International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 11, Number 2, pages 426–437, July 2021

An Image Classification Model that Learns Image Features and Numerical Information

Yuta Suzuki
Computer Software Laboratory, Kanazawa University
Kakuma, Kanazawa 920-1192, JAPAN

Toshiki Hatano
Computer Software Laboratory, Kanazawa University
Kakuma, Kanazawa 920-1192, JAPAN

Toi Tsuneda
Computer Software Laboratory, Kanazawa University
Kakuma, Kanazawa 920-1192, JAPAN

Daiki Kuyoshi
Computer Software Laboratory, Kanazawa University
Kakuma, Kanazawa 920-1192, JAPAN

Satoshi Yamane
Computer Software Laboratory, Kanazawa University
Kakuma, Kanazawa 920-1192, JAPAN

Received: February 14, 2021
Revised: May 5, 2021
Accepted: June 1, 2021
Communicated by Takashi Yokota

Abstract

In recent years, deep neural network technology has been developing rapidly, especially in the field of image recognition. However, since deep neural networks learn images based on pixel values, they can only learn the features of the image and not the meta-information that the image has. In this paper, we focused on the differences between image features and meta-information. For example, "0" and "9" are relatively similar in terms of image characteristics, but there is significant difference in terms of the numbers they actually mean. In contrast, "3" and "4" are relatively dissimilar in terms of image features, but the difference is small in terms of the values they actually mean. In order to solve problems like this example, this paper proposes a method for learning based not only on the features of the image, but also on the numerical information that the image has. Experiments were conducted on the MNIST and Kannada-MNIST datasets using three different models: DNN, CNN, and RNN. As a result, the numerical error is smaller in the proposed model than in the baseline.

Keywords: Deep Learning, Image Recognition, Meta Learning

1 Introduction

In recent years, technologies of machine learning have developed rapidly. Among them, neural network technology has had a great impact on various fields. There are various types of neural networks [1], such as deep neural networks, convolutional neural networks [2], and recurrent neural networks, each of which has shown high performance in the fields of image recognition and natural language processing.

However, when these neural networks learn the training data, they cannot also learn the meta-information that the data has. For example, when a neural network model is trained on MNIST [3], a data set of handwritten numeric images from 0 to 9, the model learns the features of the images, but not their size or meaning as numbers. For this reason, it is easy to make mistakes in image recognition in cases where the image features are relatively similar, such as "0" and "9", but there is a big difference in the actual numbers that are meant. On the other hand, it is difficult to make mistakes in image recognition in cases where the image features are relatively dissimilar, such as "3" and "4", but there is a small difference in the actual numbers that are meant. This is a problem that needs to be solved in current neural network technology. If the meta-information of the data could be utilized for learning, it would expand the possibilities of neural networks.

In this paper, we propose a neural network model that learns to reduce not only the identification error of the image but also the numerical error that the image implies in the 0-9 handwritten numeric image data set.

2 Related Works

In this section, we describe related technologies in this paper.

2.1 MNIST

MNIST (Mixed National Institute of Standards and Technology database) is an image data set that contains 60,000 handwritten numeric images for training and 10,000 images for testing. It is also a data set where the correct answer label is given to the handwritten numbers "0-9", which is a standard data set for image classification tasks.

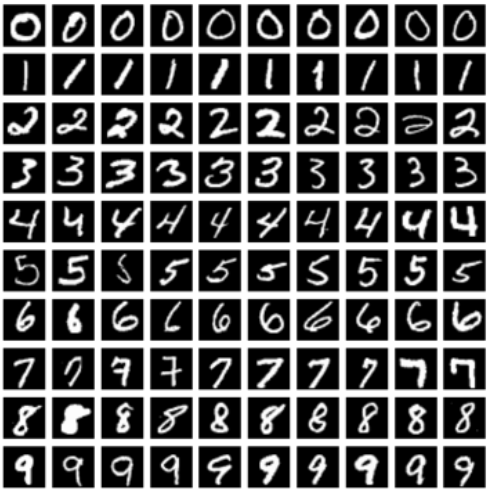

Figure 1: Some of the MNIST images. Images with labels "0", "1", "2", "3", "4", "5", "6", "7", "8", and "9" from the top row.

2.2 Kannada-MNIST

Kannada-MNIST [4] is a new handwritten numeric data set in Kannada that can serve as a direct replacement for the original MNIST data set. Kannada-MNIST has 60,000 handwritten numeric images for training and 10,240 images for testing, called Dig-MNIST.

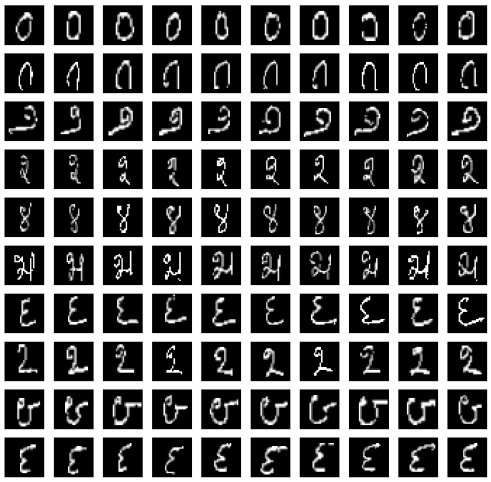

Figure 2: Some of the Kannada-MNIST images. Images with labels "0", "1", "2", "3", "4", "5", "6", "7", "8", and "9" from the top row.

There have been many years of research and discussions on using neural networks to recognize such handwritten characters [5], [6].

2.3 Deep Neural Network Model

Deep Neural Network (DNN) is a neural network with four or more layers deep corresponding to deep learning. Conventional neural networks basically consisted of three layers: one input layer, one hidden layer, and one output layer. Before the advent of deep learning, there was a problem that networks with more than two hidden layers and more than four layers in total would lose accuracy. For this reason, for a long time after the advent of neural networks, basically no more than four layers were used. However, since the advent of deep learning methods, this problem has been overcome and deep neural networks have become widely used.

Convolutional Neural Network (CNN) is a type of DNN, which is characterized by having convolutional and pooling layers inside the network, and is mainly used for image recognition tasks. CNN has become a standard technology in the field of image recognition, and many CNN models have been proposed, including AlexNet [7] in 2012, VGG [8] in 2014, ResNet [9] in 2015, EfficientNet [10] in 2019.

Recurrent Neural Network (RNN) [11] is a type of DNN that has a recurrent structure inside the network, and is mainly used in natural language processing and identification of time series data. RNN has become a standard technology in the field of natural language processing, and RNN models such as LSTM [12] in 1997 and GRU [13] in 2014 have been proposed.

When training these DNN models for the image classification task, the pixel values of the image are used as input. Then, based on the pixel values of the image, the model is trained to reduce the number of false identifications. This makes it easier to make mistakes if the image has similar characteristics, regardless of what the image actually means, such as numerical information in a numeric image. In order to solve this problem, various researches have been done on the recognition of similar images [14], [15]. Some of them incorporate a function that takes into account the similarity

of the images in the loss function to increase the accuracy of identifying similar handwritten text images [16]. There is also research into ensemble methods that combine models that can identify images that are likely to be mistaken for each other to achieve a high overall accuracy, which is very practical [17], [18], [19]. However, these models also learn from the pixel values and are not able to learn the meta-information of the image.

Due to the high utility of similar image recognition and ensemble research, there has been little work on using the meta-information of images, such as numerical information in numeric images, for learning.

3 Proposed Method

In this paper, we focused on the differences between image features and meta-information. For example, "0" and "9" are relatively similar in terms of image characteristics, but there is significant difference in terms of the numbers they actually mean. In contrast, "3" and "4" are relatively dissimilar in terms of image features, but the difference is small in terms of the values they actually mean. However, since these DNN models basically learn only from the features of the image, they cannot learn the meta-information that the image has. In this paper, we propose an image classification model that also learns numerical information in MNIST and Kannada-MNIST.

3.1 Our Proposed Model

Our proposed model is shown in Figure 3. In the usual image classification model, identification loss is calculated at the output layer and the loss is back-propagated and trained. Our proposed model calculates not only the identification loss but also the numerical loss, and learns to reduce the error. Assuming that the identification loss is L_1 and the numerical loss is L_2, the loss function of our proposed model is expressed as (1) using a hyper-parameter β with a value range of 0 to 1. This allows for the learning of numerical information.

$$L = \beta L_1 + (1 - \beta) L_2 \tag{1}$$

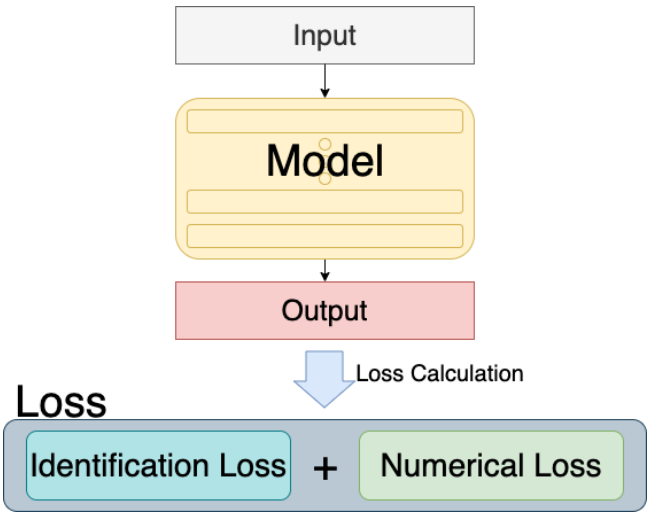

Figure 3: Our Proposed Model

3.2 Algorithm for calculating numerical loss

Algorithm 1 was proposed as a method to calculate the numerical loss. Algorithm 1 requires a one-hot vectorized correct answer label **True**, a prediction result **Predict** and numerical information for each label **Numeric**. In addition, we also provide α as a hyper-parameter to control the effect of the loss.

Algorithm 1

Require: **True**[10], **Predict**[10], **Numeric**[10], α
Ensure: Loss
1: TrueNumeric = **True** · **Numeric**
2: PredictNumeric = **Predict** · **Numeric**
3: NumericLoss = abs(TrueNumeric-PredictNumeric)
4: Loss = α · NumericLoss
5: **return** Loss

The explanation of Algorithm 1 is as follows.

1. The first line calculates the numerical size of the correct label and stores them in TrueNumeric.
2. The second line calculates the numerical size of the prediction result and stores it in Predict-Numeric.
3. The third line calculates the numerical loss between the correct label and the prediction result by calculating the difference between TrueNumeric and PredictNumeric. And it is stored in NumericLoss.
4. In the fourth line, NumericLoss is multiplied by alpha and stored in Loss to control the loss to a value that is easy to learn.
5. The fifth line returns Numerical Loss referred to in Figure 3 as the loss to be learned.

4 Experiments

In this paper, MNIST and Kannada-MNIST were used as the tasks for the evaluation experiments. MNIST was evaluated with 10,000 test images, and Kannada-MNIST was evaluated with Dig-MNIST, which has 10,240 test images.

In order to check the effectiveness of the proposed method for each type of model, three neural network models, DNN, CNN and RNN, were used in the experiments. The MNIST and Kannada-MNIST used in this study are 28x28 gray-scale images, and since the image size is small, the architecture of the models used was set to be simple. The architecture of the DNN model used in the experiment is shown in Figure 4.

For the proposed model, we adopted the loss L, which is the combination of the identification loss (categorical cross entropy) L_1 and the numerical loss L_2 calculated by Algorithm1 as the loss function. The expression of L can be expressed as (1) . When calculating L_2, α was set to 0.05 to adjust the influence of L_1 and L_2 to be comparable. When calculating L, β was set to 0.9 so as not to reduce the identification accuracy, and the percentage of L_1 was set to be large. We also adopted a baseline model with categorical cross entropy as the loss function.

The implementation was done using Python and Keras, and the experiments were conducted on Google Colaboratory.

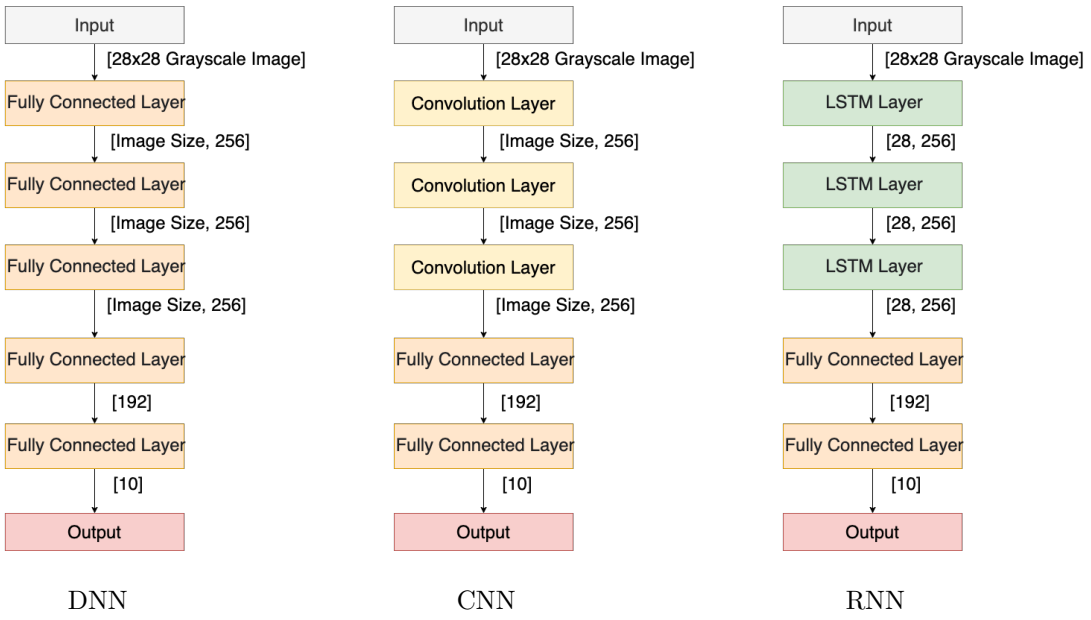

Figure 4: Architecture of the model used in the experiments

5 Results

In this section, we present the experimental results for the following three indicators, and compare and evaluate the baseline and proposed models.

- **Identification Test Error Rate**
  An indicator of the accuracy of identification. By comparing this indicator between the baseline model and our proposed model, we can see how our proposed method affects the accuracy of identification.

- **Numerical Error per One Image**
  An indicator of how much numerical error there was per one test image. By comparing this indicator between the baseline model and our proposed model, we can see whether our proposed method is able to learn to reduce the numerical error.

- **Numerical Error per One Image in the case of Misidentification**
  An indicator of how much numerical error there was per one test image that was misidentified. By comparing this indicator between the baseline model and our proposed model, we can see whether our proposed method is able to learn to reduce the numerical error. Also, since this indicator is limited to cases of misidentification, it can measure numerical errors that are not affected by accuracy.

5.1 Identification Test Error Rate

The results of identification test error rate are shown in Table 1. Table 1 shows that all models showed a higher error rate for Dig-MNIST than for MNIST. This shows that Dig-MNIST is a more difficult task to identify than MNIST. It can also be seen that for both MNIST and Dig-MNIST, the error rate is smaller for CNN, RNN and DNN in that order. There is not such a big difference in the identification test error rate between the baseline and the proposed model, but in some places, the proposed model shows a lower error rate. This can be considered as a pattern in which the error rate was improved by providing information from another direction, namely numerical errors, for images that are difficult to identify based on image features alone.

Table 1: Identification Test Error Rate. The MNIST and Dig-MNIST average and standard deviation results are reported from 3 trials.

Model	Method	MNIST(%)	Dig-MNIST(%)
DNN	baseline	$2.00_{\pm 0.037}$	$32.5_{\pm 0.458}$
	proposed	$2.00_{\pm 0.094}$	$\mathbf{31.8}_{\pm 0.323}$
CNN	baseline	$0.81_{\pm 0.057}$	$21.5_{\pm 0.273}$
	proposed	$\mathbf{0.78}_{\pm 0.053}$	$\mathbf{20.6}_{\pm 0.278}$
RNN	baseline	$0.88_{\pm 0.028}$	$21.9_{\pm 0.260}$
	proposed	$\mathbf{0.78}_{\pm 0.062}$	$22.2_{\pm 0.274}$

5.2 Numerical Error per One Image

The results of Numerical Error per One Image are shown in Table 2. From Table 2, we can see that the numerical error per image is larger for Dig-MNIST than for MNIST. The numerical error per image was similar for CNN and RNN, and the largest for DNN. Table 2 shows that the numerical error per image of MNIST and Dig-MNIST is smaller for the proposed model than the baseline for all of DNN, CNN, and RNN. This indicates that the proposed model is able to learn to reduce the numerical error.

Table 2: Numerical Error per One Image. The MNIST and Dig-MNIST average and standard deviation results are reported from 3 trials.

Model	Method	MNIST	Dig-MNIST
DNN	baseline	$0.076_{\pm 0.0022}$	$1.02_{\pm 0.030}$
	proposed	$\mathbf{0.071}_{\pm 0.0027}$	$\mathbf{0.95}_{\pm 0.023}$
CNN	baseline	$0.032_{\pm 0.0013}$	$0.65_{\pm 0.009}$
	proposed	$\mathbf{0.028}_{\pm 0.0015}$	$\mathbf{0.59}_{\pm 0.005}$
RNN	baseline	$0.033_{\pm 0.0007}$	$0.63_{\pm 0.016}$
	proposed	$\mathbf{0.028}_{\pm 0.0025}$	$\mathbf{0.61}_{\pm 0.014}$

5.3 Numerical Error per One Image in the case of Misidentification

The results of Numerical Error per One Image in the case of Misidentification are shown in Table 3. Table 3 shows that the numerical error per image in the case of misidentification is higher for MNIST than for Dig-MNIST. It can also be seen that the numerical error per image in the case of misidentification varies from model to model. This shows that the pattern of identification is different depending on the dataset and the identification model. Table 3 shows that the numerical error per image in case of misidentification in MNIST and Dig-MNIST is smaller for the proposed model than the baseline for all of DNN, CNN, and RNN. In other words, the proposed model is able to identify with a small numerical error even in the case of misidentification. This shows that the proposed model is able to learn to reduce the numerical error.

Table 3: Numerical Error per One Image in the case of Misidentification. The MNIST and Dig-MNIST average and standard deviation results are reported from 3 trials.

Model	Method	MNIST	Dig-MNIST
DNN	baseline	$3.80_{\pm 0.044}$	$3.14_{\pm 0.049}$
	proposed	$\mathbf{3.56}_{\pm 0.077}$	$\mathbf{2.99}_{\pm 0.084}$
CNN	baseline	$3.93_{\pm 0.124}$	$3.06_{\pm 0.073}$
	proposed	$\mathbf{3.59}_{\pm 0.125}$	$\mathbf{2.90}_{\pm 0.068}$
RNN	baseline	$3.78_{\pm 0.035}$	$2.91_{\pm 0.048}$
	proposed	$\mathbf{3.58}_{\pm 0.036}$	$\mathbf{2.78}_{\pm 0.032}$

5.4 Other Analysis

Figure 5 shows the number of appearances of each numerical error when the model misidentifies in MNIST and Dig-MNIST. The horizontal axis in Figure 5 is the numerical error in the case of misidentification, and the vertical axis is the number of appearances of the numerical error. From Figure 5 it is possible to read the differences in misidentification patterns between MNIST and Dig-MNIST and the differences in misidentification patterns for each model. From Figure 5, we can see that the number of appearances of numerical errors larger than 5 is lower in the proposed model than in the baseline.

Figure 6 shows the appearance rate of each numerical error when the model misidentifies in MNIST and Dig-MNIST. The horizontal axis in Figure 6 is the numerical error and the vertical axis is the appearance rate of the numerical error in the case of misidentification. From Figure 6 it is possible to read the differences in misidentification patterns between MNIST and Dig-MNIST and the differences in misidentification patterns for each model. From Figure 6, it can be seen that the proposed model has a higher appearance rate of small numerical errors such as 1 and 2. To the extent that the appearance rate of small numerical errors is high, the appearance rate of large numerical errors is relatively small.

These results also show that our proposed model is able to learn to make the numerical error smaller than the baseline.

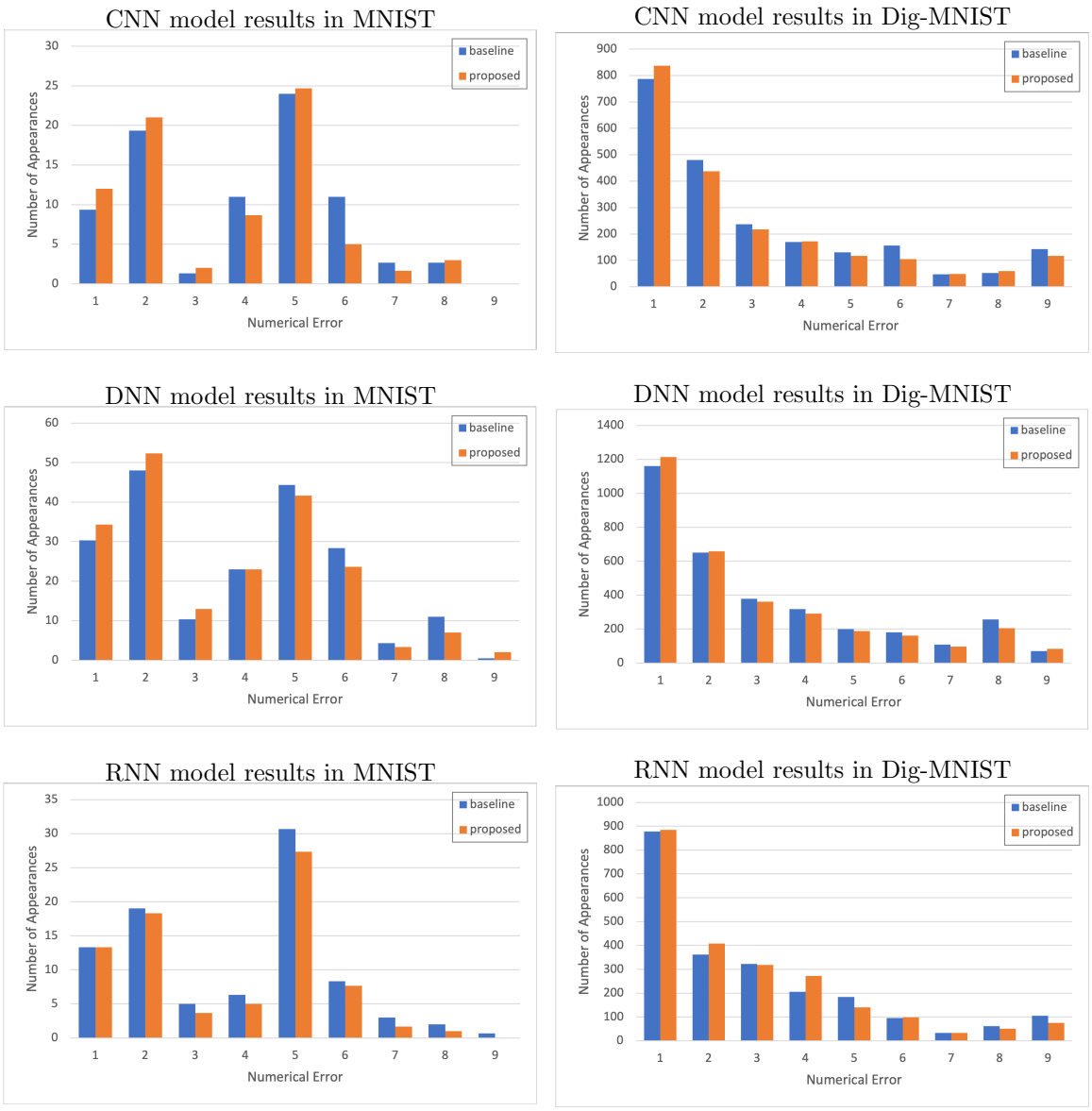

Figure 5: Number of Appearances of Each Numerical Error

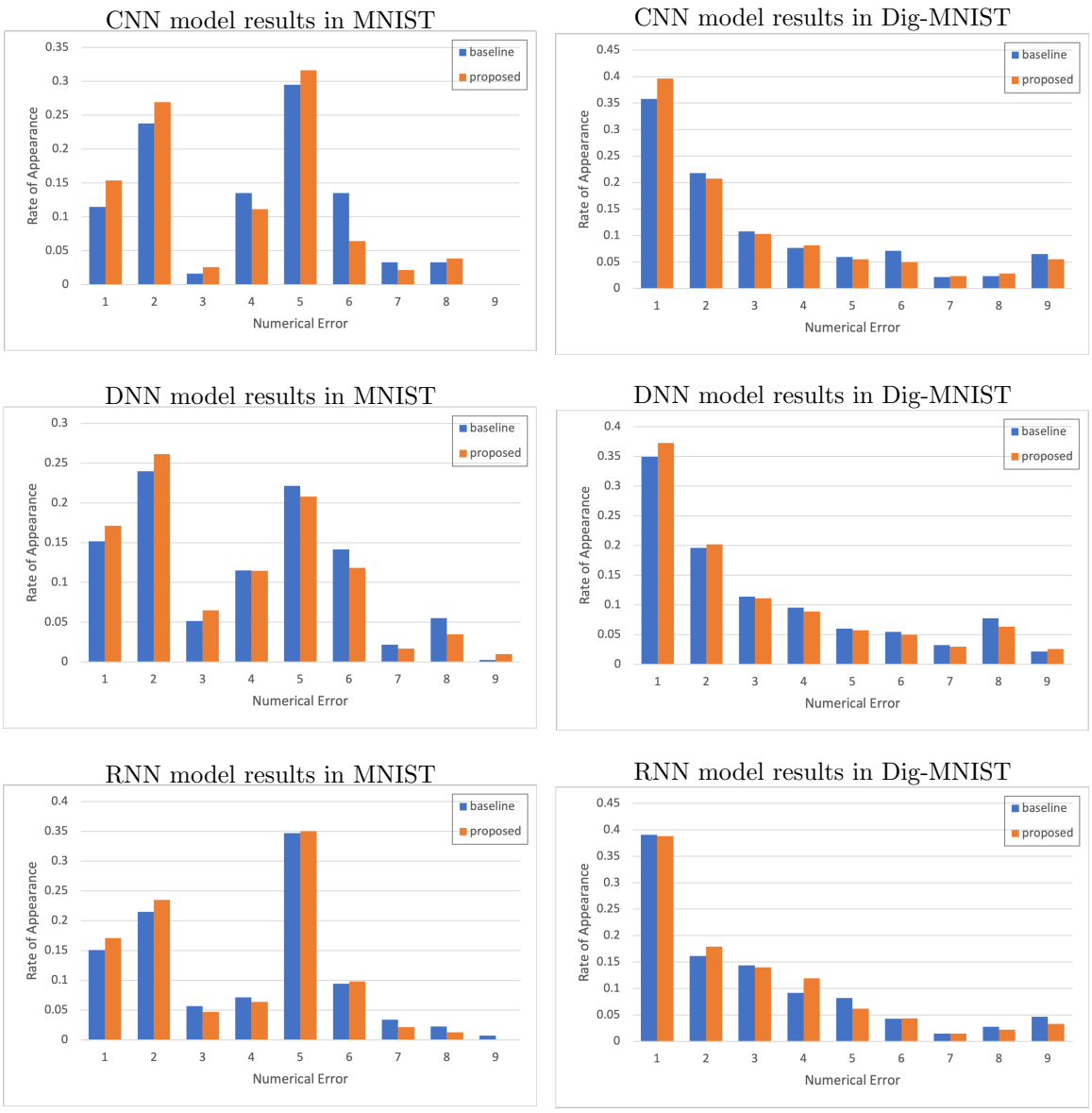

Figure 6: Rate of Appearance of Each Numerical Error

6 Conclusion and Discussion

In this paper, we propose a method for learning based not only on the features of the image, but also on the numerical information that the image has. Experiments were conducted on the MNIST and Kannada-MNIST datasets using three different models: DNN, CNN, and RNN. Identification Test Error Rate, Numerical Error per One Image and Numerical Error per One Image in the case of Misidentification were set as evaluation indicators, and the baseline model and our proposed model were compared and evaluated. As a result, our proposed model performed as well as or better than the baseline in Identification Test Error Rate. In Numerical Error per One Image and Numerical Error per One Image in the case of Misidentification, our proposed model showed higher performance than the baseline. From the results of the experiments, we can say that our proposed model is able to learn so that the numerical error is small. In addition, the proposed model is different from the usual models in that it learns information other than pixel values, so unique error patterns can be expected. Therefore, proposed model is considered to be effective for ensemble methods that combine models with various error patterns to improve performance.

References

[1] Yann LeCun, et al. "Backpropagation applied to handwritten zip code recognition." Neural computation vol. 1 no. 4, pp. 541-551, 1989.

[2] Kunihiko Fukushima: "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", Biological Cybernetics, 36[4], pp. 193-202, 1980.

[3] Yann LeCun, Corinna Cortes and Christopher J.C. Burges, "THE MNIST DATABASE of handwritten digits", in Proc. of the IEEE, 1998.

[4] Vinay Uday Prabhu, "Kannada-MNIST: A new handwritten digits dataset for the Kannada language", arXiv:1908.01242, 2019.

[5] M.Tateishi, H.Yamazaki, "A Consideration for the Hidden Layer of the Multilayered Neural Network for Hand - written Numeral Recognition", IPSJ Journal, Vol.30(10), pp.1281-1288, 1989.10 (In Japanese).

[6] Iwata et.al, "Handwritten Zip Code Recognition Using Neural Network", IEICE Technical Report, PRU95-2, pp.9-16, 1995.

[7] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", in Proc. of NIPS, 2012.

[8] Karen Simonyan, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556v6, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", arXiv:1512.03385v1, 2015.

[10] Mingxing Tan, Quoc V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", arXiv:1905.11946v3, 2019.

[11] Zachary C. Lipton, John Berkowitz, Charles Elkan, "A critical review of recurrent neural networks for sequence learning", arXiv:1506.00019, 2015.

[12] Sepp Hochreiter and Jurgen Schmidhuber , "Long short-term memory", Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", arXiv:1406.1078, 2014.

[14] T. Nakajima, T. Wakabayashi, F. Kimura, and Y. Miyake, “Accuracy Improvement by Compound Discriminant Functions for Resembling Character Recognition”, IEICE Trans., Vol. J83-D2, No. 2, pp. 623-633 (2000-2) (in Japanese).

[15] S. Chopra, R. Hadsell and Y. LeCun, ”Learning a similarity metric discriminatively, with application to face verification”, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), 2005, pp. 539-546 vol. 1, doi: 10.1109/CVPR.2005.202.

[16] Junyi Zou, Jinliang Zhang, Ludi Wang, ”Handwritten Chinese Character Recognition by Convolutional Neural Network and Similarity Ranking”, arXiv:1908.11550, 2019.

[17] Daniel Keysers, ”Comparison and Combination of State-of-the-art Techniques for Handwritten Character Recognition: Topping the MNIST Benchmark”, arXiv:0710.2231v1, 2007.

[18] Siham Tabik, Ricardo F. Alvear-Sandoval, María M. Ruiz, José-Luis Sancho-Gómez, Aníbal R. Figueiras-Vidal, Francisco Herrera, ”MNIST-NET10: A heterogeneous deep networks fusion based on the degree of certainty to reach 0.1% error rate. Ensembles overview and proposal”, Information Fusion, Volume 62, 2020, Pages 73-80, ISSN 1566-2535.

[19] Abdul Wasay and Stratos Idreos, ”More or Less: When and How to Build Convolutional Neural Network Ensembles”, International Conference on Learning Representations (ICLR2021), 2021.