

Person-Dependent Handwriting Verification for Special Education Using Deep Learning

Umut Zeki^{1,*}, Tolgay Karanfiller² and Kamil Yurtkan¹

¹Department of Computer Engineering, Cyprus International University, Haspolat, Nicosia, Northern Cyprus, Mersin 10, Turkey

²Computer and Instructional Technology Teacher Education, Cyprus International University, Haspolat, Nicosia, Northern Cyprus, Mersin 10, Turkey

*Corresponding Author: Umut Zeki. Email: umut_zeki@hotmail.com

Received: 22 May 2022; Accepted: 06 July 2022

Abstract: Individuals with special needs learn more slowly than their peers and they need repetitions to be permanent. However, in crowded classrooms, it is difficult for a teacher to deal with each student individually. This problem can be overcome by using supportive education applications. However, the majority of such applications are not designed for special education and therefore they are not efficient as expected. Special education students differ from their peers in terms of their development, characteristics, and educational qualifications. The handwriting skills of individuals with special needs are lower than their peers. This makes the task of Handwriting Recognition (HWR) more difficult. To overcome this problem, we propose a new personalized handwriting verification system that validates digits from the handwriting of special education students. The system uses a Convolutional Neural Network (CNN) created and trained from scratch. The data set used is obtained by collecting the handwriting of the students with the help of a tablet. A special education center is visited and the handwritten figures of the students are collected under the supervision of special education teachers. The system is designed as a person-dependent system as every student has their writing style. Overall, the system achieves promising results, reaching a recognition accuracy of about 94%. Overall, the system can verify special education students' handwriting digits with high accuracy and is ready to integrate with a mobile application that is designed to teach digits to special education students.

Keywords: Special education; deep learning; convolutional neural network; handwriting verification; handwriting digit verification; person-dependent training; handwriting recognition

1 Introduction

Handwriting Digit Recognition (HDR), verification, and identification are frequently confused. It is a common subject that implies both digit identification and verification. On one side, the goal of handwriting digit identification is to identify a digit based on the written digit. On the other side, handwriting digit verification is a kind of one-to-one matching that is related to validating a requested



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

identity according to the written digit, and either accepting or rejecting the identity claim. Although recognizing a handwritten digit is easy for humans, it is still a challenge for the machine because everyone's handwriting is unique and different [1]. For this reason, HDR is needed and used in many applications today [2]. As a sub-branch of HDR, handwritten digit verification is the answer to this issue; that uses the image of a digit and verifies if there is the desired number. Such technologies are also encountered in educational applications. Today, educational applications developed for individuals with special needs are on a rising trend. For example, there are many applications for people with intellectual disabilities, including Handwriting Recognition (HWR) [3]. However, it is more difficult to successfully apply the above-mentioned technologies in developed educational applications. Because individuals with special needs have difficulties in using them by touching; and speaking [4]. This makes especially HDR difficult. For example, in an application that uses an HDR system for normal peers, the system tries to recognize the digits by looking at data sets. Usually, fine handwriting is accurately predicted with a high percentage. However, to achieve successful results while studying with the individuals who need special education, instead of working with existing data, it will be healthier to work with the collected data from them. Because their handwriting can be jittery, upside down, or at a different angle. Besides, many individuals with learning difficulties cannot learn to write by following the general rules. For these reasons, HWR systems are not successful enough. Based on this problem, we developed a person-dependent handwriting digit verification system that takes the user's handwriting digit samples before starting the application as a dataset and uses these data for verification. Therefore, this paper aims to fill this gap in the literature. In this way, when the user applies the HDR feature, the comparisons will deal with the subject's dataset, and the handwriting problems such as tremors, skips, and skews can be easily overcome. We only tested it on mentally disabled individuals, but this method can also be applied to individuals with writing problems such as those that have a physical disability, hyperactivity, dyslexia, or dysgraphia. Convolutional Neural Networks (CNN) is used while developing this system; because it is a very effective model in image recognition and verification. CNN is a deep learning algorithm that can discriminate one from the other by obtaining an input image, regard to several aspects/objects in the image. ConvNet which is a network architecture for deep learning at MATLAB is employed. ConvNets identifies the objects and the faces successfully. Besides, the pre-processing needed in a ConvNet is much less than against other classification algorithms. This is important because we aim to use it in an application designed for special education to teach digits [5]. The proposed system is a person-dependent handwriting verification system. Consequently, these kinds of systems can be used in special education applications and verify the student's handwriting digit. Individuals with special education needs will be able to use such applications more effectively. The software used for the experiments is MATLAB 2018b with Deep Learning Toolbox which is specially designed to design, train, and deploy CNNs.

2 Related Works

HDR is an area of machine learning, in which a machine is trained to identify handwritten digits [6]. HDR is a very important challenge in machine learning and Computer Vision (CV) applications. In the literature, many machine learning methods have been used to solve the HDR problem in other research areas [7,8]. A backpropagation network for HDR is proposed by Naufal et al. in 2019 [9]. Later, several scientists from Bell Laboratories proposed a comparison related to the classifier methods in HDR in 1994 [10]. One year later, from Bell Laboratories again, different research groups studied the same paper to do further research [11]. At the beginning of the millennium, Byun et al. [12] approached the study using the Support Vector Machine (SVM) algorithm on the Modified National Institute of Standards and Technology (MNIST) dataset which is a large database of handwritten digits. In 2006, the Deep Belief Networks (DBN) were investigated with three layers and a greedy algorithm [13]. A LeNet5 CNN architecture-based feature extractor for the MNIST database is proposed by Lauer et al. in 2007 [14]. In

the following years, CNN has been effectively applied for HDR. In 2011, the high recognition accuracy of 99.73% on the MNIST dataset is obtained while testing with the popular committee method of combining various CNNs in a community network [15]. Later, the study was extended into the 35-net committee from the earlier 7-net committee and reported a very high accuracy of 99.77% [16]. In 2015, Gattal et al. proposed a segmentation-verification system based on SVM, for segmenting two joined handwritten digits using the oriented sliding window and SVM classifier to find the best cut for differentiating two attached digits [17]. The bend directional feature maps were investigated using CNN for air handwritten Chinese character recognition in 2018 [18]. In the study “Hybrid CNN-SVM Classifier for Handwritten Digit Recognition” published in 2019, it is developed a hybrid model of a powerful CNN and SVM for recognition of handwritten digits from the MNIST dataset. The experimental results demonstrate the effectiveness of the proposed framework by achieving a recognition accuracy of 99.28% over the MNIST handwritten digits dataset [19]. In another study by Dey, which is named “A Robust Handwritten Digit Recognition System Based on Sliding window with Edit distance” in 2020, a novel method based on a string edit distance algorithm is proposed to recognize offline handwritten digit images. This method has been tested using 5 different datasets and achieved a success rate of 88.69%–97.91% with the Chars74k (Fnt) dataset [20]. These studies reported very high recognition accuracy. The excellent performance of the study clearly shows the effectiveness of CNN’s feature extraction step. In 2021, a real-time HDR is proposed. This research paper is about the extended application of HDR. CNN is used as the model for the classification of the image and more specifically Keras sequential model is used as a classifier. The interface is created by Pygame [21]. In the same year, an ancient Urdu text recognition system is proposed a multi-domain deep CNN which integrates a spatial domain and a frequency domain CNN to learn the modular relations between features originating from the two different domain networks to train and improvise the classification accuracy. The experimental results show that the proposed network with the augmented dataset achieves an average accuracy of 97.8% which outperforms the other CNN models in this class [22].

In the literature, handwriting digit verification studies using CNN have been rarely encountered. However, there are many studies in different fields that verify the correctness of an image, face, fingerprint, or traffic sign using CNN. For example, in the field of biometrics in 2019, a study proposed by Mohapatra, was performed Offline Handwritten Signature Verification using CNN [23]. On the other hand, in the study about speaker verification published by Zhang in 2016, it is presented a novel end-to-end method for text-dependent speaker verification [24]. Besides, there are various studies in the field of face verification. In the same year, Jun-Cheng Chen presented an algorithm for unbounded face verification using convolutional features and evaluates it on the recently delivered Intelligence Advanced Research Projects Activity (IARPA) Janus Benchmark A (IJB-A) dataset as well as on the traditional Labeled Face in the Wild (LFW) dataset [25]. Verification with CNN is also used to predict whether a patient is Covid 19 or not by verifying X-ray images [26].

Most children who need special education, do not have the perceptual-motor and intellectual potential for handwriting. Also, many special education teachers have not been trained to teach handwriting or they cannot create enough time for each individual. This causes students’ handwriting to differ from normal individuals, especially in terms of conformity with the slope, format, or writing rules. Because of these reasons, most of them cannot learn good handwriting. Also, while developing an HDR, application for such individuals, it may be more appropriate to work with a dataset prepared for special education. There is a Mobile Device Application Design with Handwriting Recognition to make learning easy for students who have learning disabilities by Yilmaz in 2014 [27]. This study is a master thesis that implies HWR in a mobile application for special education students. However, the only similarity to our algorithm is, it is an HWR application for special education.

3 Theoretical Background

The implemented application in this study is handwritten digit verification. Many people think of HDR and handwriting digit verification as different problems and many of them think the same. The answer is that the idea behind both is the same, just the application area is different. Although it is explained in more detail below, the difference between these two methods can be briefly explained as follows:

Handwriting digit verification answers: Is this digit x?

Whereas, HDR answers: What is this digit?

3.1 Handwriting Digit Recognition

HDR which is also known as digit identification is the process of conversion of handwritten digits into machine-readable form. The image of the written digit may be recognized online or offline. Online character recognition is the process of recognizing handwriting recorded with a digitizer as a time sequence of pen coordinates. Off-line HWR is the process of recognizing a scanned digit and is stored digitally in the greyscale format as in the proposed system [28]. In HDR as shown in Fig. 1, the system looks for the written digit in a database of digits and tries to predict it which is a one-to-many comparison.



Figure 1: A generic digit recognition system

3.2 Handwriting Digit Verification

It can be considered a sub-branch of HDR. As the name suggests, the system tries to authenticate a handwritten digit. It is a one-to-one comparison. Here, the purpose of the system is not to guess the entered number correctly but to verify whether the entered number is correct. As seen in Fig. 2, by checking the dataset and applying an algorithm, the system verifies whether the written digit is 2 or not.



Figure 2: A generic digit verification system

3.3 Proposed Method

The proposed method is based on the CNN model and it is person-dependent. CNN is a class of deep learning and is often used in analyzing visual information. It is a model created based on the functioning of the human brain. Generally, a CNN consists of an input layer, multiple hidden layers, and one output layer. Inside the structure of hidden layers have convolutional layers, activation function layers, pooling layers, fully connected layers, and normalization layers. If we compare it with other classification algorithms, CNN needs much less preprocessing with better results as the number of training increases [29]. The network architecture of the CNN sample is shown in Fig. 3. The processing begins with feature

extraction layers and is ended by fully connected Artificial Neural Network (ANN) classification layers. Using various layers gives high recognition accuracy, and is not changeable to small geometric transformations of the input images. Because of CNN's high recognition accuracy, it is successfully used on real-world data for classification tasks.

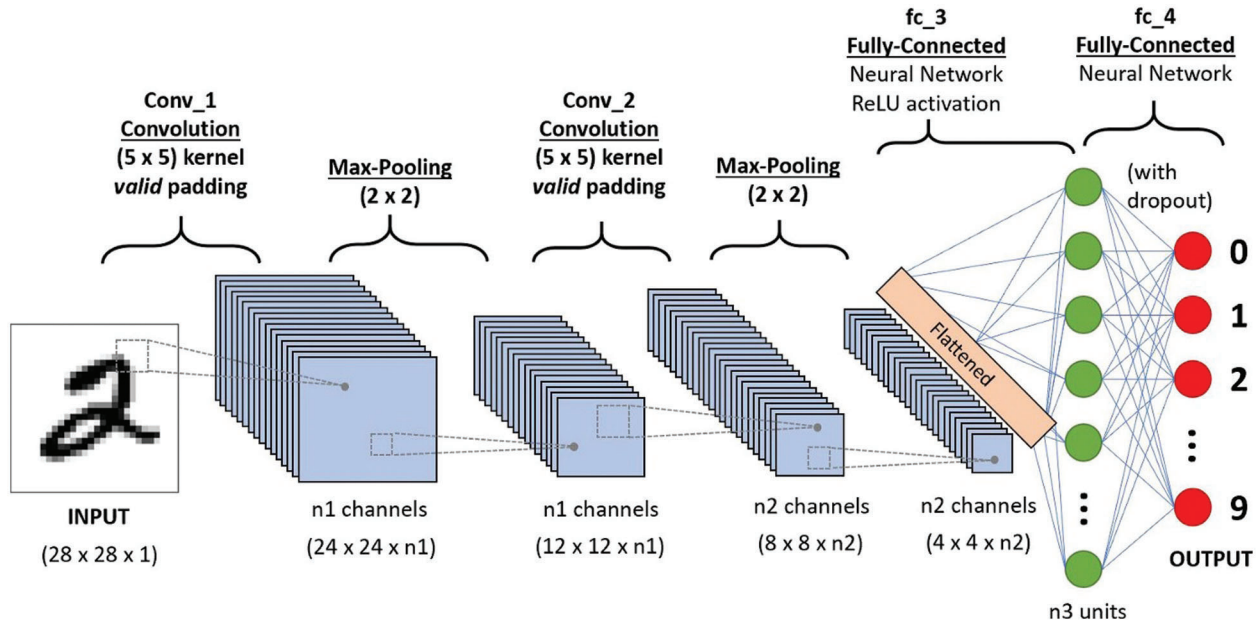


Figure 3: A CNN architecture example for a handwritten digit recognition task

The algorithm has been developed for a special education application that supports the teacher during the teaching of the digits. This algorithm aims to recognize the digit written by a user with high percentage accuracy. In one section of the application, the user should answer by writing the question. Because of that, before starting to learn a digit, each student's handwriting samples are collected as input. For example, if digit 2 is to be learned, the student is directed by the application to write down this digit several times. These data use in the designed CNN together with the student's handwriting while learning the mentioned digit in the application. What the student writes at the beginning of the application becomes a training sample, and what the student writes during the application becomes a test sample.

The person-dependent training is done in this study. That means each subject must have a separate CNN. If there are n subjects, there should be $n \times 10$ CNN architectures. This is the result of having 10 different digits (0–9) to verify. Below, in Fig. 4, there are 10 (0–9) CNN created for a single subject.

3.4 Preprocessing

In this section, the structure of the architecture will be explained in more detail. First of all, some pre-processing operations are done on the images, like resizing images, etc. When necessary pre-processing is done, data can be fed to the system. This part is will be explained in the dataset section. A CNN consists of several layers, and each layer transforms a volume of activations to the other across a differentiable function. We used 7 different layers and a total of 15 layers while building the network. These are, convolutionLayer, poolingLayer, reluLayer, softmaxLayer, classificationLayer imageInputLayer and batchNormalizationLayer. The order of layers is finalized and the architecture of the system is shown in Fig. 5.

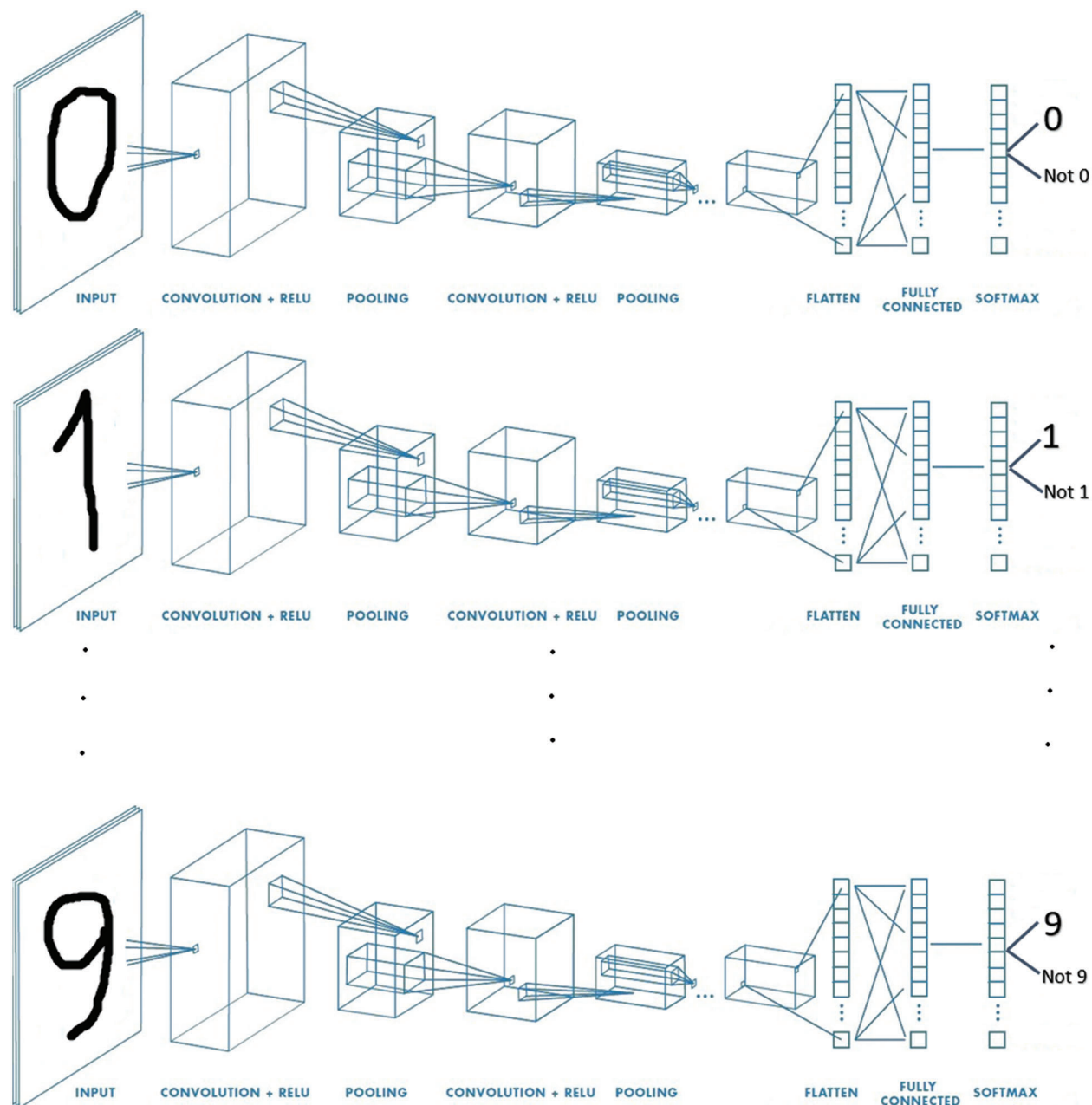


Figure 4: The architecture of our proposed system. 10 different CNNs for each subject

Because it works as a feature extractor, the initial block builds the feature of that kind of Neural Network (NN). To do so, it makes template matching by implementing convolution filtering operations. The initial layer filters the image with some convolution kernels and returns “feature maps”, which are then normalized with an activation function and/or resized. First of all, we need to give input to the system. So, it is used as an input layer as `imageInputLayer` to input 2-D images to a network and to apply data normalization. Its dimension is $100 \times 186 \times 1$ with ‘zero center’ normalization. Here 100×186 is the resolution of the input image and 1 represents a grayscale image. Then we put a Convolution layer to put the input images through a set of convolutional filters, each of which activates certain features from the

images. Here the number of filters is 8 and the filter size is 3×3 . Also, convolutions with stride [1 1] and padding [0 0 0 0]. After convolution, to normalize each input channel across a mini-batch we put a batch normalization layer. Which is used to speed up the training of CNNs and reduce the sensitivity to network initialization by putting it between convolutional and relu layers. Our fourth layer is reluLayer as the activation layer. A ReLU layer performs a threshold operation on each element of the input, where any value less than zero is set to zero as shown in Fig. 6.

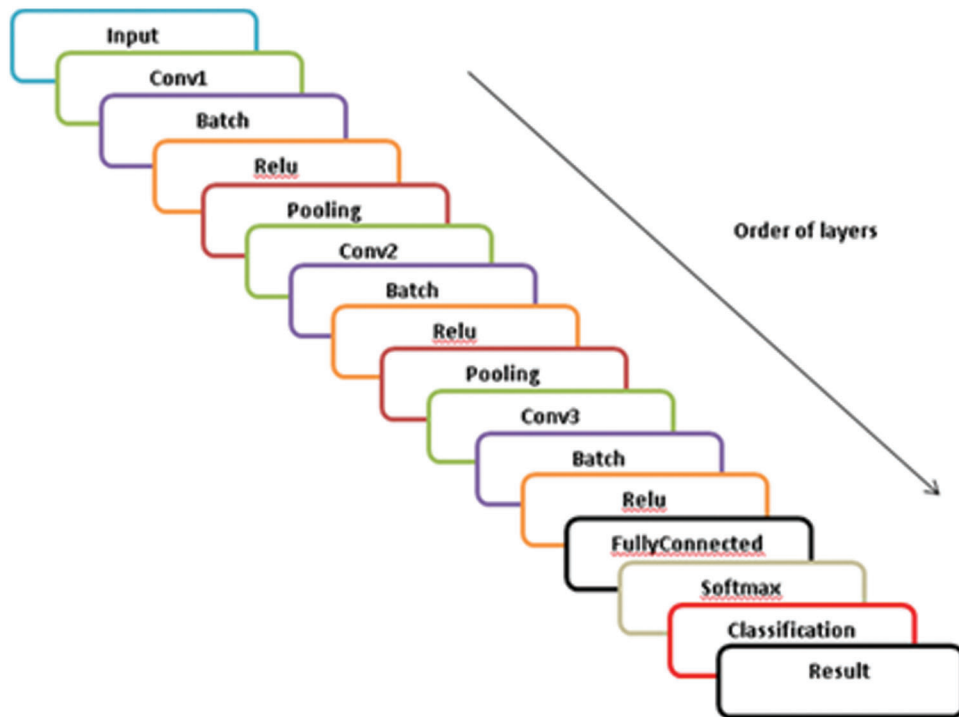


Figure 5: Demonstration of proposed CNN model

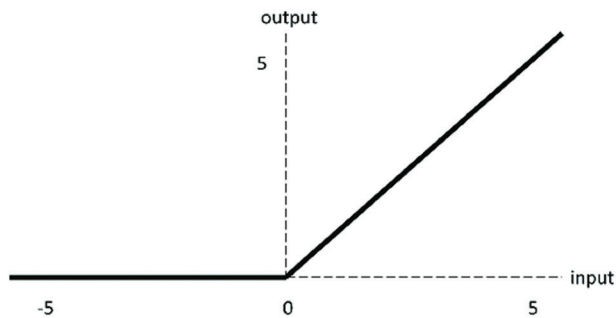


Figure 6: ReLU activation function

After that, we need to make down-sampling by dividing the input into rectangular pooling regions and computing the maximum of each region by adding a max-pooling layer with pool size [2 2] and stride [2 2]. Layers other than the input layer are repeated in the second block. Here, the only difference is the convolution layer. Its parameters are different than the first one. This time is created a 2-D convolutional layer with 32 filters of size [3 3] and ‘same’ padding. At training time, the software calculates and sets the size of

the zero paddings so that the layer output has the same size as the input. In the last block convolution layer like the second block, batchNormalizationLayer and reluLayer are repeated. Then we put a fully connected layer after the reluLayer to multiply the input by a weight matrix and add a bias vector with an output size of 2. The last two layers are output layers. The first one is the softmax layer to apply a softmax function to the input as shown in Eq. (1), where c represents the number of classes. In our model, there are two classes which are true and false as the problem is a verification problem.

$$\frac{e^{z_i}}{\sum_{i=1}^c e^{z_i}} \quad (1)$$

The last one is the classificationLayer to compute the cross-entropy loss for multi-class classification problems with mutually exclusive classes. Fig. 7 shows in detail the architecture of our CNN system for one subject and one digit.

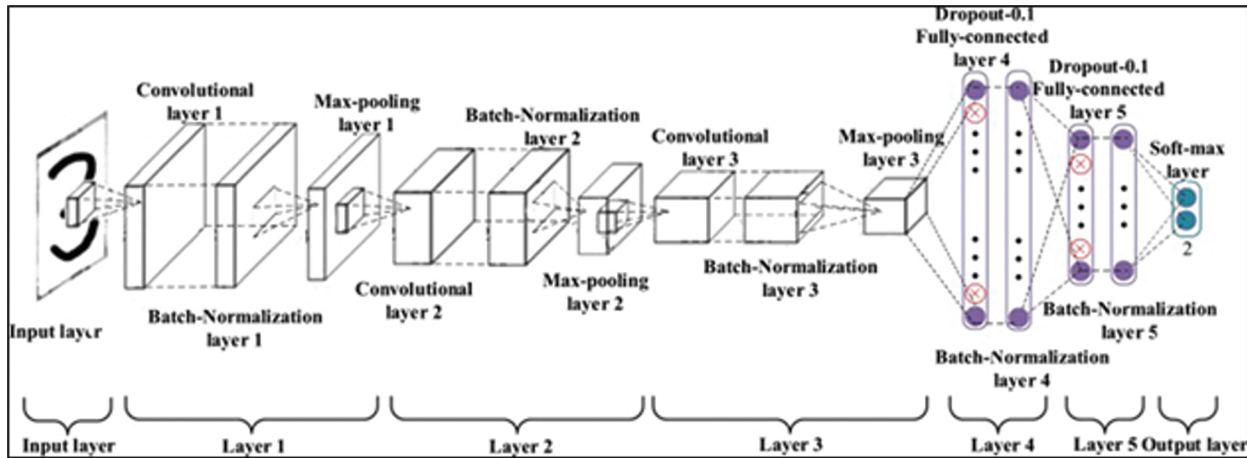


Figure 7: The architecture of our CNN system for one subject and one digit

3.5 Dataset

Since supervised learning is applied in this study, a labeled dataset is required for training [30]. To be able to person dependent verification of digits, it is needed to collect handwriting digit recognition samples of each student who would use the system. Algım Special Education Center was visited by a special education specialist to create the datasets. Since the designed HWR algorithm will be used in an application for mild mental retarded individuals, ten students were selected at that level to collect the handwriting samples. A quiet and plain room was set up to minimize distraction while collecting students' handwriting samples. A graphic tablet as shown in Fig. 8 and a computer that connected via a Universal Serial Bus (USB) cable were used to collect and transfer data to the computer. The student would write the numbers on the graphic tablet with a special pen (stylus), and the written information would be transferred to the computer. A special education specialist who collected the data and a teacher from a special education center were also present in the room. The students were taken into the room one by one and their teacher asked them to write down the numbers (0–9). The same process was repeated ten times for each student. As some of the students were quickly distracted and tired of writing the same numbers over and over again, a break was given when necessary. Since some students were excited when they first entered the room, or some of them had difficulty typing the numbers on the graphics tablet, the students were asked to rewrite the numbers from some sets. As a result, 1000 samples were collected from 10 students with mild mental retardation and a dataset to be used for this study was created.



Figure 8: Graphic tablet and the stylus

After collecting sample forms as shown in [Fig. 9](#), they were cropped and resized with 100×186 pixels resolution. Therefore, while designing, image resolutions were reduced by taking into account the inverse ratio between correct estimation rate and fast running, and the color model was changed from RGB to grayscale. The dataset is partitioned into two sets. The first set consists of 360 samples used for training and the second set has 40 samples used for testing. Collected data were turned into images, processed in MATLAB and the results were recorded.

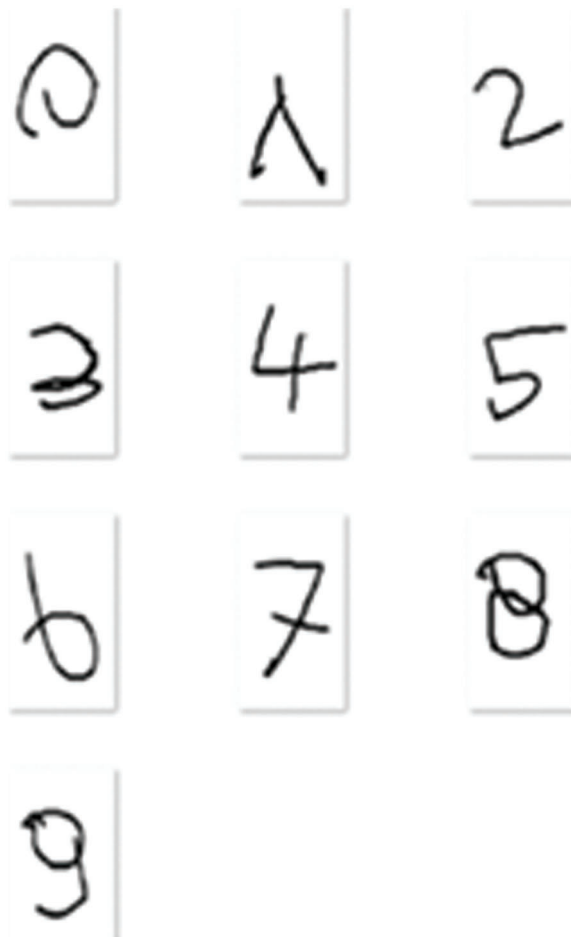


Figure 9: Handwritten digit samples (one set) of a subject

4 Results and Discussion

4.1 Experimental Setup

To implement our CNN, we used ConvNet which is a network architecture for deep learning at MATLAB. ConvNet shows the building blocks of CNN by using MATLAB functions, assuring routines for computing linear convolutions with filter banks, feature pooling, and much more. Hereby, ConvNet enables rapid prototyping of new CNN architectures; nevertheless, ConvNet promotes effective computation on the Central Processing Unit (CPU) and Graphics Processing Unit (GPU) consenting to train complicated systems on big datasets. CNN is the current state-of-art architecture for the image classification task. We trained our person-dependent handwriting digit verification system with the pixel images small enough not to slow the system down, but large enough not to reduce the recognition rate. There are 400 samples which are 360 from training and 40 from testing by a training dataset separately. While doing this, we created two classes, true and false. We put the training samples that the system will train in the false class, and in the true class the samples that the system will test. For example, if we aim to find out whether the written number is 5, in the true part we put the samples of 5's that were taken from the user as shown in Fig. 10, and the false class as shown in Fig. 11, the other numbers that are written by the user but not 5.



Figure 10: True (Test) class of a user for 5

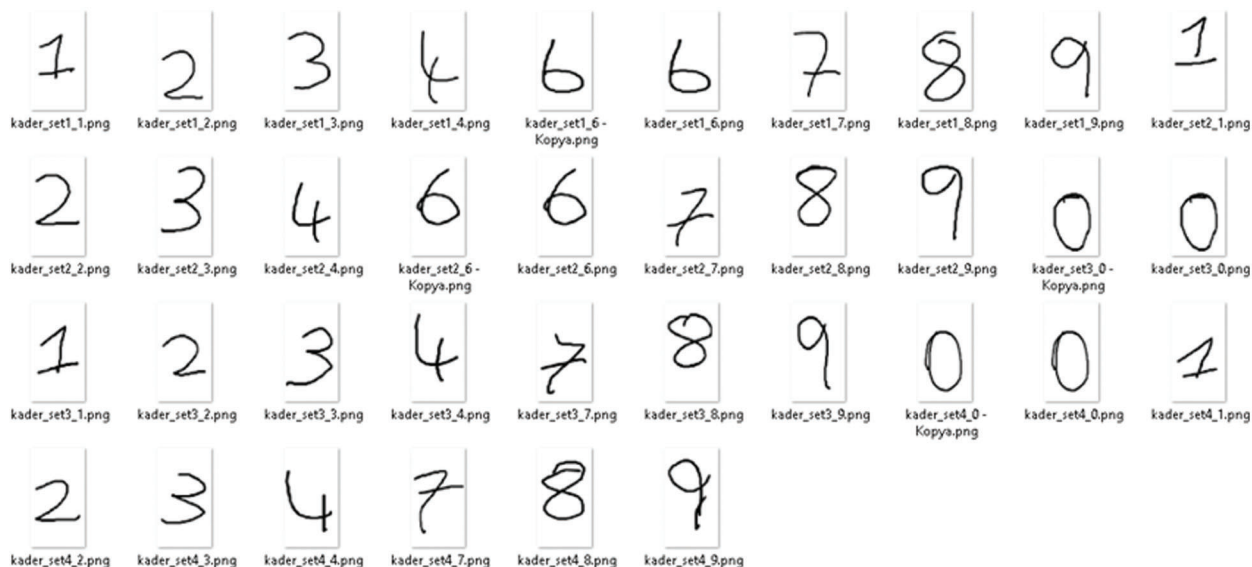


Figure 11: False (Training) class of a user for 5

4.2 Training

Our system was trained using the Stochastic Gradient Descent with Momentum (SGDM), with a learning rate of 0.001. The system was trained for 10 epochs and 30 iterations as is shown in Tab. 1.

Table 1: Training, validation, and learning rate information

Name	Value
Training time	31 s
Number of epochs	10
Validation frequency	30 iterations
Learning rate	0.001

Among 160 test cases for 4 subjects and 10 digits, the best training and validation accuracy of our handwritten digit recognition system is 100%. Each subject's handwriting digit from 0 to 9 is tested 4 times. In [Tab. 2](#), the test accuracy of each subject vs. digit is shown. Here, the values are the average of 4 tests for each digit. In the same table, the average test accuracy of each digit is listed. The average validation accuracy of the whole system is 93.5%.

Table 2: Training accuracy of each subject/digit

Subject/Digit	0	1	2	3	4	5	6	7	8	9
Subject 1	90%	90%	100%	90%	90%	90%	90%	90%	80%	100%
Subject 2	90%	80%	90%	100%	100%	100%	100%	100%	100%	90%
Subject 3	100%	100%	90%	90%	100%	90%	100%	90%	100%	100%
Subject 4	100%	100%	90%	90%	90%	80%	100%	90%	90%	90%
Average	95%	92.5%	92, 5%	92.5%	95%	90%	97.5%	92.5%	92.5%	95%
Overall average	93.5%									

In [Tab. 3](#), the validation accuracy of each subject vs. digit is shown. Here, the values are the average of 4 tests for each digit. In the same table, the average validation accuracy of each digit is listed. The average validation accuracy of the whole system is 94.5%.

Table 3: Validation accuracy of each subject/digit

Subject/Digit	0	1	2	3	4	5	6	7	8	9
Subject 1	100%	90%	100%	90%	90%	90%	100%	90%	80%	100%
Subject 2	90%	90%	90%	100%	100%	90%	100%	100%	90%	100%
Subject 3	100%	100%	90%	100%	100%	90%	100%	90%	100%	100%
Subject 4	100%	100%	90%	90%	90%	80%	100%	100%	90%	90%
Average	97.5%	95%	92, 5%	95%	95%	87.5%	100%	95%	90%	97.5%
Overall average	94.5%									

Although some digits are not good handwriting, our system will be able to classify them correctly. The above results can be considered very successful because in a normal HDR system this success rate is even below 50% for some poorly written numbers. For example, as is shown in [Fig. 12](#) below; these digits are handwriting samples of 6 and 0 of a subject. Guessing which of these numbers is 6 and which is 0 is a very difficult task, not for an HDR system, but even for a human observer. Nevertheless, the developed

system predicted such contradictory or poorly written numbers with a very high percentage. As seen in [Tab. 3](#) above, the numbers 0 and 6 written by subject1 were estimated with 100% accuracy.

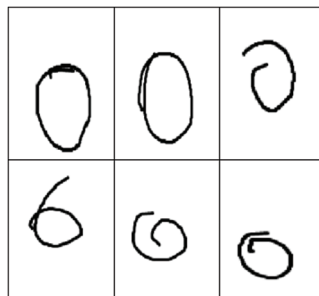


Figure 12: Confused handwriting of a subject: 0's (first row) and 6's (second row)

The handwriting of individuals with special education needs is illegible compared to their peers. In [Fig. 13](#), a set of handwriting samples (0–9) was taken from 3 mild retarded students. As can be seen, each child's handwriting character is different. Besides, many numbers are very difficult to guess when written alone rather than sequentially. For example, 7 of the first subject are similar to 4 and 6 of the second subject are similar to 0. The 0 of the third subject can be perceived as the reverse is written 9 when scanned with one of the normal HWR systems. It was also observed that the subjects were unable to follow most of the rules for writing numbers. For example, the third subject wrote the 8, which should be written without raising a hand, by interlacing the 2 rings. In the 0 samples of the 3rd subject, it is seen that the direction of writing is opposite. Looking at the number 9, which should be written without raising the hand, it is seen that the third subject wrote this by drawing a hook on the edge of the circle. In conclusion, handwriting samples of students at this level are not sufficiently successful in traditional HWR algorithms due to the reasons mentioned above. However, since the system we developed is person-dependent and creates a dataset from one's handwriting, very successful results have been obtained.

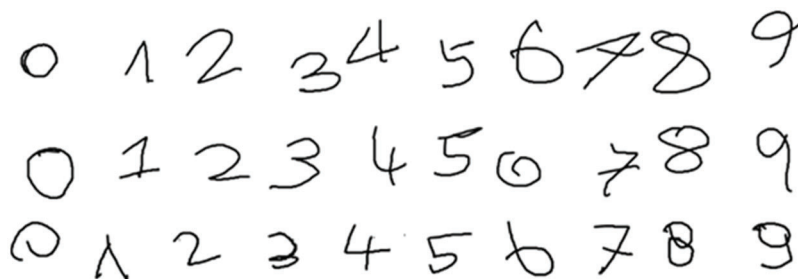


Figure 13: Handwriting character samples of 3 different students

Many studies in the literature do HDR and many of them give very successful results. Two of the examples mentioned in the related work are “Hybrid CNN-SVM Classifier for Handwritten Digit Recognition” published in 2019 and “A Robust Handwritten Digit Recognition System Based on Sliding window with Edit distance” published in 2020. In these two studies, well-written samples were used to train and test the systems developed. However, since there are no systems designed for corrupted handwriting such as handwriting samples of special education students, we realized that there is a gap in this area, and we developed a person-dependent system for special education students to recognize often

corrupted handwriting. As a result, we have achieved high success in recognizing this type of handwriting. Tab. 4 below shows the comparison of our study together with the other two studies.

Table 4: Comparison of the proposed method

Authors	Method	Dataset	Accuracy
Ciresan et al. [15]	A handwriting digit recognition system based on a sliding window with edit distance for handwritten digit images (PCA).	1016 train and 1016 for validation of each digit (0–9).	97.91%
Lauer et al. [14]	A hybrid model of a powerful Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) for recognition of handwritten digits.	6000 train and 1000 for validation of each digit (0–9).	99.28%
Proposed method	The novel feature selection procedure, spectrogram transformation, and CNN with transfer learning.	36 train and 4 for validation of each digit (0–9).	94.5%

Because the computer used for the test is called low-level in terms of GPU and also the system is designed for an educational application for tablets, the resolution of the images has been reduced and some system parameters have been changed to run the system quickly. For this reason, it is thought that the system will be more successful if it is tested on a better computer. Also, the training set size affects the accuracy and it increases as the number of data increases. Here we used a very small dataset for training and testing. The more data in the training set, the smaller the impact of training error and test error, and ultimately the accuracy can be improved.

5 Conclusion and Future Work

In this study, a novel person-dependent handwriting verification system is proposed to verify the digits from the handwriting of the students who need special education. Because the handwriting of individuals with special needs is different from their peers, the proposed system will fill this gap in the literature as an enhanced application of handwriting verification. The learning system is employed to cover all possible translations and transformations of the images of subjects supported by person-dependent training. The system employs a CNN that is created and trained from the scratch. The dataset is accomplished by collecting the handwriting of the students with the help of a tablet under the supervision of special education teachers. The system recognizes the poor handwriting with a very high percentage with the proposed method. Overall, the system reaches an average accuracy of 94.5% and can verify the handwriting of the special education students with acceptable accuracies. It is ready to be integrated with the mobile application that we are designing that is supporting the teaching of special education students. The proposed system can be further improved by applying additional feature selection procedures and transfer learning approaches as well as re-training for one vs. all systems, which will be a recognition system.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] Data Flair, “Deep learning project–Handwritten digit recognition using python,” Received from <https://data-flair.training/blogs/python-deep-learning-project-handwritten-digit-recognition>, 2020.

- [2] A. Shrivastava, I. Jaggi, S. Gupta and D. Gupta, "Handwritten digit recognition using machine learning: A review," in *2nd Int. Conf. on Power Energy, Environment and Intelligent Control (PEEIC)*, Greater Noida, India, pp. 322–326, 2019.
- [3] R. Karakaya and S. Kazan, "Handwritten digit recognition using machine learning," *Sakarya University Journal of Science*, vol. 25, no. 1, pp. 65–71, 2021.
- [4] Ö. İlker and M. A. Melekoğlu, "İlköğretim döneminde özel öğrenme güçlüğü olan öğrencilerin yazma becerilerine ilişkin çalışmaların incelenmesi," *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Özel Eğitim Dergisi*, vol. 18, no. 1, pp. 443–469, 2017.
- [5] S. Saha, "A comprehensive guide to convolutional neural networks—The eli5 way," Received from <https://towardsdatascience.com>, 2018.
- [6] T. A. Assegie and P. S. Nair, "Handwritten digits recognition with decision tree classification: A machine learning approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4446–4451, 2019.
- [7] M. A. R. Khan and M. K. Jain, "Feature point detection for repacked android apps," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1359–1373, 2020.
- [8] N. Binti, M. Ahmad, Z. Mahmoud and R. M. Mehmood, "A pursuit of sustainable privacy protection in big data environment by an optimized clustered-purpose based algorithm," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1217–1231, 2020.
- [9] E. Naufal and J. R. Tom, "Wavelength-switched passively coupled single-mode optical network," in *Proc. ICAIS*, New York, NY, USA, pp. 621–632, 2019.
- [10] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon *et al.*, "Comparison of classifier methods: A case study in handwritten digit recognition," in *Proc. of the 12th IAPR Int. Conf. on Pattern Recognition*, Jerusalem, Israel, vol. 3, pp. 77–82, 2002.
- [11] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes *et al.*, "Comparison of learning algorithms for handwritten digit recognition," in *Int. Conf. on Artificial Neural Networks*, Paris, France, vol. 60, pp. 53–60, 1995.
- [12] H. Byun and S. W. Lee, "A survey on pattern recognition applications of support vector machines," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 17, no. 3, pp. 459–486, 2003.
- [13] G. E. Hinton, S. Osindero and Y. W. The, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [14] F. Lauer, Y. C. Suen and G. Bloch, "A trainable feature extractor for handwritten digit recognition," *Pattern Recognition*, vol. 40, no. 6, pp. 1816–1824, 2007.
- [15] D. C. Cireşan, U. Meier, J. Masci, M. L. Gambardella and J. Schmidhuber, "Flexible, high-performance convolutional neural networks for image classification," in *Proc. of Twenty-Second Int. Joint Conf. on Artificial Intelligence*, Barcelona, Spain, vol. 22, pp. 1237–1242, 2011.
- [16] D. Cireşan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 3642–3649, 2012.
- [17] A. Gattal and Y. Chibani, "SVM-based segmentation-verification of handwritten connected digits using the oriented sliding window," *International Journal of Computational Intelligence and Applications*, vol. 14, no. 1, pp. 54–66, 2015.
- [18] X. Qu, W. Wang, K. Lu and J. Zhou, "Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network," *Pattern Recognition Letters*, vol. 111, pp. 9–15, 2018.
- [19] R. Dey, R. C. Balabantaray and J. Piri, "A robust handwritten digit recognition system based on a sliding window with edit distance," in *2020 IEEE Int. Conf. on Electronics, Computing and Communication Technologies (CONECCT)*, Piscataway, New Jersey, vol. 6, pp. 1–6, 2020.
- [20] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM classifier for handwritten digit recognition," *Procedia Computer Science*, vol. 167, pp. 2554–2560, 2019.
- [21] K. Senthil Kumar, S. Kumar and A. Tiwari, "Realtime handwritten digit recognition using keras sequential model and pygame," in *Proc. of 6th Int. Conf. on Recent Trends in Computing*, Singapore, pp. 251–260, 2021.

- [22] K. O. Mohammed Aarif and P. Sivakumar, "Multi-domain deep convolutional neural network for ancient urdu text recognition system," *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 275–289, 2022.
- [23] R. K. Mohapatra, K. Shaswat and S. Kedia, "Offline handwritten signature verification using CNN inspired by inception v1 architecture," in *2019 Fifth Int. Conf. on Image Information Processing (ICIIP)*, Shimla, India, pp. 263–267, 2019.
- [24] S. X. Zhang, Z. Chen, Y. Li, J. Zhao and Y. Gong, "End-to-end attention-based text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, USA, pp. 171–178, 2016.
- [25] J. C. Chen, V. M. Patel and R. Chellappa, "Unconstrained face verification using deep CNN features," in *2016 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, pp. 1–9, 2016.
- [26] Y. Edrees Almalki, M. Zaffar, M. Irfan, M. Ali Abbas, M. Khalid *et al.*, "A novel-based swin transfer based diagnosis of COVID-19 patients," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 163–180, 2023.
- [27] B. Yılmaz, "Öğrenme güçlüğü çeken çocuklar için el yazısı tanıma ile öğrenmeyi kolaylaştırıcı bir Mobil öğrenme uygulaması tasarımı/Design of a mobile device application with handwriting recognition to make learning easy for students who have learning disabilities," Master dissertation, Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2014.
- [28] B. Priyanka Barhate and G. D. Upadhye, "Handwritten digit recognition through convolutional neural network and particle swarm optimization–A review," *Mukt Shabd Journal*, vol. 8, no. 2, pp. 1–5, 2019.
- [29] G. K. Abraham, V. S. Jayanthi and P. Bhaskaran, "Convolutional neural network for biomedical applications," *Computational Intelligence and its Applications in Healthcare*, pp. 145–156, 2020.
- [30] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.