Available *online* at : http://jitce.fti.unand.ac.id/
JITCE (Journal of Information Technology and Computer Engineering)
| ISSN (Online) 2599-1663 |

Research Article

Deep Learning-Based Dzongkha Handwritten Digit Classification

Yonten Jamtsho [1], Pema Yangden [1], Sonam Wangmo[1], Nima Dema[1]

[1] *Gyalpozhing College of Information Technology, Royal University of Bhutan, Thimphu, Bhutan*

ARTICLE INFORMATION

Received: December 12th, 2022
Revised: March 16th, 2024
Available online: March 31st, 2024

KEYWORDS

Bhutan, Dzongkha digits, Deep learning, CNN, Pattern Recognition

CORRESPONDENCE

Phone: +975 17762973
E-mail: yontenjamtsho.gcit@rub.edu.bt

A B S T R A C T

In computer vision applications, pattern recognition is one of the important fields in artificial intelligence. With the advancement in deep learning technology, many machine learning algorithms were developed to tackle the problem of pattern recognition. The purpose of conducting the research is to create the first-ever Dzongkha handwritten digit dataset and develop a model to classify the digit. In the study, the 3 layer set of CONV → ReLU → POOL, followed by a fully connected layer, dropout layer, and softmax function were used to train the digit. In the dataset, each class (0-9) contains 1500 images which are split into train, validation, and test sets: 70:20:10. The model was trained on three different image dimensions: 28 by 28, 32 by 32, and 64 by 64. Compared to image dimensions 28 by 28 and 32 by 32, 64 by 64 gave the highest train, validation, and test accuracy of 98.66%, 98.9%, and 99.13% respectively. In the future, the sample of digits needs to be increased and use the transfer learning concept to train the model.

INTRODUCTION

Dzongkha is the national language of Bhutan, and its dialect is the native mother tongue of eight western districts. With the introduction of Dzongkha computing technology, a Dzongkha Linux operating system has been launched along with Optical Character Recognition (OCR), Dzongkha lexicon, and text-to-speech (TTS) [1]. OCR technology, enables the computer to convert different types of documents captured by a digital camera to editable data. OCR approaches are becoming so popular due to the rapid growth of the touch screen devices industry [2]. It is easy for the OCR technology to recognize computer-typed digits but when it comes to handwritten numbers, building a highly reliable recognition system is challenging due to the variations and distortion of handwritten characters. Handwritten digit recognition is one of the practical issues in the pattern recognition application.

So, therefore, it has become vital in Bhutan to develop an OCR application that can recognize handwritten digits. Before developing the fully functional system, it is crucial to create a Dzongkha handwritten digit dataset and develop a deep learning model which can be generalized on the new handwritten digits. Many researchers have exploited the application of handwritten OCR in various languages such as MNIST [3 - 5], Thai [6 - 8], Nepali [9 - 10], Chinese [11 - 12], etc., but no one has conducted research in the Bhutanese context, and there are no available datasets.

Therefore, this research aims to create the first-ever Dzongkha handwritten digit dataset and use deep learning technology to create a model which can classify the digits. From this research, other Bhutanese researchers can use the dataset to create their model using machine learning or deep learning algorithms. The handwritten digital classification applications are used in financial institutions, education, etc. The sample Dzongkha handwritten digit is shown in Figure 1. The equivalent of Dzongkha to English digit is shown in Table 1.

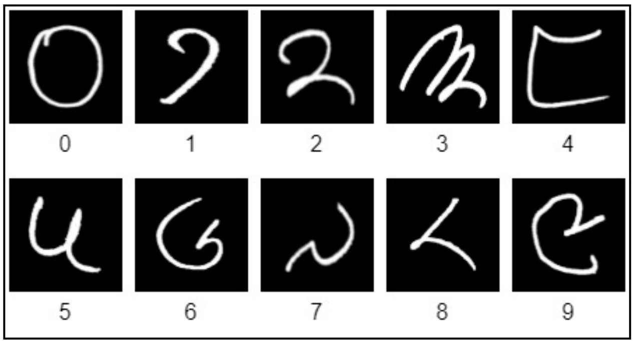

Figure 1. Sample Dzongkha handwritten digits

https://doi.org/10.25077/jitce.8.01.1-7.2024

Table 1: Equivalent Dzongkha to English digit

Dzongkha	English	Dzongkha	English
༠	0	༥	5
༡	1	༦	6
༢	2	༧	7
༣	3	༨	8
༤	4	༩	9

Related Work

The authors of [11] conducted research on the classification of handwritten Dzongkha alphabets using the concept of a convolutional neural network. The researchers proposed 4 convolutional layers with a kernel size of 3 by 3 and achieved an accuracy of 97.22%. They created the first event handwritten alphabet digit dataset consisting of 30 classes.

Khunratchasana and Treenuntharath proposed a method using convolutional neural networks to classify and recognize Thai handwritten digits. They introduced the Thai digit dataset and assessed performance via cloud computing. With a 96.88% accuracy and 0.1075 loss, their study highlights CNN's efficiency in image recognition [12]. The authors in [6] explored handwritten character recognition challenges, emphasizing unique individual handwriting and character similarities in languages. They compared deep Convolutional Neural Networks (CNNs) using the THI-C68 dataset, evaluating two training methods: Train from Scratch and Transfer Learning with VGGNet-19 and Inception-ResNet-v2. Results favored VGGNet-19 with transfer learning, achieving 99.20% recognition efficiency.

The authors in [13] introduced a modified version of the established LeNet convolutional neural network (CNN) architecture tailored for classifying Bangla and Hindi numerals. Their research specifically addressed translational invariance and to some extent rotational invariance during the recognition phase. The experimental results demonstrated impressive accuracy rates, with 98.2% achieved for Bangla numerals and 98.8% for Hindi numerals. This suggests the effectiveness of their modifications in accurately classifying numerals from these languages, showcasing promising results for practical applications such as optical character recognition (OCR) or digit classification systems. Sufian et al. built a model, BDNet for Bengali Handwritten numerical digits based on densely connected CNN [14]. The authors collected 23392 digits from ISI Bengali handwritten dataset written by 1106 persons along with 1000 datasets. From each class, 400 digits were used as a testing set and the remaining digits were used as a training set.

Alkhawaldeh et al. [15] introduced an Ensemble Deep Transfer Learning (EDTL) model for recognizing handwritten Arabic digits, addressing challenges of diverse styles and noise. By combining two pre-trained transfer learning models, the EDTL model achieves up to 99.83% accuracy through extensive dataset training. This innovative approach offers promising advancements in accuracy and robustness, validated through rigorous experimentation, with potential for various real-world applications. The authors in [16] proposed introduced a novel method for recognizing handwritten Arabic digits, which was crucial for various real-life applications. Despite extensive research in pattern recognition, Arabic digit recognition had remained challenging. Their approach utilized pre-trained convolutional neural networks, specifically the ResNet-34 model, through transfer learning. Employing the MADBase dataset, their method achieved a remarkable 99.6% accuracy, surpassing previous approaches, highlighting its significance for Arabic language applications.

The authors in [17] proposed convolutional neural networks based on LeNet-5 to train the model using MNIST datasets. The OpenCV computer vision library was used to perform image processing and feature extraction on the digits. Similarly, [18] used the same datasets for the recognition of handwritten digits using seven-layered CNN and trained the network for 15 epochs.

METHOD

System Overview

The typical overview of the proposed system is shown in Figure 2. The handwritten digit datasets needed for this research are collected from the students and then pre-processed to create a train, validation, and test sets. After splitting the dataset, the digits are trained using the convolutional neural networks, and the efficiency of the model is evaluated using the confusion matrix and classification report. The model is saved and then tested on the unseen handwritten digit to check the correctness and robustness of the trained model.

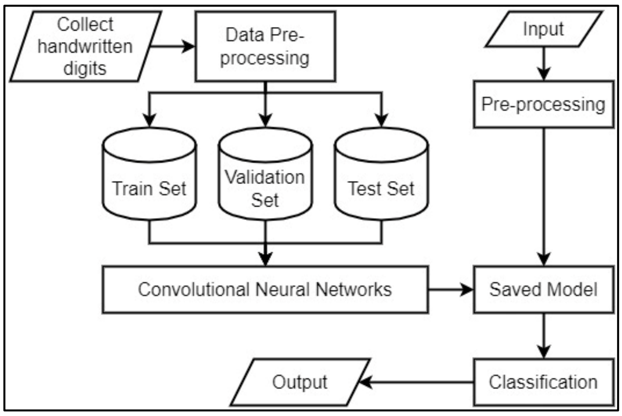

Figure 2. System architecture

Datasets

Currently, no handwritten digit dataset is available for the research. However, the researchers in [19] have created a dataset for handwritten Dzongkha alphabets which consist of 30 classes. The handwritten digit data was collected from the 648 students (primary, higher, and college students). The collected images were all converted to binary format since color information does not play any significant role in the identification of the digits and it is easy for the algorithm to train the dataset. In the datasets, each class (0 to 9) contains 1500 images (a total of 15000 images). The process involved in the dataset creation is given in Figure 3.

 https://doi.org/10.25077/jitce.8.01.1-7.2024

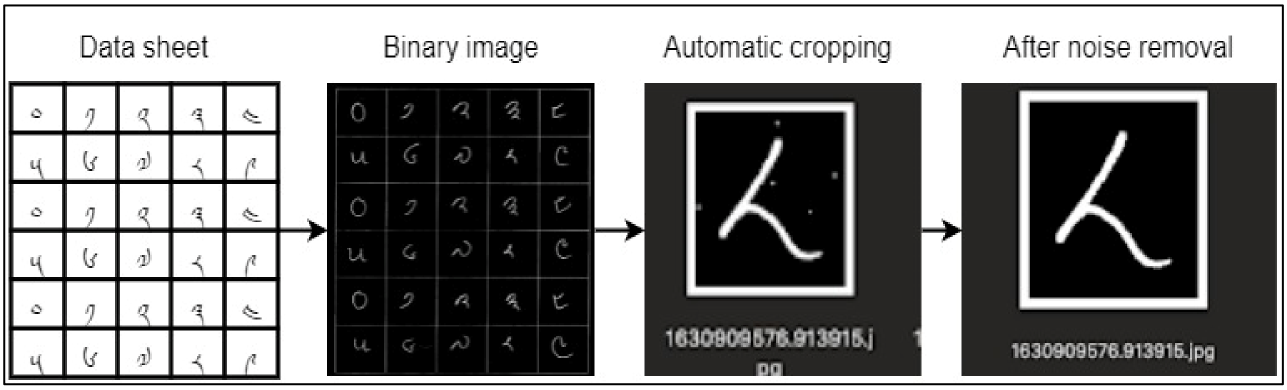

Figure 3. Dataset preparation process

For the collection of handwritten digits from the students, the datasheet containing 5 columns and 6 rows (20 small boxes on A4 size paper) was used. In the sheet, the participants were asked to write the Dzongkha digits (0 to 9) thrice using a pencil. After the collection of the data, the sheets were scanned using the scanner and saved in the image format. After scanning the datasheet, the scanned images were all converted to binary format (black background with white font) since the required digit should be in white. For image binarization, Otsu's method [20] was used since it automatically calculates the threshold value from the image histogram, for a bimodal image. Then from the binarized image, the individual digits were extracted using the following pseudo-code:

Algorithm 1: Automatic extraction of digits from the datasheet

INPUT: Binary image
OUTPUT: Individual characters (0 - 9)
img ← read the thresholded datasheet
grayscale ← convert the datasheet to grayscale
contours ← find he contours from the thresholded image
N ← total number of contours in the image
for i ← 0 to N **do**
    area ← find the area of the contour at location i
    if area > thresholded_value **do**
        boundingBox ← calculate the coordinates of the bounding box of counter i
          cropped ← crop the bounding to get the digit

After the extraction of individual digits from the datasheet, it still contains some unwanted noises (white dots) which will hamper the identification of the digits. Therefore, the noises need to be removed before the development of the model and we have to make sure that only the digit is present in the image. To accomplish the task, the following pseudo-code was used to remove the noises:

Algorithm 2: Automatic removal of noise (whit dots) from the digit

INPUT: List of extracted digits
OUTPUT: List of digits without noise
N ← total number of digits in the folder
for i ← 0 to N **do**
    img ← read each digit in location *i*
    grayscale ←convert the image to grayscale
    contours ← find the contours from the digit
    M ← find total number contours on the digit
    for j ← 0 to M **do**
        area ← find the area of the contour at location *j*
        if area < threshold_value **do**
            boundingBox ← calculate the coordinates of the bounding box
     boundingBoxFill ← fill the bounding box with black pixels

Model Development

After the preparation of the dataset, the handwritten digits dataset were split into three sets: train, validation, and test sets. For the dataset split, 10:2070 ration was used. 10% of the dataset is kept aside as the test data for testing the performance of the model after the training process. In the research, the model was trained using a convolutional neural network: three-layer sets of **CONV → ReLU → POOLING** before the fully connected layer as shown in Figure 4. To check the performance of the proposed model, three distinct experiments were performed by varying the image dimensions (28 by 28, 32 by 32, and 64 by 64). The proposed CNN architecture is shown in Table 2.

In the first and second CONV → ReLU → POOLING block, 32 filters of size 3 by 3, and in the third CONV → ReLU → POOLING block, 64 filters of size 3 by 3 were used. The CONV layers are the basic building blocks since they contain learned kernels (filters), which extract features that distinguish images from another. The pooling layer performs nonlinear down-sampling and reduces the number of parameters and computation of the network. The fully connected layers, followed by the dropout and softmax function help the models classify the digit after the feature extraction. The dropout layers prevent overfitting the training data. Dropout layers drop some of the neurons which does not add value to the next hidden layer.

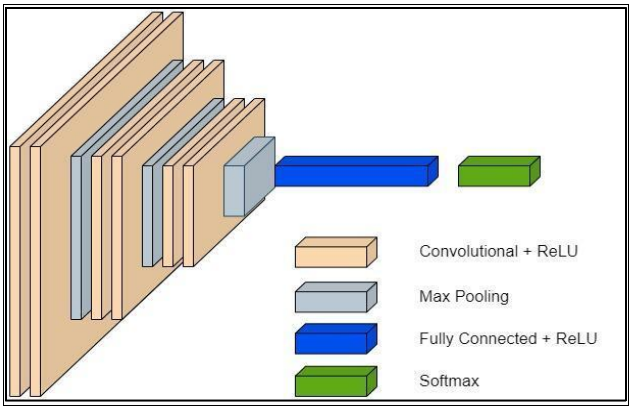

Figure 4. Dataset preparation process

Table 2. CNN architecture

First Layer Set	**Conv2D**	32 filters, 3 by 3 filter size, same padding
	ReLU	
	Max Pool	2 by 2 size
Second Layer Set	**Conv2D**	64 filters, 3 by 3 filter size, same padding
	ReLU	
	Max Pool	2 by 2 size
Third Layer Set	**Conv2D**	64 filters, 3 by 3 filter size, same padding
	ReLU	
	Max Pool	2 by 2 size
FC	**Flatten**	
	Dense	64
	ReLU	
	Dropout	0.5
	Softmax	

In this research, the OpenCV [21] library and the Keras [22] deep learning library were used for image pre-processing and deep learning tasks respectively. Both the libraries are based on Python programming language. Training a deep neural network can take a variable amount of time depending on the dataset size and processing power. In the research, the cloud-based GPU (Google Colaboratory) was used to train the model. The advantage of using Google Colaboratory: firstly, it is open-source, and secondly, it has all the required pre-requisites needed by the deep learning technology.

Evaluation Metrics

In the deep learning technology, the accuracy of the model is analysed through the use of confusion or classification matrix. It is used to describe the performance of the model based on the test data in which the true values are known. The confusion matrix is also used in measuring recall, precision, accuracy and f1-score. There are some terminologies to be known in the confusion matrix (refer Table 3).

Table 3. Confusion matrix

		Predicted Class	
		YES	**NO**
Actual Class	**YES**	True Positive (TP)	False Negative (FN)
	NO	False Positive (FP)	True Negative (TN)

Using the confusion matrix, the following evaluation metrics are calculated by using equation (1), (2), (3), and (4).

$$Accuracy = \frac{TP}{(TP + TN + FN + FP)} \tag{1}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

$$F1 - Score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{4}$$

RESULTS AND DISCUSSION

The proposed CNN architecture was trained on three image dimensions: 28 by 28, 32 by 32, and 64 by 64. The dataset was split into three sets: train, validation and test sets. The performance of the model with respect to each image dimension is shown in Table 4.

Table 4. Model performance obtained from the different image dimensions

Model performance for image dimension 28 by 28	
Train	98.59%
Validation (Seen)	98.73%
Test (Unseen)	99%
Model performance for image dimension 32 by 32	
Train	98.16%
Validation (Seen)	98.37%
Test (Unseen)	98.33%
Model performance for image dimension 64 by 64	
Train	98.86%
Validation (Seen)	98.9%
Test (Unseen)	99.13%

https://doi.org/10.25077/jitce.8.01.1-7.2024

In Table 4, the model trained on image dimensions 64 by 64 outperformed the other two models which were trained on 28 by 28 and 32 by 32. The train, validation, and test accuracy generated by the dimension is 98.66%, 98.9%, and 99.13%, respectively. The confusion matrix generated by the model for test data (unseen) is given in Table 5 (28 by 28), Table 6 (32 by 32), and Table 7 (64 by 64).

Table 5. Confusion matrix generated by model for image size 28 by 28

		Predicted Class									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Actual Class	**0**	150	0	0	0	0	0	0	0	0	0
	1	0	150	0	0	0	0	0	0	0	0
	2	0	0	146	3	0	1	0	0	0	0
	3	0	0	0	149	0	1	0	0	0	0
	4	0	1	0	0	145	0	0	1	1	2
	5	0	0	0	1	0	148	0	1	0	0
	6	0	0	0	0	0	0	145	0	0	0
	7	0	0	0	0	0	0	0	149	0	1
	8	0	0	0	0	0	0	0	0	150	0
	9	0	0	2	0	0	0	0	0	0	148

Table 6. Confusion matrix generated by model for image size 32 by 32

		Predicted Class									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Actual Class	**0**	150	0	0	0	0	0	0	0	0	0
	1	0	150	0	0	0	0	0	0	0	0
	2	0	0	146	3	1	0	0	0	0	0
	3	0	0	1	148	0	1	0	0	0	0
	4	0	1	1	0	142	0	0	0	4	2
	5	0	0	1	1	0	147	0	1	0	0
	6	0	0	0	0	0	0	150	0	0	0
	7	0	0	0	0	0	0	0	149	0	1
	8	0	0	0	0	0	0	0	0	150	0
	9	1	0	2	0	0	0	4	0	0	143

Table 7. Confusion matrix generated by model for image size 64 by 64

		Predicted Class									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Actual Class	**0**	150	0	0	0	0	0	0	0	0	0
	1	0	150	0	0	0	0	0	0	0	0
	2	0	0	149	0	0	0	0	0	1	0
	3	0	0	1	148	0	1	0	0	0	0
	4	0	1	1	0	145	0	0	0	0	3
	5	0	0	0	1	0	149	0	0	0	0
	6	0	0	0	0	0	0	150	0	0	0
	7	0	0	0	0	0	0	0	148	0	2
	8	0	0	0	0	0	0	0	0	150	0
	9	0	0	1	0	0	0	1	0	0	148

In Table 5, the highest misclassifications were found in handwritten digit 4 and then followed by digits 2 and 9. Similarly, in Table 6, digit 4 got the highest misclassification, followed by digits 2, 5, and 9. However, in Table 7, the misclassifications have been reduced drastically compared to Table 5 and Table 6. The classification report generated by the trained model for each image dimension is shown in Table 8 - 10. A classification report is a performance evaluation metric in machine learning. It gives the precision, recall, and f1-score of the trained model which takes care of the false positives and false negatives.

Table 8. Classification report for the test set from the image size 28 by 28

Class	Precision (%)	Recall (%)	F1-Score (%)
0	100	100	100
1	99	100	99
2	99	97	98
3	97	99	98
4	100	97	98
5	99	99	99
6	100	100	100
7	99	99	99
8	99	100	99
9	98	99	98

Table 9. Classification report for the test set from the image size 32 by 32

Class	Precision (%)	Recall (%)	F1-Score (%)
0	99	100	99
1	99	100	99
2	97	97	97
3	97	99	98
4	99	95	97
5	99	98	98
6	97	100	98
7	99	99	99
8	97	100	98
9	98	95	96

Table 10. Classification report for the test set from the image size 64 by 64

Class	Precision (%)	Recall (%)	F1-Score (%)
0	100	100	100
1	99	100	99
2	98	99	98
3	99	99	99
4	100	97	98
5	99	99	99
6	99	100	99
7	100	99	99
8	99	100	99
9	97	99	98

From the Tables, most of the handwritten digits have low false positive and false negative rates. Therefore, the proposed model gave the maximum correct predictions for the handwritten digits. The accuracy generated by the proposed model for the Dzongkha handwritten digit datasets cannot be compared with the previous works since the dataset used in this research was created for the first time.

The simple user interface was designed for a user to draw the digit and the system tries to identify the digit. The sample UI is given in Figure 5.

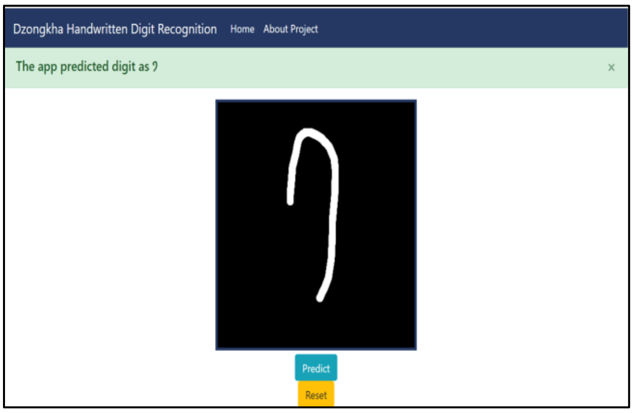

Figure 5. Interface showing the recognition of the digit

CONCLUSIONS

The purpose of this study was to create the first-ever Bhutanese handwritten digit dataset and create a model to classify the digit using deep learning technology. The proposed CNN architecture is composed of 3 3-layer sets of CONV → ReLU → POOL followed by a fully connected layer, a dropout layer, and the softmax function. The model was trained on three different image dimensions: 28 by 28, 32 by 32, and 64 by 64. The model trained on image dimensions 64 by 64 gave the highest train, validation, and test accuracy of 98.66%, 98.9%, and 99.13% compared to image dimensions 28 by 28 and 32 by 32.

The dataset created for this research will act as the benchmark for future researchers. In the future, more samples of handwritten digits need to be collected and trained using the transfer learning models. Moreover, the research can be extended to Bhutanese handwritten characters (alphabets).

ACKNOWLEDGMENT

The authors would like to thank the Royal University of Bhutan for granting funds to conduct this research.

REFERENCES

[1] U. Sherpa, D. Pemo, D. Chhoeden, A. Rugchatjaroen, A. Thangthai, and C. Wutiwiwatchai, ‘Pioneering Dzongkha Text-to-Speech Synthesis’, p. 5, 2008.

[2] M. A. Abuzaraida, A. M. Zeki, and A. M. Zeki, ‘Online Recognition System for Handwritten Hindi Digits Based on Matching Alignment Algorithm’, in *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*, Dec. 2014, pp. 168–171. doi: 10.1109/ACSAT.2014.36.

[3] E. Rani. G, Sakthimohan. M, A. Reddy. G, Selvalakshmi. D, T. Keerthi, and R. Sekar. R, ‘MNIST Handwritten Digit Recognition using Machine Learning’, in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Apr. 2022, pp. 768–772. doi: 10.1109/ICACITE53722.2022.9823806.

[4] D. Beohar and A. Rasool, ‘Handwritten Digit Recognition of MNIST dataset using Deep Learning state-of-the-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN)’, in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Mar. 2021, pp. 542–548. doi: 10.1109/ESCI50559.2021.9396870.

[5] L. Mookdarsanit and P. Mookdarsanit, ‘ThaiWritableGAN: Handwriting Generation under Given Information’, *Int. J. Comput. Digit. Syst.*, May 2021, doi: 10.12785/ijcds/100165.

[6] S. Gonwirat and O. Surinta, ‘Improving Recognition of Thai Handwritten Characters with Deep Convolutional Neural Networks’, in *Proceedings of the 3rd International Conference on Information Science and Systems*, in ICISS ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 82–87. doi: 10.1145/3388176.3388181.

[7] B. Nyaupane, R. Sah, and K. Dahal, ‘SVM, KNN, Random Forest, and Neural Network based Handwritten Nepali Barnamala Recognition’, *J. Innov. Eng. Educ.*, vol. 4, pp. 64–70, Dec. 2021, doi: 10.3126/jiee.v4i2.38254.

[8] Md. M. Hasan, A. Y. Srizon, A. Sayeed, and Md. A. M. Hasan, ‘Handwritten Numerals Recognition by Employing a Transfer Learned Deep Convolution Neural Network for Diverse Literature’, in *2020 11th International Conference on Electrical and Computer Engineering (ICECE)*, Dec. 2020, pp. 431–434. doi: 10.1109/ICECE51571.2020.9393079.

[9] R. Ameri, A. Alameer, S. Ferdowsi, V. Abolghasemi, and K. Nazarpour, ‘Classification of Handwritten Chinese

https://doi.org/10.25077/jitce.8.01.1-7.2024

Numbers with Convolutional Neural Networks', in *2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Apr. 2021, pp. 1–4. doi: 10.1109/IPRIA53572.2021.9483557.
[10] Z. Li, Q. Wu, Y. Xiao, M. Jin, and H. Lu, 'Deep Matching Network for Handwritten Chinese Character Recognition', *Pattern Recognit.*, vol. 107, p. 107471, Nov. 2020, doi: 10.1016/j.patcog.2020.107471.
[11] D. Chamling, Y. Jamtsho, and Y. Jamtsho, 'Handwritten Dzongkha Alphabet Recognition System using Convolutional Neural Network', *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 9, no. 5, pp. 20–24, 2021.
[12] K. Khunratchasana and T. Treenuntharath, 'Thai digit handwriting image classification with convolutional neural networks', *Indones. J. Electr. Eng. Comput. Sci.*, vol. 27, no. 1, p. 110, Jul. 2022, doi: 10.11591/ijeecs.v27.i1.pp110-117.
[13] J. Mukhoti, S. Dutta, and R. Sarkar, 'Handwritten Digit Classification in Bangla and Hindi Using Deep Learning', *Appl. Artif. Intell.*, vol. 34, no. 14, pp. 1074–1099, Dec. 2020, doi: 10.1080/08839514.2020.1804228.
[14] A. Sufian, A. Ghosh, A. Naskar, F. Sultana, J. Sil, and M. M. H. Rahman, 'BDNet: Bengali Handwritten Numeral Digit Recognition based on Densely connected Convolutional Neural Networks', *J. King Saud Univ. - Comput. Inf. Sci.*, Mar. 2020, doi: 10.1016/j.jksuci.2020.03.002.
[15] R. S. Alkhawaldeh, M. Alawida, N. F. F. Alshdaifat, W. Z. Alma'aitah, and A. Almasri, 'Ensemble deep transfer learning model for Arabic (Indian) handwritten digit recognition', *Neural Comput. Appl.*, vol. 34, no. 1, pp. 705–719, Jan. 2022, doi: 10.1007/s00521-021-06423-7.
[16] R. Hasan Finjan, A. Salim Rasheed, A. Abdulsahib Hashim, and M. Murtdha, 'Arabic handwritten digits recognition based on convolutional neural networks with resnet-34 model', *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, p. 174, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp174-178.
[17] J. Li, G. Sun, L. Yi, Q. Cao, F. Liang, and Y. Sun, 'Handwritten Digit Recognition System Based on Convolutional Neural Network', in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, Aug. 2020, pp. 739–742. doi: 10.1109/AEECA49918.2020.9213619.
[18] F. Siddique, S. Sakib, and Md. A. B. Siddique, 'Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers', in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, Sep. 2019, pp. 541–546. doi: 10.1109/ICAEE48663.2019.8975496.
[19] D. Chamling, Y. Jamtsho, and Y. Jamtsho, 'A Paper on Preparation of Dataset for Handwritten Dzongkha Alphabets', vol. 9, pp. 105–109, Dec. 2021.
[20] N. Otsu, 'A Threshold Selection Method from Gray-Level Histograms', presented at the IEEE Transactions on Systems, Man, and Cybernetics, IEEE, 1979, pp. 62–66.
[21] G. Bradski, 'The OpenCV Library', *Dr. Dobb's Journal of Software Tools*, 2000.
[22] Chollet, Francois, and others, 'Keras'. GitHub, 2015. [Online]. Available: https://github.com/fchollet/keras

AUTHORS BIOGRAPHY

Yonten Jamtsho

Mr. Yonten Jamtsho is currently working as a Lecturer at Gyalpozhing College of Information Technology, Royal University of Bhutan. He did his BSc (Hons) in Computer Science (2015) from Sherubtse College, Royal University of Bhutan and M.E. in Computer Engineering (2020) from Naresuan University, Phitsanulok, Thailand. He has also done a Postgraduate Certificate in Higher Education (2021) from Samtse College of Education. He has eight years of university teaching experience and his research focuses on computer visions, image processing and blockchain technology.

Pema Yangden

Ms. Pema Yangden is currently working as a faculty member at Gyalpozhing College of Information Technology (GCIT), Royal University of Bhutan. She has been teaching for the last 7 years. She has completed her MSc in Information Technology from the International Institute of Information Technology (IIIT), Hyderabad, and her Bachelor of Engineering in Information Technology from the College of Science and Technology (CST), Royal University of Bhutan. Additionally, she has achieved the status of Fellow (FHEA) from AdvanceHE, in recognition of her attainment against the UK Professional Standards Framework for teaching and learning support in higher education. Her research interests primarily lie in the field of machine learning and data science.

Sonam Wangmo

Ms. Sonam Wangmo is an educated and experienced individual, currently serving as a Lecturer at Gyalpozhing College of Information. She holds a Bachelor of Engineering degree from the College of Science and Technology, Royal University of Bhutan, a Master's degree from the International Institute of Information Technology (IIIT) in Hyderabad, India and a Postgraduate Certificate in Higher Education from Samtse College of Education. With 6 years of teaching experience, Sonam is a dedicated educator who is passionate about imparting knowledge to her students. Her research interests lie in the fields of Artificial Intelligence and full-stack development, demonstrating her expertise in cutting-edge technology and her commitment to staying at the forefront of her field.

Nima Dema

Ms. Nima Dema is currently working as a Lecturer at Gyalpozhing College of Information Technology, Royal University of Bhutan. She has completed her Bachelor of Engineering in Information Technology (2017) from the College of Science and Technology, Royal University of Bhutan and completed her Master in Information Technology (2019) from the International Institute of Information Technology (IIIT), Hyderabad, India. She has six years of university teaching experience and does research in the field of Artificial Intelligence.