

Article

# Performance Prediction of Induction Motor Due to Rotor Slot Shape Change Using Convolution Neural Network

Dong-Young Koh <sup>1</sup>, Sung-Jun Jeon <sup>1</sup> and Seog-Young Han <sup>2,\*</sup>

<sup>1</sup> Department of Mechanical Convergence Engineering, Hanyang University, Seoul 04763, Korea; dykoh77@naver.com (D.-Y.K.); jsj1026@hanyang.ac.kr (S.-J.J.)

<sup>2</sup> School of Mechanical Engineering, Hanyang University, Seoul 04763, Korea

\* Correspondence: syhan@hanyang.ac.kr

**Abstract:** We propose a method to predict performance variables according to the rotor slot shape of a three-phase squirrel cage induction motor using a convolution neural network (CNN) algorithm suitable for utilizing image data. The set of performance variables was labeled according to the images of each training dataset, and this set was generated from the efficiency, power factor, starting torque, and average torque. To verify the accuracy of the trained deep learning model, the analysis and prediction results of the CNN model were compared and verified with nine untrained double cage slot shapes and shapes optimized based on the root mean square error (RMSE). Although a large number of training data are required for high accuracy in the existing image processing deep learning model, the proposed deep learning method can predict the performance variables for various shapes with the same level of accuracy as the finite element analysis results using a small number of training data. Therefore, it is expected to be applied in various engineering fields.

**Keywords:** artificial intelligence; deep learning; convolution neural network (CNN); induction motor



**Citation:** Koh, D.-Y.; Jeon, S.-J.; Han, S.-Y. Performance Prediction of Induction Motor Due to Rotor Slot Shape Change Using Convolution Neural Network. *Energies* **2022**, *15*, 4129. <https://doi.org/10.3390/en15114129>

Academic Editor: Sheldon Williamson

Received: 28 April 2022

Accepted: 2 June 2022

Published: 4 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Motors account for more than 60% of the total power consumption, and induction motors account for more than 70% of total motor usage [1]. Induction motors can be broadly classified into squirrel cage induction motors and wire-wound induction motors, where squirrel cage induction motors have advantages in terms of maintenance and repair over wire-wound induction motors. More than 90% of electric motors in industry are three-phase squirrel induction motors, which are widely used in home appliances and various industries due to their simple structure, robustness, and low manufacturing cost [2,3]. To improve the performance of the three-phase squirrel cage induction motor, a study on the slot shape of the rotor and stator was conducted [4–6]. It has been shown that only the rotor slot shape greatly affects the characteristics of the induction motor [7–9]. In many existing induction motor design methods, it is essential to find a solution by minimizing errors through numerous iterations, which incurs an enormous computation time and high computational costs [10,11]. In addition, the previous calculation process is not saved and not used for future analysis, and it takes the same amount of time as before, even if a similar shape is analyzed. In order to solve such a problem, it is necessary to apply a deep learning technique that can predict the result value based on experience by using the trends of previous data and calculation records for future analysis.

Deep learning is a field of artificial intelligence and is a method of learning data characteristics and core contents from a large amount of data. Representative deep learning techniques include deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). In this study, a CNN is used; CNNs are composed of one or several convolution layers, and are characterized by using a pooling layer and a fully connected layer. Because of these features, CNNs are effective in extracting and learning features from image data [12–14]. The CNN was first devised by Fukushima

and Miyake [14], inspired by human neuroscience research. LeCun et al. [12] tried to recognize handwritten digits through a CNN. The network learning method using back propagation proposed by [15] became the basic structure of CNNs that are still used today. Afterwards, they were developed as a method of learning network parameters from a large amount of training data using a backpropagation method, consisting of several convolutional layers, a pooling layer, and a precoupling layer. As such, CNNs have been used in numerous image-data-based research fields, such as robots, natural language processing, and autonomous vehicles, and have been applied in various fields, such as motor design, optimization, and fault diagnosis. Sasaki and Igarashi [16] performed phase optimization using a CNN as training data using the section of the permanent magnet motor expressed in RGB. Asanuma et al. [17] performed the optimization of permanent magnet motors using transfer learning to reduce the computational cost of phase optimization based on genetic algorithms. Chattopadhyay et al. [18] proposed a CNN model with 3~10% improved performance compared with the existing method to recognize the defects of induction motors. The deep learning technique in the above studies was used to support the analysis of the data, and the final result was still obtained through simulation. To improve this, Gabdullin et al. [19] proposed a deep learning model to predict the final result without using simulation by applying CNN, and predicted the torque curve according to the shape variable of the permanent magnet motor with more than 90% accuracy.

However, in previous studies that predicted performance by applying deep learning techniques in the field of electric motors, only the performance according to shape variables was predicted for a fixed shape. That is, it was a prediction according to the sizing of a fixed shape, and no research has been conducted on a change in shape. In order to obtain prediction results for various shapes, high accuracy can be obtained by learning the same shape as the shape to be predicted. However, if the deep learning technique cannot be applied due to insufficient training data, there is a limit to learning all shapes. It was determined that the above problems could be solved by learning the minimum shape information and predicting various shapes. Accordingly, in this study, we intend to propose the predictability of a combined shape that has not been learned with learning data composed of a specific shape.

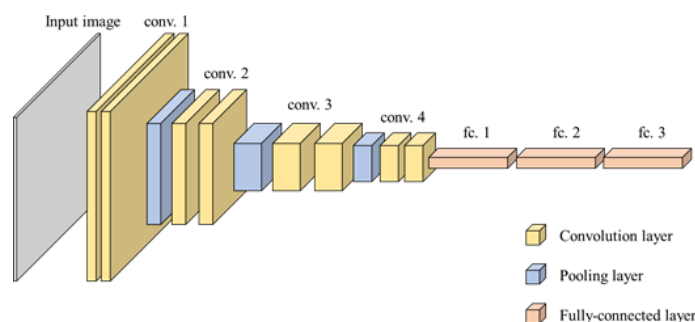
In this study, a CNN algorithm was used, and the efficiency, power factor, average torque, and starting torque were set as the performance variables. To check whether the CNN accurately learns the trends of the performance variables according to the slot shape, the training data were classified into three cases according to the shape, while the verification data were composed of the unlearned rotor slot shape. To obtain performance variables according to the shape, the MATLAB-based 2D Finite Element Analysis open-source code [20] was used. By designating the range of shape variables for each shape, the rotor shape was randomly generated within the specified range so that the image and corresponding analysis results were designated as a label. In addition, to confirm the accuracy of the predicted performance variables for the optimal shape, optimization was performed on an arbitrary shape.

We tried to predict the results based on the trend of the performance variables according to the slot shape, which has already been learned via the previous data and calculation history. This setup allows for future analysis through the application of deep learning techniques. As a result of training the CNN with 7500 training data points for 11 shapes, it was possible to obtain a prediction value that was almost the same as that of the finite element analysis of the validation data.

## 2. Convolution Neural Network (CNN)

In deep learning, an appropriate model should be selected according to the characteristics of the training data used. In this study, image data, which are suitable for expressing the various slot shapes of induction motor rotors, were used. For this purpose, a CNN, a model that can learn while maintaining spatial information of the image, was applied. To learn high-dimensional image data, a process of transforming them into a one-dimensional

planar matrix form is essential [13], and the spatial information of the data may be lost during the planarization process. That is, learning proceeds inefficiently because of a lack of information due to image spatial data loss, and there may be limitations in increasing accuracy. However, since the CNN learns while maintaining spatial information in the data-flattening process, it is possible to predict with high accuracy. CNN has a convolution layer that extracts features from image data, a pooling layer that reduces the dimension of input data and selects important information received from the convolution layer, and a fully connected layer that outputs prediction results based on the features extracted from the convolution layer. The overall structure is shown in Figure 1.



**Figure 1.** Composition of the convolution neural network.

The purpose of the convolution layer is to analyze the input image data to extract features and to learn them. To analyze the image, the feature map generated by applying a filter to the data is transferred to the next layer. The filter performs convolution from the input data, and stride is a variable that determines at what interval the filter is moved. The filter value is automatically repeated and corrected in the direction of reducing the final error value as the CNN learns.

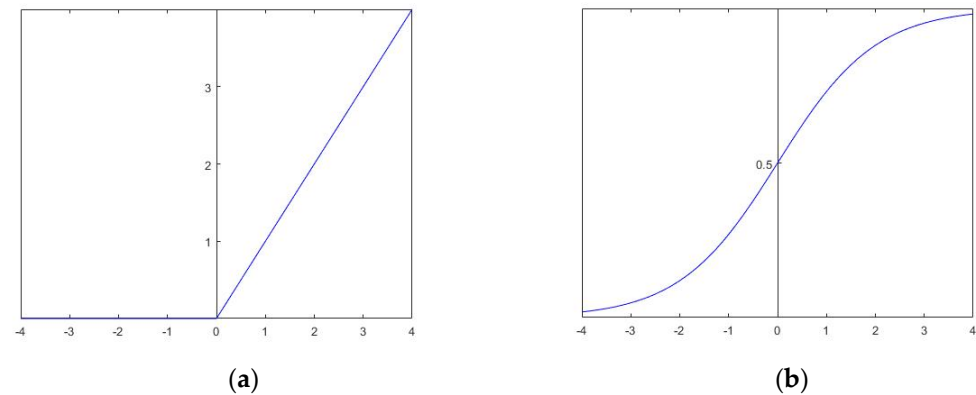
The feature map finally calculated in the convolution layer is not transmitted directly to the next layer, but only after passing through the activation function. All operations performed in the convolution layer are composed of linear functions. That is, even if the convolution layer is repeated infinitely, it is limited in solving nonlinear data problems unless nonlinearity is forcibly applied. To learn both linear and non-linear data, the process of adding non-linearity through an activation function is essential. For activation functions, ReLU and Sigmoid functions were used. Equations (1) and (2) and Figure 2 were defined by comparisons with other functions, so that the ReLU function had the advantages of fast learning, low computational cost, and very simple implementation. This function can solve the vanishing gradient problem, because it outputs 0 when the value of  $x$  is 0 or less, and a constant gradient value when it is positive [21]. The sigmoid function outputs a value between 0 and 1 according to the  $x$  value, and outputs a negative value as a value close to 0 [22].

$$\text{ReLU}(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad (1)$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

The pooling layer receives the output data of the convolution layer as the input and reduces the dimension of the feature map. The pooling technique is classified into max pooling that extracts the maximum value from a certain part of the feature map, min pooling that extracts the minimum value, and average pooling that extracts the average value. Max pooling is mainly used, rather than min pooling or average pooling, under the assumption that strong features are expressed when the pixel value is large [23]. If the convolution layer and the pooling layer are repeated, the image data are transformed into a shape that is difficult to recognize with the naked eye, but a feature map is output that extracts only the important features of the data. The fully connected layer lists the finally

obtained feature map as one-dimensional vector data and outputs the final predicted value after going through the activation function.



**Figure 2.** Activation function; (a) ReLU (b) Sigmoid.

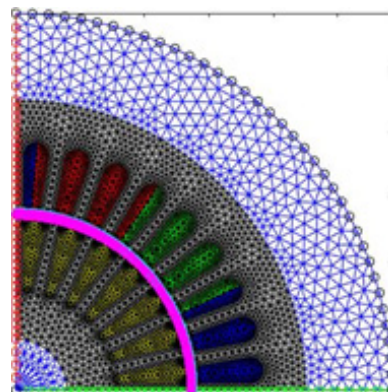
### 3. Induction Motor Performance Prediction Procedure and Results

The three-phase squirrel cage induction motor outputs various performance parameters according to the shape and size of the rotor slot. In this study, learning data were constructed by randomly changing various shapes and their associated variables for the rotor slot. The deep learning model was verified by predicting the performance variable according to the shape change, and the prediction accuracy for the optimized shape was further verified. The following procedure was performed to predict the performance variables according to shape data using deep learning as the method proposed in this study.

- Step 1: Collect induction motor analysis learning data;
- Step 2: Construct and train a deep learning model;
- Step 3: Validate the prediction accuracy of performance variables according to shape change;
- Step 4: Validate the prediction accuracy of the performance parameters for the optimized shape.

#### 3.1. Collect Induction Motor Analysis Learning Data

In this study, only the change in shape of the rotor slot was considered due to the characteristics of the induction motor, which are greatly affected by the shape and size of the rotor. Therefore, variables other than the shape and size of the rotor slot were fixed and then learning data were collected. Table 1 shows the specifications of a 2.2 kW-class three-phase squirrel cage induction motor. Figure 3 shows the results of the implementation of the shape and meshing of the three-phase squirrel cage induction motor by interlocking the MATLAB-based finite element analysis code with G-mesh [24].



**Figure 3.** Mesh image of the three-phase squirrel cage induction motor in MATLAB.




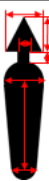


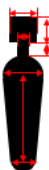


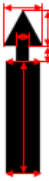

**Table 1.** Specifications of the three-phase squirrel cage induction motor.

General data	Given output power	2.2 kW
	Rated voltage	380 V
	Number of poles	4
	Frequency	60 Hz
	Length of shaft	120 mm
Winding	Turns per coil	16
	Winding layers	2
	Winding connection	Star connection
	Filling factor	41.25%
Stator	Number of slots	36
	Outer diameter	180 mm
	Inner diameter	110 mm
	Material	50PN470
Rotor	Number of slots	28
	Air gap	0.35 mm
	Inner diameter	32 mm
	Material	50PN470

To learn the shapes of various rotor slots, learning data for a total of 11 shapes were constructed, as shown in Table 2. The shape parameters used are indicated in red in Table 2. The training data of Type 1 consist of only the most basic shape, Type 2 is a shape corresponding to the upper part of a double cage slot, and Type 3 is a double cage slot that can be created by a combination of the Type 1 and Type 2 shapes. Type 1 is a basic slot shape, and to learn the characteristics of various shapes as much as possible, the range of each shape variable is considered from the minimum value. In addition, the rotor outer diameter, shaft outer diameter, and the range in which physical interference does not occur between slots were considered. To predict the performance variables for the type 3 combined shape, the training data set was classified into three cases, as shown in Table 3. In Case 1, only Type 1 data were learned; in Case 2, both Type 1 and Type 2 data were learned; and in Case 3, all data from Type 1 to Type 3 were learned. Type 1-1 and Type 1-2 generated 1500 data points each, and Type 2 and Type 3 generated 500 data points for each shape. Accordingly, 3000 pieces of training data for Case 1, 4500 pieces for Case 2, and 7500 pieces for Case 3 were used.

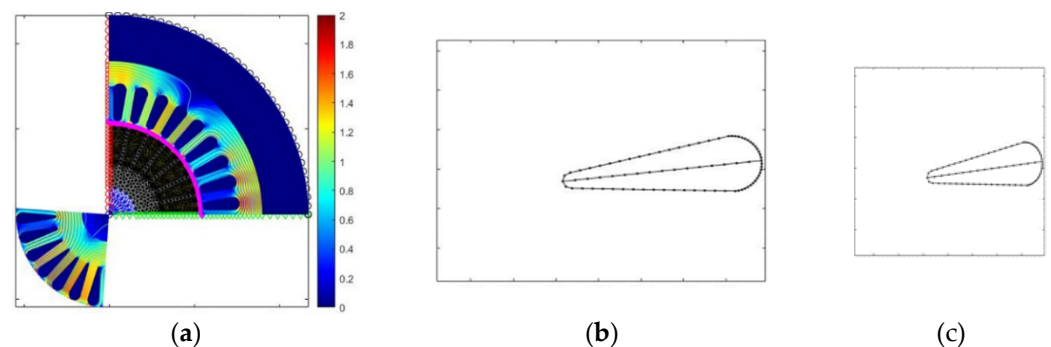
As described above, data preprocessing is essential to reduce the computational amount of deep learning that is needed to learn a large amount of training data and to increase the learning performance of the model. The data obtained through simulation are shown in Figure 4a. In this study, the performance variables for the rotor shape were considered, so Figure 4b was removed, and the existing RGB data of (875, 656, 3) are shown in Figure 4c; the amount of computation was reduced by reducing all pixels and dimensions to (128, 128, 1), expressed in black and white. After converting the image data into a numpy array for training, the efficiency, power factor, starting torque, and average torque corresponding to performance variables were labeled as labels. In addition, in order to improve the learning speed of the CNN model, image data, which were converted into a numpy array and expressed as pixel values of (0 to 255), were divided by 255 and the pixel values of the data were normalized to values between 0 and 1.

**Table 2.** Example rotor slot shapes for training data, ( ): number of data points.

Type 1	Type 2	Type 3	
			
Type 1-1 (1500)	Type 2-1 (500)	Type 3-1 (500)	Type 3-2 (500)
			
Type 1-2 (1500)	Type 2-2 (500)	Type 3-3 (500)	Type 3-4 (500)
			
	Type 2-3 (500)	Type 3-5 (500)	Type 3-6 (500)

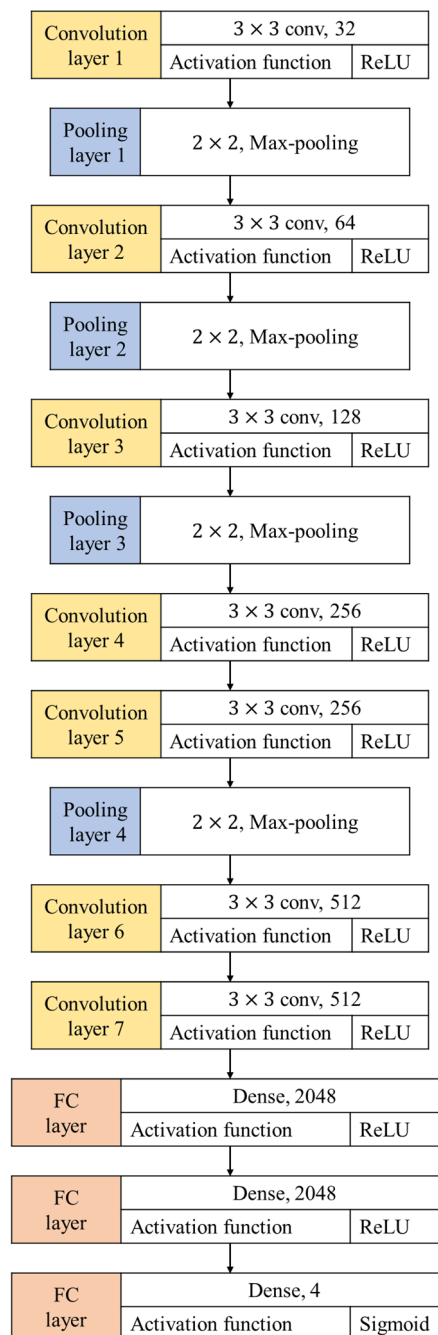
**Table 3.** Cases of the training data.

Case	Data Type	Number of Training Data Points
Case 1	Type 1	3000
Case 2	Type 1, Type 2	4500
Case 3	Type 1, Type 2, Type 3	7500

**Figure 4.** Preprocessing process of simulation data; (a) simulation data (b) rotor slot image (875, 656) (c) rotor slot image (128, 128).

### 3.2. Construct and Train a Deep Learning Model

The constructed CNN model is shown in Figure 5. This network consisted of seven convolution layers and four pooling layers, and the size of the convolution layer filter was (3, 3), which was the same for all. The CNN also had a structure in which the gradient and the pooling layer were repeated. The ReLU activation function was applied to solve the loss problem in the convolution layer. In the pooling layer, the maximum pooling technique of size (2, 2) was applied; in the output layer, a sigmoid function was used, and the learning rate was set to 0.0001. In addition, the Adam optimizer was used as an algorithm for learning, and all codes were written based on Tensorflow 2.7.0 and Keras 2.7.0. As the computing environment used in this study, the CPU was an AMD Ryzen 7 3700X, GPU was ab NVIDIA GeForce RTX 2080 TI, and RAM was 64 GB.



**Figure 5.** Architecture of the convolution neural network.

### 3.3. Validate the Prediction Accuracy of Performance Variables According to Shape Change

To verify the prediction accuracy of the trained CNN model, the shape of the validation data configured must not overlap with the training data, as shown in Table 4. For the verification data, the shape variables were set in the range where physical interference did not occur in the same way as the training data, and 50 randomly generated data for each shape were compared and verified for Cases 1 to 3.

### 3.4. Validate the Prediction Accuracy of the Performance Parameters for the Optimized Shape

To confirm the prediction accuracy of the optimal shape, optimization was performed on the shape of Type 3-1, which output the best performance variables among the configured training data. The shape parameters of Type 3-1 are shown in Figure 6. It was set

up as shown in Table 5, and the range and objective function of the shape variables were defined as shown in Table 5.

Table 4. Example shapes of the rotor slot for test data.

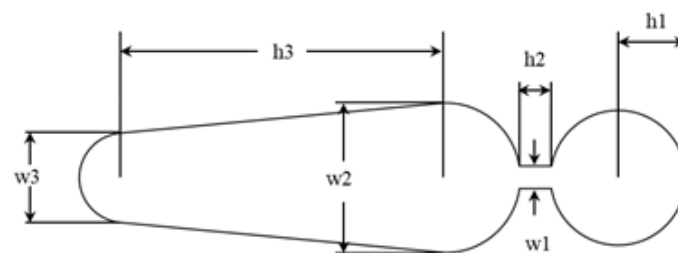
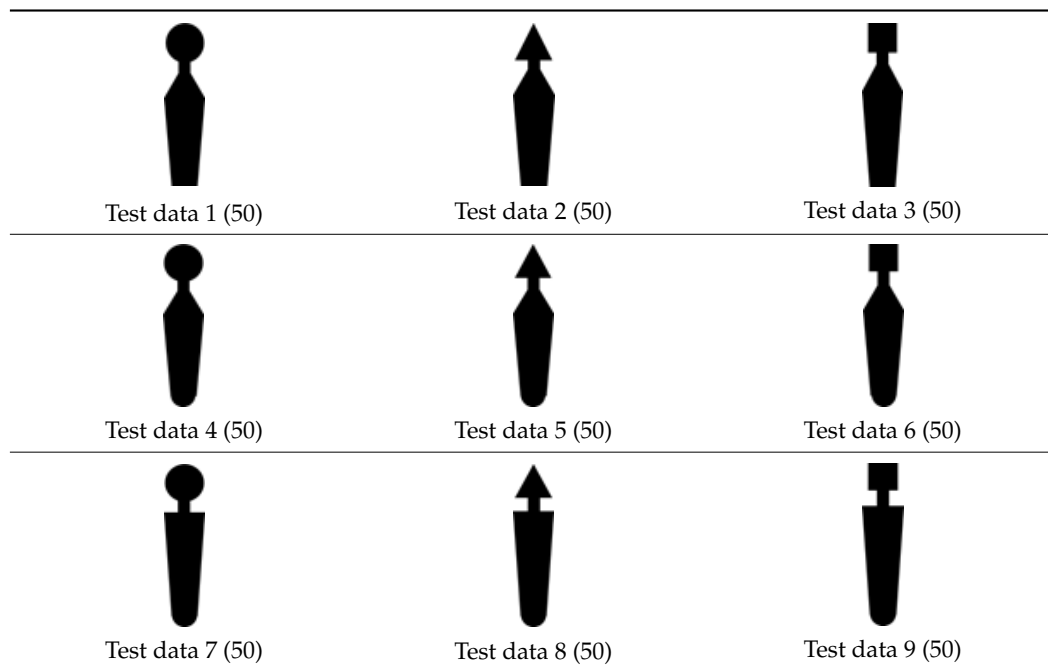


Figure 6. Parametrization of Type 3-1 rotor slot.

Table 5. Definition of the objective function and design variables.

Objective Function	Design Variables
Efficiency	$1.5 \leq h1 \leq 3.0$
	$1.4 \leq h2 \leq 7.0$
	$1.0 \leq h3 \leq 18.0$
	$0.8 \leq w1 \leq 1.0$
	$2.0 \leq w2 \leq 3.0$
	$0.5 \leq w3 \leq 2.5$

A regression function was estimated from 500 data points of Type 3-1 obtained from the simulation. A regression analysis was performed based on the constructed data to remove variables that did not affect the efficiency, which was the dependent variable. The *p*-value is used as an indicator to check the importance of each independent variable. If the *p*-value is less than 0.05, it is classified as a significant independent variable, and if it is greater than 0.1, it is classified as insignificant [25]. As a result of the regression analysis, the *p*-value of *w*1 was 0.4359 and the *p*-value of *w*2 was 0.3365, so these two variables were excluded from the independent variables.

The coefficient of determination (*R*<sup>2</sup>) is an index indicating the validity and accuracy of the estimated regression function [26], and suitability was judged through the values.

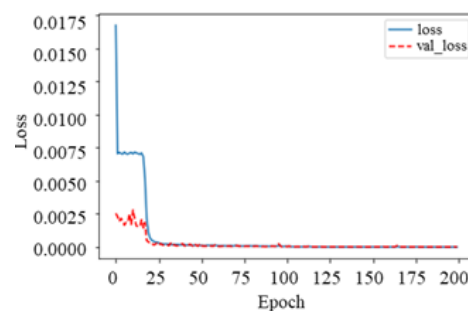


After removing insignificant variables, the regression analysis was repeated with four independent variables,  $h_1$ ,  $h_2$ ,  $h_3$ , and  $w_3$ , to estimate the coefficients of determination of the regression functions. The coefficient of determination of the estimated regression function was obtained as 0.968. Optimized values of each variable were obtained from the obtained regression function and the range of independent variables. Since  $w_1$  and  $w_2$  did not significantly affect the efficiency, they were determined as the most efficient values among the 500 data points of Type 3-1. The derived optimal values are shown in Table 6.

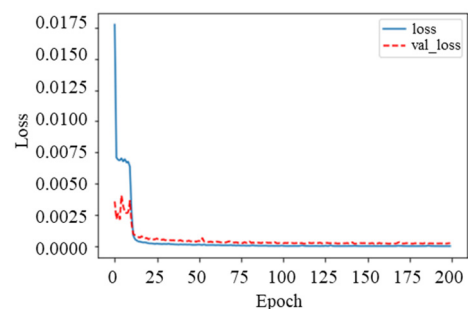
**Table 6.** Optimization result of the Type 3-1 rotor slot.

Design Variables (mm)	Efficiency (%)
$h_1 = 2.997$	89.62
$h_2 = 1.881$	
$h_3 = 12.59$	
$w_1 = 0.973$	
$w_2 = 2.589$	
$w_3 = 1.319$	

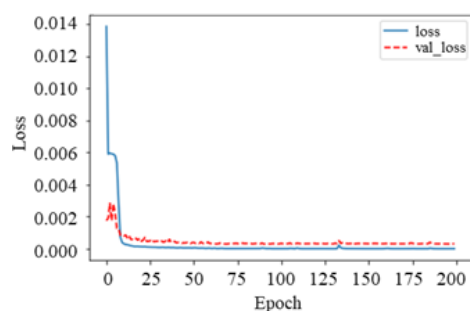
The training data for verifying the prediction accuracy of the CNN model were classified into Case 1, Case 2, and Case 3, and 200 epochs were learned for each case. Figures 7–9 are graphs of the loss according to the epoch in each case, and show the loss for both the training data and the validation data. In each case, it was confirmed that the loss converged to a value close to 0. The verification of the CNN model was conducted for the optimal shapes of Test data 1 to Test data 9 and Type 3-1 in each case. Table 7 shows a comparison between the predicted values and the analysis results for the performance variables of the optimal shape in each case. Table 7 compares the performance variables of the optimal shape and the performance values of the CNN-predicted shape for each case. Table 8 is a table that verified the accuracy of the performance values of the CNN prediction shape for each case obtained from Table 7 through the root mean square error (RSME), mean absolute percentage error (MAPE), and mean absolute error (MAE).



**Figure 7.** Loss for training data and validation data in Case 1.



**Figure 8.** Loss for training data and validation data in Case 2.



**Figure 9.** Loss for training data and validation data in Case 3.

**Table 7.** Comparison of results for Type 3-1 optimal shape and predicted shape in Cases 1, 2, and 3.

	Optimal Shape	Case 1	Case 2	Case 3
Efficiency (%)	89.62	88.85	89.38	89.60
Power factor (%)	80.69	78.48	80.35	80.62
Starting torque (Nm)	12.66	12.72	12.57	12.60
Average torque (Nm)	12.87	13.00	12.89	12.81

**Table 8.** Verification of prediction accuracy through RMSE, MAPE, and MAE.

	Method	Case 1	Case 2	Case 3
Efficiency	RMSE	0.544	0.170	0.014
	MAPE	0.430	0.134	0.011
	MAE	0.385	0.120	0.010
Power factor	RMSE	1.563	0.064	0.049
	MAPE	1.369	0.355	0.043
	MAE	1.105	0.042	0.035
Starting torque	RMSE	0.042	0.064	0.042
	MAPE	0.237	0.355	0.237
	MAE	0.030	0.045	0.030
Average torque	RMSE	0.092	0.028	0.042
	MAPE	0.505	0.155	0.233

At 200 epochs in Case 1, the loss on the training data was  $5.46 \times 10^{-6}$ , and the loss on the validation data was  $1.77 \times 10^{-5}$ . Case 1 showed a very high level of accuracy, even though only a relatively simple shape was learned. However, for the power factor, the error rate was 2.74%, confirming that there was a limit to accurately predicting the performance variables of complex shapes, such as double cage slots, with only the basic shape of Case 1. At 200 epochs in Case 2, the loss on the training data was  $4.95 \times 10^{-6}$  and the loss on the validation data was  $3.12 \times 10^{-4}$ . Case 2 was a case in which the individual data of the upper and lower parts of the double cage slot were learned to check the predictability of the combined shape. For this, in Case 2, information about the upper part of the double cage slot was additionally learned in Case 1, where only the existing simple shape was learned. It was confirmed that Case 2 accurately predicted all of the performance variables to an integer digit compared to Case 1. At 200 epochs in Case 3, the loss on the training data was  $4.51 \times 10^{-6}$ , and the loss on the validation data was  $2.33 \times 10^{-4}$ . Case 3 was a case in which the double cage slot shape was additionally learned. In Case 3, it was confirmed that predictions to the first decimal place were possible compared with Case 2, where predictions were possible only to an integer digit in all performance variables.

Among the test data composed of various rotor slot shapes from Case 1 to Case 3, the prediction accuracy of Case 3, which learned even the shape of the double cage slot, was the best. However, a considerable level of prediction accuracy was also confirmed in Case 1, which learned only the basic shape, and Case 2, where information on the upper part of the

double cage slot was additionally learned. In addition, according to the prediction results for the optimal shape of Type 3-1, Case 1 showed relatively low accuracy. However, Case 2 was able to predict accurately up to an integer number, and Case 3 was able to predict to one decimal place. Through the results up to Case 3, it was confirmed that the deep learning model constructed in this study accurately predicted the performance variables for the optimized shape compared with the finite element analysis results.

From the above results, various shapes could be predicted with almost the same level of accuracy as the finite element analysis results according to the combination of shapes and learning data learned from various shapes with a relatively small number of training data points. It was also confirmed that the performance variables of the shape could be accurately predicted.

#### 4. Conclusions

In this study, based on the CNN technique, a deep learning technique for predicting the efficiency, power factor, starting torque, and average torque according to various rotor slot types of induction motors was proposed. Predictions for various double cage slot shapes were compared using 7500 training data points classified into three cases, and performance variables were predicted with similar accuracy to the analysis results. In addition, although the basic shape of an induction motor alone had few limitations in accurately predicting the performance variables of complex shapes, learning additional information about the basic shape led to high-accuracy prediction results. Through the research results, it was confirmed that the CNN model learned the trends of the performance variables according to the slot type and made predictions with an accuracy similar to the analysis results.

Deep learning has the advantages of self-learning data, computational history, and data characteristics. According to the results derived from this study, it can be seen that, if the performance and shape variables accumulated in the deep learning model are trained, the resulting values can be predicted by self-learning the trend for the resulting values according to the characteristics of the deep learning model. Additionally, this approach can predict almost the same level of results as the data and finite element analysis values. This means that, when a deep learning model with high predictive accuracy is built, it is possible to accurately predict the analysis results without an experienced designer with high ability to utilize the simulation software. Therefore, it is expected that the proposed deep learning method can be applied to various fields with a relatively small number of experienced designers.

**Author Contributions:** Conceptualization, D.-Y.K., S.-J.J. and S.-Y.H.; methodology, D.-Y.K., S.-J.J. and S.-Y.H.; software, D.-Y.K. and S.-J.J.; validation, D.-Y.K., S.-J.J. and S.-Y.H.; formal analysis, S.-J.J.; writing—original draft preparation, D.-Y.K.; writing—review and editing, D.-Y.K., S.-J.J. and S.-Y.H.; visualization, D.-Y.K.; supervision, S.-Y.H.; project administration, S.-Y.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and granted financial resources from the Ministry of Trade, Industry, and Energy, Republic of Korea (No. 20202020800200 and No. 20192010106780).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data used for this article were collected by the research team and will be given to other researchers upon request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Koo, Y.D.; Chun, D.H. Technical trend of High efficiency MEPS in Korea. *J. Korean Inst. Illum. Electr. Install. Eng.* **2011**, *25*, 26–37.
2. Ferreira, F.J.; de Almeida, A.T. Novel multflux level, three-phase, squirrel-cage induction motor for efficiency and power factor maximization. *IEEE Trans. Energy Convers.* **2008**, *23*, 101–109. [[CrossRef](#)]
3. Lee, G.; Min, S.; Hong, J.P. Optimal Shape Design of Rotor Slot in Squirrel-Cage Induction Motor Considering Torque Characteristics. *IEEE Trans. Magn.* **2013**, *49*, 2197–2200. [[CrossRef](#)]
4. Li, Y.; Li, S.; Sarlioglu, B. Analysis of pulsating torque in squirrel cage induction machines by investigating stator slot and rotor bar dimensions for traction applications. In Proceedings of the IEEE Energy Conversion Congress and Exposition, Denver, CO, USA, 15–19 September 2013; pp. 246–253.
5. Mohamed, M.Y.; Maksoud, S.A.A.; Fawzi, M.; Kalas, A.E. Effect of poles, slots, phases number and stack length changes on the optimal design of induction motor. In Proceedings of the Nineteenth International Middle East Power Systems Conference (MEPCON), Cairo, Egypt, 19–21 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 466–471.
6. Mellah, H.; Hemsas, K.E. Design and simulation analysis of outer stator inner rotor DFIG by 2d and 3d finite element methods. *Int. J. Electr. Eng. Technol.* **2021**, *3*, 457–470.
7. Fireteanu, V.; Tudorache, T.; Turcanu, O.A. Optimal design of rotor slot geometry of squirrel-cage type induction motors. In Proceedings of the IEEE International Electric Machines & Drives Conference, Antalya, Turkey, 3–5 May 2007; Volume 1, pp. 537–542.
8. Zhang, D.; Park, C.S.; Koh, C.S. A new optimal design method of rotor slot of three-phase squirrel cage induction motor for NEMA class D speed-torque characteristic using multi-objective optimization algorithm. *IEEE Trans. Magn.* **2012**, *48*, 879–882. [[CrossRef](#)]
9. Marfoli, A.; Di Nardo, M.; Degano, M.; Gerada, C.; Chen, W. Rotor design optimization of squirrel cage induction motor-part i: Problem statement. *IEEE Trans. Energy Convers.* **2020**, *36*, 1271–1279. [[CrossRef](#)]
10. Liang, L.; Liu, M.; Martin, C. Sun, W. A deep learning approach to estimate stress distribution: A fast and accurate surrogate of finite-element analysis. *J. R. Soc. Interface* **2018**, *15*, 138. [[CrossRef](#)] [[PubMed](#)]
11. Khan, A.; Ghorbanian, V. Lowther, D. Deep learning for magnetic field estimation. *IEEE Trans. Magn.* **2019**, *55*, 1–4.
12. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
14. Fukushima, K.; Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*; Springer: Berlin/Heidelberg, Germany, 1982; pp. 267–285.
15. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
16. Sasaki, H.; Igarashi, H. Topology optimization accelerated by deep learning. *IEEE Trans. Magn.* **2019**, *55*, 1–5. [[CrossRef](#)]
17. Asanuma, J.; Doi, S.; Igarashi, H. Transfer learning through deep learning: Application to topology optimization of electric motor. *IEEE Trans. Magn.* **2020**, *56*, 1–4. [[CrossRef](#)]
18. Chattopadhyay, P.; Saha, N.; Delpha, C.; Sil, J. Deep learning in fault diagnosis of induction motor drives. In Proceedings of the 2018 Prognostics and System Health Management Conference, Chongqing, China, 26 October 2018; pp. 1068–1073.
19. Gabdullin, N.; Madanzadeh, S.; Vilkin, A. Towards End-to-End Deep Learning Performance Analysis of Electric Motors. *Actuators* **2021**, *10*, 28. [[CrossRef](#)]
20. Lehikoinen, A.; Davidsson, T.; Arkkio, A.; Belahcen, A. A high-performance open-source finite element analysis library for magnetics in MATLAB. In Proceedings of the XIII International Conference on Electrical Machines, Alexandroupoli, Greece, 3–6 September 2018; pp. 486–492.
21. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010.
22. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation functions: Comparison of trends in practice and research for deep learnin. *arXiv* **2018**, arXiv:1811.03378.
23. Zhou, Y.T.; Chellappa, R. Computation of optical flow using a neural network. In Proceedings of the IEEE 1988 International Conference on Neural Networks, San Diego, CA, USA, 24–27 July 1988; pp. 71–78.
24. Geuzaine, C.; Remacle, J.F. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *Int. J. Numer. Methods Eng.* **2009**, *79*, 1309–1331. [[CrossRef](#)]
25. Xu, H.; Sun, L.P.; Shi, Y.Z.; Wu, Y.H.; Zhang, B.; Zhao, D.Q. Optimization of cultivation conditions for extracellular polysaccharide and mycelium biomass by *Morchella esculenta* As51620. *Biochem. Eng. J.* **2008**, *39*, 66–73. [[CrossRef](#)]
26. Dodge, Y. *The Concise Encyclopedia of Statistics*; Springer Science & Business Media: Neuchâtel, Switzerland, 2008; pp. 88–90.