IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 September 2023, accepted 21 September 2023, date of publication 4 October 2023, date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3321807

RESEARCH ARTICLE

An End-to-End Air Writing Recognition Method Based on Transformer

XUHANG TAN[1], JICHENG TONG[1], TAKAFUMI MATSUMARU[1], (Senior Member, IEEE), VIBEKANANDA DUTTA[2], (Member, IEEE), AND XIN HE[1], (Student Member, IEEE)

[1]Graduate School of Information, Production and Systems, Waseda University, Wakamatsu-ku, Kitakyushu, Fukuoka 808-0135, Japan
[2]Faculty of Mechatronics, Institute of Micromechanics and Photonics, Warsaw University of Technology, 00-661 Warsaw, Poland

Corresponding author: Xuhang Tan (tanxuhang2021ips@akane.waseda.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP22K04034; and in part by the Waseda University Grant for Special Research Project 2023C-496, Project 2022C-183, and Project 2021C-589.

ABSTRACT The air-writing recognition task entails the computer's ability to directly recognize and interpret user input generated by finger movements in the air. This form of interaction between humans and computers is considered natural, cost-effective, and immersive within the domain of human-computer interaction (HCI). While conventional air-writing recognition has primarily focused on recognizing individual characters, a recent advancement in 2022 introduced the concept of writing in the air (WiTA) to address continuous air-writing tasks. In this context, we assert that the Transformer-based approach can offer improved performance for the WiTA task. To solve the WiTA task, this study formulated an end-to-end air-writing recognition method called TR-AWR, which leverages the Transformer model. Our proposed method adopts a holistic approach by utilizing video frame sequences as input and generating letter sequences as outputs. To enhance the performance of the WiTA task, our method combines the vision transformer model with the traditional transformer model, while introducing data augmentation techniques for the first time. Our approach achieves a character error rate (CER) of 29.86% and a decoding frames per second (D-fps) value of 194.67 fps. Notably, our method outperforms the baseline models in terms of recognition accuracy while maintaining a certain level of real-time performance. The contributions of this paper are as follows: Firstly, this study is the first to incorporate the Transformer method into continuous air-writing recognition research, thereby reducing overall complexity and attaining improved results. Additionally, we adopt an end-to-end approach that streamlines the entire recognition process. Lastly, we propose specific data augmentation guidelines tailored explicitly for the WiTA task. In summary, our study presents a promising direction for effectively addressing the WiTA task and holds potential for further advancements in this domain.

INDEX TERMS Air writing recognition, transformer model, human–computer interaction (HCI).

I. INTRODUCTION

A. AIR WRITING AND ITS APPLICATION

With the rapid development of computers, human-computer interaction (HCI) technology is constantly updating and has the trend of transforming traditional human-computer interaction to natural human-computer interaction. Traditional human-computer interaction generally refers to people issuing commands to machines in different ways, including touch, speech, typing, and more, using devices such as mice, keyboards, microphones, and touchscreens [1]. After receiving the information, the machine provides feedback to the user's commands, including performing specific actions or providing appropriate information.

Traditional human-computer interaction technology mainly relies on hardware devices, which limits the flexibility of HCI, increases the cost, and requires specialized learning, it is not a natural way to interact and cannot meet people's increasing needs. Natural HCI is more closely integrated with people's daily experiences than traditional HCI. In daily human-to-human communication, people communicate with other humans through voice, gestures, etc., and gather

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

information around them through their senses such as sight and touch. The core of natural human-computer interaction is to allow people to interact with computers in the same way that they interact with each other in their daily lives. With the rapid development of Sensor Systems, various new intelligent devices have been developed, such as virtual reality (VR) headsets and augmented reality (AR) glasses [2]. The rapid spread of these devices has directly led to a rapid increase in the demand for natural human-computer interaction to replace traditional human-computer interaction.

The hand is the most flexible part of the human body and can convey information more quickly and conveniently during the interaction process with computers or other intelligent devices. Therefore, hand gesture recognition technology has gradually become a popular human-computer interaction method. On the other hand, text is one of the most important media for modern human communication [3]. Compared with other forms of information carriers, text can carry a wide variety of information and transmit it accurately and efficiently to other devices. Air writing is a human-computer interaction method that combines gesture interaction with text. As a natural method of human-computer interaction, it holds great potential.

Air-writing refers to a computer directly recognizing the actions of a person's fingers in the air [4]. Compared with traditional human-computer interaction methods, air-writing recognition technology allows users to perform six degrees of freedom of movement and write in the air in a natural, unconstrained gesture, thereby providing a more intuitive, convenient, and comfortable HCI method [5]. Unconstrained movements also represent the ability to carry more information, with a learning cost far lower than that of a keyboard, which can help people such as the disabled, the elderly, and children to interact with computers more conveniently [6].

As a science-fiction-like interaction method, air-writing has broad applications in smart homes, education, virtual reality (VR), and other fields. In the future, it could be integrated into all aspects of human life by facilitating the following practical scenarios:

(a) In a smart home system, when a user wants to search for a movie on a smart TV, instead of using a remote control to enter and select each letter, the user writes the English abbreviation of the movie in the air, and the signal is transmitted to the TV via a camera, completing the typing and search;

(b) In a preschool education scenario, children can learn different vocabulary through air writing technology, they can also play the game Write and Guess with the teacher using air-writing, and the system judges correctness by combining education with entertainment;

(c) Air writing recognition can also be used in situations where direct text input is not convenient, such as when driving, where it allows the driver to reply to simple messages in a noisy environment, and in situations where small-screen electronic devices are used for input (e.g., smart watches), it allows the user to easily enter text messages. Figure 1 shows a simple example diagram of Air-writing recognition.

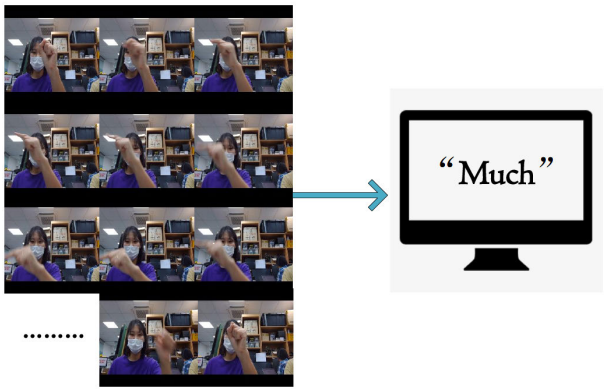


FIGURE 1. An example of Air-writing, the volunteer is writing "Much".

B. WRITING IN THE AIR (WiTA) TASK

Depending on the environment, the device, or the method of data input, Air-writing recognition can occur in various forms, such as recognizing individual characters as mentioned earlier. However, we aim to solve a different type of task called the writing in the Air (WiTA) task [7], which has the following features:

(a) Continuous air-writing recognition: Unlike previous tasks, in the WiTA task, users can write multiple characters continuously without waiting for the computer to recognize the previous character. This can greatly increase recognition efficiency.

(b) Use of visual information: Visual sensors are the most common type of sensors, and methods based on visual information can be cost-effective and easily applied in real-time situations. Technological developments in computer vision and deep learning provide even better solutions. For freehand air-writing recognition, visual-based methods are the current trend, far outmatching sensor-based methods. Visual sensors are ubiquitous and can be found on devices such as laptops and smartphones, making them simple and convenient data input devices. In practical environments, visual sensors can be deployed more easily and have the advantages of low cost, low learning cost, and non-contact measurement. Furthermore, deep learning models based on visual information have shown excellent results in recent years and can be applied to many related tasks [8]. Therefore, this study used visual devices as data input.

(c) Max identification unit is word: In the WiTA task, the maximum recognition unit is a word or sequence of characters. The evaluation criteria for the WiTA task differed, allowing for the recognition of more comprehensive information [9]. This sequence can be meaningful words or meaningless letter sequences; therefore, for our method, it is

not possible to segment an entire sentence. The minimum format for the input and output was a letter sequence.

It is worth mentioning that in this task, we did not extend the method to real-time air-writing recognition, but ensured a certain level of real-time performance by calculating the decoding frames per second (D-fps). This reflects the speed of the algorithm to a certain extent.

C. RESEARCH CHALLENGES OF CONTINUOUS AIR WRITING RECOGNITION

In previous methods for solving air-writing recognition problems, the unit of recognition was the individual characters. In contrast to the recognition of individual characters, continuous air-writing recognition has other unique characteristics, that pose significant challenges in the process of solving air-writing recognition problems. These challenges include the following.

1) POOR DATA READABILITY

For non-visual data, the sensor collects acceleration signals, angular velocity signals, or other non-textual signals, which are not spatial trajectories. It is difficult to distinguish the specific content of handwriting with the naked eye. Fig.2 shows a handwritten trajectory diagram, which makes it difficult to distinguish the original written content. This causes difficulties in data cleaning, sample segmentation, sample labeling, and other post-processing operations. For visual devices, if non-trajectory data such as spatial coordinates are obtained, this problem will also exist.

There are two main reasons for such issues in air-writing recognition.

First, information about the lifting and landing of the pen was missing. For sensors based on non-visual information, characters on the signal waveform are indistinguishable, and the connecting strokes are merged with the characters, making it difficult to perform character-level segmentation on continuous handwritten strings in the post-processing stage. In trajectory recognition methods, the connecting parts between letters can also affect the recognition results, making it sometimes difficult to distinguish whether a certain trajectory is part of a letter or a connecting stroke.

Secondly, there is interference from similar strokes. Some characters with similar strokes are prone to misidentification, such as ''a'' and ''d'', ''i'' and ''l''. For non-visual information, the difference in acceleration or angular velocity signals between writing different letters may not be significant. For trajectory recognition, the trajectories between different letters can also be easily confused, posing a challenge for air-writing recognition.

2) DIFFERENCES IN USER HABITS

Different people always have different writing habits, including stroke order, the range of hand movements, the strength and speed of writing, etc. [10]. These differences are ultimately reflected in the differences in amplitude, frequency, and trend of the sensor signal waveform or different letter-writing methods, which still exist in samples of the same category, bringing certain challenges to model recognition. Fig.3 shows the different ways of writing the letters E, F, X, and N.

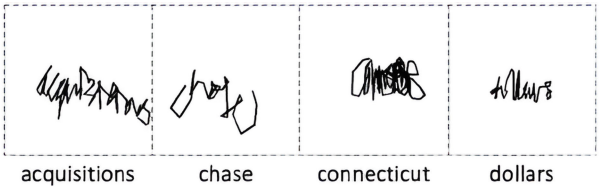


FIGURE 2. The Air-writing track sometimes is hard to recognize (newly created using the figure in [9] as material).

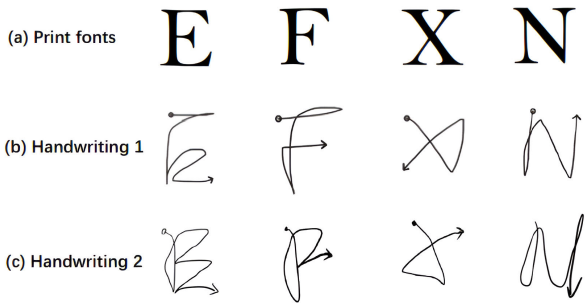


FIGURE 3. Different people have different habits for writing the same letters (newly created using the figure in [5] as material).

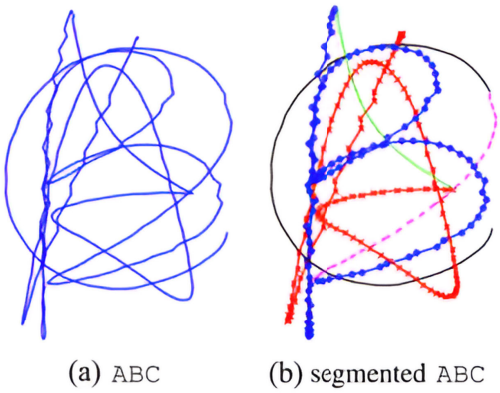


FIGURE 4. Stroke stacking occurs during continuous writing (newly created using the figure in [5] as material).

3) OVERLAPPING OF HANDWRITING SPACE

In daily life, when people write on paper, they refer to what they have written before to arrange letters in spatial order [11]. However, the characteristic of air writing is that people cannot know the previous writing content. Therefore, for trajectory recognition-based methods, once there are many characters to be detected, the pattern of each character will stack up, affecting the pattern recognition result. Figure 4 shows the stacking of strokes that occurs with continuous writing.

D. PAPER CONTRIBUTIONS

This article proposes an end-to-end air writing recognition method based on transformer models (TR-AWR), which has the following contributions:

(a) We introduce an end-to-end framework that directly takes video frame sequences as input and delivers character sequences as output, significantly reducing the complexity of the problem and avoiding information loss caused by intermediate processing steps. By training on a large benchmark dataset, we can mitigate some traditional problems of air writing to some extent.

(b) To solve the WiTA task, we employed the transformer model for the first time, which demonstrates significant contributions in other sequence-to-sequence studies. The experimental results exhibit the proposed method outperforms the baseline models in the WiTA task.

(c) We propose data augmentation principles to prevent incorrect methods from affecting the final results. By using appropriate data augmentation methods, we aim to reduce the impact of irrelevant features on the results and increase the size of the dataset, which allows us to improve the accuracy of the method further.

E. PAPER OUTLINE

This study has five main parts. Section I introduces the background of the problem, past solutions, technical difficulties, and contributions. Section II introduces the proposed method, baseline model, and contributions. Section III describes the implementation of the TR-AWR method. Section IV introduces the dataset used in the study and the specific steps of the experiments, and analyzes the experimental results. Section V presents the conclusions and discusses the possibility of further deepening this study.

II. RELATED WORK

A. AIR-WRITING RECOGNITION BASED ON NON-VISUAL SENSORS

Non-visual-based air-writing recognition mainly refers to the use of wearable devices to obtain motion information from user actions, such as inertia [12], acceleration [13], electromyography (EMG) signals [14], and WiFi signals [15]. Amma et al. developed an air-writing recognition system based on wearable gloves and inertial sensors [16], which used the signal from the inertial sensor as the input and used Hidden Markov Models (HMMs) to classify 26 English letters, achieving a recognition rate of 81.9%, Duffner et al. further integrated this method into smartphones [17], making it convenient for signature recognition and handwriting authentication.

Akl et al. proposed a motion sensor-based system, using sensors worn on the fingertips to record information such as velocity and acceleration to recognize hand gestures [18]. Fang et al. proposed a WiFi signal-based method using WiFi signal disturbances caused by hand movements for gesture recognition [15]. Tripathi et al. proposed a wearable data glove that can capture Surface Electromyography signals (MSE) from the user's body to achieve a series of human-computer interaction functions [14], including air-writing recognition. These comprehensive and diverse methods of air-writing recognition utilize analyzable data generated by human movement from multiple perspectives.

B. AIR-WRITING RECOGNITION BASED ON VISUAL SENSORS

With the development of computer vision technology, methods for air-writing recognition based on visual devices have gradually become more diverse and mature [19]. These methods primarily use visual devices to capture user-related information. The most common approach in this category is to obtain a user's hand or pen movement trajectory. The earliest attempt in this field was made in 1995 by Nabeshima et al., who proposed the MEMO-PEN system [20], which is a device equipped with a camera, pressure sensor, and microcomputer. The camera can capture each frame of the written image, and a pressure sensor is used to determine whether the pen is lifted. The trajectory of the pen tip is obtained via image processing. Xu et al. proposed a novel 3D interaction method that can recognize airborne handwritten Chinese characters using Leap Motion to accurately capture fingertip motion trajectories [21], achieving a single-character recognition rate of 90.6%.

For air-writing English word recognition, Gan et al. built an air-writing recognition system based on attention models, providing sufficient samples for training robust classifiers. They proposed a novel and effective neural network architecture based on an attention mechanism to recognize words [22], ultimately achieving a recognition accuracy of 97.74%. S. Mohammadi et al. implemented a real-time airborne system based on Kinect, using K-means clustering to detect fingertips and eliminate depth image noise [23]. They proposed a slope-change detection method to describe character features. Lee et al. proposed a visual fingertip handwriting recognition system, that uses motion and skin color detection to complete gesture segmentation. They then detected fingertips based on finger contour features and proposed an improved deep neural network to recognize handwritten characters [24]. In addition to trajectory recognition, Alam et al. used a depth camera to recognize air-writing by tracking the three-dimensional data changes of fingertip positions in space [25], Rahman et al. proposed a bone recognition-based method that uses hand key points and joint angles as input data and an RNN-LSTM structure to recognize handwritten characters [26].

Currently, air-writing recognition based on visual devices is a popular research topic that is closely integrated with the development of computer vision technology and has achieved increasingly better results.

C. PREVIOUS WORK ON CONTINUOUS AIR-WRITING RECOGNITION

As explained earlier, the WiTA task is a newly proposed problem in 2022. The only previous work on this problem was the baseline model presented in our article.

In 2022, a team of Korean researchers proposed an unconstrained air-writing recognition method using Spatio-Temporal Convolution as a baseline for this study [9]. This paper presents an end-to-end approach that directly inputs video frame data to output a sequence of letters. The authors designed a Spatio-Temporal Convolution deep learning network and proposed a judgment criterion based on the Character Error Rate (CER) to adapt to the serialized output, covering both English and Korean Air-writing recognition.

For English Air-writing recognition, the authors proposed three models: mixed 3D-2D reversed MC (ST-rMC), residual 3D convolutions (ST-R3D), and 2D convolutions followed by 1D convolutions (ST-R(2+1)D). The baseline model is the first to adopt an end-to-end approach to solve the Air-writing recognition problem and propose an evaluation criterion based on Character Error Rate (CER). However, the results of the three proposed models showed that the lowest CER data were still over 30%, indicating that there is significant room for improvement in accuracy. Therefore, this study aims to inherit the end-to-end approach and use different deep-learning methods to achieve better results.

III. PROPOSED METHOD

We propose a Transformer-based End-to-End air writing recognition method (TR-AWR), which is a concise method consisting of two main parts. The first part is data augmentation, and the second part is a deep learning network based on the transformer. It directly takes the video frame sequence as input and the character sequence as the output and calculates the CER value between the output and true sequences. Fig. 5 Shows the overall structure of the TR-AWR method, In the TR-AWR solution, the training part contains three main components, (1) data augmentation, which we only use in the training set, (2) Transformer-based deep learning model, and (3) is the calculation of Connectionist Temporal Classification (CTC) Loss. In the testing part, we used the trained transformer-based deep learning model to translate words written in the air.

A. DATA AUGMENTATION

Data Augmentation is the process of generating more representations from original data without substantially increasing the amount of data. The goal is to improve both the quantity and quality of the original data in order [27], to approximate the value of having more data. It is important to note that some general principles should be followed when using data augmentation because incorrect methods can negatively impact the final results. First, data augmentation should not generate incorrect data, such as by changing the labels of the original data or altering too many elements in the data. Second, the data should not focus on irrelevant features [28].

Based on the characteristics of air writing recognition, we propose the following principles for data augmentation for this task:

(a) Retaining important parts. The information is mainly carried by hand movements in the frame sequence of air-writing videos. This can be divided into two parts: the hand movements in a specific image, and the overall movement direction of the hand in a sequence of images. Regardless of the data augmentation method used, both parts of the information must be preserved.

(b) Using sequence as the basic unit. Because the minimum data unit in this study is a sequence of video frames, data augmentation should be performed using sequences as the basic unit. The same data augmentation method should be applied to each image within a sequence, whereas different data augmentation methods should be used randomly between different image sequences. This is done to ensure that the machine learning model focuses on hand movements within a single video frame sequence, rather than identifying different data augmentation methods as features.

Given the two principles of data augmentation mentioned above, the data augmentation methods chosen in this study differ from the traditional approaches. First, mirror augmentation was not used because the direction of hand movement carries a crucial meaning in air writing videos. Mirroring all video frames alters the hand trajectory between different frames, effectively changing the actual air writing content. If the ground truth remains the same, incorrect data will be introduced, thus affecting recognition accuracy. Similarly, methods such as frame sampling, random frame flipping, time warping, and interpolation were not utilized [29]. Another commonly used data augmentation method for videos is data mixing. However, because it is difficult to visually determine the font of the air writing, it is impossible to define the textual content written before and after a certain frame. Consequently, data augmentation through label mixing based on the video content cannot be achieved.

Next, we introduce the selected data augmentation methods in this experiment and describe how these methods reduce the influence of irrelevant features on recognition:

(a) Rotation. The images were appropriately rotated to simulate the different postures commonly used by individuals during air writing. Notably, in general, in air-writing practice applications, people interact with computers while sitting or standing, maintaining a vertical relationship with the ground. Therefore, we limited the angle range of rotation to ensure data plausibility. This data augmentation method enhances the recognition accuracy of the algorithm for air writing content with different arm angles and postures, reducing the impact of irrelevant features and improving the robustness of the model.

(b) Shift. A portion of the image was cropped and shifted, and the remaining parts were reassembled to maintain the

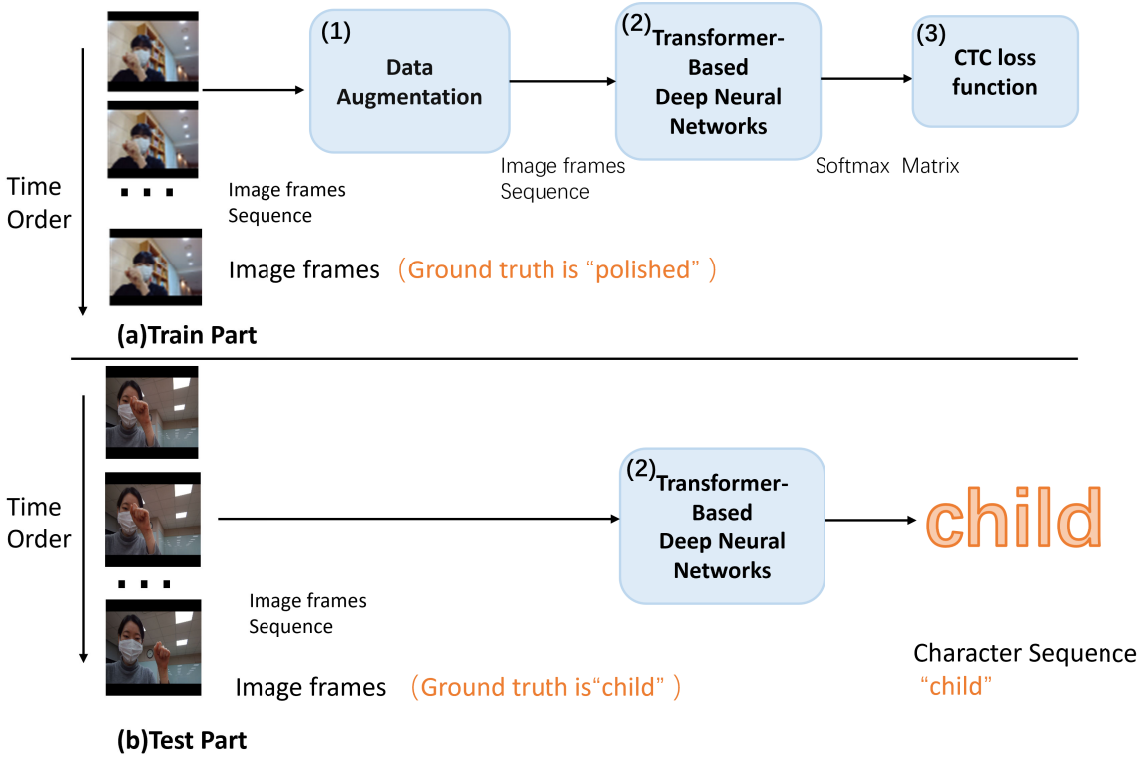


FIGURE 5. Overall structure of TR-AWR method, which includes (a) train part and (b) test part. The train part(a) includes (1) Data augmentation, (2) Transformer-based deep learning model, and (3) CTC loss function. As for the test part(b), it only involves the trained Transformer-based deep learning model.

size of the image. It is worth noting that because the individuals in the source data are generally positioned in the middle of the image, the range of cropping can be limited to preserve the integrity of the hand. Through this data augmentation method, the algorithm can better recognize the content written by volunteers in different positions within the frame, directing its attention to hand movements and reducing the influence of irrelevant features.

(c) HSV color change. The HSV color model is a color space created based on the intuitive characteristics of colors, and is also known as the hexagonal cone model [30]. The parameters in this model are the hue (H), saturation (S), and value (V). The HSV color space is more accurate for perceiving colors than the traditional RGB color space. The form of the image was altered by randomly changing these three parameters. Through this data augmentation method, the attention of the algorithm can be focused more on the hand's movement, reducing the impact of different colors and lighting conditions on the results [31].

(d) Blur. Image blurring is a common data augmentation method. Proper blurring can effectively disregard high-frequency patterns and simulate the effects of different cameras in real environments, thereby enhancing the robustness of the model.

(e) Noise. Gaussian noise refers to a type of noise whose probability density function follows a Gaussian distribution (i.e., normal distribution). By adding Gaussian noise to the image, deep neural networks can ignore certain high-frequency patterns that are not useful [32].

Data augmentation methods used in this study can be divided into two categories. First, all data will be randomly mirrored or shifted, with each sequence receiving exactly the same processing, and the ratio of the processed data to the original data is 1:1. Second, all the data will be randomly added with blur, noise, or HSV color changes, and each sequence of images will receive the same processing, with the ratio of the processed data to the original data being 1:1:1. In total, the data augmentation process results in three times the amount of data.

In conclusion, we believe that the aforementioned methods can help increase the dataset while preserving essential parts of the images, thereby enhancing the robustness of the model. These methods were carefully selected, following the two principles of data augmentation for this task, to ensure that data augmentation did not have negative effects on the experiment. Figure 6 shows five methods of data augmentation.

B. TRANSFORMER-BASED DEEP LEARNING MODEL

This study utilizes a neural network model based on transformer architecture, consisting of an image transformer for extracting visual features and a traditional text transformer

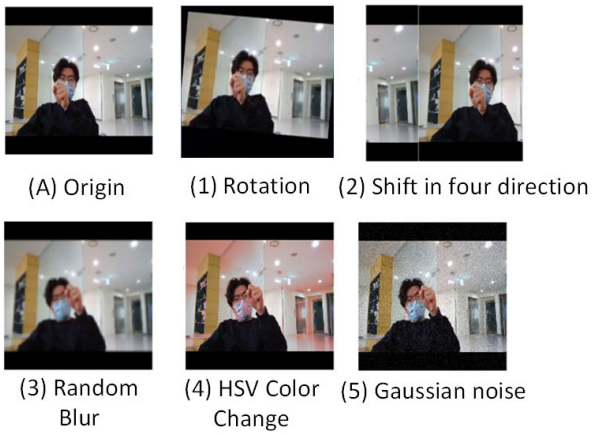


FIGURE 6. Five data augmentation methods (The images are taken from WiTA dataset [9]).

for language modeling. The structure of the article is based on a basic transformer encoder-decoder structure. The encoder is designed to extract visual features from image patches, and the decoder generates character sequences based on the visual features and previous predictions. The deep learning model mainly includes three steps: Figure 7 shows the specific structure of the transformer-based deep learning model in this study.

1) RESIZE

The WiTA task is a task that converts image sequences to character sequences. For the encoder part of the model, we referred to the ViT model's solution [33]. Because the transformer encoder cannot process image data, we input the entire image sequence and applied a fixed size to each image. Finally, the image is converted into a vector for input. The dimensions of the vector are related to the size of the image data, and data that is too large can cause calculation difficulties. Thus, the image must be resized. In this experiment, we used the bilinear method to resize a 224×224 image in a sequence into a 56×56 image and finally output a $56\times56\times$N video frame sequence, where N is the number of images in the sequence [34].

2) DATA VECTORIZATION AND POSITIONAL EMBEDDING

In addition to word embedding, the transformer also needs to use positional embedding to represent the position of words in a sentence. Because the Transformer does not adopt the structure of the RNN and uses global information, it cannot utilize word order information, which is essential for NLP. Therefore, the Transformer uses positional embedding to save the relative or absolute positions of the words in the sequence. Prior to this process, the data must be vectorized.

Any image can be represented as a $56\times56\times3$ vector because each image contains three channels. For a sequence, it can be represented as $56\times56\times$N, and the flattening process flattens the image length and height. Finally, for each image, we obtained a $3136\times$N vector.

For the traditional ViT model, large images were cut into different patches. The position of each patch in the original image will be used for positional encoding. In this study, we directly use the Visual Transformer's positional embedding [33]. However, we did not choose to cut the image but directly used the position of each image in the sequence for positional encoding, and the specific algorithm for positional encoding is consistent with ViT. Because its dimensionality is the same as that of patch embedding, the final vector length does not change.

3) TRANSFORMER ENCODER AND DECODER

In this study, we used an original transformer decoder [35]. The standard Transformer decoder also has a set of similar layers, with a structure similar to the layers in the encoder, except that the decoder inserts "encoder-decoder attention" between the multi-head self-attention and the feedforward network to allocate different attention to the encoder's output. In the encoder-decoder attention module, three input matrices are accepted: query (Q), key (K), and value (V) [35]. The key (K) and value (V) are derived from the encoder output, whereas the query is derived from the decoder input. These matrices are different representations embedded in the input through the dense or linear layers. Attention scores are evaluated by the dot product of the encoder-decoder hidden states (in the encoder-decoder attention mechanism), as shown in equation1 [35].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

In addition, the decoder uses attention masks in self-attention to prevent access to more information during training than during inference. Given that the decoder's output is shifted right by one position from the input, the attention mask ensures that the output at position "i" can only attend to the input up to position "i" in the output. We stacked three encoder layers sequentially, followed by three decoder layers. The earliest version of the transformer model used six encoder and six decoder layers. Owing to the limited training set in this study, too many layers may cause overfitting. Therefore, the depth of the deep learning model is reduced [36]. Finally, during the testing process, we use a linear layer with softmax activation to output the character sequence. For the training process, we need the softmax matrix to calculate the CTC loss. Figure 8 shows the specific structure of the Transformer encoder and decoder parts in this study.

C. FEATURES OF PROPOSED METHOD

In this study, we propose a robust and concise end-to-end method for air-writing recognition. To overcome the difficulties of traditional air writing recognition and achieve a better CER than the baseline model, our TR-AWR method has two important features. The first is that our approach

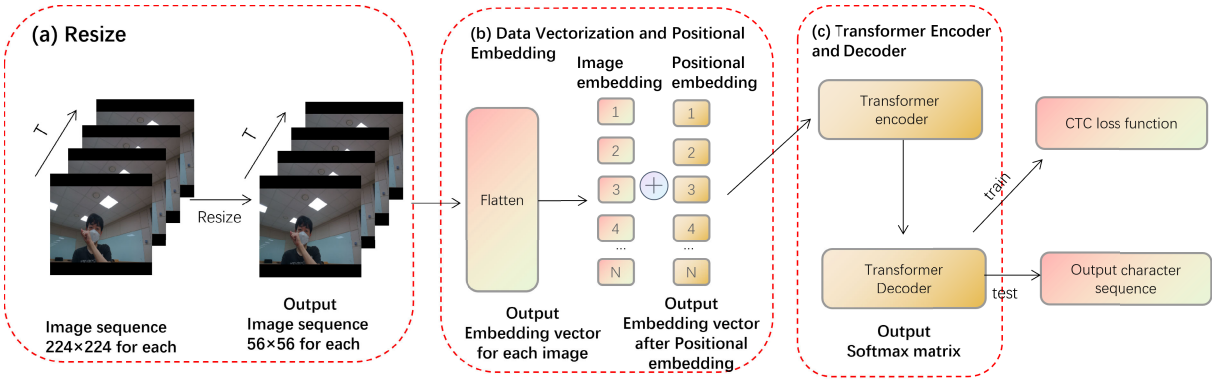


FIGURE 7. Specific structure of Transformer-based deep learning model in this study, which contains three main parts, a block: (Resize), b block: (Data Vectorization and Positional Embedding), c block: (Transformer Encoder and Decoder).

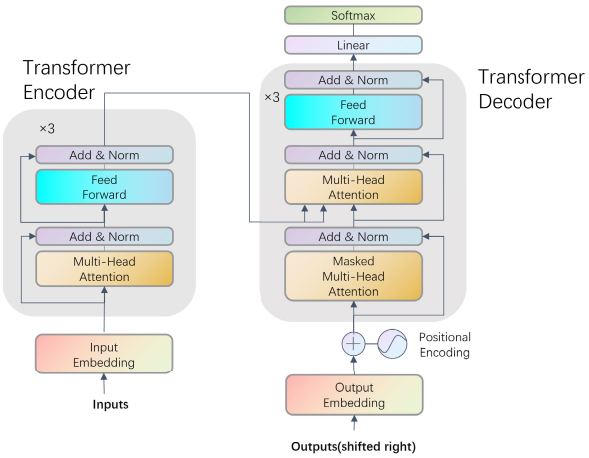


FIGURE 8. Specific structure of the Transformer encoder and decoder parts (newly created inspired from the figure in [35]). This method uses a three-layer Encoder and three-layer Decoder structure.

is based on adopting end-to-end thought, and the other important feature is that we use transformer models for recognition.

1) END-TO-END METHOD

Traditional machine learning models are often composed of multiple independent modules, and visual-based methods for freehand air writing recognition also adopt this approach. After obtaining visual information, many methods have been used to process this information. For example, trajectory images are obtained by tracking, and these images are used as input data to convert freehand air writing recognition into a handwriting recognition problem. Alternatively, hand keypoint coordinates and joint angles were extracted, or fingertip coordinates were obtained using a depth sensor, and these numbers or vectors were used as data input for air-writing recognition. Table 1 describes the intermediate steps of some methods.

TABLE 1. Intermediate steps for traditional air-wiring recognition methods.

Methods	Intermediate steps	Recognition steps
Fingertip tracking recognition [23]	Use fingertip-tracking algorithms to obtain trajectory graphs	Using a handwritten digit dataset (MNIST) as a training set to identify the content of the trajectory graph
Depth-Camera [25]	Collects fingertip coordinates (XYZ) via depth sensors	Air-writing recognition based on coordinate data using CNN networks
Hand Skeleton Recognition [26]	Obtaining hand key point coordinates and joint angles using a deep learning framework	Air writing recognition using the RNN-LSTM model based on hand key points and joint angles.

One common feature of these methods lies in their adoption of specific steps, such as hand keypoint recognition or fingertip tracking, which, however, leads to an increased chance of error in themselves, and increases the complexity of data collection and processing [37]. The difficulties mentioned in the previous section I-C also affect the recognition results.

Given these problems, we believe that an end-to-end approach should be used to solve the problem of air writing recognition. That is, the video sequence is taken as the input, the letter sequence or number sequence carried by the video is taken as the output, and the many difficulties mentioned in the previous section I-C are overcome by increasing the original data and reasonable data augmentation methods. Compared with non-end-to-end methods, this approach can avoid errors or mistakes caused by intermediate steps [38], complete retention of feature information in the image, and reduce the complexity of data collection and processing by including more different features.

2) TRANSFORMER MODEL

Based on the baseline model, we further analyzed the air writing recognition task from the end-to-end perspective. This is a sequence-to-sequence task that, inputs a video frame sequence and outputs a character sequence. The Transformer model is an excellent solution to this type kind

of problem. In recent years, the transformer model has been widely applied in computer vision tasks and has shown excellent results, such as in Optical Character Recognition (OCR) task [39]. Apart from OCR tasks, transformer models have also shown excellent results in video understanding tasks [40].

Although convolution-based methods have long held a dominant position, they also have limitations. In comparison to the model described in the baseline, we believe that the transformer model possesses several characteristics that allow for better recognition results in WiTA tasks. We believe that the following four reasons explain the transformer model's ability to achieve better results in the WiTA task.

3) SELF-ATTENTION MECHANISM

The Transformer model is based on a self-attention mechanism, that effectively captures long-range dependencies in input sequences. In air-writing recognition tasks, the input sequences often contain longer writing segments [41]. The self-attention mechanism excels in capturing the correlated information within these segments, whereas the limited receptive field of smaller convolutional operators constrains their ability to model long-range dependencies [42]. The self-attention mechanism in the transformer expands the receptive field, thereby enhancing video recognition performance.

4) PARALLEL COMPUTING CAPABILITY

The self-attention mechanism in the transformer enables direct interaction between representations at each input position and all other positions, thereby facilitating highly parallelized computation [43]. Traditional recurrent neural network (RNN) models may encounter computational efficiency limitations when handling lengthy sequences in air-writing. The parallel computing capability of the Transformer model makes it more suitable for processing long sequence data.

5) MULTI-HEAD ATTENTION MECHANISM

The Transformer model incorporates a multi-head attention mechanism, allowing simultaneous focus on different feature subspaces at distinct positions. This empowers the model to capture various types of information within the input sequences [44]. In air-writing recognition tasks, input sequences may encompass diverse stroke information and shape features. By leveraging the multi-head attention mechanism, the transformer model can effectively utilize this range of information for feature extraction and representation learning.

6) ENCODER-DECODER ARCHITECTURE

The Transformer model adopts an encoder-decoder structure, which has demonstrated outstanding performance in sequence-to-sequence tasks. In Air Writing recognition tasks, the encoder maps input sequences to high-dimensional representations, while the decoder generates corresponding recognition results based on these representations [45]. This encoder-decoder architecture enhances the model's generalization ability and recognition accuracy.

In conclusion, the transformer model's self-attention mechanism, parallel computing capability, multi-head attention mechanism, and encoder-decoder architecture contribute to its strong performance in Air Writing recognition tasks. Thus, this study applies the transformer model to air-writing recognition. Notably, the transformer model relies on a substantial amount of data. Consequently, this study proposes data augmentation criteria specifically tailored to air-writing recognition tasks and significantly increases the dataset using data augmentation methods.

D. CONNECTIONIST TEMPORAL CLASSIFICATION (CTC) LOSS FUNCTION

In this task, the input data is a sequence of images, and the output data is a string sequence. The smallest recognition unit is a word. Therefore, the air-writing recognition task in this study faced the same problem as traditional Optical Character Recognition (OCR) and machine translation tasks. It is difficult to align the input and output texts at the word level, and it is challenging to align them during pre-processing. However, if the model is trained without alignment, it will be difficult to converge owing to differences in character distances. Therefore, this study uses Connectionist Temporal Classification (CTC) [46] as the loss function.

First, we encode the text into a sequence of separate letters. We use the special character '∼' to distinguish two identical characters that appear continuously in English words. For example, the "apple" becomes (a, p, ∼, p, l, e), and "success" becomes (s, u, c, ∼, c, e, s, ∼, s).

Next, we defined the input video frame sequence as $X = [x1, x2, \ldots, XT]$, where Xi represents a certain image frame in the sequence. For such a video frame sequence X, there exists a corresponding character sequence $Y = [Y1, Y2, \ldots, YU]$, where Yi represents a specific letter in the sequence. CTC provides a solution to this problem by providing the output distribution of all possible Y for a given input sequence X. Based on this distribution, we can output the most likely result or give the probability of a certain output.

For the loss calculation, given input sequence X, we hope to maximize the posterior probability of Y. It should be differentiable such that we can gradient-descent optimization can be performed. The maximization of Y's posterior probability can be expressed in the following equation:

$$Y^* = \arg\max p(Y \mid \mathcal{X}). \tag{2}$$

After applying the CTC algorithm, the loss function for a given video sequence X and its corresponding true label Y in the training set S can be expressed in the following equation:

$$L_{ctc} = -\sum_{(\mathcal{X},\boldsymbol{Y})\in S} \ln p(\boldsymbol{Y} \mid \mathcal{X}). \tag{3}$$

TABLE 2. Comparison of traditional method, baseline method, and TR-AWR (our).

Items	Traditional method [33]	Baseline method [9]	TR-AWR method
Intermediate steps	Yes	No	No
Recognized data	Coordinate/Key Points...	Frame Sequence	Frame Sequence
Recognition format	Single character	Character Sequence	Character Sequence
Criteria for judging	Single character accuracy	CER	CER
Deep learning models	LSTM/RNN...	Spatio-Temporal Convolution	Transformer
Data Augmentation	Yes/No	No	Yes

E. SOLUTION TO RESEARCH CHALLENGES OF CONTINUOUS AIR WRITING RECOGNITION

Building upon the above-mentioned ideas, we believe that our method can solve some of the general challenges in continuous air writing recognition:

(a) To solve the issue of poor data readability, we used the most intuitive and common input format of video data, making it easy to collect, annotate, and clean data. This allowed us to quickly and conveniently increase the dataset size, avoiding the problem of poor data readability associated with sensor data acquisition.

(b) To solve the problem of user habit differences, we expanded the dataset to some extent to alleviate these issues. The dataset included air-writing videos from over 100 individuals, each with their own writing habits. Sufficient samples for similar strokes were provided by selecting an adequate vocabulary, and data augmentation was used to expand the sample size further.

(c) To solve the problem of spatial stacking, our approach recognizes the entire movement process instead of only the writing part. Unlike common methods that track fingertip trajectories and perform image recognition, our method recognizes the movement itself, which changes over time and is not stacked with previous movements.

(d) Compared to the results of the baseline model, our transformer-based method can achieve better recognition results. Table 2 presents a comprehensive comparative analysis of the key factors in the traditional air writing recognition methods [33], the baseline method [9], and TR-AWR (our).

IV. EXPERIMENT

A. DATASET

The dataset used in this study is the WiTA dataset [9], which consists of five sub-datasets in two languages (Korean and English). Only the English portion was used in this experiment, which included 10620 video sequences from 122 participants. The data were sourced from an RGB camera with a frame rate of 29 fps, and all video frames were converted to 224×224 pixel images. The data were primarily divided into two parts: English Lexical, which retrieved the top 6,000 most frequent words from the Google 1B dataset [47], accounting for 86.3% of the dataset, and English Non-Lexical, accounting for 13.7% of the dataset. The dataset authors randomly generated non-lexical words by sampling from 26 letters. The lengths of the non-lexical words range from three to seven. Figure 9 presents an example of the dataset, containing both lexical and non-lexical categories.

FIGURE 9. Examples of the air-writing text.

B. PARAMETER SETTINGS

In order to ensure rigorous training and evaluation of the models, the dataset was partitioned into three distinct groups: a training dataset comprising 80% of the total data, a validation dataset consisting of 10%, and a test dataset encompassing the remaining 10%. To conduct our experiments, we employed four NVIDIA RTX 2080ti GPUs with 11GB memory each, maintaining the same training environment as specified in the baseline model [9]. The comparison methods utilized in this study strictly adhered to the optimal settings outlined in the baseline paper [9]. In the case of our TR-AWR method, we implemented a learning rate warm-up scheme, initializing the learning rate at 1e-3. For the 6-layered Transformer models, a batch size of 128 was employed, whereas a batch size of 1 was used for measuring D-fps. We incorporated early stopping as a means of model selection, employing a stopping condition during the training procedure. It is noteworthy that all models achieved convergence within 175 training epochs.

C. COMPARISON METHODS

As stated in Section II-C, Korean researchers proposed an unconstrained air-writing recognition method using Spatio-Temporal Convolution, their solution is the comparison methods in this article. In the context of English air-writing recognition, three distinct comparison methods have been

employed. The ensuing section delineates a comprehensive exposition of these three methodologies.

1) COMPARISON METHOD 1

ST-rMC [9] means mixed reverse 3D-2D convolutions, this model contains seven main layers, the video sequence first enters the Stem Block, then passes through two 2D Conv Blocks, after which it passes through the ST-pooling layer, after which it bridges the two 3D Conv Blocks, and finally passes through the Spatial pooling layer.

2) COMPARISON METHOD 2

ST-R3D [9] means residual 3D convolutions, his model contains seven main layers, the video sequence first enters the Stem Block, then passes through two 3D Conv Blocks, after which it passes through the identity layer, after which it bridges the two 3D Conv Blocks, and finally passes through the Spatial pooling layer.

3) COMPARISON METHOD 3

ST-R(2+1)D [9] means 2D convolutions followed by 1D convolutions, this model contains seven main layers, the video sequence first enters the Stem Block, then passes through two (2+1)D Conv Blocks, after which it passes through the identity layer, after which it bridges the two (2+1)D Conv Blocks, and finally passes through the Spatial pooling layer.

D. EVALUATION CRITERIA

As the input of this experiment was a sequence of video frames and the output was a sequence of characters, we chose the Character Error Rate (CER) as the accuracy measurement instead of the accuracy of individual characters. The CER was calculated as follows:

$$CER = MCD(S, P)/length(P) \tag{4}$$

In the above equation, MCD (S, P) is the minimum character distance (Levenshtein measure [48]) between decoded phrase S and ground-truth phrase P, and length (P) is the number of characters in P. The Levenshtein distance was calculated as shown as follows.

$$\mathrm{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \mathrm{lev}(\mathrm{tail}(a), \mathrm{tail}(b)) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} \mathrm{lev}(\mathrm{tail}(a), b) \\ \mathrm{lev}(a, \mathrm{tail}(b)) \\ \mathrm{lev}(\mathrm{tail}(a), \mathrm{tail}(b)) \end{cases} & \text{otherwise} \end{cases} \tag{5}$$

The Levenshtein measure, also known as the edit distance, is a metric used to measure the difference or similarity between two sequences. It calculates the minimum number of edit operations (insertions, deletions, and substitutions) required to transform one sequence into another [49].

TABLE 3. Results of the test dataset.

Model	CER(%)			D-fps(fps)
	Lexical	N-Lexical	Overall	
TR-AWR (our)	**28.65**	**37.51**	**29.86**	**194.67**
ST-rMC [9]	95.64	96.77	95.79	460.24
ST-R3D [9]	31.30	39.48	32.41	282.36
ST-R(2+1)D [9]	91.43	93.68	91.74	267.95

The Levenshtein distance can be applied in various fields such as natural language processing, spell checking, and speech recognition [50]. It provides a universal method for measuring the similarity or difference between sequences, allowing for comparing and matching different texts or sequences. In this study, both the proposed method and the baseline models use the Levenshtein measure to calculate the Character Error Rate (CER). The CER value between the sequence and ground truth is computed for each generated sequence. The overall CER for each group is then calculated by taking the average of all the sequence results. The CER results for the lexical and N-lexical groups are weighted averages based on the number of sequences, yielding the CER values for each method.

To ensure the real-time operation of decoders, we also calculated the Average Decoding Frames per second (D-fps) performance metric. The D-fps was calculated by averaging the total number of frames decoded in one second. This metric may vary depending on the device used; however, as long as the output results are not too low, it can be ensured that the model has a certain real-time performance. The D-fps was calculated as shown as follows.

$$D - fps = \frac{Decoding\ frames\ Number}{Time(s)} \tag{6}$$

E. EXPERIMENTAL RESULTS AND ANALYSIS

According to Table 3, our method (TR-AWR) achieved CER values of 28.65 and 37.51 in the lexical and N-Lexical groups, respectively. The overall CER value was 29.86, which was significantly lower than that of the ST-rMC model (CER: 95.79) and ST-R (2+1)D model (CER: 91.74) in the baseline model. Compared to the best-performing method, ST-R3D, in the baseline model, our comprehensive error rate was reduced by approximately 3 percentage points. Additionally, our method achieved better experimental results in both lexical and N-Lexical groups, particularly in the lexical classification task.

Regarding real-time performance, we compared our method with three baseline models using the same machine. Although our D-fps data are lower than the three baseline models, the result of 194.67 still ensures a reasonable recognition speed. Overall, the proposed method outperformed the baseline models in terms of accuracy and maintained a certain level of real-time capability.

Figure 10 shows three examples of correct recognition in the results, the first image sequence is the letter "c" in the sequence "child", the second image sequence is the letter

FIGURE 10. True case of air-writing recognition by three examples. The pictures on the upper row are the sequence of images corresponding to a particular letter in the action of drawing a word. The text on the middle row shows the actual correct character and the character output by the proposed method on the lower row, which is correct.

''jet'' in the sequence ''map'', and the third image sequence is the letter ''m'' in the sequence ''map''. The first image sequence is the letter ''c'' in the sequence ''child'' and the second image sequence is the letter ''j'' in the sequence ''jet''. The third image sequence is the letter ''m'' in the sequence ''map''.

Figure 11 shows two examples of recognition errors. The first video clip is for the letter ''i'' in the sequence ''smiles'', but the result is ''l'', and the second video clip is for the letter ''c'' in the sequence ''economy'', but the result is ''e''. The second video clip is the letter ''c'' of the letter sequence ''economy'', but the result is the letter ''e''. The reason for this result may be related to the difficulty in recognizing continuous air writing that we analyzed in section I-C, where the two falsely recognized letters are very similar, and the links between the letters also interfere with the accuracy of the recognition.

F. METHOD LIMITATIONS

The TR-AWR method proposed in this study represents a further exploration of the WiTA task; therefore, it still has several limitations:

(a) Recognition unit and accuracy: In this study, the minimum recognition unit was a word, rather than a complete sentence. If this method is to be applied in practical environments, further adjustments must be made to ensure coherence between words. In addition, the accuracy of the current method still requires improvement.

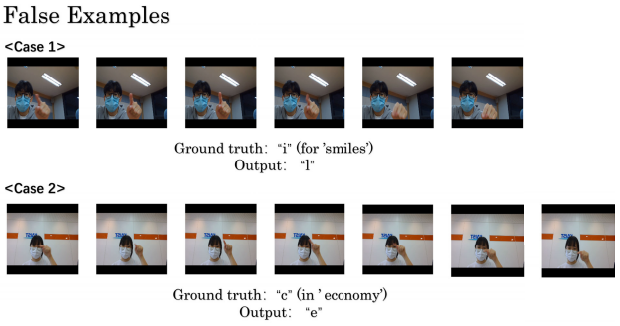


FIGURE 11. False case of air-writing recognition by two examples. The pictures on the upper row are the sequence of images corresponding to a particular letter in the action of drawing a word. The text on the middle row shows the actual correct character and the character output by the proposed method on the lower row, which is false.

(b) Real-time recognition: In this task, we did not extend the method to real-time air writing; instead, we ensured a certain level of real-time capability by calculating the decoding frames per second (D-fps). Transformer models, as they typically require global attention computations over the entire input sequence, may introduce latency when processing longer input sequences. This may result in a delay in displaying recognition results within certain time windows during the user's writing, thus reducing the user's interactive experience.

(c) Computational complexity: The Transformer model may require more computational resources and memory when processing longer input sequences [51]. Particularly in the case of continuous writing processes, if the real-time execution of large-scale transformer models is necessary, it may impose increased demands on computing devices. This can pose limitations in embedded devices or resource-constrained environments.

V. CONCLUDING REMARKS

A. CONCLUSION

This study presents a novel end-to-end air-writing recognition method based on the transformer model, that effectively addresses the WiTA task, which entails continuous air-writing recognition. By leveraging the transformer model, we construct an end-to-end Air-Writing model that successfully transforms sequences of Air-Writing video frames into corresponding character sequences. Our model employs an attention mechanism within an encoder-decoder framework, allowing it to automatically recognize handwritten characters by learning the mapping relationship between frame sequences and characters. Evaluation of the model's performance is based on the character error rate (CER), while ensuring specific real-time performance by calculating D-fps. This study achieved a CER value of 29.86%, currently standing as the best-reported data. Furthermore, the algorithm demonstrates a D-fps result of 186.75 fps, guaranteeing its real-time performance. Comparative experiments with similar end-to-end methods confirm that our model achieves

superior CER results while maintaining a specific level of real-time performance, thereby enhancing overall accuracy. The contributions of this article lie in two significant contributions. Firstly, we introduce the Transformer model to the WiTA task, surpassing the performance of the baseline method. Secondly, we pioneer the use of data augmentation techniques specifically tailored for the WiTA task, along with the proposal of criteria for effective data augmentation. Collectively, our study provides valuable insights and guidance for other researchers aiming to address the WiTA task.

B. FUTURE WORK

The Transformer model is a concise and efficient deep learning model, and the size of the dataset is an important factor affecting its performance. In future work, the most important aspect will be to establish a larger and more diverse dataset in multiple dimensions. We focus on improving the model's performance in handling different writing styles, languages, and environments to enhance its practicality [52]. For example, we will collect more sample data for different age groups, education levels, and professions and analyze and pre-process the data to improve the model's performance. In addition, we will attempt to apply this model to other fields such as numerals, graphics, and Chinese characters, thereby expanding its application scope. With further development of computer vision methods, more advanced computer vision methods should also be applied to this problem to further improve the model's accuracy and real-time performance.

REFERENCES

[1] B. R. Gaines and M. L. G. Shaw, "From timesharing to the sixth generation: The development of human–computer interaction. Part I," *Int. J. Man-Mach. Stud.*, vol. 24, no. 1, pp. 1–27, Jan. 1986.

[2] M. B. Rosson and J. M. Carroll, *Usability Engineering: Scenario-Based Development of Human Computer Interaction*. San Francisco, CA, USA: Morgan Kaufmann, 2009.

[3] J. Katona, "A review of human–computer interaction and virtual reality research fields in cognitive InfoCommunications," *Appl. Sci.*, vol. 11, no. 6, p. 2646, Mar. 2021.

[4] F. Gurcan, N. E. Cagiltay, and K. Cagiltay, "Mapping human–computer interaction research themes and trends from its existence to today: A topic modeling-based review of past 60 years," *Int. J. Hum.–Comput. Interact.*, vol. 37, no. 3, pp. 267–280, Feb. 2021.

[5] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognition—Part I: Modeling and recognition of characters, words, and connecting motions," *IEEE Trans. Hum.-Mach. Syst.*, vol. 46, no. 3, pp. 403–413, Jun. 2016.

[6] W. Roldan, X. Gao, A. M. Hishikawa, T. Ku, Z. Li, E. Zhang, J. E. Froehlich, and J. Yip, "Opportunities and challenges in involving users in project-based HCI education," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–15, doi: 10.1145/3313831.3376530.

[7] Y. Song, D. Demirdjian, and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction," *ACM Trans. Interact. Intell. Syst.*, vol. 2, no. 1, pp. 1–28, Mar. 2012.

[8] D. D. Mohan, B. Jawade, S. Setlur, and V. Govindaraju, "Deep metric learning for computer vision: A brief overview," in *Handbook of Statistics*. Buffalo, NY, USA: Univ. Buffalo, Feb. 2018, pp. 59–79.

[9] U.-H. Kim, Y. Hwang, S.-K. Lee, and J.-H. Kim, "Writing in the air: Unconstrained text recognition from finger movement using spatio-temporal convolution," *IEEE Trans. Artif. Intell.*, early access, Oct. 11, 2022, doi: 10.1109/TAI.2022.3212981.

[10] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognit.*, vol. 70, pp. 163–176, Oct. 2017, doi: 10.1016/j.patcog.2017.05.012.

[11] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 1135–1139, doi: 10.1109/icdar.2011.229.

[12] S. Xu, Y. Xue, X. Zhang, and L. Jin, "A novel unsupervised domain adaptation method for inertia-trajectory translation of in-air handwriting," *Pattern Recognit.*, vol. 116, Aug. 2021, Art. no. 107939.

[13] S. Agrawal, I. Constandache, S. Gaonkar, R. R. Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services*, Jun. 2011, pp. 15–28, doi: 10.1145/1999995.1999998.

[14] A. Tripathi, A. P. Prathosh, S. P. Muthukrishnan, and L. Kumar, "SurfMyoAiR: A surface electromyography-based framework for air-writing recognition," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[15] Y. Fang, Y. Xu, H. Li, X. He, and L. Kang, "Writing in the air: Recognize letters using deep learning through WiFi signals," in *Proc. 6th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Jul. 2020, pp. 8–14.

[16] C. Amma, M. Georgi, and T. Schultz, "Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3D-space handwriting with inertial sensors," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 52–59.

[17] S. Duffner, S. Berlemont, G. Lefebvre, and C. Garcia, "3D gesture classification with convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 5432–5436.

[18] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6197–6205, Dec. 2011.

[19] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognition—Part II: Detection and recognition of writing activity in continuous stream of motion data," *IEEE Trans. Hum.-Mach. Syst.*, vol. 46, no. 3, pp. 436–444, Jun. 2016, doi: 10.1109/THMS.2015.2492599.

[20] S. Nabeshima, S. Yamamoto, K. Agusa, and T. Taguchi, "MEMO-PEN: A new input device," in *Proc. Conf. Companion Hum. Factors Comput. Syst.*, 1995, pp. 256–257.

[21] N. Xu, W. Wang, and X. Qu, "Recognition of in-air handwritten Chinese character based on leap motion controller," in *Proc. 8th Int. Conf., Image Graph.*, in Lecture Notes in Computer Science, Jan. 2015, pp. 160–168.

[22] J. Gan and W. Wang, "In-air handwritten English word recognition using attention recurrent translator," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 3155–3172, Jul. 2019.

[23] S. Mohammadi and R. Maleki, "Real-time Kinect-based air-writing system with a novel analytical classifier," *Int. J. Document Anal. Recognit.*, vol. 22, no. 2, pp. 113–125, Jun. 2019.

[24] C.-C. Lee, C.-Y. Shih, and B.-S. Jeng, "Fingertip-writing alphanumeric character recognition for vision-based human computer interaction," in *Proc. Int. Conf. Broadband, Wireless Comput., Commun. Appl.*, Nov. 2010, pp. 533–537.

[25] M. S. Alam, K.-C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, and N. Kim, "Trajectory-based air-writing recognition using deep neural network and depth sensor," *Sensors*, vol. 20, no. 2, p. 376, Jan. 2020.

[26] A. Rahman, P. Roy, and U. Pal, "Air writing: Recognizing multi-digit numeral string traced in air using RNN-LSTM architecture," *Social Netw. Comput. Sci.*, vol. 2, no. 1, pp. 1–13, Feb. 2021.

[27] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[28] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 6382–6388, doi: 10.18653/v1/d19-1670.

[29] N. Cauli and D. Reforgiato Recupero, "Survey on videos data augmentation for deep learning models," *Future Internet*, vol. 14, no. 3, p. 93, Mar. 2022, doi: 10.3390/fi14030093.

[30] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," in *Proc. Int. Conf. Image Process.*, 2002, pp. 1–4, doi: 10.1109/icip.2002.1040019.

[31] H.-T. Duong and V. T. Hoang, "Data augmentation based on color features for limited training texture classification," in *Proc. 4th Int. Conf. Inf. Technol. (InCIT)*, Oct. 2019, pp. 208–211, doi: 10.1109/incit.2019.8911934.

[32] R. Gontijo Lopes, D. Yin, B. Poole, J. Gilmer, and E. D. Cubuk, "Improving robustness without sacrificing accuracy with patch Gaussian augmentation," 2019, *arXiv:1906.02611*.

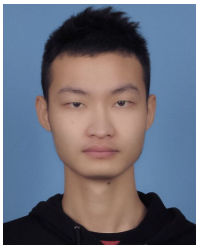
[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[34] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 317–326, doi: 10.1109/CVPR.2016.41.
[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.
[36] W. Wang and Z. Tu, "Rethinking the value of transformer components," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 6019–6029, doi: 10.18653/v1/2020.coling-main.529.
[37] C. Coleman, D. Narayanan, D. Kang, and T. Zhao. *DawnBench: An End-to-End Deep Learning Benchmark and Competition*. Accessed: Jun. 10, 2023. [Online]. Available: https://dawn.cs.stanford.edu/benchmark/papers/nips17-dawnbench.pdf
[38] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1.
[39] D. C. Bui, D. Truong, N. D. Vo, and K. Nguyen, "MC-OCR challenge 2021: Deep learning approach for Vietnamese receipts OCR," in *Proc. RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, Aug. 2021, pp. 1–6.
[40] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clapés, "Video transformers: A survey," 2022, *arXiv:2201.05991*.
[41] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 464–468, doi: 10.18653/v1/n18-2074.
[42] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, p. 29.
[43] C. Liang, M. Xu, and X.-L. Zhang, "Transformer-based end-to-end speech recognition with residual Gaussian-based self-attention," in *Proc. Interspeech*, Aug. 2021, pp. 2072–2076, doi: 10.21437/interspeech.2021-427.
[44] M. India, P. Safari, and J. Hernando, "Self multi-head attention for speaker recognition," in *Proc. Interspeech*, Sep. 2019, pp. 4305–4309, doi: 10.21437/INTERSPEECH.2019-2616.
[45] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder–decoder networks," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1875–1886, Nov. 2015.
[46] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.
[47] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," in *Proc. Interspeech*, 2014, pp. 2635–2639, doi: 10.21437/interspeech.2014-564.
[48] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1091–1095, Jun. 2007, doi: 10.1109/TPAMI.2007.1078.
[49] D. Li, C. R. Opazo, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1448–1458, doi: 10.1109/wacv45572.2020.9093512.
[50] R. Haldar and D. Mukhopadhyay, "Levenshtein distance technique in dictionary lookup methods: An improved approach," 2011, *arXiv:1101.1232*.
[51] Z. Zhang, Z. Gong, and Q. Hong, "A survey on: Application of transformer in computer vision," in *Proc. 8th Int. Conf. Intell. Syst. Image Process.*, 2021, pp. 21–28, doi: 10.12792/icisip2021.006.
[52] V. Carbune, P. Gonnet, T. Deselaers, H. A. Rowley, A. Daryin, M. Calvo, L.-L. Wang, D. Keysers, S. Feuz, and P. Gervais, "Fast multi-language LSTM-based online handwriting recognition," *Int. J. Document Anal. Recognit.*, vol. 23, no. 2, pp. 89–102, Jun. 2020, doi: 10.1007/s10032-020-00350-4.

XUHANG TAN was born in Heilongjiang, China, in 2000. He received the bachelor's degree, in 2021. He is currently pursuing the master's degree in information, production, and systems engineering with Waseda University, Fukuoka, Japan. His main research interests include computer vision, deep learning, and human–computer interaction.

JICHENG TONG was born in Inner Mongolia, China, in 1998. He received the bachelor's degree, in 2021. He is currently pursuing the master's degree in information, production, and systems engineering with Waseda University, Fukuoka, Japan. His main research interests include data mining, deep learning, and human–computer interaction.

TAKAFUMI MATSUMARU (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Waseda University, Tokyo, Japan, in 1985, 1987, and 1998, respectively. He was with Toshiba Corporation, Kawasaki, Japan, and Shizuoka University, Hamamatsu, Japan. He is currently a Professor with Waseda University anf the Head of the Bio-Robotics and Human-Mechatronics Laboratory (BRHM Laboratory) (http://www.waseda.jp/sem-matsumaru/). He was a Visiting Professor with the Warsaw University of Technology, in 2023. He is the author of "Seitai-Kinou Kougaku (Biological Function Engineering) (Tokyo Denki University Press, ISBN 9784501417505). His research interests include bio-robotics and human-mechatronics, especially in human–robot interaction.

VIBEKANANDA DUTTA (Member, IEEE) received the M.Sc. degree in computer science, with specialization in artificial intelligence from the Central University of Rajasthan, Rajasthan, India, in 2012, and the Ph.D. degree in automation and robotics from the Warsaw University of Technology, Warsaw, Poland, in 2019. His research interests include human–robot interaction, machine vision, and 3D imaging.



XIN HE (Student Member, IEEE) received the master's degree from the Graduate School of Information, Production and System, Waseda University, in 2021, where he is currently pursuing the Ph.D. degree with the Graduate School of Information, Production and System. His research interests include deformable objects manipulation, object shape modeling, point cloud data process, and human gesture recognition.