SFO Science of Fisheries & Oceanography
ISSN 2734-1011 (Print)
ISSN 2671-4310 (Online)
Vol. 34, No. 1, pp. 21-27 (2025)
https://doi.org/10.22714/SFO.2025.34.1.3

Original Research Article

Nonlinear Normalization for Improved CNN Performance in Fish Disease Diagnosis

Dae-Hyon KIM[1,*]

[1]Department of Culture and Tourism Management, Chonnam National University, Yeosu 59626, Republic of Korea

* Correspondence to Dae-Hyon KIM
E-mail: daehyon@chonnam.ac.kr

Received November 26, 2025
Revised February 28, 2026
Accepted February 28, 2026

Abstract : The accelerating growth of aquaculture has heightened the challenge of accurately diagnosing fish diseases, leading to the increasing application of deep learning-based machine learning approaches. In particular, the predictive performance of deep learning models is significantly influenced by the normalization methods applied to the input data. This study investigates the impact of various normalization techniques on improving the accuracy of fish disease classification and seeks to determine the most effective method. To achieve this, a Convolutional Neural Network (CNN) model was trained using Power Normalization, Logistic Normalization, and Z-score Normalization (Standardization), each evaluated over 30 experimental trials. The experimental results showed that the average prediction accuracy with the conventional linear normalization method was 92.8%, while the accuracy improved slightly to 93.4% when using Power Normalization with an exponent of 1.2. Although the improvement in predictive accuracy is modest, these results suggest that the choice of normalization technique can have a tangible impact on model performance. This study contributes to a deeper understanding of how normalization strategies affect CNN-based fish disease classification.

Keywords : Fish disease classification, Convolutional Neural Network (CNN), Normalization techniques, Deep learning, Aquaculture diagnostics

Introduction

The aquaculture industry has experienced unprecedented growth over the past few decades, becoming a vital source of food security and economic stability worldwide. However, this rapid expansion has brought about significant challenges, particularly in the area of fish health management. Among these challenges, the accurate and timely diagnosis of fish diseases has emerged as a critical issue. Fish diseases can spread rapidly in densely populated aquaculture environments, leading to substantial economic losses and threatening the sustainability of the industry. Therefore, developing reliable diagnostic tools for early detection and classification of fish diseases is of paramount importance (Islam *et al.*, 2024; Kim, 2024).

In recent years, the application of deep learning-based machine learning methods has shown great promise in addressing the complexities of fish disease diagnosis. Convolutional Neural Networks (CNNs), in particular, have demonstrated superior performance in image classification tasks, making them a natural choice for diagnosing fish diseases based on visual data (Dash *et al.*, 2024). However, the success of deep learning models in such applications is not solely dependent on the architecture of the network but is also significantly influenced by the preprocessing techniques applied to the input data (Kim, 2010).

Normalization, as a preprocessing step, plays a crucial role in ensuring that the input data is scaled appropriately, thereby facilitating efficient learning and improving model performance (Kim, 1999, 2004; Kim *et al.*, 2010). Traditionally, linear normalization techniques have been widely used in deep learning applications, primarily due to their simplicity and ease of implementation. However, recent studies suggest that non-linear normalization methods may offer advantages in certain contexts, potentially leading to improved predictive accuracy (Kim, 2020).

This study investigates the impact of various normalization techniques on the performance of a CNN model for classifying fish diseases. The effectiveness of Power normalization, Logistic normalization, and Z-score normalization (Standardization) is compared to that of the conventional linear normalization method. Through a series of 30 trials, the goal is to identify the normalization technique that achieves the highest predictive accuracy, offering valuable insights into the optimal preprocessing strategies for deep learning models in fish disease diagnosis.

Research Methodology

1. Basics of convolutional neural networks and training strategies

Convolutional Neural Networks (CNNs) are a specialized class of deep learning models that have revolutionized the domains of computer vision and image analysis. Inspired by the biological mechanisms of visual processing in animals, CNNs excel at identifying spatial patterns within images (LeCun *et al.*, 2015). One of the key advantages of CNNs is their ability to retain the spatial structure of input data during the learning process. In contrast to traditional multilayer feedforward neural networks (MFNNs), which necessitate the flattening of image data into one-dimensional vectors—potentially leading to a loss of essential spatial information—CNNs work with data in its original matrix form. This capability allows CNNs to effectively capture the strong correlations between adjacent pixels, which are prevalent in image data. By applying filters directly to the matrices, CNNs maintain the integrity of the input images' shape and structure, thus avoiding the information loss that can occur with vectorization (Kim, 2024).

A standard CNN is composed of several layers, each of which is vital for processing and extracting features from input images. As shown in Fig. 1 (Kim, 2024), the architecture of CNNs typically includes the following key components:

(a) Convolutional layer: This layer applies a set of filters, also known as kernels, to the input image. Through elementwise multiplication with these filters, it generates feature maps that enable the network to detect local features such as edges and textures (Krizhevsky *et al.*, 2012).

(b) Activation function: After the convolution operation, an activation function introduces non-linearity into the model. The Rectified Linear Unit (ReLU) is widely used due to its simplicity and effectiveness in speeding up the training process (Nair and Hinton, 2010).

(c) Pooling layer: This layer reduces the spatial dimensions of the feature maps, minimizing the number of parameters and computational load while ensuring that the detected features are invariant to minor transformations and distortions (Zeiler and Fergus, 2014).

(d) Fully connected layer: This layer is responsible for high-level reasoning, connecting every neuron in one layer to every neuron in the next, similar to a traditional MLP. It integrates the features learned through previous layers to make the final predictions (Goodfellow *et al.*, 2016).

(e) Output layer: The final predictions are generated by this layer, often using a softmax activation function in classification tasks to provide a probability distribution across various classes (Bishop, 2006).

Training CNNs involves adjusting the network's weights to reduce the difference between its predictions and the actual outcomes. This optimization is typically performed using backpropagation in combination with optimization techniques such as Stochastic Gradient Descent (SGD) or the Adam optimizer (Kingma and Ba, 2015). Throughout the training process, CNNs iteratively refine the weights of their filters to focus on the most important features in the training data. In the initial stages, the filters learn to identify simple patterns, such as edges and corners. As training progresses, the deeper layers of the network start to capture more complex patterns and high-level features, allowing the model to accurately classify images. The effectiveness of training CNNs is influenced by several factors, including the network's architecture, the size and quality of the training dataset, regularization techniques like dropout, and hyperparameters such as learning rate and batch size. Achieving optimal

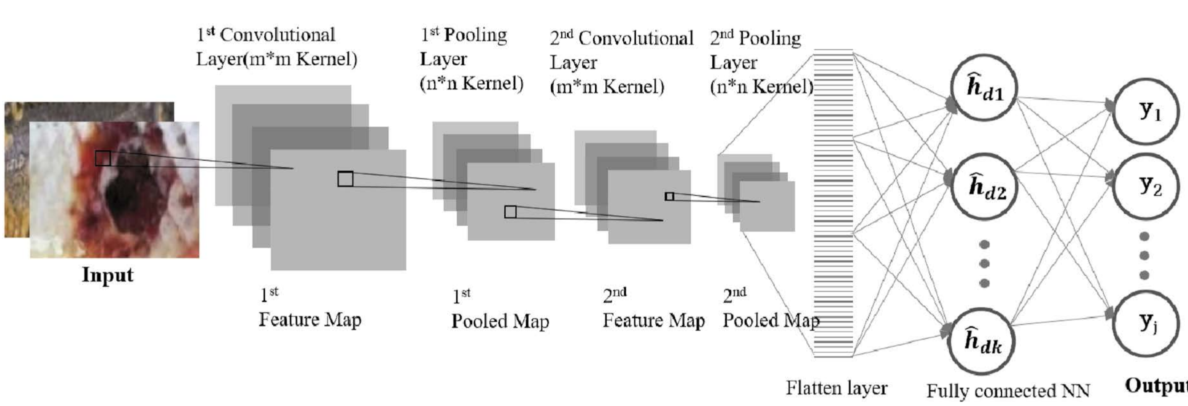


Fig. 1. CNN model structure: Layers and data flow (adapted from Kim, 2024).

performance and generalization to new data often requires careful tuning and experimentation.

2. Normalization methods for input vectors

1) Linear normalization

In machine learning, there are generally two types of linear normalization methods for input vectors. The first type involves using the maximum and minimum values of the input vectors within each dataset. This method of input normalization is represented by Equation (1).

$$\widetilde{{}_tx}_k = \frac{{}_tx_k - {}_tx_{\min}}{{}_tx_{\max} - {}_tx_{\min}}, \tag{1}$$

where $t(t=1,2,\ldots,n)$ is the training and test pattern sets, $x_{\min} = \min({}_tx_k;\, k=1,2,\ldots,m)$ and ${}_tx_{\max} = \max({}_tx_{ki};\, k=1,2,\ldots,m)$. In Equation (1), ${}_tx_{\min}$ and ${}_tx_{\max}$ are the minimum and maximum values, respectively, of the input data and $\widetilde{{}_tx}_k$ denotes the normalized value of the unit k of the input vector ${}_tx = ({}_tx_1, {}_tx_2, {}_tx_3, \ldots, {}_tx_m)$. This type of linear normalization method will use the range 0 to 1 for each input data and treats two linearly dependent data sets identically.

When applying artificial neural network models, Kim (1999) showed that an alternative linear normalization method, defined by Equation (2), can be more effective and superior to the earlier method described in Equation (1).

$$\widetilde{{}_tx}_k = \frac{{}_tx_k - {}_Tx_{\min}}{{}_Tx_{\max} - {}_Tx_{\min}}, \tag{2}$$

where ${}_Tx_{\min}$ and ${}_Tx_{\max}$ are the minimum and maximum values, respectively, of the input unit across all the input data sets, i.e.:

$${}_Tx_{\min} = \min\{d_1(x_1,x_2,\ldots,x_m),\ldots,d_n(x_1,x_2,\ldots,x_m)\} \text{ and}$$

$${}_Tx_{\max} = \max\{d_1(x_1,x_2,\ldots,x_m),\ldots,d_n(x_1,x_2,\ldots,x_m)\}, \tag{3}$$

where $\{d_1(x_1,x_2,\ldots,x_m),\ldots,d_n(x_1,x_2,\ldots,x_m)\}$ are the input data sets. The normalization described in Equation (2) uses a range of 0 to 1 across the entire input dataset and is one of the most widely used normalization methods.

2) Logistic normalization

Although linear-based normalization methods are common, various types of normalization techniques can be developed, with nonlinear models in particular offering potentially better performance than linear regularization methods. Kim (2020) proposed a nonlinear normalization method based on a logistic function to enhance the learning performance of neural network models.

As a method for input vector normalization, the logistic function is defined by Equation (4).

$$\widetilde{{}_tx}_k = \frac{1}{1+\exp^{(-{}_tx_k + c)}}, \tag{4}$$

where ${}_tx_k$ is a variable representing the original pixel value, and c is constant with

$$c = \frac{({}_tx_{\max} - {}_tx_{\min})}{2}, \tag{5}$$

where ${}_tx_{\min}$ and ${}_tx_{\max}$ are the minimum and maximum values, respectively, of the input units across all input data sets. In Equations (4) and (5), c is a constant that is calculated using two fixed values from the data sets. For classification problems involving image datasets, it is common to set ${}_tx_{\min} = 0$, ${}_tx_{\max} = 255$, and $c = 128$.

Logistic functions for input vector normalization have specific characteristics: they expand values near the median, c, which is the value separating the upper and lower halves of the dataset. This normalization method tends to emphasize the median of the dataset while reducing the influence of extreme values near the maximum and minimum.

3) Power normalization

Power normalization, when applied to data that has already been linearly normalized, modifies the distribution of the input values by raising the normalized values to a specified power. This approach combines the benefits of linear normalization with a nonlinear transformation that can either compress or amplify the range of the data depending on the chosen exponent. The general form of power normalization is defined by Equation (6).

$$\widetilde{{}_tx}_k = \left(\frac{{}_tx_k}{T^x_{\max}}\right)^{Power}, \tag{6}$$

where Power is the exponent used in the transformation.

When the exponent is less than 1, the transformation compresses the normalized values, particularly those closer to 1. This reduces the relative impact of larger values, effectively decreasing the influence of outliers and spreading the smaller values over a larger range. This can be beneficial in datasets where the values are skewed or where the influence of extreme values needs to be mitigated. Conversely, when the power is set to a value greater than 1, the transformation amplifies the differences between values, particularly those that are already larger. This approach emphasizes the larger normalized values, making distinctions among them more pronounced. It is useful in scenarios where it is important to highlight larger values or when greater separation between high and low values is desired.

The combination of linear normalization followed by power transformation ensures that all input values are initially brought into a common scale (0 to 1), and then the power transformation either compresses or expands specific ranges within this scale. This method provides a flexible way to address data distributions that are not ideally handled by simple linear normalization alone.

4) Z-score normalization

Z-score normalization, also known as standardization, is a crucial data preprocessing technique in machine learning and statistics. It is particularly useful for datasets with features that have different units or scales, ensuring that each feature contributes equally to the model's learning process.

Z-score normalization is a method that transforms data so that the distribution of each feature has a mean of zero and a standard deviation of one. This process is particularly beneficial when dealing with data that may contain outliers, as it helps to mitigate their impact. In the study by Dinç *et al.* (2014), it was reported that this technique can improve model accuracy.

The general form of Z-score normalization is defined by the following Equation (7).

$$\widetilde{{}_t x_k} = \frac{{}_t x_k - \mu}{\sigma}, \tag{7}$$

where μ is the mean of the input data. σ is the standard deviation of the input data.

By centering the data around zero and scaling it, standardization makes it easier for machine learning algorithms to converge during training and improves the performance of models that are sensitive to the scales of features, such as those that use distance metrics.

Research Results

1. Experimental data

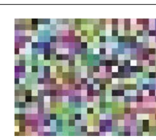
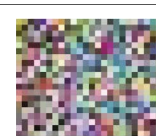
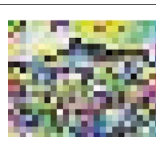
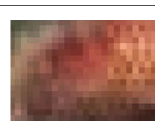
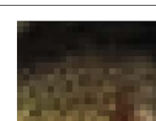
In the experiments conducted for this study, data from a previous research by Kim (2024) was utilized. This dataset, titled 'Freshwater Fish Disease Aquaculture in South Asia' (Biswas, 2024), was sourced from Kaggle and consists of several components. A total of 198 images were extracted from this dataset, including 108 images of diseased fish and 90 images of healthy fish. For experimental purposes, the dataset was divided into training and testing sets. The training set comprised 54 images of diseased fish and 45 images of healthy fish, making a total of 99 images. The remaining 99 images were allocated for testing.

Table 1 illustrates the data used for training and testing, resized to 20×25 pixels, normalized, and visualized as a color image map.

2. Network architecture and hyperparameters

The CNN model employed in the experiments was configured with the following hyperparameters. It consists of three Conv2D layers, each with a 3×3 kernel. The first layer util-

Table 1. Normalized image visualization using a color map

Normalization methods	Training data		Testing data	
	Healthy fish	Diseased fish	Healthy fish	Diseased fish
Power = 0.8				
Power = 1.2				
Logistic				
Z-score (Standardization)				
Linear				

izes 32 filters, while the second and third layers each employ 64 filters to extract features from the input images. The ReLU activation function is applied across all layers to introduce nonlinearity.

To reduce the spatial dimensions of the feature maps, two MaxPooling2D layers with 2×2 pooling windows are incorporated. The input vector for the CNN model consists of 20×25 pixel image data with three color channels (RGB), processed using various normalization methods.

A Flatten layer is employed to convert the 2D feature maps into a 1D vector when transitioning to the fully connected layers. Following this, two Dense layers are used: the first with 64 neurons and the second with 2 neurons, corresponding to the binary labels (0 and 1) for detecting the presence or absence of disease. A Dropout rate of 0.5 is applied to mitigate overfitting.

The output layer uses the softmax activation function to facilitate binary classification. The model's performance is assessed based on mean accuracy and standard deviation, with comparisons made across different image sizes to identify the optimal dimensions.

For training, the model was run for 100 epochs with a batch size of 32. These settings were consistently maintained across multiple runs to ensure a reliable assessment of the model's performance.

3. Experimental results

Table 1 presents the results of training and testing the CNN model on images of diseased and healthy fish, using various normalization methods. Table 2(A) shows the average prediction performance across 30 experiments conducted with different initial weights, while Table 2(B) and Table 2(C) display the best and worst cases from these experiments, respectively.

In this study, five normalization methods were compared: Linear normalization, Logistic normalization, Power (Power = 0.8) normalization, Power (Power = 1.2) normalization, and Z-score normalization (Standardization). For each method, the model was trained and tested 30 times with different initial weights. The average performance of the CNN model over 30 experiments showed that the Power (Power = 1.2) normalization method achieved the highest mean prediction accuracy of 93.4% and the lowest error rate of 6.6%. In contrast, the widely used Linear normalization method yielded a slightly lower accuracy of 92.8%. Additionally, the variance in prediction results across the 30 trials was lower with Power (Power = 1.2) normalization at 2.12, compared to 2.64 for Linear normalization, indicating that Power (Power = 1.2) normalization is less sensitive to initial weight conditions. However, despite showing good results in previous studies, Logistic normalization and Z-score normalization (Standardization) performed worse than Linear normalization in this study.

An ANOVA test was conducted to evaluate the statistical significance of performance differences between normalization methods, resulting in an F-statistic of 32.938 and a p-value of 1.658e-57. These results highlight the significant impact of the chosen normalization method on the performance of the CNN model. The trends observed in the average performance were also reflected in the best and worst cases across the 30 experiments. Specifically, in the best-case scenario, Power (Power = 1.2) normalization achieved a prediction accuracy of 96.0%, outperforming the 94.9% accuracy obtained with Linear normalization.

In conclusion, for the task of pattern recognition in the images of diseased and healthy fish studied here, Power (Power = 1.2) normalization proved to be the most effective normalization method.

Conclusions

This study explored the impact of various normalization techniques on the performance of CNNs for fish disease classification. The results demonstrate that the choice of normalization method can significantly influence the predictive accuracy and robustness of deep learning models in this context. Among the five normalization techniques evaluated—Linear normalization, Logistic normalization, Power normalization (Power = 0.8 and Power = 1.2), and Z-score normalization (Standardization)—the Power normalization with an exponent of 1.2 consistently outperformed the others.

The Power normalization (Power = 1.2) method achieved the highest mean prediction accuracy of 93.4% and exhibited the lowest error rate of 6.6%. This method also demonstrated a lower variance in prediction results across multiple trials, indicating greater stability and reduced sensitivity to initial weight conditions. In contrast, the commonly used Linear normalization, while still effective, delivered a slightly lower accuracy of 92.8% and exhibited higher variance, suggesting that it may be more susceptible to variations in initial conditions.

Interestingly, despite their effectiveness in other contexts, Logistic normalization and Z-score normalization (Standardization) did not perform as well in this study, yielding lower accuracy compared to Linear and Power normalization methods. The results from the ANOVA test further confirmed the statistical significance of these differences, emphasizing the critical role that normalization techniques play in enhancing the performance of CNN models for fish disease diagnosis.

Overall, the findings of this study suggest that Power normalization, particularly with an exponent of 1.2, is the most effec-

Table 2. Comparison of predictive accuracy across different normalization techniques

(A) Average performance on test data across 30 runs

Category		Normalization method				
		Linear	Logistic	Power normalization		Z-score (Standardization)
				Power = 0.8	Power = 1.2	
Prediction errors	Healthy fish	3.3	5.7	5.3	3.1	3.6
	Diseased fish	3.8	6.6	3.5	3.4	5.3
Total errors		7.1	12.3	8.8	6.5	8.9
Variance		2.64	7.11	10.25	2.12	4.44
Prediction error rate (%)		7.2	12.4	8.9	6.6	9.0
Prediction accuracy (%)		92.8	87.6	91.1	93.4	91.0
ANOVA result		F-statistic: 132.938; p-value: 1.658e-57				

(B) Best performance on test data across 30 runs

Category		Normalization method				
		Linear	Logistic	Power normalization		Z-score (Standardization)
				Power = 0.8	Power = 1.2	
Prediction errors	Healthy fish	3	3	1	1	2
	Diseased fish	2	6	4	3	4
Total errors		5	9	5	4	6
Prediction error rate (%)		5.1	9.1	5.1	4.0	6.1
Prediction accuracy (%)		94.9	90.9	94.9	96.0	93.9

(C) Worst performance on test data across 30 runs

Category		Normalization method				
		Linear	Logistic	Power normalization		Z-score (Standardization)
				Power = 0.8	Power = 1.2	
Prediction errors	Healthy fish	9	11	17	9	8
	Diseased fish	2	8	2	2	7
Total errors		11	19	19	11	15
Prediction error rate (%)		11.1	19.2	19.2	11.1	15.2
Prediction accuracy (%)		88.9	80.8	80.8	88.9	84.8

tive normalization method for the given dataset and task. These results offer valuable insights for researchers and practitioners in the field of aquaculture and machine learning, highlighting the importance of carefully selecting normalization methods to optimize model performance. Future research could expand on these findings by exploring the applicability of Power normalization to other types of datasets and machine learning tasks, as well as investigating the potential benefits of combining multiple normalization techniques.

References

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

Biswas, S., 2024. Freshwater Fish Disease Aquaculture in South Asia. Kaggle. Accessed July 28, 2024. https://www.kaggle.com/datasets

Dash, S., S.Ojha, R.K. Muduli, S.P. Patra and R.C. Barik, 2024. Fish type and disease classification using deep learning model-based customized CNN with ResNet-50 technique. *J. Adv. Zool.*, 45(3), 32-43.

Dinç, İ., M. Sigdel, S. Dinç, M.S. Sigdel, M.L. Pusey and R.S. Aygün, 2014. Evaluation of normalization and PCA on the performance of classifiers for protein crystallization images. *Proc. IEEE Southeastcon*, 2014, 1-6. https://doi.org/10.1109/SECON.2014.6950744

Goodfellow, I., Y. Bengio and A. Courville, 2016. Deep Learning. MIT Press.

Islam, S.I., F. Ahammad and H. Mohammed, 2024. Cutting-edge technologies for detecting and controlling fish diseases: Current status, outlook, and challenges. *J. World Aquacult. Soc*., 55(2), e13051. https://doi.org/10.1111/jwas.13051

Kim, D., 1999. Normalization methods for input and output vectors in backpropagation neural networks. *Int. J. Comput. Math*., 71(2), 161-171.

Kim, D., 2004. Prediction performance of support vector machines on input vector normalization methods. *Int. J. Comput. Math.*, 81(5), 547-554.

Kim, D., 2010. Pre-processing of inputs to a neural network model for better performance in traffic scene analysis. *Civ. Eng. Environ. Syst*., 27(1), 23-31.

Kim, D., 2020. Nonlinear Normalization Model to Improve the Performance of Neural Networks. *APJCRI*, 6(11), 183-192. https://doi.org/10.47116/apjcri.2020.11.16

Kim, D., 2024. Analyzing the Impact of Image Size on CNN Model Performance for Fish Disease Diagnosis. *SFO*, 33(10), 31-37. https://doi.org/10.22714/SFO.2024.33.1.0

Kim, D., J.-M. Kim and S.-H. Mun, 2010. Normalisation methods on neural networks for predicting pavement layer moduli. *Road & Transport Research*, 19(3), 38-46.

Kingma, D.P. and J. Ba, 2015. Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980

Krizhevsky, A., I. Sutskever and G.E. Hinton, 2012. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst*., 25, 1097-1105.

LeCun, Y., Y. Bengio and G. Hinton, 2015. Deep learning. *Nature*, 521(7553), 436-444.

Nair, V. and G.E. Hinton, 2010. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, pp. 807-814.

Zeiler, M.D. and R. Fergus, 2014. Visualizing and understanding convolutional networks. In European Conference on Computer Vision. Springer, pp. 818-833.