Received March 6, 2019, accepted March 27, 2019, date of publication April 9, 2019, date of current version April 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909573

The Study of a Classification Technique for Numeric Gaze-Writing Entry in Hands-Free Interface

SANGBONG YOO, DAE KYO JEONG, AND YUN JANG

Department of Computer Engineering, Sejong University, Seoul 05006, South Korea

Corresponding author: Yun Jang (jangy@sejong.edu)

This work was supported in part by the MSIT (Ministry of Science, ICT), South Korea, under the ITRC (Information Technology Research Center) support program under Grant IITP-2019-2016-0-00312, and in part by the grant (Development of Complex Fast Stream Big Data Processing based on In-memory Technology in Distributed Environment) supervised by the IITP (Institute for Information and communications Technology Promotion) under Grant R0190-16-2016.

ABSTRACT Recently, many applications are developed in numerous domains with various environments. Since some environments require hands-free applications, new technology is needed for the input interfaces other than the mouse and keyboard. Therefore, to meet the needs, many researchers have begun to investigate the gaze and voice for the input technology. In particular, there are many approaches to render virtual keyboards with the gaze. However, since the virtual keyboards hide the screen space, this technique can only be applied in limited environments. In this paper, we propose a classification technique for gaze-written numbers as the hands-free interface. Since the gaze-writing is less accurate compared to the virtual keyboard typing, we apply the convolutional neural network (CNN) deep learning algorithm to recognize the gaze-writing and improve the classification accuracy. Besides, we create new gaze-writing datasets for training, gaze MNIST (gMNIST), by modifying the MNIST data with features of the gaze movement patterns. For the evaluation, we compare our approach with the basic CNN structures using the original MNIST dataset. Our study will allow us to have more options for the input interfaces and expand our choices in hands-free environments.

INDEX TERMS Gaze-writing, input technique, MNIST, Eye tracking, machine learning.

I. INTRODUCTION

The keyboard and mouse are the most familiar typing tools connected to our computers. Input tools should be provided to meet the needs of users according to the given environment. Recently, as many applications are developed in various environments such as virtual reality, smart device, and mobile, input tools using various technologies such as haptics, voice recognition, and gaze-tracking technology, have been invented. However, there is an increasing demand for input tools that free our hands as we are not able to use our hands or have to manipulate multiple input tools simultaneously. Therefore, many input tools using voice and gaze have been devised. Typical examples are the virtual keyboard for the gaze and the mobile voice recognition function.

However, speech recognition and gaze keyboard input techniques are not always accessible. Since the speech recognition technology is weak against noise, it cannot be used in environments where there is noise or inter-user voice chat. Some applications offer voice services by speaking specific keywords as a solution, but they often malfunction due to the pronunciation or intonation of a user. Silent speech technology [1]–[3] similar to speech recognition has been proposed to overcome the noise disturbance. While the silence speech technology allows users to utilize hands-free interfaces, the technology has limitations on data collection and certain phoneme classification [4]. In machine learning for the silent speech technology, manual labeling for a huge volume of silent voice data is impractical. Therefore, the data collection process is artificially slowed down by gathering data in the form of reasonable duration tokens for the statistical alignment. In the silent speech technology, tongue musculature information is an important feature to distinguish certain phonemes. However, the sEMG sensors placed on the skin surface are not appropriate to receive the tongue musculature information. The gaze virtual keyboard technology cannot be employed in an environment where continuous interaction is required since the virtual keyboard tends to hide

The associate editor coordinating the review of this manuscript and approving it for publication was Waleed Alsabhan.

the screen and a user needs to look at the keyboard for a long time. Nonetheless, the gaze virtual keyboard technology is more attractive than the speech recognition technology since it can be easily used by people who do not speak properly due to muscle stiffness. If we can overcome the disadvantages caused by the gaze virtual keyboard, it is possible to apply it to more various environments since our hands are freed from the interactions.

The simplest way as a hands-free input device without using the virtual keyboard is to write directly with the eyes. However, writing with the eyes is not attractive because it is difficult to recognize the writing due to the low accuracy. Therefore, there is a need for a technique of classifying gaze-written texts with high accuracy. In general, we use a learning algorithm to make the computer classify information. An example of classifying handwritings is the classification of the Modified National Institute of Standards and Technology (MNIST) database with the Convolutional Neural Network (CNN) deep learning algorithm. However, in addition to the classification accuracy, there remains a challenge for training datasets. As the hand-written characters have various patterns, many datasets are needed for the proper training. In order to collect the gaze-writing data, it is necessary to keep the same environment with the gaze-tracking device and to recruit many participants for various gaze-writing patterns. However, collecting the gaze-writing data is more difficult and costs more than collecting hand-writings. It is, therefore, necessary to device a new way to build the gaze-writing dataset instead of collecting the gaze-writings directly.

In this paper, we propose a gaze-writing input technique for classifying gaze-written numbers as a hands-free interface. In order to recognize the gaze-writing, we create a gaze MNIST (gMNIST) training dataset by converting the MNIST dataset. For the conversion of the MNIST dataset, we utilize the features of the gaze-written characters. For the evaluation, we collect the test dataset of the numbers from 0 to 9 written by the gaze of the participants. Also, we compare two basic CNN models and one customized model to demonstrate that the gaze-write classification accuracy using the gMNIST dataset is as high as the hand-write classification accuracy using MNIST. The contributions of our study are as follows:

- We propose a numeric gaze-writing classification technique as an interface that enables a hands-free environment.
- We create a gMNIST training dataset that is transformed by applying the gaze-writing features to the MNIST dataset.
- We evaluate the accuracy of the gaze-write classification with the proposed gMNIST dataset by comparing the accuracy of existing hand-writing techniques utilizing three different CNN models.

Our research goal is to provide a hands-free numeric gaze-writing technique that can be used in more diverse environments.

II. BACKGROUND AND RELATED WORK

We review input technologies as a hands-free interface, CNN algorithms, and MNIST research trends in this section. Also, we examine the problems of insufficient training data and classification accuracy, which are the main issues of gaze-writing input technique, from the existing technologies.

A. INPUT TECHNOLOGIES AS A HANDS-FREE INTERFACE

The gaze input technology has been initially developed for people with limited movement [5]. In recent years, however, hardware such as Google [6], Tobii [7], and gaze-tracking VR headphones from FOVE [8] has been released to utilize the gaze data, which applies to a broader range of applications including games, virtual reality, and user convenience. Especially, as the performance of gaze-tracking hardware stabilizes, the virtual keyboard becomes a critical communication tool for people who cannot move their body. The gaze input technique is divided into two types, dwell-based and dwell-free, depending on the input method. Dwell-based techniques [9], [10] provide users with conditions for recognizing key input, thereby reducing typing errors and providing high accuracy. Typically, the most commonly used input constraint is time, and a user needs to look at the virtual keyboard for a certain amount of time to enter a key. In contrast, the dwell-free method [11]–[13] is a technique of inserting a key only by the eye motion. While the input accuracy is low, the key can be added quickly. Accuracy and speed are a matter of choice, therefore, finding a balance between these two is the biggest challenge of the gaze typing research. Also, the virtual keyboard requires a lot of effort and time for a user to get used to it and multitasking is almost impossible since the virtual keyboard obscures the screen. Therefore, the gaze gesture as an input technique has been studied in the environment where the interaction is required. The gaze gesture [14], [15] is accurate and simple to use because it provides only a few input patterns to the user. However, input techniques using the gaze gestures require the user to memorize the patterns and use only limited pattern combinations. Therefore, it can be used only for simple interactions.

Speech recognition technology has been widely used in modern devices. In the past, while using speech recognition technology, many errors occurred due to individual differences such as accent and pronunciation. However, lately, it has become a core technology that can be easily found in mobile, smart TV, and IoT equipment due to the studies for performance improvement [16]–[18]. A feature of recently introduced speech technology is that it uses a specific word as a trigger for the input. For example, Smart speaker [18] uses a specific word as a trigger. However, even though the user does not intend to use a trigger word, speech recognition technology may be executed due to noise. In order to resolve this problem, a silent speech technique has recently been proposed, which reads lips [1] or an EMG around lips [2]. However, in the silent speech technology, the data collection process for learning is slow. Also, in the speech recognition technology, the tongue musculature information is needed to identify specific phonemes. However, the sEMG sensors attached to the skin surface are not suitable for obtaining tongue musculature information [4].

In addition to the techniques using gaze and voice, numerous interfaces using face gesture [19], nose [20], head direction [21], foot [22], and EMG [23], [24] have been developed. However, these techniques are difficult to employ in various applications due to the limited environment and usability limitations.

B. CONVOLUTIONAL NEURAL NETWORKS

The Convolutional Neural Networks (CNN, or ConvNets) is one of the machine learning algorithms and is specialized in image classification [25]. The CNN structure is divided into two steps of extracting features and classification based on the training data. In particular, CNN has the advantage of showing good performance with fewer preprocessing, fewer parameters, and simple training compared to other deep learning algorithms. Since Tensorflow [26] and Keras [27] allow us to change the CNN structure easily, many researchers choose them for the image classification. The CNN algorithm is mainly used in the field of computer vision. There are many studies to classify videos or images as a classification of human behavior [28], large-scale video classification [29], and image classification using ImageNet [30]. Also, the CNN is used in research for the hand-writing classification. The CNN algorithm is mainly used for Chinese characters such as Chinese and Japanese [31]–[33]. These studies apply a CNN model with at least six layers to distinguish complicated characters for the high accuracy. However, it is inefficient to utilize a CNN model with a deep hierarchy for classifying complex characters in order to classify relatively simple English alphabets or numbers.

C. MNIST DATABASE

The Modified National Institute of Standards and Technology (MNIST) database contains an extensive volume of hand-written number images [34]. The size of each image is 28x28, and the format is the grey scale. Since the MNIST database contains 60,000 training data and 10,000 test data, we can classify the hand-written number images with various patterns, such as fonts, through machine learning algorithms. The MNIST database is freely accessible [35]; therefore, many researchers utilize this database for their studies. Research using the MNIST database aims to classify the hand-written numbers with high accuracy. In previous studies, the hand-writing has been classified by applying various learning algorithms such as CNN [36], Support Vector Machine (SVM) [37], Restricted Boltzmann Machine (RBM) [38], and LImited Receptive Area (LIRA) [39]. The MNIST data is also employed to classify objects other than numbers by transforming the database. Cohen *et al.* [40] create EMNIST by adding more characters in the MNIST database and classify the alphabets. Xiao *et al.* [41] create the MNIST data format database for the garment classification. After investigating existing studies, we believe that the MNIST database transformation enables us to resolve the problem of the lack of training data in various fields. However, there is no database to classify the gaze-writing. In this work, we transform the MNIST dataset to gMNIST and classify the gaze-writing with CNN algorithm and the gMNIST data for the high recognition accuracy.

III. COLLECTING GAZE-WRITING DATA

We collect the gaze-writing data for the test dataset and for the feature analysis of the gaze-writing patterns, which is applied to create gMNIST data.

A. IMPLEMENTATION

We have collected the gaze-writing data to classify the numbers written by the gaze. We have used Tobii Pro X2-30 [42] for the data collection, which is a screen-based gaze tracker that collects the gaze information at 30 Hz. We have developed a gaze-writing framework for data collection with Python 3.5, PyQt5 [43] for GUI, and Tobii Pro SDK [44] to trace the gaze on Windows 10.

B. DATA COLLECTION

Ten participants have been engaged in the data collection. The environment for the data collection includes a head-free, a constant light, and a 30-inch monitor with 3840x2160 resolution as shown in Figure 1. Data collection proceeds in two phases. The first phase is to learn how to use the gaze-tracking device and practice it. Every participant writes numbers from 0 to 9 to adapt to the data collection environment during the exercise phase. At this stage, a number image is provided on the canvas to assist participants who are not accustomed to the gaze-writing. The second phase is the collection phase. In the collection phase, we remove the number background image from the canvas provided in the previous phase, and the participants perform the gaze-writing. We collect gaze point coordinates, (x, y), and time-stamps of each number written by the gaze. Note that for the gaze-writing, we provide

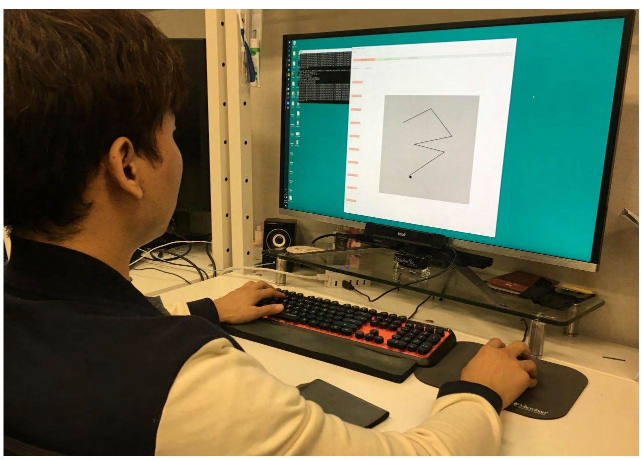


FIGURE 1. An environment for the numeric gaze-writing data collection.

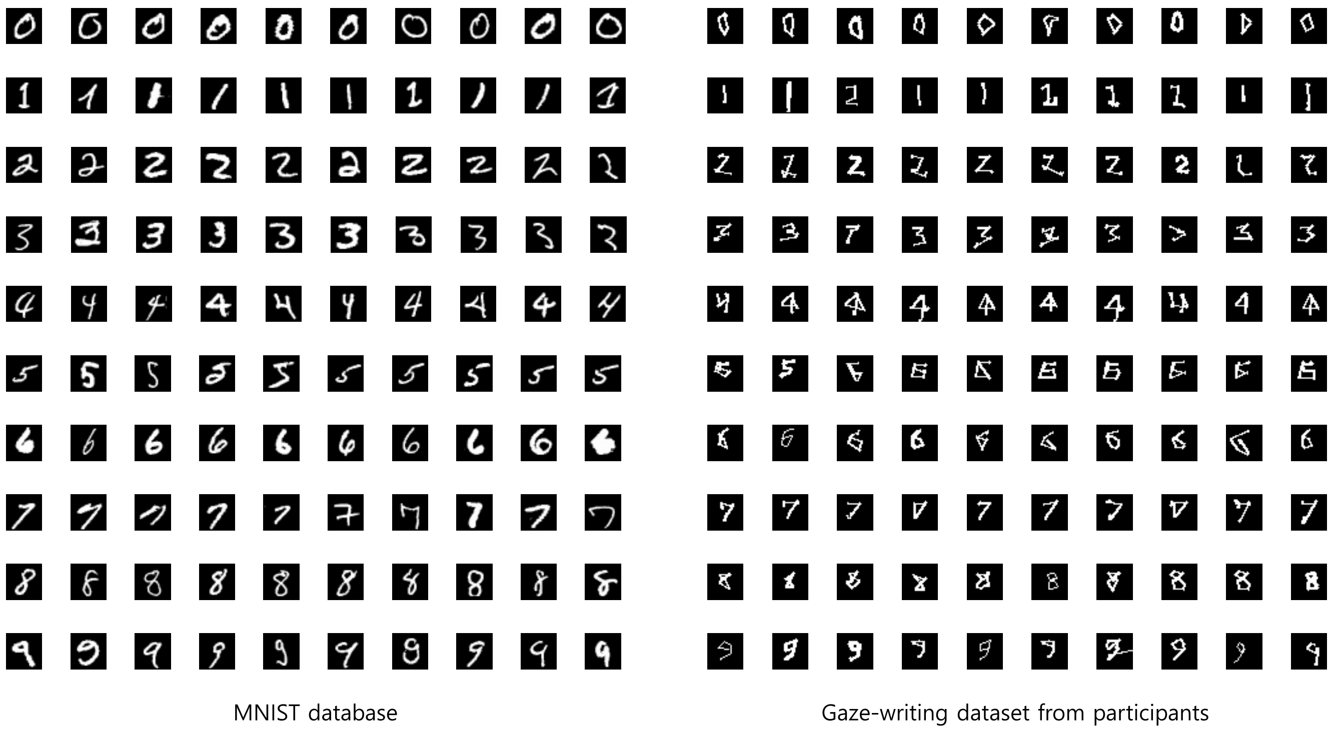


FIGURE 2. The comparison of MNIST database and the gaze-writing dataset. The hand-written MNIST database images are on the left and our collected gaze-writing dataset images are shown on the right.

participants with a stroke key to define the beginning and end of gaze-writing. We also ask participants to write numbers one by one in order to analyze the numeric writing patterns.

C. IMAGE DATA GENERATION

We convert the collected gaze-writing data into images in order to classify them with the CNN model using the MNIST database. We make gray-scale images of 28×28 size for the conversion to the MNIST-NN-format. We remove datasets that are out of the canvas as outliers. The generated image dataset is presented in Figure 2. The left-hand side in Figure 2 is a hand-written numeric data from the MNIST database, and the right-hand side is a gaze numeric image data that we have collected. We utilize 1,329 images for the tests, and there is no significant difference in the amount of data for each number. Finally, the collected gaze-writing images are converted into the MNIST-NN-format [45]. We use the gaze-writing data collected from this work as the test dataset to verify the recognition performance of our gaze-writing input technology and the feature analysis for creating gMNIST dataset.

IV. GAZE MNIST

In order to provide an input device using the gaze-writing technique, it is necessary to have a capability of classifying the input numbers with high accuracy. High classification accuracy can be achieved by modifying a learning algorithm, but there is no training data for the learning of the gaze-writing features. For the gaze-writing data collection, it is necessary to have a space with an eye-tracking device, many participants, and long-term labeling work, but if we can find a way to replace these procedures, it is possible to speed up the use of the gaze input tools. In this section, we, therefore, analyze the gaze-writing data to obtain a database for the gaze-writing classification in a non-direct manner.

The hand-writing and the gaze-writing have many similar characteristics as shown in Figure 2. Therefore, if we apply the gaze-writing characteristics to the hand-writing MNIST database, we can obtain the training data for the desired gaze-writing classifications. Besides, the MNIST database has been used widely as a ground truth dataset [46] for the hand-writing in the machine learning. Therefore, the MNIST dataset saves time and effort by eliminating the need to create an entirely new dataset for overlapping patterns since the gaze-writing has many similar patterns to the hand-writing as shown in Figure 2. The characters obtained by the gaze-writing have rather angular shapes than curved shapes compared to the letters in the MNIST database. Also, there are various shapes in the gaze-written letters. Angled shapes in the gaze-writing seem to appear since the eye tracker cannot detect all eye movements. Besides, the reason why various shapes of a letter are observed is that people use different fonts or they write the same letter in different orders or directions. Therefore, we analyze the features caused by the gaze-writing and the characteristics of the writing styles. We, also, ask the participants about their writing style as shown in Figure 3 for the analysis.

Four patterns are extracted after analyzing the features of the gaze-writing and the characteristics of the writing style.

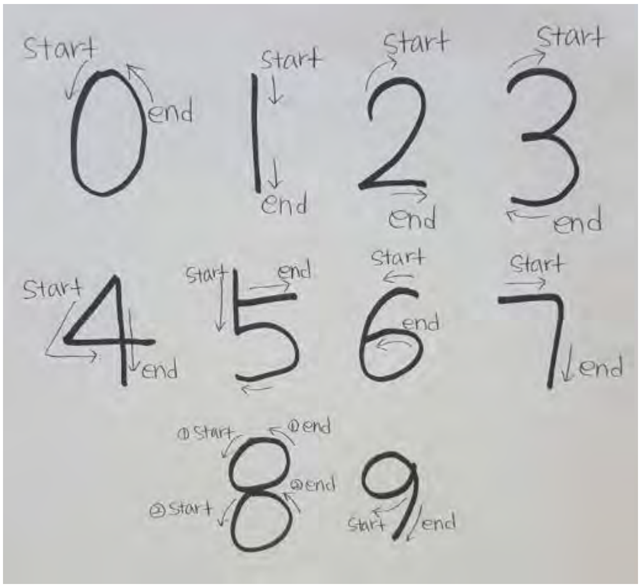


FIGURE 3. Numeric writing styles of a participant.

Figure 4 summarizes the method of classifying the letters into four patterns by comparing the MNIST data with the gaze-writing data. The first pattern is that the letters can be classified with the MNIST data with minimal variation regardless of the writing style. The numbers, 0, 2, 3, 6, 8, 9, belong to this pattern as presented in Figure 4 (a). These numbers are written in angular forms regardless of the direction or order in which the letters are written. Since the MNIST also has partially angled hand-written letters, it is possible to classify these numbers easily. The number 1 belongs to the second pattern. The number 1 can be written in three ways depending on how we write as shown in Figure 4 (b). However, since all forms in the gaze-writing are the same even if it is written by hand, we can classify the number 1 with the MNIST data.

The third pattern is to create a letter that does not exist in MNIST depending on the writing styles as seen in Figure 4 (c). The numbers, 5 and 7, belong to this pattern. The number 5 has two ways of writing depending on where we start writing. The number written as in Figure 4 (c-1) tends to be classified as the number 6 with the MNIST data since the gaze path is marked as the red line in the figure. The red line occurs during the process of moving the eyes while writing numbers. This red line changes the original shape of the numbers even though it is not even intended. On the other hand, in the case of (c-2) and (c-3), the number 5 is classified correctly with the MNIST data. To improve the classification accuracy in the case of (c-1), we create new data for the number 5 by applying the features of the gaze-writing as shown in Figure 5 (a). In the first step, we divide the 28x28 size square into six subspaces. In the second step, we create six random points within the bounds of each subspace so that various shapes of the number 5 can be drawn. In the third step, we connect all points to make a line in the order of 1 to 6. In the last step, a line connecting the points p2 and p6 is drawn to express a line caused by the gaze movement. Then, we replace a part of the data for the number 5 in

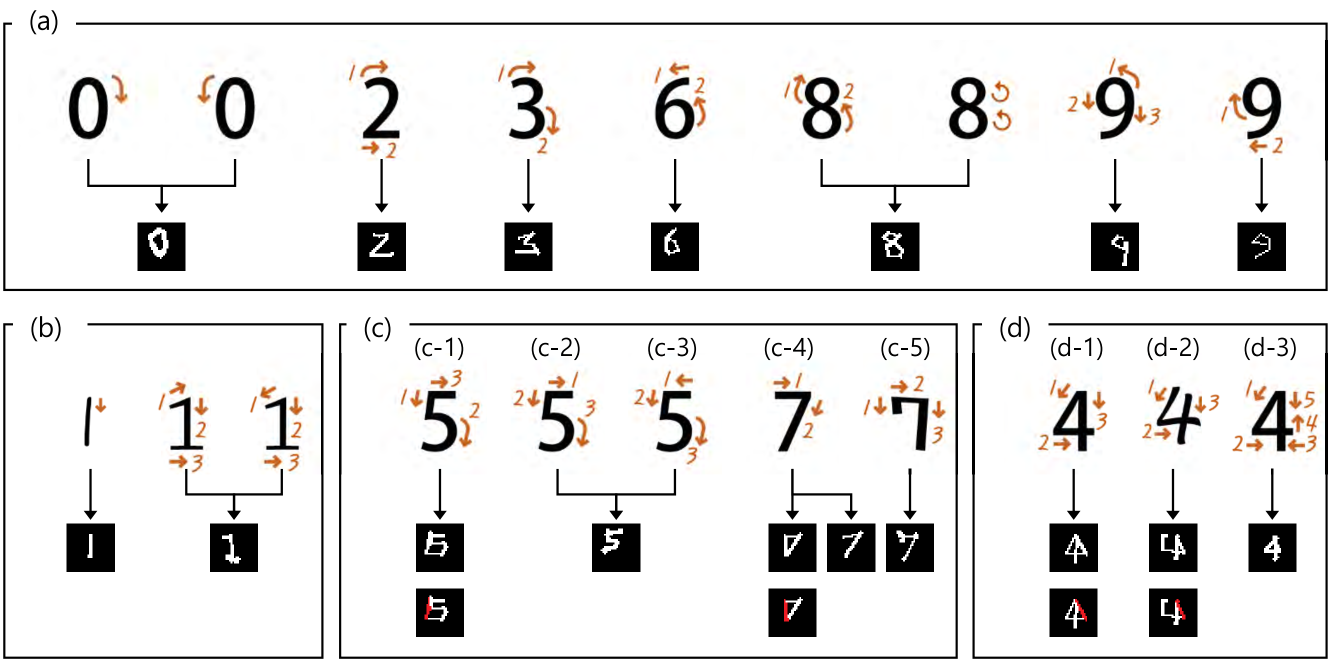


FIGURE 4. Four numeric gaze-writing patterns. Different styles of the numeric gaze-writing data are analyzed. (a) The gaze-writing classification is possible using the MNIST data with minimum variation regardless of the writing style. (b) There are various writing styles, but the number can be classified with the MNIST database. (c) Some writing styles can be classified with the MNIST data but some characters do not exist in the MNIST data. (d) Regardless of the writing style, the gaze-writing is different from the MNIST data. Note that the orange arrows depict the writing styles and the red line indicates the shape change of a number due to the eye movement.

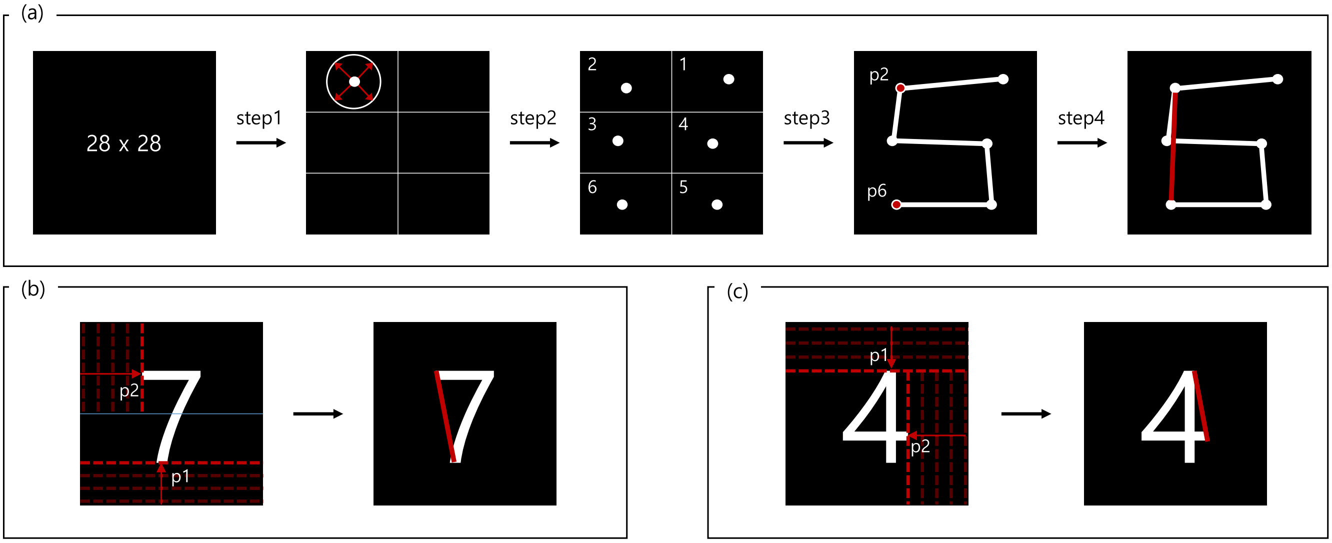


FIGURE 5. A method for applying the gaze-writing feature to the MNIST database. The process of applying for the three numbers 5 in (a), 7 in (b), and 4 in (c) to the MNIST database. The red lines indicate the lines caused by the gaze movements, and the dotted lines represent the processes of finding the two points to create the red lines.

the MNIST with the newly created number 5 images. The number 7 can be written as Figure 4 (c-4) and (c-5), but it cannot be correctly classified with the MNIST data if the number 7 is drawn as a triangle. To recognize the number 7, which resembles a triangle, we transform the MNIST data with the process illustrated in Figure 5 (b). In order to transform the data, we search for two points in the image of the number 7 and these two points tend to be connected by the gaze movement. The first point is found by scanning the first white that appears from the bottom up, whereas, the second point is picked by the search of the first white pixel from left to right. Since the first point may be located on the left of the second point in some cases, only the top half region of the image is scanned for the search of the second point. Then, the number 7 data with the gaze-writing feature is generated once we connect the two points. We replace some of the number 7 images in the MNIST data with this newly created number 7 images.

The fourth pattern is that the shapes of the number are very different from the shapes in the MNIST data regardless of the writing style. The number 4 belongs to this pattern. The number 4 can be written in three cases. Regardless of the methods in Figure 4 (d-1) and (d-2), the gaze path for the number 4 is marked as the red line, and the shape is very close to the shape of an arrow. We also find a participant writing a number as shown in (d-3). The way of (d-3) is the same as (d-1) and (d-2), but the habit of unconsciously returning the eye to the center is observed before moving the gaze upward. The number 4 requires a new set of training data for the classification since there is no similar shaped character in the MNIST data except for the special case in (d-3). Therefore, we apply an algorithm illustrated in Figure 5 (c) to classify the number 4 correctly. Since the number 4 can also be transformed in the same way of transforming the number 7, we find two points from the image of the MNIST database. In the case of the number 4, we can obtain one point by the search from top to bottom and one by the search from right to left. Once we connect the two points, it is possible to change the shape of the number 4 into an arrow shape, which is a transformed shape with the gaze-writing features.

In this paper, we apply the characteristics of the gaze-writing patterns to the MNIST database to transform some letters into gMNIST datasets. The advantage of our work is to reuse the hand-write data from the MNIST database, which are considerably similar to the gaze-writing. This approach allows us to minimize the time and effort of creating a gMNIST dataset. From our study, not all numbers and writing styles in the gaze-write are affected by the gaze movements. Therefore, we combine the transformed MNIST data with the original MNIST data. In this work, we create four different gMNIST training datasets according to the ratio of MNIST data, including 10%, 20%, 30%, and 40%. Note that the total amount of images in the gMNIST data remains the same to avoid performance differences due to the amount of data.

V. CNN FOR GAZE-WRITING CLASSIFICATION

In this section, we introduce a customized CNN model for classifying the gaze-written numbers and compare the customized CNN model with other conventional CNN models. Figure 6 shows the customized CNN structure used in our study. Since the format of the gMNIST data is the same as MNIST-NN-format, the size of the input layer is $28 \times 28 \times 1$. Our CNN model utilizes two convolution layers and two max-pooling layers. The first convolution layer employs a 32-size filter, a 5×5 kernel, and the ReLU activation function. The second convolution layer is identical to the first convolution layer except that the filter size is 128. The max-pooling

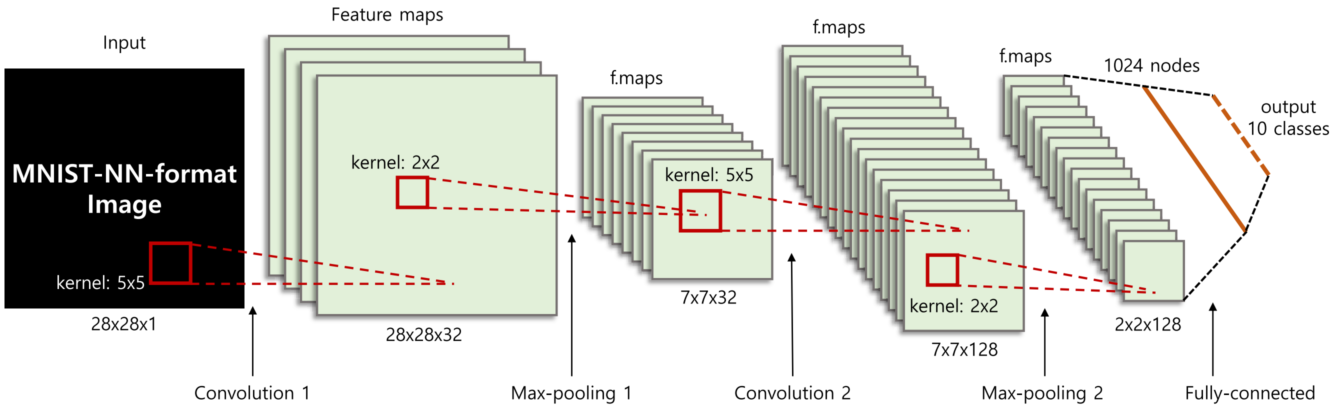


FIGURE 6. Customized CNN model consists of two convolution layers and two max-pooling layers.

layer follows each convolution layer. The stride and kernel size of the two max-pooling layers is 2×2. Finally, through the fully-connected layer of 1024 nodes, the number 0 to 9 are classified into 10 classes. We proceed to drop out at a ratio of 0.3 in each convolution layer and fully-connected layer step to prevent the overfitting.

VI. EVALUATION

Our research goal in this paper is to provide a hands-free numeric gaze-writing technology in more diverse environments. We classify numbers written by the gaze as the first step in recognizing the gaze-written letters. We have collected gaze-written numerical data, created a gMNIST training database transformed by applying the features of the gaze-writing to MNIST. In this section, we evaluate the classification performance of CNN models with the gMNIST dataset by comparing with conventional CNN models and MNIST dataset.

The models and datasets used for the performance evaluation of the classification accuracy are summarized in Table 1. We use two test datasets including the MNIST test data and our gaze-writing data. The training datasets include MNIST training datasets and four different gMNISTs by applying the gaze-writing characteristics to 10-40% of the MNIST data. The reason for using test datasets and training datasets from the MNIST database is to compare the classification accuracy of our approach with the accuracy of the existing hand-writing classification models. Also, the comparison of the four gMNIST datasets is intended to compare how much the gaze-writing features need to be applied to the MNIST database to achieve a good performance. The CNN models compared in our evaluation is the basic keras MNIST CNN model [27], the basic tensorflow MNIST CNN model [26], and the customized CNN model. The keras and tensorflow models are chosen since they are the most basic and popular CNN model using the MNIST database.

For the evaluation, we create 17 cases for the model and dataset combinations summarized in Table 2. The case 1 and case 2 are the existing MNIST classification models. Although they are simple models, they produce a high accuracy of 0.9891 and 0.992, respectively. However, it is seen from the case 3 and case 4 that the classifiers cannot learn the characteristics of the gaze-writing by using the

TABLE 1. Test datasets, training datasets, and CNN models list used for the evaluation.

Test dataset	MNIST test dataset
	Collected gaze-written numeric data
Training dataset	MNIST training dataset
	gMNIST_10
	gMNIST_20
	gMNIST_30
	gMNIST_40
CNN model	Conventional keras MNIST CNN model
	Conventional tensorflow MNIST CNN model
	Customized CNN model

TABLE 2. Comparison of the numeric gaze-writing classification accuracy according to the data and models. We use MNIST and gMNIST data for training dataset, the gaze-written data from participants and the MNIST data for test dataset. Also, we use conventional Keras and Tensorflow, and the customized CNN model for the evaluation.

	Training dataset	Test dataset	CNN model	Accuracy
Case 1	MNIST	MNIST	Keras	0.9891
Case 2	MNIST	MNIST	Tensorflow	0.9920
Case 3	MNIST	gaze-written	Keras	0.5359
Case 4	MNIST	gaze-written	Tensorflow	0.8392
Case 5	gMNIST_10	gaze-written	Keras	0.5461
Case 6	gMNIST_20	gaze-written	Keras	0.5944
Case 7	gMNIST_30	gaze-written	Keras	0.5622
Case 8	gMNIST_40	gaze-written	Keras	0.5631
Case 9	gMNIST_10	gaze-written	Tensorflow	0.9908
Case 10	gMNIST_20	gaze-written	Tensorflow	0.9893
Case 11	gMNIST_30	gaze-written	Tensorflow	0.9896
Case 12	gMNIST_40	gaze-written	Tensorflow	0.9887
Case 13	MNIST	gaze-written	Customized	0.9885
Case 14	gMNIST_10	gaze-written	Customized	0.9916
Case 15	gMNIST_20	gaze-written	Customized	0.9921
Case 16	gMNIST_30	gaze-written	Customized	0.9916
Case 17	gMNIST_40	gaze-written	Customized	0.9886

gaze-writing test data on the MNIST training dataset. Also, in the cases 5-12 using gMNIST as a training data, it is seen that the keras model is not good for learning the features of the gaze-writing compared to the tensorflow model. Comparing only the existing CNN models, the case 9 using the gMNIST_10 as a training dataset with the tensorflow CNN model produces the highest accuracy at 0.9908. In the case of using the gaze-writing data as the test data, the model with the highest classification accuracy is the case 15 using gMNIST_20 as the training dataset and applying our CNN model. The accuracy is 0.9921. The case 15 shows that the classification model for the gaze-writing is as good as the classification model for the hand-writing. In addition, we find that it is appropriate to apply the gaze-writing features to the MNIST database in a ratio of 20% from the case 15.

However, when we compare the accuracy of the case 13 to 17 in Table 2, there is no significant difference between the accuracy of gaze-write classification using MNIST database and the accuracy of gaze-write classification using the proposed gMNIST database. Therefore, it is necessary to verify the reliability of the proposed approach. Consequently, we analyze a part of the error cases caused by the difference of training datasets using MNIST and gMNIST as shown in Figure 7 to verify the reliability of the proposed approach. Figure 7 shows the classification error cases with MNIST and gMNIST datasets. The numbers 2, 5, and 6 are the misclassified labels for the gaze-written yellow numbers. The red and orange boxes indicate classification errors that occur only using MNIST database as training data. Since the numbers written by the gaze include the eye movements, the shapes compared with the hand-written numbers are unintentionally changed. The modified pattern is a feature that is not included in the MNIST database; therefore, the feature learning becomes faulty. Thus, as shown in the orange box, the numbers such as 7 or 4 written like hand-writing are misclassified. In the case of the red boxes, the error occurs since the gaze-written 5 and the hand-written 6 are similar in shape. This is also a classification error that happens because the characteristic of the gaze-written 5 is not present in the MNIST database. On the other hand, errors in the gray boxes are common since numbers are merely similar in shape. Therefore, the classification errors presented in Figure 7 show that there is a performance difference depending on the training data model even though the accuracy is similar. We have verified through this evaluation that our gaze-write classification accuracy using the proposed gMNIST database is similar to the accuracy of hand-write classification using MNIST. Besides, it verifies that the gMNIST database is better suited for the gaze-write classification even though there is no significant difference between the classification accuracy using the MNIST database and the classification accuracy using the gMNIST database.

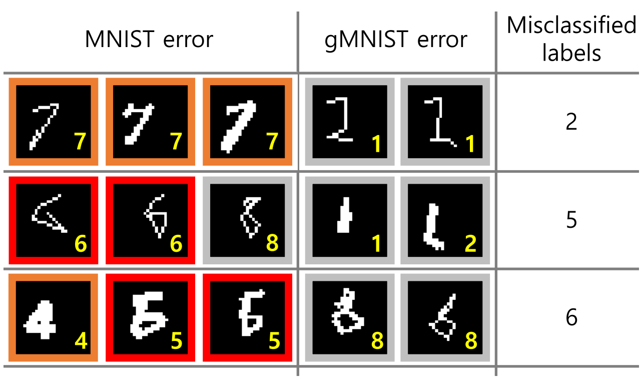


FIGURE 7. Classification error cases for misclassified label 2, 5, 6 using MNIST and gMNIST datasets. 2, 5, and 6 on the right column are the misclassified labels. The white patterns are the gaze-written numbers while writing the yellow numbers. The gaze-written numbers in the red and orange boxes are misclassified since there is no gaze-writing feature within the MNIST database. The gaze-writing in the gray boxes are typical errors that occur when there are many similar features because of the similarity of the number shapes. This figure tells that using the gMNIST dataset causes only a general classification error, and the gaze-writing features must be included in the training dataset.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a numeric gaze-writing input technique as a hands-free environments. We collected gaze-writing number data from the participants. We analyzed the gaze-writing data and discovered four different features in the gaze data. Then, we applied the data features to transform the existing MNIST dataset into a new gMNIST data for the gaze-writing classification. Through the evaluation, we have verified that our model for the gaze-writing classification performs the same as the hand-writing classification with the MNIST data. Besides, we found that the gMNIST data transformed by applying the gaze-writing features to 20% of the MNIST data is the best training data that features the gaze-writing characters.

However, we did not investigate an algorithm for efficient recognition of key-input trigger signals during the gaze-writing. Since the gaze-tracking technology continues to generate data, it is difficult to recognize when a user has entered a key and finished. Moreover, it is not easy to produce signals such as an eye blinking or specific gaze movement patterns that can be used to recognize keystrokes. However, these techniques can cause malfunctions or inconveniences while using applications. Therefore, in this work, we focused only on the training data and accuracy, which are the biggest challenge for pure gaze-writing input technology. However, in addition to the training data and accuracy, finding an efficient way to recognize keystrokes is also an essential task for applying the gaze-writing to various environments. We, therefore, plan to study an efficient keystroke recognition technique for the gaze-writing in the future. In addition, since the numeric gaze-writing may have various patterns such as hand-writing styles by cultural differences, we will investigate the various patterns including exceptional cases appearing in the numeric gaze-writing. Finally, for a hands-free interface, we should be able to classify letters as well as numbers. We will explore ways to adopt a similar pattern analysis of gaze-writing and hand-writing employing hand-written text databases such as

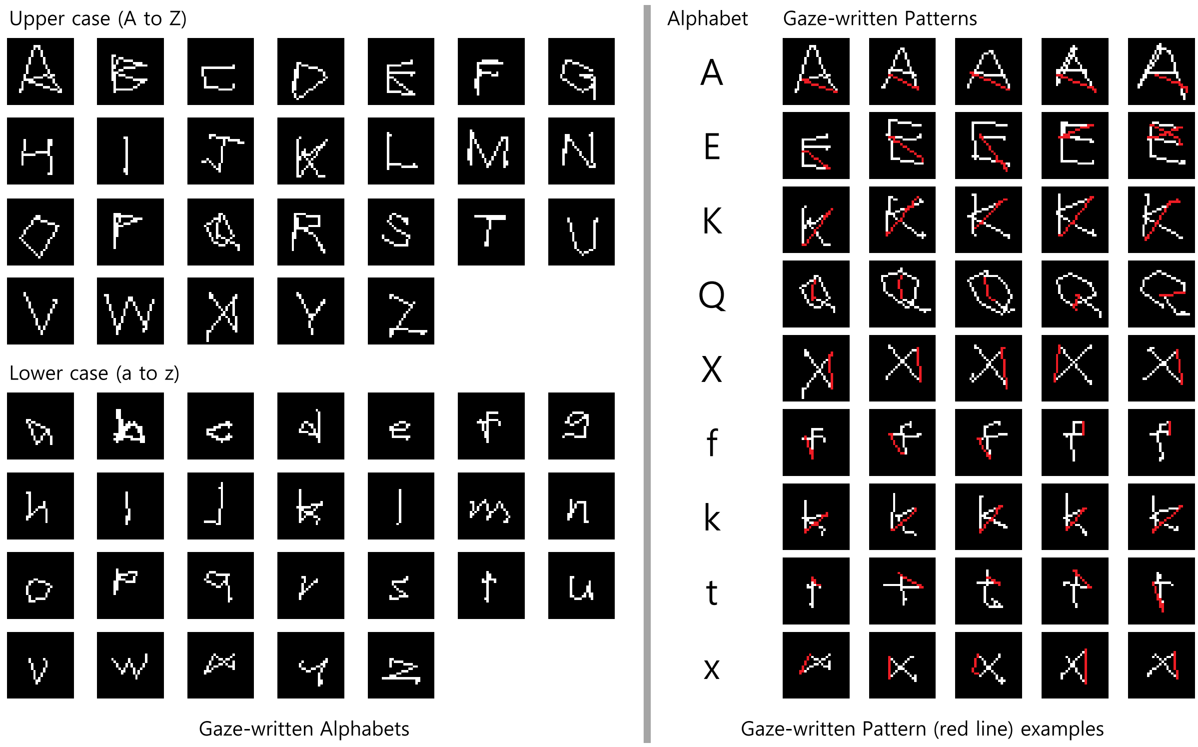


FIGURE 8. Examples of gaze-written alphabets. The red lines indicate gaze movement patterns that appear only in the gaze-writes unlike the hand-writes.

EMNIST [40] in the future. Figure 8 presents the examples of gaze-written upper and lower case alphabets. The red lines indicate the eye movements appearing only in the gaze-writes. To classify gaze-written alphabets properly, there are more issues besides the gaze movement patterns. We need to distinguish upper and lower case alphabets with similar shapes. Also, we need to identify the upper points of the lower case *i* and *j*, which are not visible in the gaze-writes. We will investigate the deformation patterns as shown in Figure 8 and expand them to recognize various shapes of gaze-written alphanumeric characters.

REFERENCES

[1] G. S. Meltzner, J. T. Heaton, Y. Deng, G. De Luca, S. H. Roy, and J. C. Kline, "Development of sEMG sensors and algorithms for silent speech recognition," *J. Neural Eng.*, vol. 15, no. 4, 2018, Art. no. 046031.

[2] A. Kapur, S. Kapur, and P. Maes, "Alterego: A personalized wearable silent speech interface," in *Proc. 23rd Int. Conf. Intell. Interfaces*, 2018, pp. 43–53.

[3] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg, "Silent speech interfaces," *Speech Commun.*, vol. 52, no. 4, pp. 270–287, 2010.

[4] G. S. Meltzner, J. T. Heaton, Y. Deng, G. De Luca, S. H. Roy, and J. C. Kline, "Silent speech recognition as an alternative communication device for persons with laryngectomy," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 12, pp. 2386–2398, Dec. 2017.

[5] P. Majaranta and K.-J. Räihä, "Twenty years of eye typing: Systems and design issues," in *Proc. Symp. Eye Tracking Res. Appl.*, 2002, pp. 15–22.

[6] (2018). *Google Glass—Wikipedia, the Free Encyclopedia*. Accessed: Sep. 19, 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Google_Glass&oldid=859694488

[7] (2018). Tobii Technology. *Tobii is the World Dleader in Eye Tracking*. Accessed: Sep. 19, 2018. [Online]. Available: https://www.tobii.com/

[8] (2014). Fove. *Home—Fove Eye Tracking Virtual Reality Headset*. Accessed: Sep. 19, 2018. [Online]. Available: https://www.getfove.com/

[9] M. E. Mott, S. Williams, J. O. Wobbrock, and M. R. Morris, "Improving dwell-based gaze typing with dynamic, cascading dwell times," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 2558–2570.

[10] I. S. MacKenzie and X. Zhang, "Eye typing using word and letter prediction and a fixation algorithm," in *Proc. Symp. Eye Tracking Res. Appl.*, 2008, pp. 55–58.

[11] P. O. Kristensson and K. Vertanen, "The potential of dwell-free eye-typing for fast assistive gaze communication," in *Proc. Symp. Eye Tracking Res. Appl.*, 2012, pp. 241–244.

[12] A. Kurauchi, W. Feng, A. Joshi, C. Morimoto, and M. Betke, "EyeSwipe: Dwell-free text entry using gaze paths," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 1952–1956.

[13] D. Pedrosa, M. D. G. Pimentel, A. Wright, and K. N. Truong, "Filteryedping: Design challenges and user performance of dwell-free eye typing," *ACM Trans. Accessible Comput.*, vol. 6, no. 1, 2015, Art. no. 3.

[14] H. Drewes and A. Schmidt, "Interacting with the computer using gaze gestures," in *Proc. 11th IFIP TC Int. Conf. Hum.-Comput. Interact. (INTERACT)*, Rio de Janeiro, Brazil. Berlin, Germany: Springer-Verlag, 2007, pp. 475–488. [Online]. Available: http://dl.acm.org/citation.cfm?id=1778331.1778385

[15] H. Istance, A. Hyrskykari, L. Immonen, S. Mansikkamaa, and S. Vickers, "Designing gaze gestures for gaming: An investigation of performance," in *Proc. Symp. Eye-Tracking Res. Appl.*, 2010, pp. 323–330.

[16] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Commun.*, vol. 16, no. 3, pp. 261–291, 1995.

[17] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 6645–6649.

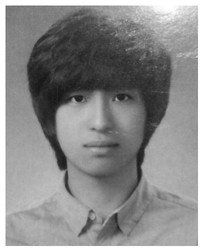
[18] (2018). *Smart Speaker—Wikipedia, the Free Encyclopedia*. Accessed: Feb. 1, 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Smart_speaker&oldid=875086073

[19] Y. Gizatdinova, O. Špakov, and V. Surakka, "Face typing: Vision-based perceptual interface for hands-free text entry with a scrollable virtual keyboard," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2012, pp. 81–87.

[20] S. K. Chathuranga, K. C. Samarawickrama, H. M. L. Chandima, K. G. T. D. Chathuranga, and A. M. H. S. Abeykoon, "Hands free interface for human computer interaction," in *Proc. 5th Int. Conf. Inf. Automat. Sustainability (ICIAFs)*, Dec. 2010, pp. 359–364.

[21] A. Grinshpoon, S. Sadri, G. J. Loeb, C. Elvezio, and S. K. Feiner, "Hands-free interaction for augmented reality in vascular interventions," in *Proc. IEEE Conf. Virtual Reality User Interfaces (VR)*, Mar. 2018, pp. 751–752.
[22] T. Kawai, M. Fukunishi, A. Nishikawa, Y. Nishizawa, and T. Nakamura, "Hands-free interface for surgical procedures based on foot movement patterns," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2014, pp. 345–348.
[23] C. S. L. Tsui, P. Jia, J. Q. Gan, H. Hu, and K. Yuan, "EMG-based hands-free wheelchair control with EOG attention shift detection," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2007, pp. 1266–1271.
[24] Z. Yi, D. Lingling, L. Yuan, and H. Hu, "Design of a surface EMG based human-machine interface for an intelligent wheelchair," in *Proc. IEEE 10th Int. Conf. Electron. Meas. Instrum. (ICEMI)*, vol. 3, Aug. 2011, pp. 132–136.
[25] (2018). *Convolutional Neural Network—Wikipedia, the Free Encyclopedia*. Accessed: Sep. 19, 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network%&oldid=859466798
[26] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.
[27] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras
[28] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[29] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[31] Z. Zhong, L. Jin, and Z. Xie, "High performance offline handwritten Chinese character recognition using Googlenet and directional feature maps," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, Aug. 2015, pp. 846–850.
[32] C. Tsai, "Recognizing handwritten Japanese characters using deep convolutional neural networks," Stanford, CA, USA, Tech. Rep., 2016. Accessed: Dec. 6, 2018. [Online]. Available: https://www.studocu.com/en-us/document/stanford-university/convolutional-neural-networks-for-visual-recognition/other/recognizing-handwritten-japanese-characters-using-deep-convolutional-neural-networks/751923/view
[33] W. Yang, L. Jin, Z. Xie, and Z. Feng. (2015). "Improved deep convolutional neural network for online handwritten chinese character recognition using domain-specific knowledge." [Online]. Available: https://arxiv.org/abs/1505.07675
[34] (2018). *MNIST Database—Wikipedia, the Free Encyclopedia*. Accessed: Sep. 19, 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=MNIST_database&oldid=8577332%86
[35] Y. LeCun. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[36] S. Tabik, D. Peralta, A. Herrera-Poyatos, and F. Herrera, "A snapshot of image pre-processing for convolutional neural networks: Case study of MNIST," *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 555–568, 2017.
[37] L. S. Oliveira and R. Sabourin, "Support vector machines for handwritten numerical string recognition," in *Proc. 9th Int. Workshop Frontiers Handwriting Recognit.*, Oct. 2004, pp. 39–44.
[38] A. W. Savich and M. Moussa, "Resource efficient arithmetic effects on RBM neural network solution quality using MNIST," in *Proc. Int. Conf. Reconfigurable Comput. (FPGAs)*, Nov./Dec. 2011, pp. 35–40.
[39] E. Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database," *Image Vis. Comput.*, vol. 22, no. 12, pp. 971–981, 2004.
[40] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. (2017). "Emnist: An extension of MNIST to handwritten letters." [Online]. Available: https://arxiv.org/abs/1702.05373
[41] H. Xiao, K. Rasul, and R. Vollgraf. (2017). "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms." [Online]. Available: https://arxiv.org/abs/1708.07747
[42] Tobii Technology. (2018). *Tobii Pro X 2–30 Screen-Based Eye Tracker*. Accessed: Sep. 19, 2018. [Online]. Available: https://www.tobiipro.com/product-listing/tobii-pro-x2-30/
[43] Riverbank | Software | PyQt | PyQt5 Download. (2018). *Tobii Pro SDK—Develop Eye Tracking Applications for Research*. Accessed: Sep. 19, 2018. [Online]. Available: https://riverbankcomputing.com/software/pyqt/intro
[44] Tobii Technology. (2018). *Tobii Pro SDK—Develop Eye Tracking Applications for Research*. Accessed: Sep. 19, 2018. [Online]. Available: https://www.tobiipro.com/product-listing/tobii-pro-x2-30/
[45] G. S. Kielian. (2015). *JPG-PNG-to-MNIST-NN-FORMAT*. Accessed: Sep. 19, 2018. [Online]. Available: https://github.com/gskielian/JPG-PNG-to-MNIST-NN-Format
[46] (2018). *Ground Truth—Wikipedia, the Free Encyclopedia*. Accessed: Feb. 1, 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Ground_truth&oldid=856664364

SANGBONG YOO received the bachelor's degree in computer engineering from Sejong University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include gaze analysis, eye tracking technique, mobile security, and data visualization.



DAE KYO JEONG received the B.S. and M.S. degrees in computer engineering from Sejong University, Seoul, South Korea, in 2017 and 2019, respectively. His research interests include virtual reality, data visualization, and brain signal analysis.

YUN JANG received the bachelor's degree in electrical engineering from Seoul National University, South Korea, in 2000, and the master's and Ph.D. degree in electrical and computer engineering from Purdue University, in 2002 and 2007, respectively. He was a Postdoctoral Researcher with CSCS and ETH Zürich, Switzerland, from 2007 to 2011. He is currently an Associate Professor of computer engineering with Sejong University, Seoul, South Korea. His research interests include interactive visualization, volume rendering, and visual analytics.