# GERMIC: Application of Gesture Recognition Model with Interactive Correction to Manual Grading Tasks

Kohei Yamamoto†, Fumiya Kan†, Kazuya Murao†, Masahiro Mochizuki‡, and Nobuhiko Nishio†

† Ritsumeikan University, College of Information Science and Engineering
‡ Research Organization of Science and Technology, Ritsumeikan University
Noji-higashi. 1-1-1, 525-8577 Shiga, Japan
{moi,fumiya}@ubi.cs.ritsumei.ac.jp,murao@cs.ritsumei.ac.jp
moma@ubi.cs.ritsumei.ac.jp,nishio@cs.ritsumei.ac.jp

**Abstract.** Gesture-based recognition is one of the most intuitive methods for inputting information and is not subject to cumbersome operations. Recognition is performed on human's consecutive motion without reference to retrial or alternation by user. We propose a gesture recognition model with a mechanism for correcting recognition errors that operates interactively and is practical. We applied the model to a setting involving a manual grading task in order to verify its effectiveness. Our system, named GERMIC, consists of two major modules, namely, handwritten recognition and interactive correction. Recognition is materialized with image feature extraction and convolutional neural network. A mechanism for interactive correction is called on-demand by a user-based trigger. GERMIC monitors, track, and stores information on the user's grading task and generates output based on the recognition information collected. In contrast to conventional grading done manually, GERMIC significantly shortens the total time for completing the task by 24.7% and demonstrates the effectiveness of the model with interactive correction in two real world user environments.

**Key words:** Handwriting Recognition, Recognition Error Correction

## 1 Introduction

Human activity recognition has received much attention because it is considered one of the most natural methods for improving quality of life by monitoring and supporting human life and work[1][2][3][4]. Some famous systems include a system that monitors a nurse provider and automatically outputs the nurse's notes[5] and a system that monitors an assembly worker and displays procedures[6][7]. These systems recognize human motions based on sensor values, store them as data memory in the virtual world, and then output the information in the real world. However, recognition is performed on a user's consecutive motion without regard to retrial or alternation by the user even though that is

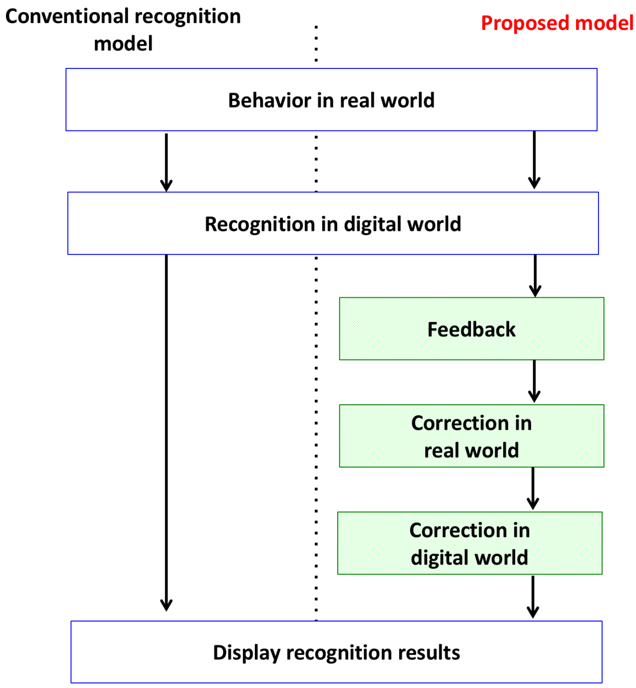


**Fig. 1.** Different process flows between the conventional recognition model and the proposed recognition model.

likely to occur. Thus, a mechanism that allows the user to interactively correct recognition errors is needed to better fit an existing human activity model by allowing repeat motion, alternation, and suchlike unanticipated behavior.

Hence, we propose a gesture-based recognition model with a mechanism for correcting recognition errors by the user without affecting the real world. The different process flows between the conventional recognition model and the proposed recognition model are described in Fig. 1. We developed a recognition system based on the proposed model to support manual grading tasks, and verified its effectiveness. Our system, named GERMIC, consists of two major modules: handwritten recognition and interactive correction. The handwritten module recognizes diagrams such as " ◯ ", " △ ", and " / " drawn by a user with a pen-shaped mouse; moreover, the module recognizes numbers drawn by user too. Diagrams are recognized by image feature extraction and numbers are recognized using convolution neural network (CNN) on a PC. The interactive correction module is called on-demand by a user-based trigger, i.e., clicking a button embedded in the pen-shaped mouse. The interactive correction mechanism is then activated over voice feedback and the user can correct any occurrence of recognition errors or any recognition of unintended action. In addition, voice feedback enables the user to make corrections without slowing down or distracting the user by having to look at the PC screen. Each recognition result is stored in the system to be used to generate an output spreadsheet. Hence, we designed GERMIC to assist with grading tasks without impacting the user's conventional way of grading manually while reducing the user's mental and physical workload.

## 2 Existing Recognition Systems

There is a number of research on the systems and services that support graders. For instance, paper-based automated grading systems like "Glyph"[8] by Xerox[1] using a formatted sheet or systems using Optical Mark Recognition (OMR)[2][9][10][11] help score papers, tests, and surveys automatically, reducing the burden on graders or evaluators. However, these systems require a rich infrastructure: formatted sheets, software, hardware, optical recognition capabilities, and so on. The sheets themselves are severely constraining as these systems do not accept responses in just any format, such as handwritten characters or diagrams, which take away flexibility and convenience for users.

There are also tablet, cloud, and web-based learning and grading technologies to assist graders. A project called CLP[12][13][14], conducted by MIT (Massachusetts Institute of Technology), is one of the most famous tablet-based learning and grading systems, which focuses on student-teacher interaction using a pen and tablet with the capability of accepting various answer formats. However, utilizing the tablet requires a lengthy and cumbersome setup including inputting all the types of questions and answers that will appear on each tablet. The system is thus focused on recognizing and collecting various type of answers efficiently without regard for errors in recognition so that it is hardly used in the real environment. In addition, the infrastructure and costs for supporting the use of tablets are considered prohibitive. With respect to cloud and web-based learning and grading, there is a lot of research and development on expanding Web-CAT (Web-based Center for Automated Testing)[15][16], which is the most widely known open source automated grading system for programming. These types of systems can accept richly expressive codes but are limited to programming assessments. Some other systems[17][18][19] that focus on more generic uses, such as automated scoring of students' writing, appears to be highly flexible with the ability to evaluate complex natural language but requires installation of a huge infrastructure, requires every user to own a PC, and does not recognize handwritten formats.

There has been a lot of research and development on recognition systems of handwritten characters[21, 22, 23, 24]. Christoph et al.[21] proposed an interactive handwriting input method using motion sensors such as accelerometer and gyroscope. They focus on the modality and intuitivity of their 3D recognition system but the system has very limited practical application. On recognition algorithms, Abdul et al.[22] proposed using a support vector machine to recognize handwritten characters, while others following a main current in handwriting

[1] Xerox: `https://www.xerox.com/`

[2] Remark: `http://remarksoftware.com/products/office-omr/`

recognition systems are utilizing deep neural network architectures such as recurrent neural network[23]. To investigate recognition accuracy, Ching et al.[24] introduced eight different classifiers for identifying handwritten digit errors. Despite these developments, none of the handwriting recognition systems have yet to be applied to the task of grading student work.

Research which assists graders investigated thus far has been forcing the user to drastically change their attitudes to grading tasks, even if the research is based on mobile assisting system, this point must be burdensome for the user. Furthermore, although it is a major way for graders and teachers to perform grading manually, a system to assist such scenes have not been investigated thus far and consequently it requires them to score each paper and to tally the final results with burdens. Therefore, a practical system that supports manual graders without affecting their conventional attitudes to grading is needed.

## 3 Proposed System

This section describes system requirements and we developed the system which meets the requirements named GERMIC (GEsture Recognition Model with Interactive Correction).

### 3.1 User Application Requirements

Currently, grading is done manually because students are still required put their answers down on paper. There is a huge demand for a system that supports manual grading, that is able to store and query the results for output and analysis, and that provides a mechanism for users to interactively correct recognition errors by the system. We propose a system that supports manual grading tasks in a class environment defined by paper-based assignments and exams.

For instance, the user will be able to grade a student paper using a pen-shaped mouse to manually draw a diagram such as a “ ◯ ” for a correct answer, “ △ ” for a partially correct answer, and “ / ” for an incorrect answer. The user can also draw a number as partial point after recognition of “ △ ”. The results are collected and stored in a database and to be used for output and analysis (e.g., automatic calculation of an individual student’s scores to obtain a total score for that student). The system is set up to provide information to the user on the recognition results to enable the user to easily correct any recognition errors. The system physically consists of a PC for storing recognition results to a database and the pen-shaped mouse for reading the trail of hand gesture while their grading papers.

### 3.2 System Overview

A system flowchart of GERMIC is shown in Fig. 2. To begin, GERMIC reads essential information from a csv file set (e.g., number of answers, number of

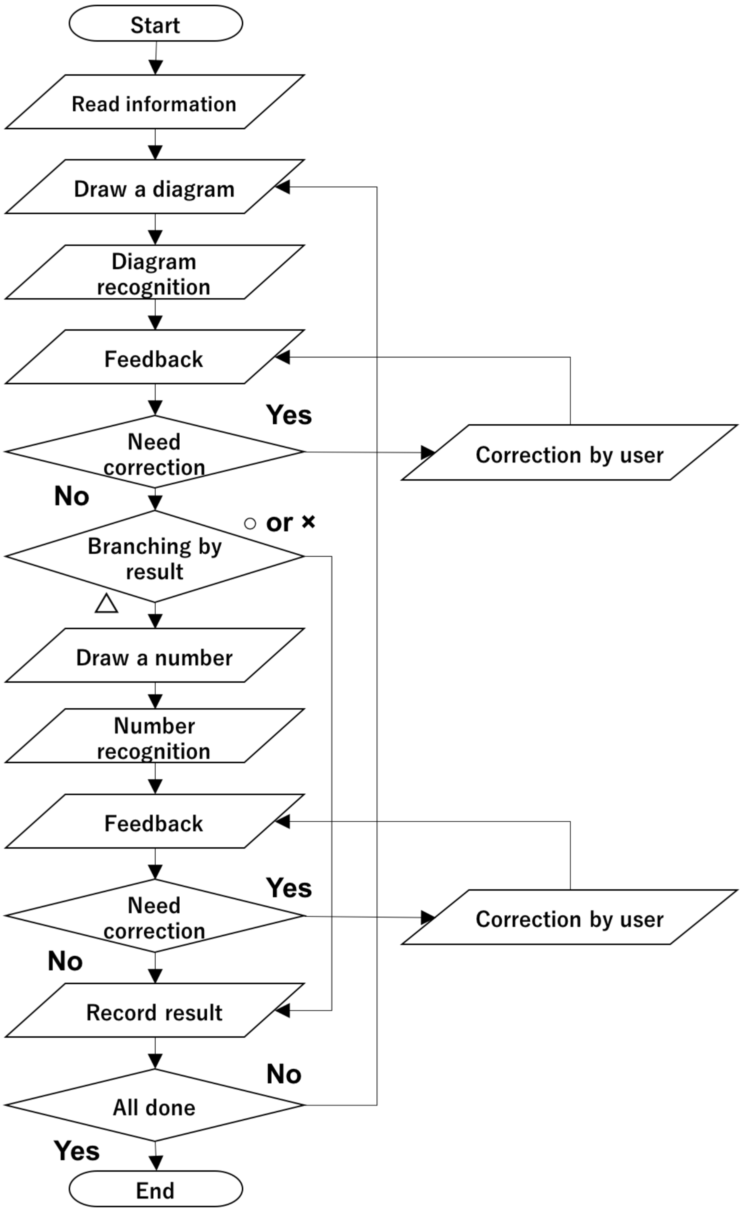


**Fig. 2.** Flowchart of GERMIC.

questions, and allotment of scores). GERMIC then classifies diagrams drawn by the user into three groups: " ◯ " (correct answer), " △ " (partially correct answer), and " / " (incorrect answer) by image feature extraction. GERMIC provides results of the recognition to the user over voice feedback. Whenever the

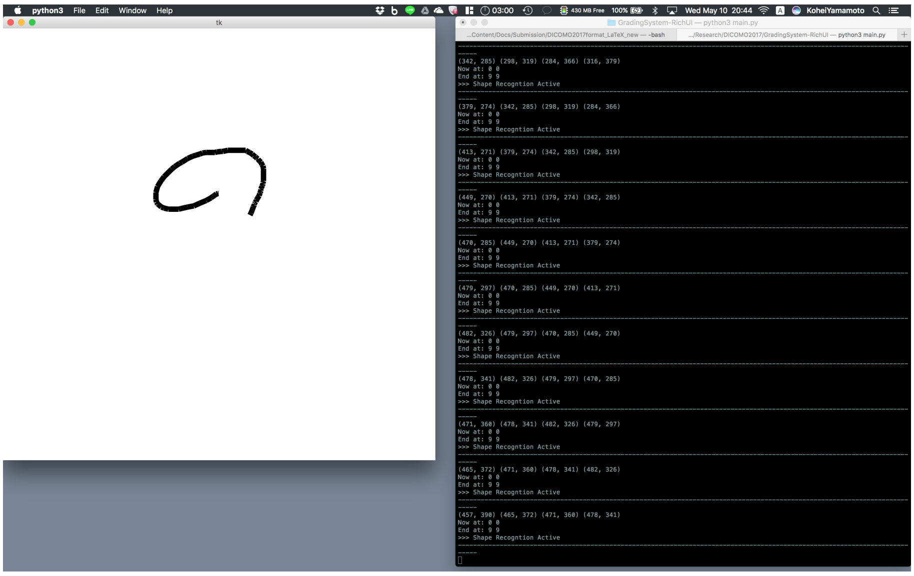

**Fig. 3.** Screen shot of the application window opened on the desktop.

user catches a recognition error, the user can correct it by pressing the button embedded in the pen-shaped mouse. The user is notified of the correction through voice feedback too. If the system recognizes a " ◯ " or " / " the score is stored in a database according to how the information was initially set up in the csv file for GERMIC. If the system reads a " △ " the user is enjoined to write a number as a grade for a partially correct answer, which is recognized by the CNN trained system. The user can also correct faulty recognition of numbers by pressing the button on the pen. When the recognition conditions are satisfied, the user is instructed to proceed to the next grading task.

Once the user is done with grading, the system automatically calculates the score and output the results in a spreadsheet. The following provides details on how GERMIC recognizes the user's handwritten notation, how recognition is achieved through image extraction, and how voice feedback works with the mechanism for interactive correction.

### 3.3 Acquisition of the User's Drawing

GERMIC, which is implemented in Python (ver.3.6.1) comprising libraries related to automated computation, is as an application that run on macOS Sierra (ver.10.12.4). After the start-up of GERMIC, a window appears on the desktop of the computer and the cursor is automatically positioned at the center of the window on the left side as shown in Fig. 3. As the user draws a form with the cursor, the trailing coordinates of the form are recorded by GERMIC from beginning to end. The entire window is stored as image data once the drawing is done. GERMIC then performs image recognition of the trailing coordinates and

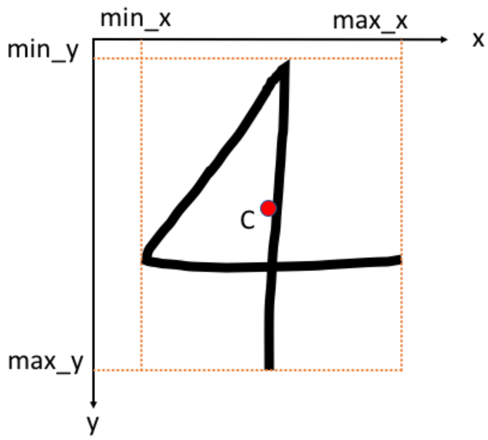


**Fig. 4.** The algorithm to cutout a number from the entire window at the time of number recognition.

the form is classified within the diagram or number category. When the user draws something, the user does not have to push or hold the button.

### 3.4 Image Recognition

GERMIC performs image recognition on diagrams and numbers drawn by the user once the entire window is converted to image data. How image recognition is materialized is described below.

**Cutout a Drawing Trail** When image recognition is performed on diagrams, the system processes the entire window since diagram recognition involves classifying the diagram based on the number of feature points contained in the window. For number recognition, the input must match the scale ratio of the image data (square image of a number) used in CNN training (to be discussed later). Thus, number recognition involves the process of cutting out the square image of the hand-drawn number on the window frame.

Fig. 4 shows the algorithm for a number cutout. First, the entire window is converted to gray scale and each pixel is scanned as the top left is zero. Then the maximum $x$ coordinate $max\_x$, the minimum $x$ coordinate $min\_x$, the maximum $y$ coordinate $max\_y$, the minimum $y$ coordinate $min\_y$, and the center coordinate $C$ of the hand-drawn number are calculated. If the drawn number is longer on the $x$-axis than the $y$-axis, then the number is squared by the size $max\_x - min\_x$ centered on $C$. If the drawn number is longer on the $y$-axis than the $x$-axis, the drawn number is squared by the size $max\_y - min\_y$.

**Diagram Recognition** Diagram recognition classifies drawn diagrams into one of three diagram types: " ◯ ", " △ ", and " / ". Thus, the expected output of a diagram recognition is any one of the three diagram types. There are several ways

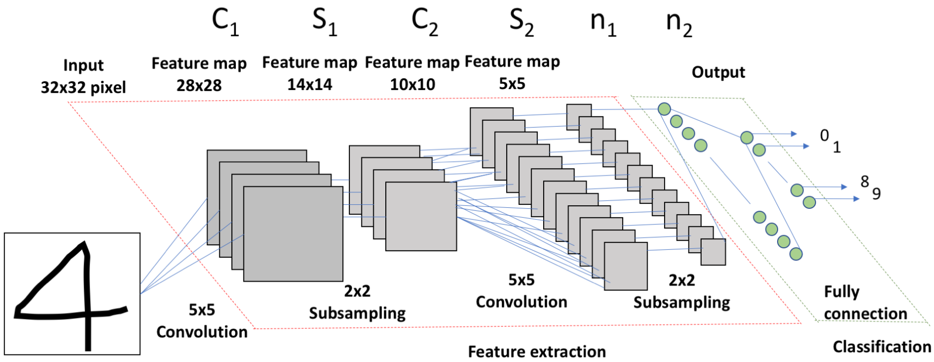


**Fig. 5.** Diagram of convolution neural network used in number recognition.

to classify diagrams but most do not execute fast enough to provide real-time feedback. Therefore, we adopted the FAST algorithm[20] for performing diagram recognition. FAST algorithm detects feature points based on the condition as to whether a certain pixel $p$ is continuously lighter or darker than the circumference of the surrounding pixels. If the condition can be satisfied, pixel $p$ is registered as a feature point. Details of the algorithm is not provided herein.

GERMIC uses the number of feature points for classifying diagrams. Image feature extraction using FAST algorithm is implemented with OpenCV3.0 (ver.3.2.0) library[3]. From our preliminary experiment, a diagram is recognized as " / " if the number of feature points is less than nine, as " △ " if greater than or equal to nine but less than 23, and as " ◯ " if greater than or equal to 23. In addition, if the number of feature points is less than three, the diagram is read as an unintended motion error and thus ignored.

**Number Recognition** When GERMIC instructs the user to draw a number, the user is required to draw a one-digit number from one to nine as credit for a partially correct answer. The number zero is not recognized by GERMIC because " / " is consider its equivalent. Number recognition is based on the CNN model as shown in Fig. 5.

The CNN model used in this paper consists of two sets of a convolution layer and a sampling layer besides a fully connected layer. Relu is used as an activation function as it is often used in the field of image recognition, moreover, softmax function is used as activation function at an output layer and probability is calculated for each number. Learning session adjusts weight (parameters) with Adam using back propagation method. Cross-entropy is used as loss function to calculate the difference between the prediction and the truth. In testing session, the CNN model calculates probability for each number and extracts a number with the highest probability as a number output.

[3] OpenCV: `http://opencv.org/opencv-3-2.html`

**Fig. 6.** “ 4 ”, “ 5 ”, and “ 7 ” drawn with one stroke.

Number recognition using CNN is implemented with a machine learning library called TensorFlow[4] which Google developed and released as open source. Besides, the CNN model is trained 20,000 times in advance with the datasets of one-digit written numbers called MNIST[5]. In the process of number recognition, the user is asked to draw a number with one stroke even if it is ”4”, ”5”, and ”7” as shown in Fig. 6 since the user draws a number without utilizing the button. In addition, feature point extraction is performed parallelly in number recognition. Then if the number of feature points is less than or equal to three, the drawing is recognized as unintended motion then ignored so to secure the redundancy.

**Interactive Correction Mechanism** When recognition is conducted, GERMIC provides the user voice feedback. For instance, when the recognition result is “ ◯ ” GERMIC pronounces ”circle”, when the result is ”1” GERMIC says ”one”. In that way, voice feedback enables the user to correct any faulty recognition on the fly without requiring the user to slow down in order to confirm the result on the computer screen.

To correct a recognition error, the user presses the button embedded in the pen-shaped mouse to trigger the correction mechanism. If the error relates to a diagram misrecognition, then the user can swap the diagram with the correct one by clicking. If the error relates to a number misrecognition, then the user can input the correct number by the number of clicks. For instance, if the intended number is ”3” but is incorrectly recognized by the system, then the user can click the button three times to input the correct number.

## 4 Evaluation

We conducted grading task experiments to evaluate the effectiveness of GERMIC. We recruited five subjects to serve as graders. They were given a tutorial and lesson to familiarize them with GERMIC. Subjects were then asked to draw three types of diagrams (“ ◯ ”, “ △ ”, and “ / ”) and nine different numbers (numbers 1 to 9) for a total of 10 times per item. We then evaluated the recognition rate. Following that, subjects were asked to utilize GERMIC during the

[4] TensorFlow: `https://www.tensorflow.org/`

[5] The MNIST database of handwritten digits: `http://yann.lecun.com/exdb/mnist/`

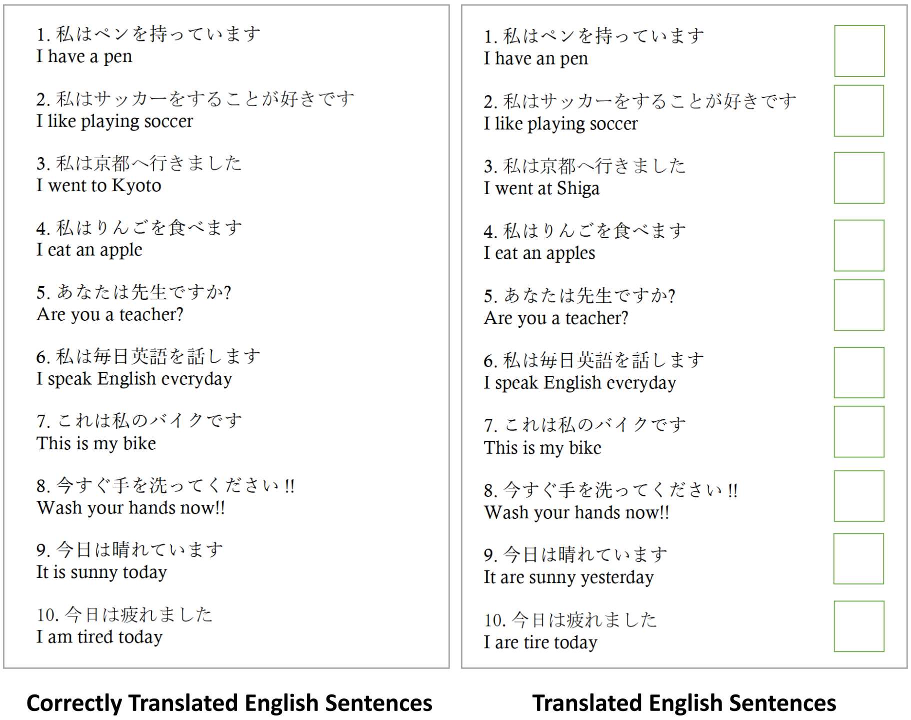


**Fig. 7.** Translated English sentences (right) are compared with the correct English sentences (left) in the grading task experiments.

grading task. As shown in Fig. 7, subjects were given a paper to grade, which contained 10 sentences that were translated from Japanese to English. Subjects were asked to score the 10 translated sentences by comparing them against the correctly translated English sentences provided them for a maximum perfect score of 100. In addition, subjects were asked to tally their results on a spreadsheet. We measured the total amount of time it took for subjects to complete their grading tasks.

Two types of experiments were performed. In the first experiment, subjects were required to grade the sentences based on the following criteria. Each English-translated sentence consisted of four words. If the four words in the translated sentence matched all four words of the correct English sentence, subjects were to draw a " ◯ " inside a square next to the sentence. If the number of un-matched words were greater than or equal to one, they were asked to draw " / ". In the second experiment, if all four words of the translated sentence matched the words in the correct English sentence, they were to draw a " ◯ "; if just one word did not match, they were to draw a " △ "; and if the total number of

**Table 1.** Confusion matrix of diagram recognition for each subject.

| O / I | Subject 1 | | | Subject 2 | | | Subject 3 | | | Subject 4 | | | Subject 5 | | | Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ◯ | △ | / | ◯ | △ | / | ◯ | △ | / | ◯ | △ | / | ◯ | △ | / | |
| ◯ | 10 | | | 8 | 2 | | 10 | | | 8 | 2 | | 9 | 1 | | 90% |
| △ | 2 | 8 | | 1 | 9 | | | 10 | | 1 | 9 | | 3 | 7 | | 86% |
| / | | | 10 | | | 10 | | | 10 | | | 10 | | | 10 | 100% |

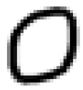

**Fig. 8.** Case that drawn circle " ◯ " misrecognized as triangle " △ ".

**Fig. 9.** Case that drawn triangle " △ " misrecognized as circle " ◯ ".

unmatched words were greater than one, they were to draw a " / ". Whenever they drew a " △ ", they were to draw a one-digit number as partial credit.

We compared the case that the subjects conducted grading without utilizing GERMIC with the case with utilizing GERMIC. In order to maintain a fairness in the comparison, we divided the subjects into the two groups: the group conducts grading with GERMIC after grading manually, the other group conducts grading manually after grading with GERMIC.

### 4.1 Accuracy of Diagram Recognition

The results of diagram recognition for each subject compared to each subjects' diagram drawing is shown in Table. 1. The recognition rate for " / " was 100% while recognition errors occurred between " ◯ " and " △ ". It appeared that " ◯ " tended to be misrecognized as " △ " because the number of discernible feature points fell below 23 whenever the circle was drawn too small as shown in Fig. 8. In contrast, " △ " tended to be misrecognized as " ◯ " whenever the number of feature points were greater than or equal to 23 such that the triangle resembled more like a circle as shown in Fig. 9.

**Table 2.** Confusion matrix of number recognition for each subject.

| I \ O | Subject 1 | | | | | | | | | Subject 2 | | | | | | | | | Subject 3 | | | | | | | | | Subject 4 | | | | | | | | | Subject 5 | | | | | | | | | Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| 1 | 10 | | | | | | | | | 10 | | | | | | | | | 8 | | | 1 | | | | | 1 | 10 | | | | | | | | | 10 | | | | | | | | | 96% |
| 2 | 1 | 9 | | | | | | | | | 9 | 1 | | | | | | | | 10 | | | | | | | | | 9 | | 1 | | | | | | | 10 | | | | | | | | 94% |
| 3 | | | 10 | | | | | | | | 2 | 7 | | | | | 1 | | | | 10 | | | | | | | | | 10 | | | | | | | | 1 | 8 | | 1 | | | | | 90% |
| 4 | | | | 8 | | | | | 2 | | | | 8 | | | | | 2 | 2 | | | 4 | | | 1 | | 3 | 1 | | | 8 | 1 | | | | | | | | 8 | | | | | 2 | 72% |
| 5 | | | | 1 | 7 | 2 | | | | | | 1 | | 9 | | | | | | | 2 | | 5 | 2 | | 1 | | | | | | 10 | | | | | | | 1 | 1 | 8 | | | | | 78% |
| 6 | | | | | | 10 | | | | | | | 1 | | 9 | | | | | | | | | 10 | | | | | | | | | 10 | | | | | | 1 | | | 9 | | | | 96% |
| 7 | 1 | | | | | | 8 | | 1 | | | 1 | | | | 8 | | | 1 | | | | | | 8 | | 2 | 2 | | | | | | 8 | | | | 1 | 1 | | | | 7 | 1 | | 70% |
| 8 | | | | | | | | 9 | 1 | | | | | | | | 10 | | | | | | | | | 10 | | | | | | | | | 10 | | | | | | | | | 9 | 1 | 96% |
| 9 | | | | 1 | | | 2 | | 7 | | | | | | | | | 10 | | | | | | | | | 10 | | | | | | | 1 | | 9 | | | | | | | 1 | 1 | 8 | 88% |

**Table 3.** Comparison of the time taken to complete grading tasks only with " ◯ " and " / " between the case without GERMIC and the case with GERMIC.

| | Without GERMIC | | | With GERMIC |
|---|---|---|---|---|
| | Scoring (sec) | Tallying (sec) | Total Grading (sec) | Grading (sec) |
| Subject 1 | 258 | 216 | 474 | 309 |
| Subject 2 | 243 | 197 | 440 | 277 |
| Subject 3 | 248 | 148 | 396 | 385 |
| Subject 4 | 183 | 230 | 413 | 284 |
| Subject 5 | 204 | 139 | 343 | 301 |

### 4.2 Accuracy of Number Recognition

The results of number recognition for each subject compared to each subject's number drawing is shown in Table. 2. Correct recognition rates for "4," "5," and "7" are lower relative to the other numbers. It is assumed that users were not used to drawing such three numbers with one stroke: "4" tended to be misrecognized as "9" whenever the horizontal line was too short as circled in red in Fig. 10, "5" tends to be misrecognized as "3" or "6" when the lines shown in a red circle in Fig. 11 stick together, and "7" tends to be misrecognized as "9" when the entire number is drawn diagonally as shown in Fig. 12 and as "1" when the width of the number is too short. The accuracy rate is lower than the 99.3% accuracy of the CNN model. This is because GERMIC automatically reads trailing without pressing the button, thus, the beginning and the end of the trail tends to get limp. At this point, we can expect further improvement of recognition accuracy by removing such fluctuation of a line.

### 4.3 Effectiveness of GERMIC in Real Environment

We evaluated GERMIC in two different real environment simulations and measured its effectiveness. Hereafter, the results are described.

**Effectiveness in Grading Tasks with " ◯ " and " / " Only.** The results of a GERMIC-based grading task when using only diagram types " ◯ " and " / " for scoring is shown in Table. 3. A grading task conducted manually without GERMIC involves scoring each sentence by marking a " ◯ " and " / " and tallying the results in a spreadsheet; whereas a grading task using GERMIC only requires the scoring each sentence with a mark since GERMIC does the rest in

**Fig. 10.** Case that drawn number “ 4 ” tends to be misrecognized as “ 9 ”.

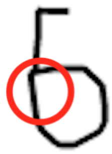

**Fig. 11.** Case that drawn number “ 5 ” tends to be misrecognized as “ 3 ” or “ 6 ”.

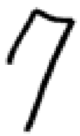

**Fig. 12.** Case that drawn number “ 7 ” tends to be misrecognized as “ 1 ” or “ 9 ”.

terms of recognition, storage, and output of the results in a spreadsheet. This difference is reflected in the results, which shows that all of subjects completed their scoring task faster without GERMIC but on the whole completed their grading task faster (102 second or 24.7% faster on average) with GERMIC.

**Effectiveness in Grading Tasks with “ ◯ ”, “ / ”, and “ △ ”** The result of a GERMIC-based grading tasks when using diagram types “ ◯ ”, “ / ”, and “ △ ” for scoring is shown in Table. 4. All subjects completed their scoring task faster without GERMIC but on the whole completed their grading task faster with GERMIC by 107.6 seconds or 14.9% on average.

We also found that the subjects especially who were good at calculations could complete their grading tasks much faster than the others and the subjects especially who were good at operating the PC could complete their tallying tasks much faster than the others.

**Table 4.** Comparison of the time taken to complete grading tasks with " ◯ ", " / ", and " △ " between the case without GERMIC and the case with GERMIC.

| | Without GERMIC | | | With GERMIC |
|---|---|---|---|---|
| | Scoring (sec) | Tallying (sec) | Total Grading (sec) | Grading (sec) |
| Subject 1 | 593 | 238 | 831 | 694 |
| Subject 2 | 407 | 353 | 760 | 555 |
| Subject 3 | 395 | 259 | 654 | 651 |
| Subject 4 | 467 | 268 | 735 | 608 |
| Subject 5 | 417 | 212 | 629 | 563 |

**Feedback from Subjects** Subjects commonly reported that they would not have completed their grading tasks as quickly without GERMIC, and that they did not feel burdened by the interactive correction mechanism once they became familiar with using it. One subject felt he to wait a little while for voice feedback (feedback was given within 1 second after starting processing the handwritten character). We intend to continue improving GERMIC to reduce delay time. Another subject stated that she would have preferred voice over manual correction of a recognition error. We intend to look for a better interface to perform corrections and to find other applications for GERMIC.

## 5 Conclusion

The usage of conventional systems which support graders is limited from the view of environment and infrastructure e.g., automatic grading machines and tablet-based scoring systems require rich infrastructure. Besides, the system which supports manual grading hardly exists. When it comes to materializing such system, we must think of applying hand gesture recognition as there is a large number of systems perform gesture recognition. However, in such systems, recognition is performed on a user's consecutive motion without regard to retrial or alternation by the user even though that is likely to occur.

We proposed GERMIC as a gesture-based recognition system to assist with manual grading tasks. The important feature of GERMIC is its interactive correction mechanism, which is the integration of handwritten character recognition and voice feedback to enable users to correct recognition errors. We evaluated the effectiveness of GERMIC by conducting grading task experiments using the system. Subjects were asked to use GERMIC as they were grading a translation exercise. We found that all subjects completed their grading tasks much faster using GREMIC than when they were manually grading. In addition, handwritten diagrams and numbers were recognized by the system with high accuracy. Subjects indicated that they did not feel burdened by using the interactive correction mechanism once they became familiar with using it. Therefore, GERMIC significantly shortened the total time for completing the grading tasks without burdening the user and demonstrated the effectiveness of the interactive correction mechanism based on the gesture recognition model.

## Acknowledgement

This research has been supported by the Kayamori Foundation of lnformational Science Advancement.

## References

1. A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors", **ACM Computing Surveys (CSUR)**, Vol. 46, No. 3, pp. 33 (2014).
2. J. L. R. Ortiz, L. Oneto, A. Samá, X. Parra, and D. Anguit, "Transition-Aware Human Activity Recognition Using Smartphones", **Neurocomputing)**, Vol. 171, No.C pp. 754–767 (2016).
3. Z. Ren, J. Meng, J. Yuan, and Z. Zhantg, "Robust Hand Gesture Recognition with Kinect Sensor", in Proc. of **19th ACM International Conference on Multimedia**, pp. 759–760 (2011).
4. Z. Ren, J. Yuan, and Z. Zhang, "Robust Hand Gesture Recognition Based on Finger-earth Mover's Distance with a Commodity Depth Camera", in Proc. of **19th ACM International Conference on Multimedia**, pp. 1093–1096 (2011).
5. N. Kuwahara, K. Kogure, A. Ohmura, and H. Noma, "Wearable Sensors for Auto-Event-Recording on Medical Nursing - User Study of Ergonomic Design -", **2012 16th International Symposium on Wearable Computers**, pp. 8–15 (2004).
6. G. Westerfield, A. Mitrovic, and M. Billinghurst, "Intelligent Augmented Reality Training for Motherboard Assembly", **International Journal of Artificial Intelligence in Education**, Vol. 25, NO. 1, pp. 157–172 (2015).
7. R. Radkowski, J. Herrema, and J. Oliver, "Augmented Reality-Based Manual Assembly Support With Visual Features for Different Degrees of Difficulty", **Industrial and Production Engineering**, Vol. 34, No. 5, pp. 362–374 (2015).
8. W. Johnson, H. Jellinek, L. Klotz Jr., R. Rao, and S. Card, "Bridging the Paper and Electronic Worlds", in Proc. of **INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems**, pp. 507–512 (1993).
9. G. Bayar, "The Use of Hough Transform to Develop an Intelligent Grading System for the Multiple Choice Exam Papers", in Proc. of **Karaelmas Science & Engineering**, Vol. 6, No. 1, pp. 100–104 (2016).
10. J. L. P. Benedito, E. Q. Aragón, J. A. Alriols, and L. Medic, "Optical Mark Recognition in Student Continuous Assessment", **IEEE Revista Iberoamericana de Tecnologias del Aprendizaje**, Vol. 9, No. 4, pp. 133–138 (2014).
11. H. Atasoy, E. Yildirim, Y. Kutlu, and K. Tohma, "Webcam Based Real-Time Robust Optical Mark Recognition", in Proc. Part II, of **22nd International Conference on Neural Information Processing**, Vol. 9490, pp. 449–456 (2015).
12. K. Koile, K. Chevalier, C. Low, S. Pal, A. Rogal, D. Singer, J. Sorensen, K. S. Tay, and K. Wu, "Supporting PenBased Classroom Interaction: New Findings and Functionality for Classroom Learning Partner", in Proc. of **International Workshop on Pen-Based Learning Technologies (PLT '07)**, pp. 1–7 (2007).
13. K. Koile, K. Chevalier, M. Rbeiz, A. Rogal, D. Singer, J. Sorensen, A. Smith, K. S. Tay, and K. Wu, "Supporting Feedback and Assessment of Digital Ink Answers to In-Class Exercises", in Proc. of **22nd National Conference on Artificial Intelligence**, pp. 1787–1794 (2007).

14. K. S. Tay and K. Koile, “Improving Digital Ink Interpretation through Expected Type Prediction and Dynamic Dispatch”, in Proc. of **19th International Conference on Pattern Recognition**, pp. 1–4 (2008).
15. S. Edwards, “Work-in-progress: program grading and feedback generation with Web-CAT”, in Proc. of **1st ACM Conference on Learning @ Scale Conference**, pp. 215–216 (2014).
16. J. Caiza and J. M. D. Alamo, “Programming assignments automatic grading: review of tools and implementations”, in Proc. of **7th International Technology, Education and Development Conference (INTED2013)**, pp. 5691–5700 (2013).
17. M. Yamamoto, N. Umemura, and H. Kawano, “Automated Essay Scoring System Based on Rubric”, **International Conference on Applied Computing and Information Technology**, Vol. 727, pp. 177–190 (2017).
18. M. Liu, Y. Li, W. Xu, and L. Liu, “Automated Essay Feedback Generation and Its Impact in the Revision”, **IEEE Transactions on Learning Technologies**, Vol. PP, No. 99, pp. 1–1 (2016).
19. D. M. Nguyen, J. Hsieh, and G. D. Allen, “The impact of web-based assessment and practice on students’ mathematics learning attitudes”, **Mathematics and Science Teaching**, Vol. 25, No. 3, pp. 251–279 (2006).
20. E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection”, in Proc. of **9th European Conference on Computer Vision (ECCV’06)**, Vol. 1, pp. 430–443 (2006).
21. C. Amma, M. Georgi, and T. Schultz, “Airwriting: a wearable handwriting recognition system”, **Personal and Ubiquitous Computing**, Vol. 18, No. 1, pp. 191–203 (2014).
22. A. R. Ahmad, M. Khalia, C. V. Gaudin and E. Poisson, “Online handwriting recognition using support vector machine”, in Proc. of **2004 IEEE Region 10 Conference TENCON 2004**, Vol. 1, pp. 311–314 (2004).
23. P. Doetsch, M. Kozielski and H. Ney, “Fast and Robust Training of Recurrent Neural Networks for Offline Handwriting Recognition”, in Proc. of **2014 14th International Conference on Frontiers in Handwriting Recognition**, pp. 279–284 (2014).
24. C. Y. Suen and J. Tan, “Analysis of errors of handwritten digits made by a multitude of classifiers”, **Pattern Recognition Letters**, Vol. 26, No. 3, pp. 369–379 (2005).