sharpening filters. Integrating performance metrics with theoretical analysis offers valuable guidance for researchers and practitioners in selecting activation functions tailored to specific tasks. The findings underscore the importance of balancing stability, accuracy, and computational efficiency, paving the way for further exploration into activation functions' role in optimizing deep learning architectures.

5. Conclusion

In this paper, the researchers conducted a comparative analysis of five activation functions (ReLU, SELU, ELU, sigmoid, and tanh) applied in the feature extraction layer of convolutional neural networks (CNNs) using sharpening filters. The findings indicate that the choice of activation function plays a crucial role in feature extraction performance. ReLU demonstrates superior performance in achieving the smallest error values across light and heavy sharpening filters. This result highlights the suitability of ReLU for feature extraction tasks requiring sharp contour detection and efficient convergence during training. On the other hand, tanh exhibits significant fluctuations in performance, underscoring the importance of activation function stability for consistent results. These findings contribute to the growing knowledge of neural network optimization by emphasizing the necessity of aligning activation function characteristics with specific filter requirements in feature extraction layers.

The implications of this research are twofold. The study provides practical guidance for selecting activation functions in real-world CNN applications, particularly for image-processing tasks requiring sharpening filters. Secondly, it lays the groundwork for further exploration into optimizing other architectural elements of CNNs, such as filter types, hyperparameter tuning, or layer configurations. In future research, the researchers aim to expand this study by evaluating the interaction of activation functions with other factors, such as loss functions and optimization algorithms, to develop robust and adaptable models for diverse artificial intelligence tasks. This work aims to bridge the gap between theoretical insights and practical implementations, contributing to more powerful and efficient neural network models for real-world applications.

Acknowledgement

The researchers appreciate the reviewers for their insightful feedback to improve the quality of this paper. Our gratitude goes to the Ministry of Education, Culture, Research, and Technology for supporting us in completing this paper. We wish to thank Electronic Engineering Polytechnic Institute of Surabaya (EEPIS) and the Department of Information and Computer Engineering for funding this research and providing assistance during this research. We would also like extend our gratitude to all teams who helped this research until its completion.

References

Bogdan, V., Bonchiş, C., & Orhei, C. (2024). An Image Sharpening Technique Based on Dilated Filters and 2D-DWT Image Fusion. *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. https://doi.org/10.5220/0012416600003660

Chattopadhyay, T., & Gayen, D. K. (2023). Sigmoid activation function generation by photonic artificial neuron (PAN). *Optical and Quantum Electronics*, 56(2). https://doi.org/10.1007/s11082-023-05618-7

Chegeni, M. K., Rashno, A., & Fadaei, S. (2022). Convolution-layer parameters optimization in Convolutional Neural Networks. *Knowledge-Based Systems*, 261, 110210. https://doi.org/10.1016/j.knosys.2022.110210

Choi, J., Hong, S., Park, N., & Cho, S. (2023). Blurring-Sharpening Process Models for Collaborative Filtering. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1096–1106. https://doi.org/10.1145/3539618.3591645

Cococcioni, M., Rossi, F., Ruffaldi, E., & Saponara, S. (2020). A novel posit-based fast approximation of elu activation function for deep neural networks. *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*, 244–246. https://doi.org/10.1109/smartcomp50058.2020.00053.

Demir, Y., & Kaplan, N. (2023). Low-light image enhancement based on sharpening-smoothing image filter. *Digital Signal Processing*, 138, 104054. https://doi.org/10.1016/j.dsp.2023.104054

Eckle, K., & Schmidt-Hieber, J. (2019). A comparison of deep networks with ReLU activation function and linear spline-type methods. *Neural Networks, 110*, 232–242. https://doi.org/10.1016/j.neunet.2018.11.005.

Eom, G., & Yun, K. (2024). Deep Learning-Based Computational Time-Domain Homogenization Method for Viscoelastic Composites. *Transactions of the Korean Society of Mechanical Engineers A*, 48(10), 681–687. https://doi.org/10.3795/ksme-a.2024.48.10.681

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press,Massachusetts.

Herwanto, H. W., Handayani, A. N., Wibawa, A. P., Chandrika, K. L., & Arai, K. (2021). Comparison of Min-Max, Z-Score and Decimal Scaling Normalization for Zoning Feature Extraction on Javanese Character Recognition. *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, 1–3. https://doi.org/10.1109/iceeie52663.2021.9616665

Hu, X., Niu, P., Wang, J., & Zhang, X. (2019). A dynamic rectified linear activation units. *IEEE Access*, *7*, 180409–180416. https://doi.org/10.1109/access.2019.2959036.

Ismail, A. H., Soliman, T. A., Rihan, M., & Dessouky, M. I. (2023). Deep Learning-Based Beamforming for Millimeter-Wave Systems Using Parametric ReLU Activation

Function. *Wireless Personal Communications*, 129(2), 825–836. https://doi.org/10.1007/s11277-022-10157-7

Job, M. S., Bhateja, P. H., Gupta, M., Bingi, K., & Prusty, B. R. (2022). Fractional rectified linear unit activation function and its variants. *Mathematical Problems in Engineering, 2022*(1), 1860779. https://doi.org/10.1155/2022/1860779.

K. Jermsittiparsert et al. (2020). Pattern recognition and features selection for speech emotion recognition model using deep learning. *International Journal of Speech Technology, 23*(4), 799–806. https://doi.org/10.1007/s10772-020-09690-2.

Kalyanam, L. K., & Katkoori, S. (2023). Unstructured Pruning for Multi-Layer Perceptrons with Tanh Activation. 2021 IEEE International Symposium on Smart Electronic Systems (iSES). https://doi.org/10.1109/ises58672.2023.00025

Khaskhoussy, R., & Ayed, Y. B. (2023). Improving Parkinson's disease recognition through voice analysis using deep learning. *Pattern Recognition Letters*, *168*, 64--70. https://doi.org/10.1016/j.patrec.2023.03.011.

Lee, D. (2020). Comparison of reinforcement learning activation functions to improve the performance of the racing game learning agent. *Journal of Information Processing Systems*, *16*(5), 1074–1082. https://doi.org/10.3745/JIPS.02.0141

Lin, S., Chi, K., Wei, T., & Tao, Z. (2021). Underwater image sharpening based on structure restoration and texture enhancement. *Applied Optics*, 60(15), 4443. https://doi.org/10.1364/ao.420962

Liu, K., Shi, W., Huang, C., & Zeng, D. (2023). Cost effective Tanh activation function circuits based on fast piecewise linear logic. *Microelectronics Journal*, 138, 105821. https://doi.org/10.1016/j.mejo.2023.105821

Liu, X., Zhou, Z., & Qian, H. (2019). Comparison and Evaluation of Activation Functions in Term of Gradient Instability in Deep Neural Networks. *2019 Chinese Control And Decision Conference (CCDC)*. https://doi.org/10.1109/ccdc.2019.8832578

Narmadha, S., & Vijayakumar, V. (2019). An improved stacked denoise autoencoder with elu activation function for traffic data imputation. *Int J Innov Technol Explor Eng*, *8*, 977–980. https://doi.org/10.35940/ijitee.k2022.0981119.

Nguyen, T. D., Kim, D. H., Yang, J. S., & Park, S. Y. (2021). High-Speed ASIC Implementation of Tanh Activation Function Based on the CORDIC Algorithm. *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 1–3. https://doi.org/10.1109/itc-cscc52171.2021.9501440

Pappas, C., Kovaios, S., Moralis-Pegios, M., Tsakyridis, A., Giamougiannis, G., Kirtas, M., Van Kerrebrouck, J., Coudyzer, G., Yin, X., Passalis, N., Tefas, A., & Pleros, N. (2023). Programmable Tanh-, ELU-, Sigmoid-, and Sin-Based Nonlinear Activation Functions for Neuromorphic Photonics. *IEEE Journal of Selected Topics in Quantum Electronics*, 29(6: Photonic Signal Processing), 1–10. https://doi.org/10.1109/jstqe.2023.3277118

Pham, T. D. (2022). Kriging-Weighted Laplacian Kernels for Grayscale Image Sharpening. *IEEE Access*, 10, 57094–57106. https://doi.org/10.1109/access.2022.3178607

Qiumei, Z., Dan, T., & Fenghua, W. (2019). Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function. *IEEE Access*, *7*, 151359–151367. https://doi.org/10.1109/access.2019.2948112.

Rachmawati, O. C. R., & Darmawan, Z. M. E. (2024). The Comparison of Deep Learning Models for Indonesian Political Hoax News Detection. *CommIT (Communication and Information Technology) Journal*, 18(2), 123–135. https://doi.org/10.21512/commit.v18i2.10929

Ramadijanti, N., Barakbah, A., & Husna, F. A. (2018). Automatic breast tumor segmentation using hierarchical k-means on mammogram. *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 170–175. https://doi.org/10.1109/kcic.2018.8628467.

Sakketou, F., & Ampazis, N. (2019). On the invariance of the SELU activation function on algorithm and hyperparameter selection in neural network recommenders. *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24--26, 2019, Proceedings 15*, 673–685.

https://doi.org/10.1007/978-3-030-19823-7_56.
Salam, A., Hibaoui, A. ., & Saif, A. (2021). A comparison of activation functions in multilayer neural network for predicting the production and consumption of electricity power. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*(1), 163. https://doi.org/10.11591/ijece.v11i1.pp163-170.
Shatravin, V., Shashev, D., & Shidlovskiy, S. (2022). Sigmoid Activation Implementation for Neural Networks Hardware Accelerators Based on Reconfigurable Computing Environments for Low-Power Intelligent Systems. *Applied Sciences*, *12*(10), 5216. https://doi.org/10.3390/app12105216.
Shodiq, M. N., Kusuma, D. H., Rifqi, M. G., Barakbah, A. R., & Harsono, T. (2017). Spatial analisys of magnitude distribution for earthquake prediction using neural network based on automatic clustering in Indonesia. *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 246–251. https://doi.org/10.1109/kcic.2017.8228594.
Subhan, M., Sudarsono, A., & Barakbah, A. R. (2017). Classification of Radical Web Content in Indonesia using Web Content Mining and k-Nearest Neighbor Algorithm. *EMITTER International Journal of Engineering Technology*, *5*(2), 328–348. https://doi.org/10.24003/emitter.v5i2.214.
Suganya, M., & Sasipraba, T. (2023). Stochastic Gradient Descent long short-term memory based secure encryption algorithm for cloud data storage and retrieval in cloud computing environment. *Journal of Cloud Computing Advances Systems and Applications*, 12(1). https://doi.org/10.1186/s13677-023-00442-6
Szandała, T. (2020). Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. *Bio-Inspired Neurocomputing*, 203–224. https://doi.org/10.1007/978-981-15-5495-7_11.
Tian, Y., Zhang, Y., & Zhang, H. (2023). Recent Advances in Stochastic Gradient Descent in Deep Learning. *Mathematics*, 11(3), 682. https://doi.org/10.3390/math11030682
Vaidya, B., & Paunwala, C. (2019). Deep Learning Architectures for Object Detection and Classification. *Smart Techniques for a Smarter Planet*, 53–79. https://doi.org/10.1007/978-3-030-03131-2_4.
Wong, K., Dornberger, R., & Hanne, T. (2022). An analysis of weight initialization methods in connection with different activation functions for feedforward neural networks. *Evolutionary Intelligence*. https://doi.org/10.1007/s12065-022-00795-y
Xiangyang, L., Xing, Q., Han, Z., & Feng, C. (2023). A Novel Activation Function of Deep Neural Network. *Scientific Programming*, *2023*, 1–12. https://doi.org/10.1155/2023/3873561.
Yan, W., Xu, G., Du, Y., & Chen, X. (2022). SSVEP-EEG Feature Enhancement Method Using an Image Sharpening Filter. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 30, 115–123. https://doi.org/10.1109/TNSRE.2022.3142736
Ye, S., & He, Q. (2024). Mean squared error bound for learning-based multi-target localization and its application in learning network architecture design. *Digital Signal Processing*, 151, 104559. https://doi.org/10.1016/j.dsp.2024.104559.