IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 December 2023, accepted 7 January 2024, date of publication 15 January 2024, date of current version 23 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353781

RESEARCH ARTICLE

A Multilingual Handwriting Learning System for Visually Impaired People

MUHAMMAD NAZRUL ISLAM[1], RAIYAN JAHANGIR[1], (Member, IEEE), NASIF SHAHRIAR MOHIM[1], MD. WASIF-UL-ISLAM[1], ANIKA ASHRAF[1], NAFIZ IMTIAZ KHAN[1], MOHAMMAD RATUL MAHJABIN[1], ABU SALEH MUSA MIAH[2], AND JUNGPIL SHIN[2], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Military Institute of Science and Technology, Mirpur Cantonment, Dhaka 1216, Bangladesh
[2]School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan

Corresponding author: Jungpil Shin (jpshin@u-aizu.ac.jp)

This work was supported by the Competitive Research Fund of The University of Aizu, Japan.

ABSTRACT Visually impaired people have previously been brought into learning and educational systems through various forms of assistive technology, such as haptic feedback systems. Haptic systems generally need expensive equipment and support from sighted teachers. Moreover, the learning has always been carried out with letters of different alphabets mapped into some tactile pattern. Writing is a big concern for the visually impaired as most official work, like signing, is still carried out by conventional handwriting methods. Most of the existing systems are limited to teaching a single language's alphabet and basic grammar or may not provide feedback to let the learners know of their learning progress. Therefore, the objectives of this research are to develop an efficient system that includes voice-over guidance to teach writing in multiple alphabets to visually impaired people and to evaluate the performance of the proposed system. As such, a system was developed for teaching multilingual alphabets to visually impaired people with voice instructions. With the aid of a voice-over guide, learners were able to write letters with a stylus on a graphics pad. The progress assessment of the learners is carried out by an image processing algorithm and scored by a machine learning (ML) model. The Random Forest model was used due to its high accuracy (f1-score of 99.8% on test data) among the existing ten different ML algorithms. Finally, the performance and usability of this system were evaluated through an empirical study replicated with 16 participants, including four teachers and twelve visually impaired people. It was found that visually impaired people made fewer attempts to learn handwriting with the proposed system than with the normal handwriting teaching system. 100% of the participants agreed to recommend the system in the future.

INDEX TERMS Visually impaired, voice-over guide, machine learning, assistive technology.

I. INTRODUCTION

Today, due to more equitable access to educational facilities, visually impaired people can succeed in academia and the workplace [1]. According to an estimation by WHO, there are 285 million visually impaired people globally, of whom 39 million have limited vision, and 246 million have poor vision [2]. The visually impaired are essential members of society, and their education is not only a fundamental right but also a societal imperative [3]. Furthermore, an educated, visually impaired population challenges stereotypes and misconceptions, promoting a culture of understanding, empathy, and accessibility for all [4]. Thus, to empower them and equip them with the tools and skills to navigate the world, education is a must for them [5]. As such, education or learning will provide employment opportunities, independence, and self-confidence, enabling them to contribute actively to the workforce and society [6].

Therefore, the development of information and communication technology for visually impaired individuals is an

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

emerging area of research that has garnered significant attention [7], [8], [9], [10]. Although many researchers have been working to develop sign language recognition systems for hearing-impaired people [11], [12], [13], [14], few research works have been done for visually impaired individuals. However, to help visually impaired people interact with systems based on information and communication, assistive technologies such as refreshable Braille displays, screen readers, and digital screen magnification are used.

Assistive technology refers to any device or software designed to help visually impaired individuals use computers, cell phones, and tablets. SmartTouch is one of the many technologies that are mainly a sensor placed on the skin, and it can collect the stimulated nerves beneath the skin [15]. In the same way, to improve human-system communication, Optacon [16] and tongue display unit [17] can play a crucial unit for the fingertips and tongue tactile visual, respectively. In addition, smartphone-based tactile vision substitution, namely the Hamsatouch [18] system, is also considered to represent cutting-edge technology for human system communication. However, many challenges arise with touch-based mobile devices besides the mentioned advancement [19]. The challenges become especially acute for visually impaired individuals, given the absence of tactile indications and the high visual demands associated with these technologies. Although certain web and smartphone apps have been created for visually impaired learners and have successfully been used to teach Braille [20], [21], learning to write is more complicated due to a lack of visual feedback. Despite rapid advancements in digital technology, handwriting is essential to daily life. Intelligent applications are in development, incorporating machine learning (ML) algorithms to predict and inform the visually impaired community about their handwriting accuracy [22], [23], [24]. However, it's important to note that while these technologies can predict and generate findings on the precision of handwritten letters, they do not guarantee that users will learn to write correctly. Even though they cannot see, people with visual impairments have highly developed senses of hearing, touch, and other kinds of perception [25]. This suggests that voice-over technology can teach handwriting to visually impaired people by giving them step-by-step instructions as they write.

To tackle the challenges outlined earlier, the objectives of this study are as follows First, to develop a system that includes voice-over guidance for teaching visually impaired people handwriting in multilingual alphabets. Secondly, it will explore some machine learning algorithms to find the best model for evaluating the progress of visually impaired learners. Finally, to evaluate the performance of the proposed system. We note that an earlier version of this article was published in a conference proceeding [26]. However, substantial additional work has been done, and the results are reported in this article.

The remainder of the paper is organized as follows. Related work is discussed in Section II. Sections III and IV cover the development of the system and the ML model, respectively. Section V describes the evaluation of the system. Section VI contains the conclusion.

II. RELATED WORK

impaired individuals in learning handwriting. The primary method for teaching visually impaired individuals has been the Braille system [27]. Despite proficiency in the Braille system, individuals with visual impairments often face challenges when it comes to signing official or legal documents. Therefore, it is clear that those who are imited vision or visually impaired need to learn how to write. Stocker [28] proposed a strategy in which visually impaired people need to give a mental image in the first step of the English letters and then practised writing based on muscle memory. However, learning in this manner was challenging for visually impaired learners. Another strategy described by Huckins [29] instructed visually impaired students community in writing using the general method of handwriting training. During the learning process, a pen was held by both the teacher and the student, and the student attempted to understand how the teacher was writing a given letter. This feedback system eventually introduced the haptic system for writing, and the subject of cursive writing came up. McCoy and Leader [30] undertook the task of instructing visually impaired learners to sign their names in cursive. They developed a system wherein a cardboard replica of the student's name was created and provided to the student. This innovative approach allowed the learners to perceive the shape of the signature and gain a sense of how to write it. Although this method aided in teaching signatures, more was needed for handwriting.

Likewise, several researchers have endeavoured to create programs aimed at enhancing the handwriting skills of visually impaired individuals. One such example is EdgeWrite [31], designed specifically for individuals with motor and visual impairments. In this system, users typed text by navigating along diagonals and edges of a square hole positioned over a standard text input area. While this system utilized a distinctive set of patterns to represent letters, it is noteworthy that it was limited to English letters. McSig was another multimodal teaching and learning environment developed by Plimmer et al. [32], which allowed visually impaired people to learn character shapes, handwriting, and signatures in collaboration with teachers. It combined haptic and audio output to implement the teacher's pen input in parallel non-visual modes. Without visual cues, it can be exceedingly challenging for vision-impaired individuals to learn how to write letters (and eventually signatures). This system was therefore enhanced by Reid and Plimmer [33] by including tactile feedback, sonification, and haptic assistance to help educate visually challenged children to form letters and eventually a signature. The system also aimed to teach the spatial motor skills needed for handwriting to visually impaired people, as these are challenging to develop. Further improvements to the system were carried out by Plimmer et al. [34] by providing kinesthetic information to the student through a force-feedback haptic pen that imitated the movement of the teacher's stylus. Special mechanical devices were necessary to use the EdgeWrite and McSig systems. Similarly, haptic-based systems have also been developed to teach Arabic [35], and Kanji [36] handwriting. To avoid this problem, Wu et al. [37] created a mobile application called LightWrite to teach lowercase English letters and Arabic numerals in a specially designed font using voice-based detailed lessons. All additional equipment and teaching assistants were optional with this technique. Voice-based instructions were given to visually impaired learners to enable them to write letters on a touch screen. The learners were then informed of their progress in writing letters by handwritten text recognition.

Machine learning algorithms have been widely used for handwritten text recognition. Parres and Paredes [38] developed an approach for handwritten text recognition on historical documents. They fine-tuned a vision encoder-decoder transformer and trained it on 3 different datasets. Their model achieved a word error rate of 6.9% on the ICFHR 2014 Bentham dataset, 14.5% on the ICFHR 2016 Ratsprotokolle dataset, and 17.3% on the Saint Gall dataset. Davoudi and Traviglia [39] suggested a deep neural network architecture built with encoder-decoder networks with an added quantization layer. The model extracts discrete representation from input-text line images and predicts the output. The model's performance showed a decrease in error rate by 22% and 21.1% on IAM and ICFHR18 datasets, respectively. Kumari et al. [40] proposed LexiconNet, a handwritten paragraph text recognition system. The system employed Vertical Attention Network and Word Beam Search technologies to recognize handwritten texts. The character error rate is 3.24% on the IAM dataset, 1.13% on RIMES, and 2.43% on the READ-16 dataset. The word error rate is 8.29% on the IAM dataset, 2.94% on the RIMES dataset, and 7.35% on the READ-2016 dataset. In summary, this literature review has identified several issues. Firstly, most previous research has applied haptic feedback to teach visually impaired people how to write letters, while only one study has focused on voice-over instructions. Secondly, the majority of researchers have concentrated on instructing letters of a single language, typically English, which were predefined in the system. This implies that no additional alphabets or symbols from other languages could be introduced to the system at a later stage. Thirdly, most existing studies have involved the creation of language-specific algorithms (for example, English or Arabic) in their system to teach letters of only those alphabets. Fourthly, only a limited number of researchers have developed automated systems utilizing machine learning (ML) to ascertain whether a learner has acquired the correct handwriting skills by recognizing patterns in handwritten characters. In most studies, human assistants have been employed to assess whether visually impaired learners are acquiring correct writing skills. Finally, not all research endeavours have thoroughly evaluated their systems in terms of efficiency, effectiveness, and user satisfaction. Additionally, not all studies have conducted comparisons with previous works in the field. Therefore, this research aims to develop a system that can assist visually impaired people in writing an alphabet in any language using voice-over guidance. An algorithm is also designed to assess the learners' progress and give them feedback. A usability evaluation was conducted to determine the system's usability and compare the system's performance with previous systems.

Algorithm 1 Data Acquisition Algorithm

Input: Continue the stroke until the user says "finish."
Init :
  list of pairs of co-ordinates named dataset
  list of pairs of co-ordinates named stroke
  Initialize threshold value t

i=0;
Capture the initial stroked point as the starting point (x_2,y_2), and add it to the 'stroke' list: stroke.append(Starting point);
while *true* **do**
  $x_1, y_1 = stroke[i]$;
  $x_2, y_2 = present\ coordinate\ of\ the\ user$
  stroke.append($\{x_2, y_2\}$)
  $distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
  if *distance* $>$ *t* **then**
    dataset.append(stroke);
    i=i+1;
  end
  else
    continue
  end
end

III. SYSTEM DEVELOPMENT

The proposed system comprises a graphics pad and a software program (Figure 1). In this setup, a teacher inputs letters into the system, and a visually impaired learner engages in practice to acquire the skills of writing these letters on a graphics pad. The graphics pad is connected to a PC/laptop/notebook where the proposed software has been installed. The system incorporates voice-based navigation, allowing visually impaired learners to operate it independently without the need for assistance. A screenshot of the system's user interface is presented in Figure 2.

The proposed system consists of three sequential phases:

- Dataset acquisition: A sighted person or teacher introduces standard letters to the system.
- Incorporation of voice-over guide: The proposed system provides guidance for visually impaired learners as they write and practice multilingual letters.

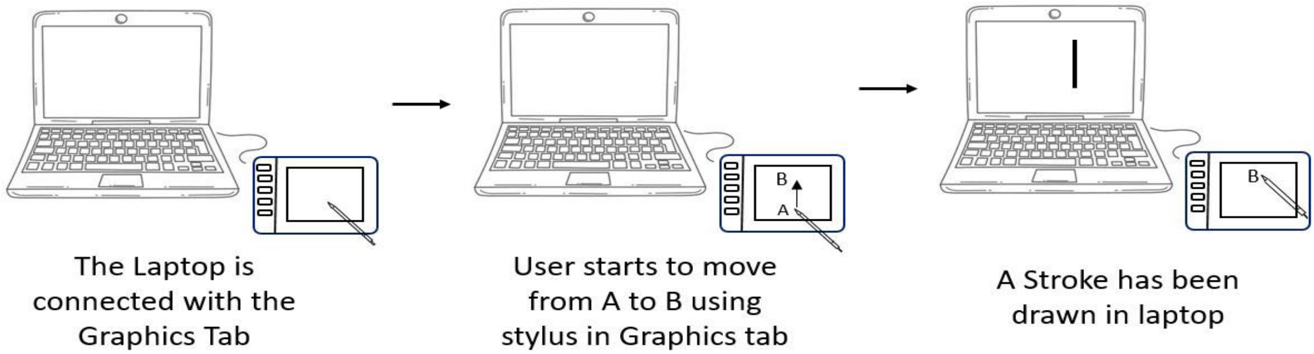


FIGURE 1. Components of the proposed system.

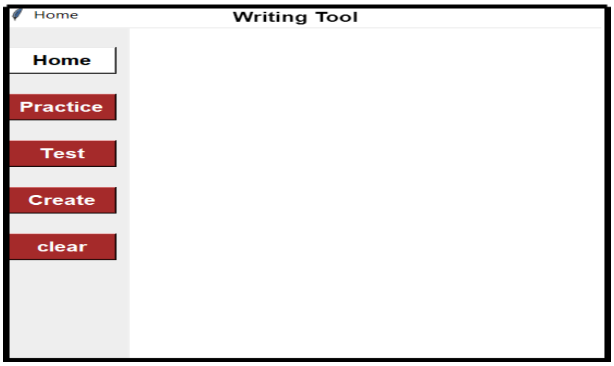


FIGURE 2. User interface of the proposed system.

- Progress assessment: The writing progress of each visually impaired learner is assessed through an ML algorithm.

A. DATASET ACQUISITION

A visually sighted teacher used a graphics pad to write the letter. Teachers teach the writing procedure to visually impaired people. Figure 3 demonstrated the workflow of this phase. The teacher begins the process by vocalizing the command "Create" to access the Create option. Afterwards, the system awaits input from the instructor for the letter's name. Subsequent to this, the teacher utilizes a stylus to write the letter on the graphics pad. The system records the written letter as a sequence of coordinate points and captures an image of the letter, subsequently storing both the image and the corresponding coordinate points. The process employed to acquire and store the coordinate points of each letter is succinctly outlined in Algorithm 1. The system maps the entire interface as a coordinate system and remains in a state of readiness until the instructor commences writing. It establishes two lists in the backend, denoted as Dataset and Stroke, and incorporates a constant value known as the threshold (t). The coordinates of the letter are stored in the Dataset sequentially in the way the teacher writes. All the coordinate points of the teacher's Stroke are kept in the Stroke, and the threshold value t determines the distance of two consecutive strokes. The t-value is set to 15 to maintain an optimal distance between adjacent coordinates, ensuring they are neither too close nor too far from each other. The initial point stroked by the instructor is recognized as the starting point (x2, y2) [Line 2] and is subsequently stored in the Stroke list [Line 3]. As the teacher continues writing, the last stored point in Stroke is set as (x1, y1) [Line 5], and the newest point is set as (x2, y2) [Line 6] and appended to Stroke [Line 7]. Subsequently, the distance between the two points (x1, y1) and (x2, y2) is determined using the Euclidean distance formula [Line 8]. The new point (x2, y2) is added to the Dataset only if the distance between the points exceeds the threshold value t [Line 9]. The method continues till the letter writing is completed. inally, the Dataset list is stored, and an image of the letter is captured. The image is then cropped to the region of interest and stored as the standard letter.

An ML model was also developed to assess the progress of visually impaired learners through the progress assessment module. The steps used to build the ML model were as follows:

- The image of each letter was augmented to give 53 more images by applying image augmentation operations (such as rotation, scaling, dilation, and erosion) for the model's training.
- The images were then transformed into an array of pixels for processing.
- Feature extraction was carried out after normalizing the pixel values.
- An ML model was built based on all these images, as discussed in Section.

B. INCORPORATION OF VOICE-OVER GUIDE

In this phase, the implementation of a voice-over guide serves to offer necessary guidance to visually impaired users. Figure 4 delineates the workflow of this stage. Visually impaired people can vocalise the command to select the "Practice" option from the menu. The system then awaits the learners' spoken input for the letter they wish to practice. Upon the learner vocalizing the letter, the system loads the pre-entered coordinate points of that letter, as provided

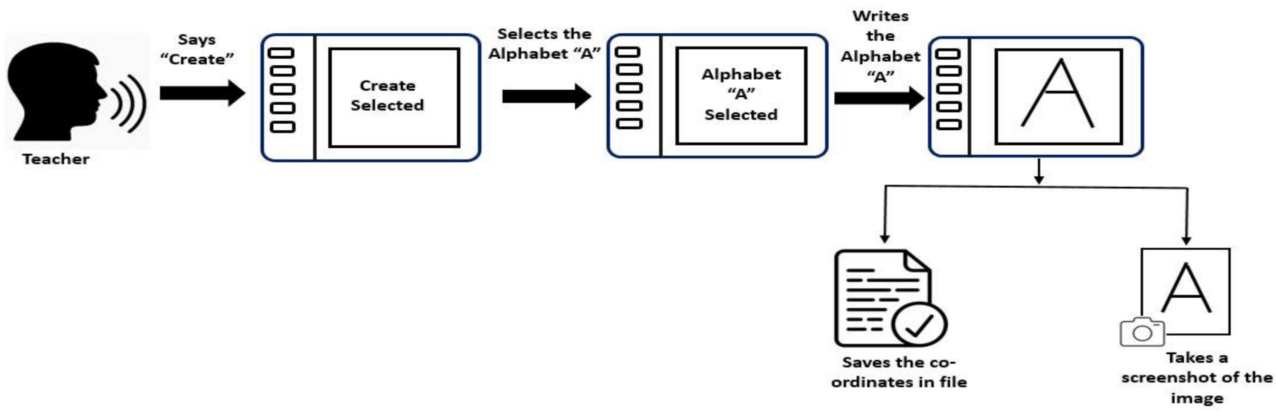


FIGURE 3. Workflow of the dataset acquisition.

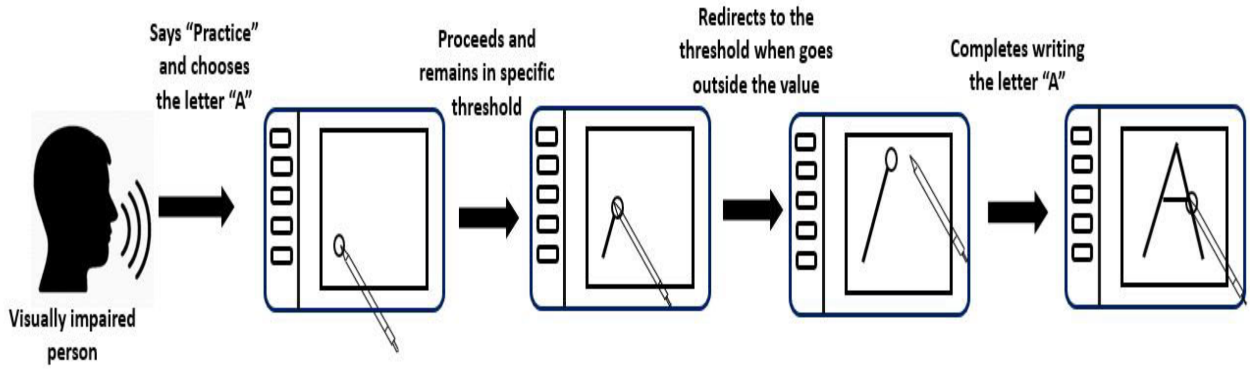


FIGURE 4. Workflow of the incorporation of voice-over guide.

by a sighted teacher, and awaits the learner's subsequent writing. The system guides the learner to start from a different position if they place their stylus in an area of the interface where it is impossible to complete the entire letter. Once the stylus is appropriately positioned, the system delivers a voice-over guide to assist the learner in writing the letter. Positive feedback, such as "Keep going," is provided by the system when the learner moves the stylus correctly until they initiate a stroke in a different direction. Due to the higher likelihood of visually impaired learners making mistakes and generating incorrect strokes, a margin surrounding each coordinate point is deemed acceptable if the learner's strokes deviate from the correct trajectory. The size of this margin is determined by the threshold value 't'. Any deviations beyond this acceptable area prompt the system to provide feedback, such as "Threshold crossed: move to the right," guiding the learner back to the correct size. The practice session concludes with the system signalling the end by saying "Done" once all the coordinates have been traversed. Learners can choose to further practice writing the letter or assess their progress through the progress assessment module.

C. PROGRESS ASSESSMENT

The visually impaired learner selects the "Test" phase by saying the command. Upon the learner vocalizing a letter,

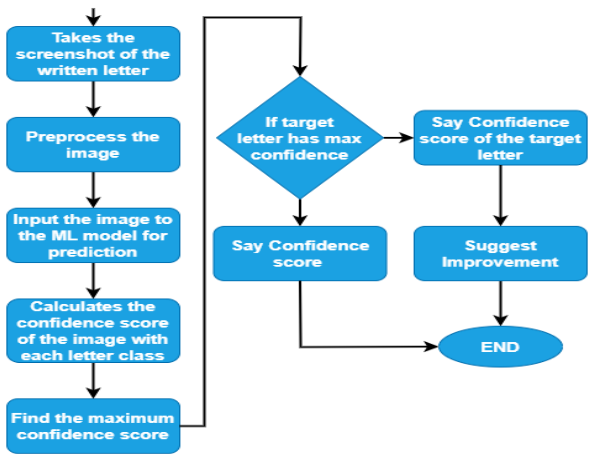


FIGURE 5. Workflow of the progress assessment phase.

the system awaits the learner's initiation of writing. The system remains inactive until the learner utters "Assess" after completing the letter. The system then evaluates the written letter and gives feedback on the learner's progress. Figure 5 shows the workflow of this phase. The workflow is described below:

- The system captures an image of a letter the learner wrote and stored this image for comparison with the standard letter.

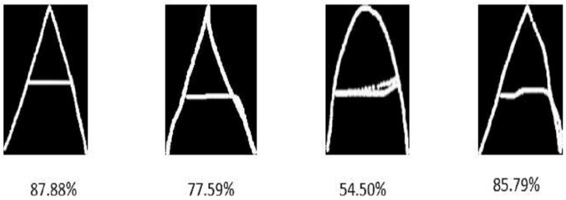


FIGURE 6. Pictorial view of different "A" written by visually impaired learner with score.

- The system performs preprocessing tasks such as cropping the image to the region of interest and resizing it to a suitable size for assessment.
- The image is then fed to the ML model, which compares this image with all the letters it has been trained.
- A confidence score is generated as a percentage for each letter.
- It then finds the letter with the maximum confidence and the confidence score for the letter drawn by the learner.
- If the letter written by the learner has the maximum confidence, then the model states the confidence score.
- If the letter written by the learner does not have the maximum confidence, then the model states the confidence score and notes that the letter written by the learner is more like the letter with the maximum score.
- If the evaluation result is 70% or more, the system considers that the letter was appropriately written; otherwise, the system will suggest practicing more. An example of the scoring is shown in Figure 6.

IV. DEVELOPMENT OF THE ML MODEL

Ten different conventional ML algorithms were used to train the model for assessing the progress of visually impaired students in learning to write a letter, and the performance of each model was evaluated in terms of the precision, recall, and F1 scores and compared. The models were developed using scikit-learn [41], and the different algorithms considered were support vector machine (SVM), K-nearest neighbors (KNN), logistic regression, Gaussian naïve Bayes, decision tree, random forest, AdaBoost, gradient boosting, XGBoost, and an artificial neural network (ANN). These algorithms were selected because they have been used in previous works and represent some basic supervised ML algorithms [42]. The proposed system predicts the letter written by a visually impaired learner and determines the score by comparing the written letter with the standard letter. The ML models were created through data collection, data preprocessing, and the development of progress assessment models with hyperparameter tuning. Finally, the performance of each model was evaluated, and the best model was selected for deployment in the system.

A. DATA COLLECTION

A dataset was formed from the images created in the dataset acquisition phase of the proposed system. In this phase, a sighted person (such as a teacher) enters letters into the system, and the system saves the pattern used to write the letter and an image of the completed letter. The dataset contained images of 26 uppercase English letters, six lowercase English letters, and 18 Bengali letters, giving 50 letters from two different alphabets. Therefore, the number of labels used by our model corresponds to 50 classes.

B. DATA PREPROCESSING

The images were cropped to the region of interest, and augmentation operations were applied to the image of each letter, such as rotation, scaling, dilation, and erosion. Each of these operations was done to create slightly different versions of each letter, as the letters written by the visually impaired learners will not be identical to the standard letters and may be slanted, smaller or larger, and thicker or thinner. However, despite these differences, the learners may have written the letters correctly, and the ML model needs to be trained to recognize them. Rotation is applied to change the orientation of a letter and is essential when the letter in the image is not aligned with the image of the standard letter. Scaling is used to adjust the size of a letter and represents letters that need to be resized to fit the standard letter. Dilation and erosion are mathematical operations used to modify the shape of an image: dilation is used to thicken or expand a letter, while erosion is used to remove or thin out a letter. Some examples of the letters generated by the image augmentation process are given in Figure 7. In this way, we obtained a total of 57 images (one standard and 56 augmented) for each letter, giving a total of $(57 \times 50) = 2850$ images in our dataset (as shown in Table 1). The images were also blurred to create smoother images since these are better for training models [43]. A random training/test split was applied to the dataset, where 80% (2280 instances) of the samples were used for the training dataset and 20% (570 instances) for the test dataset.

C. HYPERPARAMETER TUNING

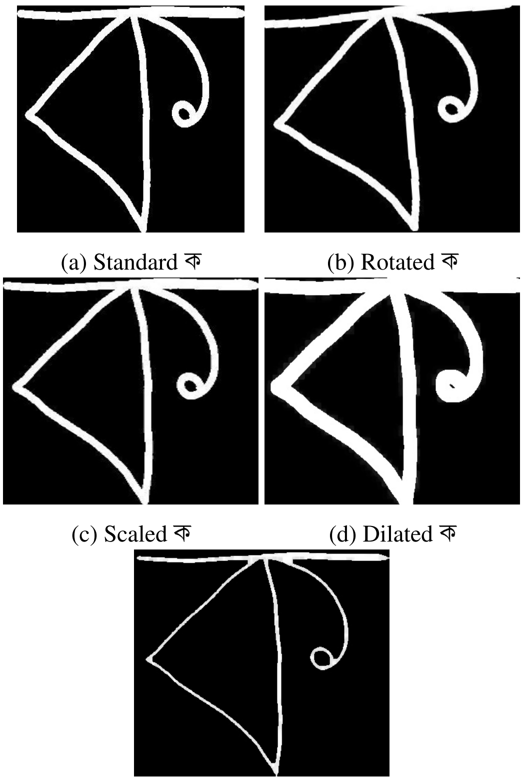
In ML, choosing a learning algorithm's ideal hyperparameters is known as hyperparameter optimization (or tuning). A hyperparameter is a parameter whose value regulates learning [44]. Different constraints, weights, or learning rates may be needed to generalize a given ML model to various data patterns. An ideal model is produced through hyperparameter optimization: a tuple of hyperparameters can be identified to minimize a predetermined loss function on independent data [45], and the goal function returns the associated loss based on this tuple of hyperparameters. The generalization performance is frequently estimated using cross-validation [46]. For our models, hyperparameter tuning was carried out with the grid search algorithm [47]. The models were trained with the hyperparameter values obtained from this process to ensure the best performance. Table 2 summarizes the hyperparameter settings for the developed models.

D. ANALYSIS OF THE MODELS

In this study, prediction models were generated using the training dataset, and their performance was evaluated on

TABLE 1. Image augmentation operation to generate images.

Operation	Hyperparameter Setting	No. of generated images
Rotation	Rotated at Degree (-5, -4.5, -4, -3.5, -3, -2.5, -2, -1.5, -1, -0.5, +0.5, +1, +1.5, +2, +2.5, +3, +3.5, +4, +4.5, +5)	20
Scaling	Scaled with ((0.1, 0.1), (0.1, 0.6), (0.1, 1.1), (0.1, 1.6), (0.6, 0.1), (0.6, 0.6), (0.6, 1.1), (0.6, 1.6), (1.1, 0.1), (1.1, 0.6), (1.1, 1.1), (1.1, 1.6), (1.6, 0.1), (1.6, 0.6), (1.6, 1.1), (1.6, 1.6)) along horizontal and vertical direction respectively	16
Dilation	Dilated with n x n kernels where value of n= (1,2,3,4,5,6,7,8,9,10)	10
Erosion	Eroded with n x n kernels where value of n= (1,2,3,4,5,6,7,8,9,10)	10



(a) Standard ক (b) Rotated ক (c) Scaled ক (d) Dilated ক (e) Eroded ক

FIGURE 7. Example images for the Bengali letter ক: (a) Standard Letter Image (b)-(e) Augmented Images.

TABLE 2. Hyper-parameter settings for the proposed ML models.

Model	Parameter	Value
SVM	Regularization parameter	10
	Kernel	Radial Basis Function
	Gamma	0.01
KNN	Number of neighbours	3
	Power parameter	Euclidean distance
Gaussian Naive Bayes	Var-smoothing	1e-10
Logistic Regression	Regularization	L2
	Inverse regularization strength	1
	Solver	Liblinear
Decision Tree	Maximum depth of tree	10
	Minimum samples for split	2
	Minimum samples for leaf node	1
Random Forest	Number of estimators	100
	Minimum samples for split	2
	Minimum samples of leaf node	1
Adaboost	Number of estimators	500
	Base estimator	Random forest
	Learning rate	0.1
Gradient Boosting	Number of estimators	500
	Learning rate	0.1
	Maximum depth of a tree	3
	Minimum samples for split	2
	Minimum samples of leaf node	1
XGBoost	Objective	Multi-softmax
ANN	Parameter solver	LBFGS
	Learning rate	1e-5
	Hidden layer	16 x 16

both the training set and an unknown dataset (test set) regarding the precision, recall, and F1 score. This study did not consider accuracy, as prior research has suggested that this is misleading when data augmentation is performed on the dataset [48]. It is also unsuitable when there is a class imbalance in the dataset. The results are shown in Table 3.

E. CROSS-VALIDATION RESULTS

Cross-validation, like the repeated random sub-sampling procedure, has no overlap between any two test sets. The learning set is divided into k-disjoint subsets of roughly equal size in k-fold cross-validation [49]. One subset is kept isolated for testing only, while the other k-1 subsets are utilized for training the model. The training and testing subsets are chosen again until every subset is tested. The testing subsets are used to estimate how well each fold performs. The 10-fold cross-validation method was used in this study on both the train and test datasets. Precision, recall, and f1-score were used to evaluate each model's performance.

TABLE 3. Performance measures for the developed models.

Model	Precision (train)	Recall (train)	F1 Score (train)	Precision (test)	Recall (test)	F1 Score (test)
SVM	0.999	0.999	0.999	0.993	0.989	0.99
Adaboost	1	1	1	0.987	0.978	0.979
Gradient Boosting	1	1	1	0.983	0.98	0.98
Random Forest	1	1	1	0.998	0.999	0.998
Decision Tree	1	1	1	0.979	0.975	0.976
KNN	1	1	1	0.998	0.999	0.998
ANN	1	1	1	0.989	0.985	0.985
Gaussian Naive Bayes	1	1	1	0.996	0.993	0.994
Logistic Regression	1	1	1	0.998	0.999	0.998

Table 4 illustrates the cross-validation performance for the created models considering 10 data folds. It is noticed that algorithms like Random Forest, SVM, Decision Tree, Gradient Boosting, KNN, and Logistic Regression have excellent performances on train data. All of their f1-scores are more than 97.5% on all cross-validation folds. However, when it comes to test data, only Random Forest consistently performs by having an f1-score of more than 94.2% on all cross-validation folds.

F. INTEGRATING THE BEST MODEL FOR THE SYSTEM

Based on the results for the performance of each model (Table 3) and the cross-validation process (Table 4), it was observed that Random Forest gave the best results in prediction and progress assessment. Therefore, a model with this algorithm was integrated into the progress assessment system.

V. SYSTEM EVALUATION

The system's usability was evaluated by measuring its effectiveness, efficiency, and user satisfaction.

A. PARTICIPANT PROFILES

A user study was conducted with 16 participants who were asked to complete several tasks. All participants were male, and the group included four teachers from a school for imited vision people. The other 12 participants were visually impaired students. Six were fully imited vision, and the remainder had limited vision. The teachers were aged between 38 and 45, while the students were at higher/secondary education level and were aged between 16 and 19. These students were undergoing education via the Braille system. None of the participants had been involved at any stage of the system development, and none had ever used a voice over guided system of this sort before.

B. STUDY PROCEDURE

The participants were briefed about the purpose of the study, which formed part of a usability evaluation, and were informed that participation was voluntary. Written consent was obtained from all of them, and biographical data were also recorded. The system was then demonstrated to the participants, and a briefing was given.

The participants were given instructions on the overall system and the working procedure. They were then given a set of tasks, as summarized in Table 5. Before the formal data collection from this process began, they were allowed to explore the system's functionalities for 15 20 minutes. The participants were observed while they were carrying out the required tasks. First, the four partially sighted participants were asked to complete Task 1. The time taken to complete the task in seconds, the number of attempts, the number of times help was requested, and the success rate were calculated for each participant. The 12 imited vision participants were asked to carry out Tasks 2 and 3. They were first given a set of English and Bengali letters made from corkboards to enable them to perceive the shape of each letter and how it should be written before starting their tasks. In a similar way to Task 1, the task completion time, the number of attempts, the number of times help was requested, and the success rate were calculated for each participant for Tasks 2 and 3.

When these tasks were complete, a comparison of the proposed system with the traditional method of teaching handwriting (where the teacher and learner hold a pen together) was carried out. In this experiment, the 12 visually impaired participants were divided into four groups of three members. Two of these groups contained three fully imited vision participants, and the other two each had three participants with partial vision. They were asked to practice writing four English letters (C, O, A, D) and two Bengali letters (য, ব, ক, ও). One group of three imited vision participants and

TABLE 4. Cross validation result for all developed models.

Model	Test	1	2	3	4	5	6	7	8	9	10	Mean
SVM	Train Precision	0.997	0.996	0.996	0.996	0.991	0.986	1	0.997	0.993	0.997	0.995
	Train Recall	0.995	0.995	0.996	0.995	0.983	0.98	1	0.996	0.986	0.995	0.992
	Train F1 Score	0.995	0.995	0.996	0.995	0.985	0.98	1	0.996	0.987	0.995	0.992
	Test Precision	0.951	0.989	0.969	1	0.989	0.94	0.94	0.969	1	1	0.975
	Test Recall	0.946	0.989	0.979	1	0.989	0.957	0.96	0.98	1	1	0.98
	Test F1 Score	0.936	0.986	0.972	1	0.986	0.946	0.947	0.973	1	1	0.975
KNN	Train Precision	1	0.994	1	0.991	0.996	0.994	0.997	0.997	0.994	1	0.996
	Train Recall	1	0.991	1	0.987	0.995	0.99	0.995	0.995	0.991	1	0.994
	Train F1 Score	1	0.992	1	0.987	0.995	0.99	0.995	0.995	0.992	1	0.995
	Test Precision	1	1	0.972	0.931	0.947	0.911	0.968	0.993	0.969	0.958	0.965
	Test Recall	1	1	0.979	0.946	0.947	0.936	0.979	0.989	0.979	0.969	0.972
	Test F1 Score	1	1	0.974	0.931	0.943	0.918	0.972	0.989	0.972	0.958	0.966
Logistic Regression	Train Precision	1	1	0.997	1	1	1	1	0.997	0.997	1	0.999
	Train Recall	1	1	0.996	1	1	1	1	0.995	0.996	1	0.999
	Train F1 Score	1	1	0.996	1	1	1	1	0.995	0.996	1	0.999
	Test Precision	0.969	0.967	0.956	0.967	0.965	1	0.963	0.94	1	1	0.972
	Test Recall	0.979	0.978	0.956	0.978	0.968	1	0.969	0.96	1	1	0.979
	Test F1 Score	0.972	0.97	0.956	0.97	0.961	1	0.962	0.947	1	1	0.974
Gaussian Naive Bayes	Train Precision	0.99	0.997	0.993	1	0.997	0.993	0.997	0.997	0.986	1	0.995
	Train Recall	0.983	0.995	0.992	1	0.996	0.985	0.995	0.996	0.973	1	0.992
	Train F1 Score	0.984	0.995	0.992	1	0.996	0.987	0.995	0.996	0.975	1	0.992
	Test Precision	0.94	0.91	1	0.968	1	1	1	0.989	1	0.965	0.977
	Test Recall	0.957	0.938	1	0.979	1	1	1	0.989	1	0.979	0.984
	Test F1 Score	0.946	0.919	1	0.972	1	1	1	0.986	1	0.968	0.979
Decision Tree	Train Precision	0.974	0.941	0.962	0.963	0.971	0.971	0.989	0.967	0.98	0.966	0.968
	Train Recall	0.958	0.933	0.955	0.953	0.979	0.965	0.981	0.949	0.97	0.968	0.959
	Train F1 Score	0.952	0.918	0.941	0.951	0.969	0.968	0.979	0.969	0.96	0.976	0.964
	Test Precision	0.812	0.723	0.858	0.841	0.816	0.646	0.812	0.801	0.851	0.795	0.775
	Test Recall	0.851	0.883	0.851	0.859	0.87	0.777	0.883	0.714	0.875	0.854	0.836
	Test F1 Score	0.808	0.808	0.85	0.805	0.709	0.783	0.778	0.785	0.769	0.821	0.794
Random Forest	Train Precision	1	1	1	0.996	0.993	0.996	1	1	1	1	0.999
	Train Recall	1	0.995	1	0.996	0.995	0.995	1	1	1	1	0.998
	Train F1 Score	1	1	1	1	0.991	0.995	1	1	1	1	0.999
	Test Precision	1	0.972	1	0.935	0.964	0.936	0.957	1	0.969	0.969	0.976
	Test Recall	1	0.979	1	0.946	0.989	0.936	0.979	1	0.979	1	0.985
	Test F1 Score	1	0.974	1	0.942	0.986	0.943	0.972	1	0.972	0.972	0.977
AdaBoost	Train precision	0.993	0.985	0.985	0.988	0.983	0.991	0.991	0.994	0.991	1	0.99
	Train recall	0.985	0.968	0.965	0.969	0.964	0.987	0.986	0.992	0.988	1	0.98
	Train F1 Score	0.986	0.972	0.97	0.973	0.967	0.987	0.987	0.992	0.987	1	0.982
	Test precision	0.889	0.946	1	0.933	0.935	0.911	0.933	0.744	0.965	0.969	0.922
	Test recall	0.906	0.957	1	0.956	0.946	0.933	0.947	0.776	0.969	0.979	0.937
	Test F1 Score	0.892	0.949	1	0.941	0.934	0.919	0.933	0.744	0.962	0.972	0.925
Gradient Boosting	Train Precision	0.997	0.988	0.991	0.997	0.997	0.993	1	0.993	0.993	0.994	0.994
	Train Recall	0.995	0.983	0.988	0.996	0.996	0.992	1	0.992	0.99	0.992	0.992
	Train F1 Score	0.995	0.984	0.988	0.996	0.996	0.991	1	0.992	0.99	0.992	0.992
	Test Precision	1	0.967	1	0.967	0.967	1	1	0.969	0.969	1	0.984
	Test Recall	1	0.978	1	0.978	0.978	1	1	0.98	0.98	1	0.989
	Test F1 Score	1	0.971	1	0.97	0.971	1	1	0.973	0.973	1	0.986
XGBoost	Train precision	0.996	0.996	0.997	0.996	0.997	0.993	0.993	0.986	0.982	0.993	0.993
	Train recall	0.996	0.996	0.996	0.995	0.996	0.992	0.99	0.983	0.978	0.991	0.991
	Train F1 Score	0.996	0.996	0.996	0.995	0.996	0.992	0.99	0.983	0.978	0.991	0.991
	Test precision	0.99	0.957	0.959	0.942	0.969	1	0.91	0.908	0.978	0.879	0.949
	Test recall	0.99	0.957	0.967	0.959	0.979	1	0.938	0.936	0.978	0.904	0.961
	Test F1 Score	0.986	0.957	0.959	0.948	0.972	1	0.919	0.918	0.97	0.883	0.951
ANN	Train Precision	0.99	0.964	0.987	0.99	1	0.989	0.989	0.993	1	0.989	0.989
	Train Recall	0.987	0.951	0.982	0.986	1	0.986	0.988	0.991	1	0.987	0.986
	Train F1 Score	0.987	0.95	0.982	0.987	1	0.986	0.987	0.991	1	0.987	0.986
	Test Precision	0.931	0.958	0.969	0.958	0.961	0.989	0.906	0.87	0.94	0.989	0.947
	Test Recall	0.948	0.969	0.98	0.969	0.968	0.989	0.935	0.9	0.957	0.989	0.96
	Test F1 Score	0.933	0.958	0.973	0.958	0.96	0.985	0.916	0.877	0.946	0.986	0.949

TABLE 5. List of tasks for user study.

Task No.	Task Description
T1	Introduce a letter to the system
T2	Practice writing to learn how to write a letter with the help of the voice-over guide of the system
T3	Assess the learning progress when a letter will be written by a participant without the help of the voice-over guide and check its accuracy

TABLE 6. The average results of user study.

Task No.	Task Completion Time (sec)(M±SD)	No. of Attempts (M±SD)	No. of help (M±SD)	Success
T1 (n=4)	12.75±1.92	1±0.0	0±0.0	100%
T2 (n=12)	65.167±39.19	1.83±0.79	0.75±0.83	54.54%
T3 (n=12)	21.167±7.66	1.4167±0.49	0.58±0.64	70.58%

TABLE 7. Effectiveness and efficiency of the developed system.

Usability Metrics	Data Type	Task No.	Mean and SD	Min	Max
Effectiveness	No. Of Attempts	Task 1	1±0.0	1	1
		Task 2	1.83±0.79	1	3
		Task 3	1.4167±0.49	1	2
Efficiency	Task Completion Time (sec)	Task 1	12.75±1.92	10	15
		Task 2	65.167±39.19	15	120
		Task 3	21.167±7.66	12	36
	No. of help	Task 1	0±0	0	0
		Task 2	0.75±0.83	0	2
		Task 3	0.58±0.64	0	2

TABLE 8. Satisfaction survey of the system.

Data Type	Answer
Overall Satisfaction	93.75% of the participants are satisfied
Easy to Use	81.25% of the participants are comfortable to use this application
Easy to Learn	87.5% of the participants agreed
Future Use	75% of participants said they would use the app again in the future.
Recommend Others	The application will be recommended by 100% of the participants.

one group of three partially sighted participants were given the proposed system to practice. In contrast, the other two groups were helped to write via the traditional process. The number of attempts needed before they could write a letter correctly was recorded for each participant, and the data were then compared. Finally, when the user study was complete, questionnaires were provided to the participants to collect feedback about the developed system. They were asked to answer 'yes' or 'no' in response to each question, and the answers were then calculated as a percentage. Both objective and subjective data were collected to assess the usability and performance of the system in terms of its effectiveness, efficiency, and user satisfaction [50], [51], [52].

C. ANALYSIS OF RESULTS

The average values of the results were calculated for each task, and the mean and standard deviation for each data type were determined. The data are summarized in Table 6.

Data on the number of attempts, task completion time (in seconds), and the number of times help was needed is presented in Table 7. The results showed that participants took an average of 12.75, 65.167, and 21.167 s to complete Tasks 1, 2, and 3, respectively, with a minimum of 10 s and a maximum of 15 s for Task 1, a minimum of 15 s and a maximum of 120 s for Task 2, and a minimum of 12 s and a maximum of 36 s for Task 3. Participants made an average of one, 1.83, and 1.4167 attempts to complete Tasks 1, 2, and 3, respectively, with a minimum of one and a maximum of one attempt for Task 1, a minimum of one and a maximum of three for Task 2, and a minimum of one and a maximum of two for Task 3. None of the participants asked for help on Task 1, whereas the average number of times help was needed for Task 2 was 0.75, and for Task 3, it was 0.58. From Table 8, it can be seen that 93.75% of the participants were satisfied with using the system, 81.25% agreed that the system was easy to use, and 87.5% said that the system was easy to learn. 75% of the participants said they would like to use it in the future, and all reported that they would recommend the system to others. The conclusions that can be drawn from this analysis are that the proposed system is effective, efficient, and satisfactory to its targeted users.

1) COMPARISON BETWEEN THE PROPOSED SYSTEM AND THE TRADITIONAL TEACHING PROCESS

The results of comparing our system and the traditional process of teaching writing are shown in Table 9. It can be seen that participants with poor vision required fewer attempts to learn to write a letter than fully imited vision participants. Participants who used our system needed fewer attempts to learn to write letters than their counterparts taught by the traditional method. The fully imited vision participants who used our system required an average of 3.33, 2.67, 4.33, 3.33, 6.33, 4, 5.67, and 7.33 attempts to learn the letters C, O, A, D, য, ব, ক, and ও, respectively. In contrast, the fully imited vision participants using the traditional method required 5.67, 5, 7.33, 5.67, 8, 4.33, 5.33, and 9.66 attempts on average, which in each case was higher than the former group. The partially sighted participants who used our system required 1, 1, 1, 1, 1.67, 1, 1.33, and 2.33 attempts on average for the letters C, O, A, D, য, ব, ক, and ও, respectively. At the same time, those learning via the traditional method took 1, 1.33, 1.67, 1, 2, 1, 1.67, and 3 attempts on average. This analysis proves that the proposed system performed better in terms of teaching handwriting of multilingual letters to visually impaired people based on the smaller number of attempts needed.

TABLE 9. Comparative analysis of attempts taken to learn writing with the help of system and the traditional method by different groups.

Type of participants	Average attempts taken by each group to learn to write a letter (Mean±SD)							
	C	O	A	D	য	ব	ক	ও
Full blind via proposed system	3.33±0.47	2.67±0.47	4.33±0.47	3.33±1.24	6.33±0.94	4±0.0	5.67±1.24	7.33±0.47
Full blind via traditional method	5.67±0.47	5±0.81	7.33±0.94	5.67±1.24	8±0.81	4.33±0.47	5.33±1.24	9.66±1.24
Low-vision via proposed system	1±0.0	1±0.0	1±0.0	1±0.0	1.67±0.47	1±0.0	1.33±0.47	2.33±0.47
Low-vision via traditional method	1±0.0	1.33±0.47	1.67±0.47	1±0.0	2±0.0	1±0.0	1.67±0.47	3±0.0

TABLE 10. Comparison between existing and proposed system.

System Features	LightWrite [37]	McSig [34]	EdgeWrite [31]	Li et al. [53]	Li et al. [54]	Hsu et al. [55]	Proposed system
Interface	Mobile App	Force Feedback Device	Palm PDA	-	-	-	Graphics Pad
Machine Learning Based	YES	NO	NO	YES	YES	YES	YES
Voice over guidance	YES	Limited usage	NO	-	-	-	YES
Language or Alphabets	English lowercase letters and digits	Primarily shapes and alphabets	English alphabet	Braille	Braille	Braille	Multilingual alphabets
Haptic Feedback Based	NO	YES	YES	-	-	-	NO
Can learn signatures from the system	NO	YES	YES	-	-	-	YES
Self assessment	YES	NO	NO	NO	NO	NO	YES
Evaluation Study	YES	YES	YES	YES	YES	YES	YES
Heavy Setup	NO	YES	YES	NO	NO	NO	NO
ML algorithm used	CNN	-	-	SVM	BraUNET	CNN	Random Forest
ML Performance	91.8% accuracy	-	-	69.6% accuracy	98.98% f1-score	98.73% accuracy	99.8% f1-score

VI. DISCUSSION AND CONCLUSION

This study has presented a design for a voice-over-guided system that visually impaired people can use to learn to write multilingual letters. The system constantly monitors the strokes the learner is making and tracks the strokes being made, and a voice-over guide gives instructions accordingly. It will also alert if the learner makes a wrong stroke or moves the stylus out of the acceptable range. This process can be used successfully to teach any alphabet and language, enabling visually impaired learners to enjoy writing. The main contributions of this study include:

- The development of a language-independent algorithm that can help visually impaired people write multilingual alphabets.
- A voice-over guide mechanism in the teaching process that saves the system from having any heavy or costly device setups.
- The incorporation of ML algorithm to assess the progress of the learners.
- A proof of an effective and user-friendly system by usability evaluation.

A comparison with prior work shows that another mobile application, LightWrite [37], offers voice-over assistance rather than haptic feedback and uses an ML-based algorithm to teach lower-case English letters and digits. Only the basic shapes and letters are covered by the McSig system [34], which offers haptic input to assist visually impaired people in learning to create a signature. Similarly to McSig, a program called EdgeWrite [31] uses specially created strokes mapped to the English alphabet to ensure adequate comprehension for those who are imited vision or visually impaired. A comparison between the proposed system and alternative schemes is given in Table 10.

The limitations of this research include the following: (i) our method has only been proven for uppercase English letters and a few Bengali letters; (ii) the voice-over guide is fixed in the proposed system for a specific alphabet; (c) the assessment module of the progress assessment algorithm was occasionally observed to be inconsistent; (d) the participants who evaluated the system were few; in future, more participants will be used for an evaluation of the system; and (e) only the conventional ML algorithms have been used in this research.

In future work, we intend to demonstrate writing in upper and lowercase English letters, all Bengali letters, numeric digits, letters from other alphabets, and various symbols. The complexity of handling a large number of classes will be solved by increasing the system's computational resources. The voice-over guide could also include artificial intelligence to give more intelligent and accurate guidance. Another potential scope of future study could be exploring the possibilities of modifying or extending machine learning or deep learning models [56] for enhancing the accuracy of progress assessment.

REFERENCES

[1] R. Singh, "Blind handicapped vs. technology: How do blind people use computers?" *Int. J. Sci. Eng. Res.*, vol. 3, no. 4, pp. 1–7, 2012.

[2] D. Pascolini and S. P. Mariotti, "Global estimates of visual impairment: 2010," *Brit. J. Ophthalmol.*, vol. 96, no. 5, pp. 614–618, May 2012.

[3] P. M. Ajuwon and B. Ruth Bieber, "Vision impairment and quality of life," *Int. Public Health J.*, vol. 6, no. 4, p. 341, 2014.

[4] P. Lynch, P. Lund, and B. Massah, "Identifying strategies to enhance the educational inclusion of visually impaired children with albinism in Malawi," *Int. J. Educ. Develop.*, vol. 39, pp. 216–224, Nov. 2014.

[5] K. Gill, R. Sharma, and R. Gupta, "Empowering visually impaired students through e-learning at higher education: Problems and solutions," *IOSR J. Humanities Social Sci.*, vol. 22, no. 8, pp. 27–35, 2017.

[6] G. B.-D. Akyeampong, "Getting in and getting on: An intersectional approach to the careers of the visually impaired in the workplace," Ph.D. dissertation, Brunel Bus. School, Brunel Univ. London, U.K., 2023.

[7] F. Habib, T. Fatema, M. Khan, N. I. Khan, and M. N. Islam, "Exploring design attributes and development of an acoustic VR game to improve ethical values of visually impaired people," in *Proc. IEEE 7th Int. Conf. Converg. Technol. (I2CT)*, Apr. 2022, pp. 1–8.

[8] F. L. Barsha, Z. Tasneem, S. Mojib, M. Afrin, N. Jahan, M. Tasnim, U. Habiba, and M. N. Islam, "An IoT based automated door accessing system for visually impaired people," in *Proc. IEEE Int. WIE Conf. Electr. Comput. Eng. (WIECON-ECE)*, Nov. 2019, pp. 1–4.

[9] M. N. Islam, T. T. Inan, N. T. Promi, S. Z. Diya, and A. Islam, "Design, implementation, and evaluation of a mobile game for blind people: Toward making mobile fun accessible to everyone," *Inf. Commun. Technol. for humanitarian services*, pp. 291–310, 2020.

[10] M. N. Islam, *Information and Communication Technologies for Humanitarian Services*. Edison, NJ, USA: IET, 2020.

[11] A. S. M. Miah, J. Shin, M. A. M. Hasan, and M. A. Rahim, "BenSignNet: Bengali sign language alphabet recognition using concatenated segmentation and convolutional neural network," *Appl. Sci.*, vol. 12, no. 8, p. 3933, Apr. 2022.

[12] A. S. M. Miah, Md. A. M. Hasan, and J. Shin, "Dynamic hand gesture recognition using multi-branch attention based graph and general deep learning model," *IEEE Access*, vol. 11, pp. 4703–4716, 2023.

[13] J. Shin, A. S. M. Miah, M. A. M. Hasan, K. Hirooka, K. Suzuki, H.-S. Lee, and S.-W. Jang, "Korean sign language recognition using transformer-based deep neural network," *Appl. Sci.*, vol. 13, no. 5, p. 3029, Feb. 2023.

[14] A. S. M. Miah, J. Shin, M. A. M. Hasan, M. A. Rahim, and Y. Okuyama, "Rotation, translation and scale invariant sign word recognition using deep learning," *Comput. Syst. Sci. Eng.*, vol. 44, no. 3, pp. 2521–2536, 2023.

[15] H. Kajimoto, N. Kawakami, and S. Tachi, "Optimal design method for selective nerve stimulation and its application to electrocutaneous display," in *Proc. 10th Symp. Haptic Interface Virtual Environ. Teleoperator Syst.*, 2002, pp. 303–310.

[16] J. G. Linvill and J. C. Bliss, "A direct translation reading aid for the blind," *Proc. IEEE*, vol. 54, no. 1, pp. 40–51, Feb. 1966.

[17] S. Kercel and P. Rita, "Human nervous system, noninvasive coupling of electronically generated data into," in *Wiley Encyclopedia of Biomedical Engineering*. Wiley, 2006.

[18] H. Kajimoto, M. Suzuki, and Y. Kanno, "HamsaTouch: Tactile vision substitution with smartphone and electro-tactile display," in *Proc. CHI Extended Abstr. Hum. Factors Comput. Syst.*, Apr. 2014, pp. 1273–1278.

[19] A. Abdolrahmani, R. Kuber, and A. Hurst, "An empirical investigation of the situationally-induced impairments experienced by blind mobile device users," in *Proc. 13th Int. Web All Conf.*, Apr. 2016, pp. 1–8.

[20] L. R. Milne, C. L. Bennett, R. E. Ladner, and S. Azenkot, "BraillePlay: Educational smartphone games for blind children," in *Proc. 16th Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2014, pp. 137–144.

[21] G. Chomchalerm, J. Rattanakajornsak, U. Samsrisook, D. Wongsawang, and W. Kusakunniran, "Braille dict: Dictionary application for the blind on Android smartphone," in *Proc. 3rd ICT Int. Student Project Conf.*, Mar. 2014, pp. 143–146.

[22] T. M. Mitchell and T. M. Mitchell, *Machine Learning*, vol. 1. New York, NY, USA: McGraw-Hill, 1997.

[23] R. Jahangir, "CNN-SCNet: A CNN net-based deep learning framework for infant cry detection in household setting," *Eng. Rep.*, vol. 2023, Oct. 2023, Art. no. e12786.

[24] M. R. Sheikh, T. H. Masud, N. I. Khan, and M. N. Islam, "An efficient transfer learning model for predicting forged (handwritten) signature," in *Proc. Int. Conf. Comput., Commun., Chem., Mater. Electron. Eng. (IC4ME2)*, Dec. 2021, pp. 1–4.

[25] J. D. Cho, "A study of multi-sensory experience and color recognition in visual arts appreciation of people with visual impairment," *Electronics*, vol. 10, no. 4, p. 470, Feb. 2021.

[26] R. Jahangir, M. Wasif-Ul-Islam, N. S. Mohim, A. Ashraf, N. I. Khan, and M. N. Islam, "Towards developing a voice-over-guided system for visually impaired people to learn writing the alphabets," in *Proc. 25th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2022, pp. 1015–1020.

[27] J. Jiménez, J. Olea, J. Torres, I. Alonso, D. Harder, and K. Fischer, "Biography of Louis Braille and invention of the Braille alphabet," *Surv. Ophthalmol.*, vol. 54, no. 1, pp. 142–149, Jan. 2009.

[28] C. Stocker, "A new approach to teaching handwriting to the blind," *J. Vis. Impairment Blindness*, vol. 57, no. 6, pp. 208–210, Jun. 1963.

[29] A. P. Huckins, "Teaching handwriting to the blind student," *J. Vis. Impairment Blindness*, vol. 59, no. 2, pp. 63–65, Feb. 1965.

[30] K. M. Mccoy and L. A. Leader, "Teaching cursive signatures to the blind: A task analytic approach," *J. Vis. Impairment Blindness*, vol. 74, no. 2, pp. 69–71, Feb. 1980.

[31] J. O. Wobbrock, B. A. Myers, and J. A. Kembel, "EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion," in *Proc. 16th Annu. ACM Symp. User Interface Softw. Technol.*, Nov. 2003, pp. 61–70.

[32] B. Plimmer, A. Crossan, S. A. Brewster, and R. Blagojevic, "Multimodal collaborative handwriting training for visually-impaired people," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2008, pp. 393–402.

[33] P. Reid and B. Plimmer, "A collaborative multimodal handwriting training environment for visually impaired students," in *Proc. 20th Australas. Conf. Comput.-Hum. Interact., Designing Habitus Habitat*, Dec. 2008, pp. 195–202.

[34] B. Plimmer, P. Reid, R. Blagojevic, A. Crossan, and S. Brewster, "Signing on the tactile line: A multimodal system for teaching handwriting to blind children," *ACM Trans. Comput.-Hum. Interact.*, vol. 18, no. 3, pp. 1–29, Jul. 2011.

[35] A. S. Bayousuf, H. S. Al-Khalifa, and A. S. Al-Salman, "Towards the development of haptic-based interface for teaching visually impaired Arabic handwriting," in *Proc. 15th Int. ACM SIGACCESS Conf. Comput. Accessibility*, Oct. 2013, pp. 1–2.

[36] Y. Murai, M. Kawahara, H. Tatsumi, and M. Miyakawa, "Kanji writing training with haptic interface for the visually impaired," in *Proc. World Autom. Congr.*, Sep. 2010, pp. 1–6.

[37] Z. Wu, C. Yu, X. Xu, T. Wei, T. Zou, R. Wang, and Y. Shi, "LightWrite: Teach handwriting to the visually impaired with a smartphone," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2021, pp. 1–15.

[38] D. Parres and R. Paredes, "Fine-tuning vision encoder–decoder transformers for handwriting text recognition on historical documents," in *Proc. Int. Conf. Document Anal. Recognit.* Cham, Switzerland: Springer, 2023, pp. 253–268.
[39] H. Davoudi and A. Traviglia, "Discrete representation learning for handwritten text recognition," *Neural Comput. Appl.*, vol. 35, no. 21, pp. 15759–15773, Jul. 2023.
[40] L. Kumari, S. Singh, V. V. S. Rathore, and A. Sharma, "A comprehensive handwritten paragraph text recognition system: Lexiconnet," in *Proc. Int. Conf. Document Anal. Recognit.* Cham, Switzerland: Springer, 2023, pp. 226–241.
[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and M. Blondel, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[42] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2016, pp. 1310–1315.
[43] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8107–8116.
[44] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning*. Cham, Switzerland: Springer, 2019, pp. 3–33.
[45] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," 2015, *arXiv:1502.02127*.
[46] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 1–15, 2012.
[47] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*.
[48] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data Mining and Knowledge Discovery Handbook*. New York, NY, USA: Springer, 2010, pp. 875–886.
[49] M. W. Browne, "Cross-validation methods," *J. Math. Psychol.*, vol. 44, no. 1, pp. 108–132, 2000.
[50] N. Bevan, J. Carter, and S. Harker, "ISO 9241–11 revised: What have we learnt about usability since 1998?" in *Proc. Int. Conf. Hum.-Comput. Interact.* Cham, Switzerland: Springer, 2015, pp. 143–151.
[51] R. Jahangir, Z. Sultana, N. S. Mohim, M. T. Islam, M. Kamal, I. Islam, and M. N. Islam, "Identifying design guidelines of interactive whiteboard interfaces through usability evaluation," in *Proc. Int. Conf. Design Digit. Commun.* Cham, Switzerland: Springer, 2023, pp. 94–107.
[52] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human–Computer Interaction*. London, U.K.: Pearson, 2003.
[53] T. Li, X. Zeng, and S. Xu, "A deep learning method for Braille recognition," in *Proc. Int. Conf. Comput. Intell. Commun. Netw.*, Nov. 2014, pp. 1092–1095.
[54] R. Li, H. Liu, X. Wang, J. Xu, and Y. Qian, "Optical Braille recognition based on semantic segmentation network with auxiliary learning strategy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 2362–2368.
[55] B.-M. Hsu, "Braille recognition for reducing asymmetric communication between the blind and non-blind," *Symmetry*, vol. 12, no. 7, p. 1069, Jun. 2020.
[56] F. M. Gouveia, B. L. D. Bezerra, C. Zanchettin, and J. R. J. Meneses, "Handwriting recognition system for mobile accessibility to the visually impaired people," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2014, pp. 3918–3923.

MUHAMMAD NAZRUL ISLAM received the B.Sc. degree in computer science and information technology from the Islamic University of Technology, Bangladesh, in 2002, the M.Sc. degree in computer engineering from Politecnico di Milano, Italy, in 2007, and the Ph.D. degree in information systems from Åbo Akademi University, Finland, in 2014. He is currently an Associate Professor with the Department of Computer Science and Engineering, Military Institute of Science and Technology (MIST), Mirpur Cantonment, Dhaka, Bangladesh. Before joining MIST, he was a Visiting Teaching Fellow with Uppsala University, Sweden; and a Postdoctoral Research Fellow with Åbo Akademi University. He was also a Lecturer and an Assistant Professor with the Department of Computer Science and Engineering, Khulna University of Engineering and Technology (KUET), Bangladesh, from 2003 to 2012. He is the author of more than 175 peer-reviewed publications in international journals and conferences. His research interests include human–computer interaction (HCI), humanitarian technology, health informatics, military information systems, information systems usability, and computer semiotics. He is a member of the Institution of Engineers, Bangladesh (IEB).

RAIYAN JAHANGIR (Member, IEEE) received the B.Sc. degree in Computer Science and Engineering from the Military Institute of Science and Technology (MIST), Dhaka, Bangladesh. He has authored more than ten peer-reviewed publications in reputed conferences and journals. His research interests include artificial intelligence, image processing, machine learning, deep learning, computer vision, bioinformatics, health informatics, HCI, and the IoT.

NASIF SHAHRIAR MOHIM is currently pursuing the B.Sc. degree with the Department of Computer Science and Engineering, Military Institute of Science and Technology (MIST), Dhaka, Bangladesh. He has four peer-reviewed publications in reputed conferences. His research interests include web app development, machine learning, cloud, DevOps, and HCI.

MD. WASIF-UL-ISLAM is currently pursuing the B.Sc. degree with the Department of Computer Science and Engineering, Military Institute of Science and Technology (MIST), Dhaka, Bangladesh. He has one peer-reviewed publication in a reputed conference. His research interests include web app development, image processing, and the IoT.

ANIKA ASHRAF received the B.Sc. degree from the Military Institute of Science and Technology (MIST), Dhaka, Bangladesh. She is a Lecturer of computer science and engineering with MIST. She has two peer-reviewed publications in reputed conferences. Her research interests include web app development, machine learning, image processing, and robotics.

NAFIZ IMTIAZ KHAN received the B.Sc. degree from the Military Institute of Science and Technology (MIST), Dhaka, Bangladesh. Demonstrating his academic prowess, he has authored over 30 peer-reviewed publications featured in esteemed international journals and conferences. His areas of expertise encompass a diverse range, spanning data science, machine learning, image processing, edge computing, and bioinformatics.

MOHAMMAD RATUL MAHJABIN received the B.Sc. degree in software engineering from the Islamic University of Technology (IUT), in 2023. He is currently a Research Assistant with the Military Institute of Science and Technology (MIST). His research interests include human–computer interaction (HCI), deep learning in computer vision, and health informatics.

ABU SALEH MUSA MIAH received the B.Sc.Eng. and M.Sc.Eng. degrees in computer science and engineering from the Department of Computer Science and Engineering, University of Rajshahi, Rajshahi, Bangladesh. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, The University of Aizu, Japan, in 2021, under a scholarship from the Japanese Government (MEXT). He became a Lecturer and an Assistant Professor with the Department of Computer Science and Engineering, Bangladesh Army University of Science and Technology (BAUST), Saidpur, Bangladesh, in 2018 and 2021, respectively. He has authored or coauthored more than 35 publications published in widely cited journals and conferences. His research interests include human–computer interaction (HCI) and deep learning in computer vision.

JUNGPIL SHIN (Senior Member, IEEE) received the B.Sc. degree in computer science and statistics and the M.Sc. degree in computer science from Pusan National University, South Korea, in 1990 and 1994, respectively, and the Ph.D. degree in computer science and communication engineering from Kyushu University, Japan, in 1999, under a scholarship from the Japanese Government (MEXT). He was an Associate Professor, a Senior Associate Professor, and a Full Professor with the School of Computer Science and Engineering, The University of Aizu, Japan, in 1999, 2004, and 2019, respectively. He has coauthored more than 350 published papers for widely cited journals and conferences. His research interests include pattern recognition, image processing, computer vision, machine learning, human–computer interaction, non-touch interfaces, human gesture recognition, automatic control, Parkinson's disease diagnosis, ADHD diagnosis, user authentication, machine intelligence, bioinformatics, handwriting analysis, recognition, and synthesis. He is a member of ACM, IEICE, IPSJ, KISS, and KIPS. He served as the program chair and a program committee member for numerous international conferences. He serves as a reviewer for several major IEEE and SCI journals. He serves as an Editor for IEEE journals, Springer, Sage, Taylor and Francis, *Sensors* and *Electronics* (MDPI), and Tech Science.