# Writing in the Air: Unconstrained Text Recognition From Finger Movement Using Spatio-Temporal Convolution

Ue-Hwan Kim, Yewon Hwang, Sun-Kyung Lee, and Jong-Hwan Kim, *Fellow, IEEE*

***Abstract*—In this article, we introduce a new benchmark dataset for the challenging writing in the air (WiTA) task—an elaborate task bridging vision and natural language processing (NLP). WiTA implements an intuitive and natural writing method with finger movement for human–computer interaction (HCI). Our WiTA dataset will facilitate the development of data-driven WiTA systems, which, thus, far have displayed unsatisfactory performance—due to lack of dataset as well as traditional statistical models they have adopted. Our dataset consists of five subdatasets in two languages (Korean and English) and amounts to 209 926 video instances from 122 participants. We capture finger movement for WiTA with red-green-blue (RGB) cameras to ensure wide accessibility and cost-efficiency. Next, we propose spatio-temporal residual network architectures inspired by 3-D ResNet. These models perform unconstrained text recognition from finger movement, guarantee a real-time operation [>100 frames per second (FPS)], and will serve as an evaluation standard.**

***Impact Statement*—Writing in The Air (WiTA) is a technology that enables a new form of HCI. As more advanced technologies integrate into human's daily lives through various ways, the need for new types of text entry systems suitable for those technologies has grown. Most text entry methods presented today, however, are not entirely inclusive for all types of users and entail their own shortcomings, which we explain further in our discussion section. WiTA recognition system which we introduce in this article overcomes previous limitations and allows fully unconstrained HCI. With an overall character error rate of 29.24% in English and the ability to process 697 FPS, our network can serve as a good starting point to further research on WiTA. WiTA provides contact-free means for humans to communicate with computers and has a great potential to be utilized in many applications, such as augmented reality (AR) and virtual reality.**

Manuscript received 24 April 2022; revised 1 August 2022 and 20 September 2022; accepted 2 October 2022. Date of publication 10 October 2022; date of current version 22 November 2023. This article was recommended for publication by Associate Editor C. L. P. Chen upon evaluation of the reviewers' comments. This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) under Grant 2020-0-00842, Development of Cloud Robot Intelligence for Continual Adaptation to User Reactions in Real Service Environments and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant NRF-2022R1C1C1009989. *(Ue-Hwan Kim and Yewon Hwang contributed equally to this work.) (Corresponding author: Jong-Hwan Kim.)*

Ue-Hwan Kim is with the AI Graduate School, Gwang-ju Institute of Science and Technology, Gwang-Ju 61005, South Korea (e-mail: uehwan@gist.ac.kr).

Yewon Hwang, Sun-Kyung Lee, and Jong-Hwan Kim are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: ywhwang@rit.kaist.ac.kr; sklee@rit.kaist.ac.kr; Johkim@rit.kaist.ac.kr).

Our dataset and the source codes are available at https://github.com/Uehwan/WiTA.

Digital Object Identifier 10.1109/TAI.2022.3212981

***Index Terms*— 3-D convolution, air-writing, human–computer interaction (HCI), spatio-temporal (ST) convolution, text-entry, writing-in-the-air (WiTA).**

## I. Introduction

As NEW types of technologies integrate into people's daily lives, the need for text-entry systems that suit modern mobile devices has emerged [1]. Among various advanced text-entry methods, writing in the air (WiTA), in which people write letters with finger movement in free space, has drawn much attention [2]. Ideal WiTA systems enable people to write text without focusing on the keyboard layout on a tiny screen and implement a natural and intuitive text-entry system while securing privacy. Applications that would benefit from WiTA by immensely improving user experience include remote signatures and intelligent system controls.

Developing feasible WiTA systems is challenging due to the interdependence among the involved gestures and lack of concrete anchors or reference positions [3]. Furthermore, understanding the correlation between various writing patterns and the corresponding characters is complicated—leading to an elaborative task bridging vision and natural language processing (NLP) in the long run. As a result, contemporary WiTA systems hardly achieve a satisfactory performance, which prevents their deployment into real-world applications. Conventional WiTA systems, in general, rely on traditional statistical models with hand-crafted features, which restricts their performance [4], [5]. Although researchers have attempted to apply data-driven approaches for designing WiTA systems, the current datasets available possess multiple limitations. For instance, the work in [2], [3], and [6] used expensive motion sensors to capture users' writing pattern, the work in [3] and [7] forced users to follow predefined unistroke writing patterns, and the work in [8] and [7] only collected videos capturing a single English lower-case letter, which are not comprehensive enough for WiTA systems. Moreover, Huang et al. [9] adopted an egocentric view that demands users to wear a motion capturing device.

To overcome the limitations mentioned above, we collect a benchmark dataset in this work; Fig. 1 shows an example data instance.[1] Among multiple modalities for capturing finger

[1]Fig. 2 displays the annotated result of Fig. 1. The tracking of the fingertip reveals the text written in the air—though the tracking is hardly possible for laypersons when viewed in real-time, ensuring a private HCI tool.

Fig. 1. Example instance of the dataset collected in this work. The person is writing "re" from the word "recognized." WiTA offers a private communication tool for HCI.

movement in the air, we choose red-green-blue (RGB) cameras as the sensing device due to their superior accessibility, monetary-wise low cost, and generality compared to other sensing modalities, such as depth or gyro sensors. In addition, we adopt a third-person view rather than an egocentric view to improve user experience by removing the possibility of attaching additional devices on users [10]. We also allow users to follow their natural handwriting patterns to maximize usability. Finally, we collect five subdatasets—to ensure universality and actualize unconstrained text recognition from finger movement—in two languages: Korean lexical, English lexical, Korean nonlexical, English nonlexical, and the mixture of the two languages in a nonlexical format. As far as we are aware, our dataset is the most comprehensive benchmark dataset for the WiTA task, and we expect our dataset would facilitate the research on WiTA.

Next, we propose baseline models for the WiTA task, which will serve as an evaluation standard for forthcoming WiTA systems. The baseline models receive a sequence of image frames and transform the input into a sequence of characters written in the air. The proposed baseline models perform the decoding process in an end-to-end manner—performing unconstrained text recognition from finger movement. For developing the baseline models, we propose spatio-temporal (ST) residual network architectures inspired by 3-D ResNet [11]. The proposed ST residual networks effectively deal with both spatial and temporal contexts within the WiTA input signals. Furthermore, we conduct a thorough ablation study to examine the effect of each design choice and offer insights for the development of more advanced WiTA systems.

Specifically, the main contributions of our work are as follows.

1) *Benchmark dataset:* We explicitly define the task of WiTA and collect five types of the dataset in two languages (Korean and English) for the development and evaluation of WiTA systems.
2) *Baseline models:* We propose ST residual network architectures that translate an input video sequence into a sequence of characters written in the air as baseline models.
3) *Ablation study:* We conduct a comprehensive ablation study by varying the training conditions in multiple ways and analyze the effect of each design choice in a thorough manner.
4) *Open source:* We contribute to the corresponding research society by making the source code of the data collection tool, the WiTA dataset, the proposed WiTA baseline networks, and the pretrained network parameters public.

The rest of this article is structured as follows. Section II reviews previous research outcomes related to WiTA. Section III illustrates the data collection process and the data statistics. Section IV describes the proposed WiTA baseline architectures. Section V delineates the evaluation setting and examines the evaluation results with corresponding analysis. Section VI discusses future research directions for further improvement of WiTA. Finally, Section VII concludes this article.

## II. Related Works

### A. Writing Recognition Systems

*1) Handwriting:* Handwriting recognition aims to interpret handwritten input from a specified source, such as pen strokes or paper documents and output the best text corresponding to the input [12]. In the general setting of handwriting recognition systems, humans grasp input devices, and write on recording devices [13], [14]. Examples of input and recording device pairs include electric pens and touch screens as well as pens and paper.

Depending on system designs, temporal information is incorporated for online recognition [12], [15]; otherwise, the systems become analogous to optical character recognition systems [16], [17], [18]. Temporal information provides recognition algorithms a richer context, and recognition algorithms without temporal context employ computer vision techniques to process visual input. Generally, convolution neural network (CNN) is used to extract visual features and recurrent neural network (RNN) is used to model temporal information. Recently, there have been attempts to make optical character recognition (OCR) models compact. For instance, Ding et al. [19] compressed CNN in an effort to reduce the footprint and runtime latency of the model, and Ding et al. [20] further compressed the model using teacher-student learning.

Another distinction classifies handwriting recognition systems into segmentation-based or segmentation-free algorithms [21], [22]. The performance of segmentation affects the performance of recognition systems, and segmentation-free systems could achieve robust performance in certain use cases [23],

TABLE I
SUMMARY OF THE PARTICIPANT STATISTICS

| Metric | Type | Value |
|---|---|---|
| **Gender** | Male | 74/122 |
| | Female | 48/122 |
| | Neutral | – |
| **Age** | Range | 19–42 |
| | Average | 24.33 |
| | s.t.d. | 2.39 |
| **Comfort-hand** | Left | 1/122 |
| | Right | 119/122 |
| | Both | 2/122 |
| **Korean Fluency** | Reading | 4.82/5.00 (0.47) |
| | Writing | 4.61/5.00 (0.43) |
| | Overall | 4.70/5.00 (0.45) |
| **English Fluency** | Reading | 4.33/5.00 (0.39) |
| | Writing | 4.16/5.00 (0.37) |
| | Overall | 3.45/5.00 (0.30) |

[24]. Recently, Yousef and Bishop [25] expanded segmentation-free single line recognition into multiline recognition. Lastly, the restriction on writing styles categorizes handwriting systems into constrained or unconstrained systems [22]. The unconstrained systems in general utilize the connectionist temporal classification setting [12], [26].

Despite their effectiveness in certain use cases, e.g., document analysis [27], handwriting recognition systems' requirements of expensive and volumetric input and recording devices limit their application to mobile computing environments as text-entry. Moreover, design choices including input sequence encoding affect the performance of handwriting recognition systems [14], [15], [28], which complicates the system design process.

*2) Finger Writing:* In finger writing recognition systems, users write text in the air with right or left index finger. Then, recognition systems capture and interpret the finger movement. For capturing finger movement, recognition systems integrate various types of sensors. One category of sensors get attached to users' body and gather the finger movement information. Examples of such sensors include smartwatches [29], [30], [31] and custom-manufactured sensors [32], [33]. This category of sensors lessens the usability since users have to carry these sensors for text-entry, and physical contacts could cause discomfort [10].

A few research groups have attempted to improve the usability of WiTA by excluding body-installed sensors. One of the approaches encodes each character or word into a set of actions and formulates WiTA as action recognition [34], [35]. Accordingly, users have to learn the new encoding systems, which in turn degrades usability. Typing in the air is another example of this approach [36]. Moreover, another group of researchers has employed Kinect (depth) [2], [5], [37] or motion sensors [3], [6], [38], [39] to avoid body-installed sensors. However, users do not always have access to these high-cost sensors due to their limited availability.

RGB cameras, which omit physical contacts, offer an easy-to-deploy and low-cost way for capturing finger movement. Contemporary approaches utilizing RGB cameras for WiTA focus on fingertip tracking to formulate WiTA as handwriting recognition [9], [10], [40] or treat WiTA as gesture recognition by performing word-based recognition of written text [41], [42]. In contrast, we propose end-to-end baseline models for the WiTA task—recognizing the text written in the air on a character level. The end-to-end architectures for unconstrained text recognition lead to simplification of the design process as well as enhancement of the performance. In addition, the proposed baselines improve usability since users do not need to slow down their writing for finger detection and tracking.

### B. Convolutions for ST Data

Some of the representative applications that utilize convolution over ST data are video action recognition [43], [44], [45], video classification [46], video super-resolution [47], and flow prediction [48]. Typically these applications utilize ST networks, which are composed of blocks of ST convolutions. In video action recognition, convolution deals with macroscopic semantics within a sequence of images. Among various convolution architectures [49], [50], 3-D ResNet and its variants have exhibited satisfactory performance in video action recognition [11], [51], [52]. Furthermore, researchers improved the performance of convolution by devising various structures and architectures: deformable kernels [53], a multiscale temporal window [54], two-path architectures [55], and attentions [56].

Recently, researchers have applied convolution to the hand gesture recognition task [57], [58], [59]. These works concentrate on recognizing a set of predefined simple hand gestures. Contrary to these works, we aim to recognize the text written in the air with ST convolution. Achieving this goal requires architectures that maintain the temporal structure of input and generate a sequence of vectors rather than a single vector for classification as in previous works. Moreover, the WiTA task involves more complex hand gestures than the simple hand gesture recognition task. We design ST convolution architectures for the WiTA baseline models that could retain temporal structure and handle complex hand gestures.

## III. WiTA DATASET

### A. Data Collection Procedure

*1) Participants:* Table I summarizes the statistics of the participants. In total, we recruited 122 participants (74 male and 48 female). The participants aged from 19 to 42 (average $=$ 24.33, std $=$ 2.39). One of the participants is left-handed, two participants are ambidextrous, and the rest are right-handed. All of them use Korean as their mother tongue, and they could read and write both Korean and English without any difficulties.

*2) Environment and Apparatus:* We collected our data in nine environments to ensure the robustness to background variations (see Fig. 3): three seminar rooms, three resting areas, one lab environment, and two outdoor areas. These open spaces result in dynamic backgrounds. Moreover, we modified the viewpoints (camera's distance, angle, and position) for different data collection processes to diversify the backgrounds in our

Fig. 2. Annotated example instance of the dataset collected in this work. We visualize the trajectory of the fingertip of the person in the example. The person is writing "re" from the word "recognized."

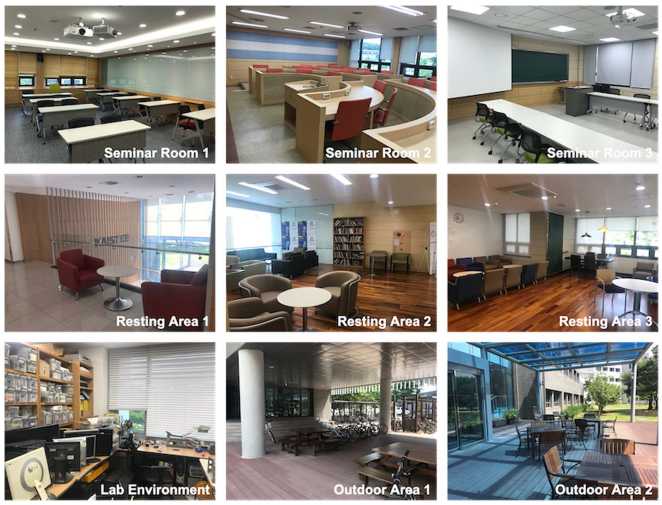


Fig. 3. Data collection environments. We varied the background for each data collection process to remove the performance dependency on the background variation.

dataset. We set up a laptop (MS Surface) equipped with an RGB camera [29 frames per second (FPS)] on a desk or a table in each data collection environment. We captured image sequences with the resolution of 224×224.

*3) Writing Interface:* We implemented the data collection interface using PyQT5,[2] which supports cross-platform application development. The beginning page of the interface collects the demographics of participants. Next, the main page of the interface displays the text to write at the top center area, and the middle area shows the current video. The right middle area contains a group of buttons for controlling the data collection process: "start," "stop," "next," and "redo."

[2]https://riverbankcomputing.com/software/pyqt/download5


Fig. 4. Examples of text for writing.

*4) Procedure:* First, we informed the participants the data collection procedure and gathered the demographics. Then, we asked the participants to assume that a perfect AI system will decode their WiTA and write as naturally as possible. As a warm-up, the participants familiarized themselves with the writing interface using the first ten phrases. Then, the participants wrote 75 phrases of lexical Korean and English texts, and 15 phrases of nonlexical Korean, English and the Mixture texts. For every data collection process, we varied the camera view to account for different angles and positions. Each participant wrote and captured 195 $(=75\times 2+15\times 3)$ phrases and the total data we collected includes 209 926 video instances. Moreover, each participant spent approximately 50 min.

*5) Text for Writing:* To verify the generality of the proposed WiTA task among multiple languages at least in a preliminary manner, we collect five subdatasets in two languages. We randomly sample a word at every data collection process—resulting in very few numbers of duplicated text—and compose the text for each dataset as follows (Fig. 4 shows example texts):

1) *Korean*[3] *lexical:* We utilize the dataset[4] collected by the National Institute of Korean Language (NIKL), the most frequent 6000 Korean words dataset.
2) *Korean nonlexical:* We randomly generate nonlexical words by sampling from the most common 1989 syllables

[3]A Hangul (Korean syllable), which is the basic building block of Korean words, consists of two to the following three letters: first letter, middle letter, and optional last letter. Consonants can be placed at the first and last letter positions, while vowels at the middle letter position. For example, the Hangul '대' consists of two letters ('ㄷ' and 'ㅐ') while '한' of three letters ('ㅎ,' 'ㅏ,' and 'ㄴ').

[4]https://www.korean.go.kr/front/reportData/reportDataView.do?mn_id=45&report_seq=1

TABLE II
COMPARISON OF DATASETS

| Dataset | People | Videos | Frames | Text | C/V | Sem | Sensor | View | Environment | Access |
|---|---|---|---|---|---|---|---|---|---|---|
| [61] | 69 | 1 794 | – | E | 1 | – | RGB | ego | Indoor | – |
| [8] | 21 | 1 290 | – | E | 1 | – | RGB | Third | Indoor | – |
| [2] | – | 375 | 44 522 | ECN | 1 | – | RGB-D | Third | – | – |
| [3] | 22 | 11 120 | – | E | $\leq$3 | ✓ | Motion | – | Indoor | – |
| [9] | 24 | – | 93 729 | EC | 1 | – | RGB-D | ego | Indoor+Outdoor | – |
| [7] | 5 | 26 000 | – | E | 1 | – | WiFi | – | Indoor | – |
| [10] | 5 | 1 800 | – | EN | 1 | – | RGB | Third | – | – |
| **WiTA** (ours) | 122 | 209 926 | 1 757 307 | KE | $\geq$3 | ✓ | RGB | 3rd | Indoor+Outdoor | ✓ |

Ours is the most comprehensive and provides rich types of data. Our dataset supplies videos containing semantic text written in the air, which capture the interdependence between gestures for different characters. C/V, Sem, K, E, C, and N stand for character/video, inclusion of semantic words, Korean, English, Chinese, and numbers, respectively.

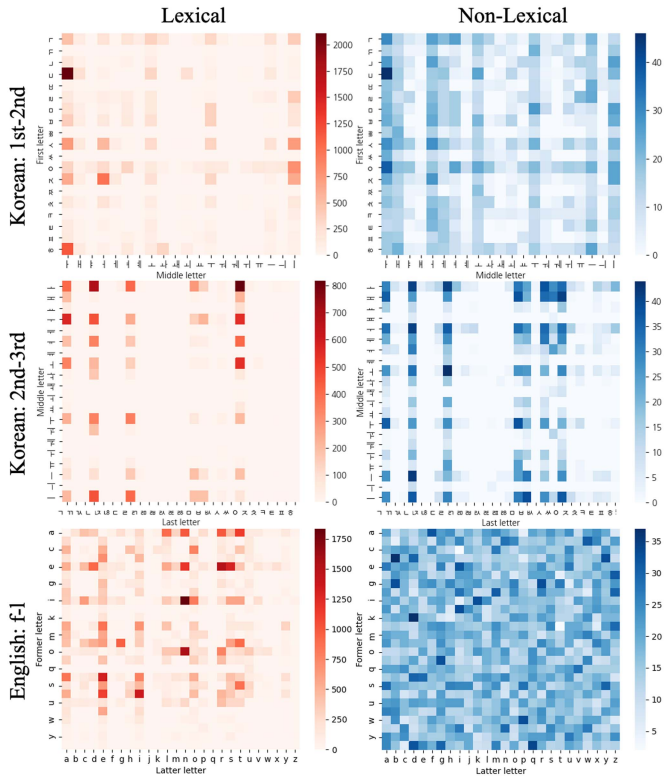


Fig. 5. Co-occurrence statistics of our WiTA dataset.

(Hangul) dataset.[5] We restrict the lengths of the generated words to range from one to three.

3) *English lexical:* We retrieve the top 6000 most-frequent words from the Google 1B dataset [60].
4) *English nonlexical:* We randomly generated nonlexical words by sampling from 26 alphabets. The lengths of the nonlexical words are between 3 and 7.
5) *Mixture nonlexical:* For testing multilingual WiTA systems, we generate nonlexical words using both Korean and English syllables.[6]

## B. Data Statistics and User Behavior Analysis

*1) Data Statistics:* Table II summarizes the statistics of the WiTA dataset collected in this work and compares it with those of previous studies. In respect of dimension, our dataset is the most comprehensive. Moreover, our dataset covers both Korean and English in addition to lexical and nonlexical phrases, while other datasets simply provide single-letter or less-than-three-letter videos. Since our dataset supplies videos containing semantic words, they capture the complex interdependence between gestures for different characters (C/V$\geq$3); it would foster the development of WiTA systems for real-world applications. Furthermore, our dataset is the only dataset accessible to the public at this time.

Fig. 5 visualizes the co-occurrence statistics.[7] The lexical data is more biased than the nonlexical data in both languages. Especially, the nonlexical English shows a well-scattered distribution. In the case of Korean, the nonlexical data is more biased than that of English since only about 2000 pairs out of 11 172 possible Hanguls are practically used—though the nonlexical Korean data shows more even distribution than that of the lexical Korean data. Thus, the nonlexical data would play a vital role in the development of unconstrained text recognition from finger movement.

Table III summarizes video and text statistics. We utilized the number of Hanguls to measure the number of characters for Korean WiTA and the number of characters for English WiTA. Since a Hangul consists of two to three letters, the number of characters for English WiTA is approximately 2.5 times larger than that of Korean WiTA. Furthermore, the participants spent slightly longer time to write Korean in the air than English. We presume that the temporal difference occurred since Korean

[5]In theory, 11 172 distinct Korean syllables (Hanguls) exist, but about 2000 of them are practically used [28]. NIKL provides this dataset as well.

[6]We expect we could verify the performance of a unified WiTA model for multiple languages with this dataset in the future.

[7]As a Hangul consists of two to three letters, we analyzed the co-occurrence between the first and the second letters, and between the second and the third letters. For English, we analyzed the co-occurrence between the former and the latter letters of every pair.

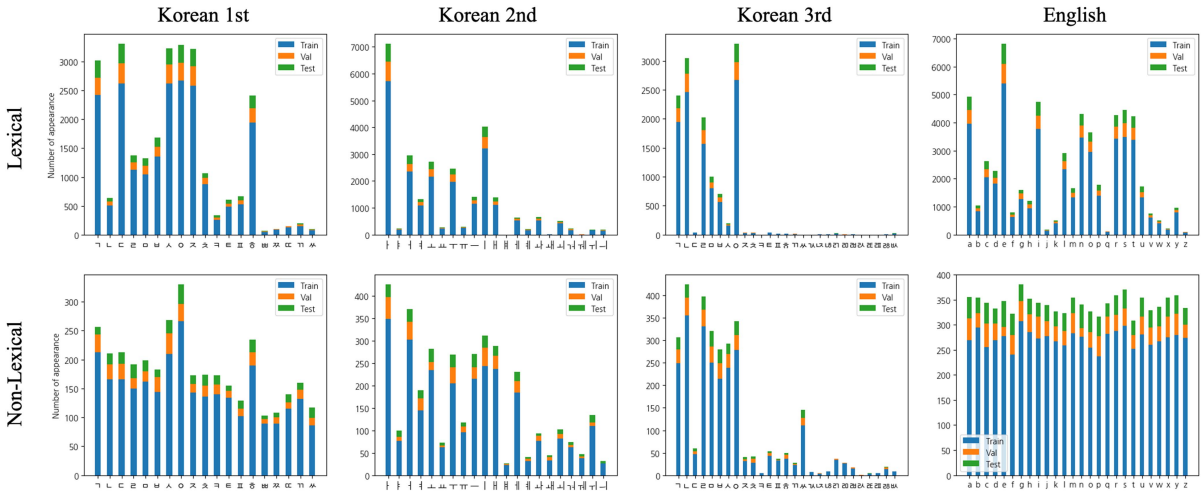


Fig. 6. The histogram of each character by dataset split. The nonlexical datasets display more even distribution than the lexical datasets in both languages.

TABLE III
SUMMARY OF VIDEO AND TEXT STATISTICS

| Language | Type | #Frames | #Characters |
|---|---|---|---|
| **Korean** | Lexical | 87.82/32.72 | 3.05/1.08 |
| | N-Lexical | 79.36/31.02 | 2.00/0.82 |
| **English** | Lexical | 78.75/28.65 | 6.59/2.54 |
| | N-Lexical | 68.08/21.49 | 5.03/1.41 |

The numbers in each cell indicate (average/std).

TABLE IV
SUMMARY OF USER BEHAVIOR ANALYSIS

| Language | Metric | Avg. | Std. | Range |
|---|---|---|---|---|
| **Korean** | HPS | 3.98 | 1.06 | $(2.11, 7.11)$ |
| | $x$-Scale | 43.56 | 12.58 | $(21.96, 99.78)$ |
| | $y$-Scale | 38.26 | 18.45 | $(11.47, 147.22)$ |
| **English** | CPS | 3.57 | 0.86 | $(1.82, 5.74)$ |
| | $x$-Scale | 18.35 | 7.27 | $(7.32, 52.78)$ |
| | $y$-Scale | 14.39 | 8.64 | $(3.73, 57.46)$ |

The unit of the metrics in the table is pixel. HPS and CPS stand for *Hangul-per-second* and *character-per-second*, respectively.

requires an additional movement of up-and-down for a set of second letters and last (third) letters of Hanguls.

Fig. 6 shows the histogram of characters in each dataset split. The lexical datasets are biased toward certain characters. For example, in the English data, the most frequent character (i.e., "e") appeared approximately 70 times more than the least-frequent character (i.e., "z"). On the other hand, the English nonlexical data shows a well-balanced data distribution within each dataset as well as across train, validation, and test datasets. Combining all the characters in each dataset, every character appears within 300 to 400 times, and the appearance of the most-frequent character was approximately only 10% more than that of the least appeared character. Likewise, the Korean nonlexical data are more fairly distributed compared to the Korean lexical data. In particular, the first Korean nonlexical characters are well spread out, while the lexical bar graph shows a drastic difference between the most-frequent character and the least-frequent character. Although following the general distribution of the lexical data, the second and the third Korean nonlexical data are relatively more spread out. The drastic difference in the number of appearances of more-likely-to-appear characters and less-likely-to-appear characters in Korean nonlexical data is inevitable because less appearing characters are simply not used frequently in the Korean language in general.

*2) User Behavior Analysis:* For the analysis of user behaviors in WiTA, we selected 12 participants for each language and analyzed the data by manually labeling (hand-annotating) the fingertips since off-the-shelf fingertip detectors [62], [63] failed to recognize fingertips due to fast finger movement. Fig. 7 exemplifies a set of WiTA patterns. In both languages, users tend to squeeze characters to fit the whole word within the screen though not consistent for all cases. Moreover, most of the patterns are challenging and complicated since we asked users to write given text freely and naturally.

Next, Table IV displays the quantitative analysis result. The participants wrote the Korean text faster than the English text and revealed a larger deviation in the case of Korean. We consider the difference in writing speed could have resulted from the fact that the participants were more familiar with Korean than English. Next, the scales of the two languages show a very distinctive difference. We utilized the number of Hanguls for measuring the scale of Korean WiTA while the number of characters for English WiTA. Since a Korean Hangul consists of two or three letters, the Korean scale is approximately 2.5 times larger than that of English.

## IV. METHODOLOGY

### A. Problem Formulation

We formulate the WiTA decoding for unconstrained text recognition as follows. Given a sequence of image frames that capture user's WiTA $\mathcal{I} = (\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n)$ where $\boldsymbol{I}_i$ $(1 \leq i \leq n)$ is


Fig. 7. Examples of WiTA patterns. Users' natural writing patterns are complex and challenging. Korean gets written in the order of left-to-right, top-to-bottom and first-to-middle-to-last-letters.

an image frame, a WiTA decoding algorithm aims to find the labeling $l^*$ with the highest conditional probability

$$l^* = \arg\max_{l} p(l|\mathcal{I}). \tag{1}$$

For the labeling, we adopt the concept of connectionist temporal classification (CTC) [26] where there is a mapping between a labeling and paths denoted as $\pi$s. An operator $\mathcal{B}$ maps a set of paths onto a labeling, i.e., multiple label sequence paths reduce to the same labeling by $\mathcal{B}$. For instance, $\mathcal{B}(a,-,a,a,b) = \mathcal{B}(-,a,-,a,-,b,-) = (a,a,b)$, where—indicates a blank. Thus, the conditional probability can be evaluated as follows:

$$p(l|\pi) = \sum_{\pi \in \mathcal{B}^{-1}(l)} p(\pi|\mathcal{I}) \tag{2}$$

where

$$p(\pi|\mathcal{I}) = \prod_{t=1}^{T} p(\pi_t, t|\mathcal{I}) = \prod_{t=1}^{T} y_{\pi_t}^t \tag{3}$$

where $\pi_t$ is the label observed at time $t$ along path $\pi$ and $y_{\pi_t}^t$ is the softmax-normalized output.

In practice

$$p(l|\pi) = \sum_{s}^{|l'|} \alpha_s^t \beta_s^t \tag{4}$$

where $l'$ is a modified labeling for which blanks get added at the beginning and the end of $l$ as well as between every pair of consecutive labels, $\alpha_s^t$ and $\beta_s^t$ are forward and backward variables defined for searching paths, and $s$ indicates steps.

Finally, given pairs of input $\mathcal{I}$ and target label $\boldsymbol{z}$ in a training set $S$, the objective loss function becomes

$$L_{\text{ctc}} = -\sum_{(\mathcal{I},\boldsymbol{z}) \in S} \ln p(\boldsymbol{z}|\mathcal{I}). \tag{5}$$

The loss function accomplishes maximum likelihood training, which simultaneously maximizes the log probabilities of all the correct labeling classifications in the training set.

### B. Text Encoding

We encode text into a sequence of separate letters. Moreover, we employ a special character "∼" to distinguish consecutive Hanguls for Korean and two identical characters that appear adjacent to each other for English. For example, "대한" and "success" becomes (ㄷ, ㅐ, ∼, ㅎ, ㅏ, ㄴ) and (s, u, c, ∼, c, e, s, ∼, s), respectively.

### C. ST Residual Network

We propose ST residual network architectures (see Fig. 8) inspired by convolutional residual blocks without bottlenecks [11], [64] as our baseline model. The proposed architectures preserve the temporal structure of input, which is crucial for unconstrained text recognition; the architectures can handle input sequences of arbitrary lengths since the CTC loss evaluates the training loss from sequences of arbitrary lengths. Each convolutional residual block consists of two convolution layers followed by an rectified linear unit (ReLU) nonlinearity [65]. The output of the $i$th residual block becomes

$$\boldsymbol{x_i} = \boldsymbol{x_{i-1}} + \mathcal{F}(\boldsymbol{x_{i-1}}; \theta_i) \tag{6}$$

where $\boldsymbol{x_i}$ denotes the tensor computed by the $i$th convolutional block and $\mathcal{F}(;\theta_i)$ implements the composition of two convolutions with the parameters $\theta_i$ and the application of the ReLU nonlinearity.We consider four types of convolution blocks to design the proposed ST residual network architectures (see Table V): 3D–2D mixed convolutions (ST-MC), reversed MC (ST-rMC), residual 3-D convolutions (ST-R3D) and 2-D convolutions followed by 1-D convolutions (ST-R(2+1)D).

We place a 3-D pooling layer in the middle of the ST-MC and ST-rMC networks to better capture both spatial and temporal contexts. In the cases of ST-R3D and ST-R(2+1)D, we omit the 3-D pooling layer since a sufficient amount of temporal contexts are captured via a number of ST convolutions. Next, we employ an adaptive spatial pooling layer [66] at the end of each ST residual network. The spatial pooling layer preserves the temporal structure of the input tensor which gets transformed into a sequence of characters. While ST-MC, ST-rMC, and ST-R3D contain a pair of convolutions in each convolution block, the ST-R(2+1)D architecture includes two pairs of convolutions in

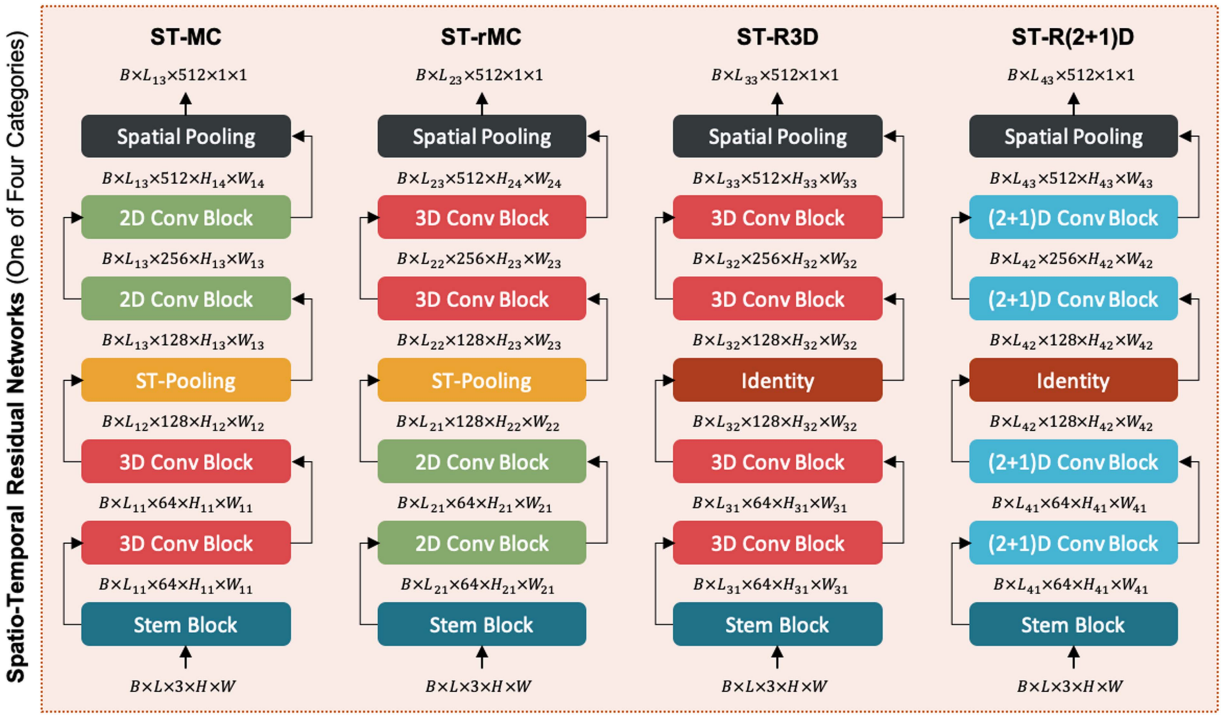


Fig. 8. Overall architecture of the WiTA baseline models. We design four ST residual network architectures and each model conducts unconstrained text recognition. ST-Pooling, $B$, $L$, $H$, and $W$ stand for ST pooling, batch size, length, height, and width, respectively.

TABLE V
ST RESIDUAL NETWORK ARCHITECTURES

| Layer Name | ST-MC | ST-rMC | ST-R3D | ST-R(2+1)D |
|---|---|---|---|---|
| Stem Block | $3 \times 7 \times 7$, stride $1 \times 2 \times 2$ | | | $1 \times 7 \times 7$, stride $1 \times 2 \times 2$, $3 \times 1 \times 1$, stride $1 \times 1 \times 1$ |
| Conv Block 1 | $\begin{bmatrix} 3 \times 3 \times 3, & 64 \\ 3 \times 3 \times 3, & 64 \end{bmatrix} \times n$ | $\begin{bmatrix} 1 \times 3 \times 3, & 64 \\ 1 \times 3 \times 3, & 64 \end{bmatrix} \times n$ | $\begin{bmatrix} 3 \times 3 \times 3, & 64 \\ 3 \times 3 \times 3, & 64 \end{bmatrix} \times n$ | $\begin{bmatrix} 1 \times 3 \times 3, 144 \\ 3 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 144 \\ 3 \times 1 \times 1, 64 \end{bmatrix} \times n$ |
| Conv Block 2 | $\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times n$ | $\begin{bmatrix} 1 \times 3 \times 3, 128 \\ 1 \times 3 \times 3, 128 \end{bmatrix} \times n$ | $\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times n$ | $\begin{bmatrix} 1 \times 3 \times 3, 230 \\ 3 \times 1 \times 1, 128 \\ 1 \times 3 \times 3, 230 \\ 3 \times 1 \times 1, 128 \end{bmatrix} \times n$ |
| Pooling (Middle) | Spatio-temporal pooling (maximum or average) | | - | - |
| Conv Block 3 | $\begin{bmatrix} 1 \times 3 \times 3, 256 \\ 1 \times 3 \times 3, 256 \end{bmatrix} \times n$ | $\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times n$ | $\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times n$ | $\begin{bmatrix} 1 \times 3 \times 3, 460 \\ 3 \times 1 \times 1, 256 \\ 1 \times 3 \times 3, 460 \\ 3 \times 1 \times 1, 256 \end{bmatrix} \times n$ |
| Conv Block 4 | $\begin{bmatrix} 1 \times 3 \times 3, 512 \\ 1 \times 3 \times 3, 512 \end{bmatrix} \times n$ | $\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times n$ | $\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times n$ | $\begin{bmatrix} 1 \times 3 \times 3, 921 \\ 3 \times 1 \times 1, 512 \\ 1 \times 3 \times 3, 921 \\ 3 \times 1 \times 1, 512 \end{bmatrix} \times n$ |
| Pooling (Last) | Global adaptive spatial pooling (maximum or average) | | | |
| Fully connected 1 | $512 \times 256$ fully connections | | | |
| Fully connected 2 | $256 \times$number of classes fully connections | | | |

$n$ is 1 for the 10-layered models and 2 for the 18-layered models.

each convolution block. Except for the ST-R3D architecture, all other architectures entail 2-D convolutions. The proposed ST residual networks offer a way to scale-up or scale-down the model depths. For ten-layered models, $n$ in the table is 1 while $n$ is 2 for 18-layered models. Although models with more than 18 layers are possible, it is highly probable that such models would hit the hardware memory limit during the training procedure.

## V. Experiments

### A. Settings

*1) Data Split:* For training, validation, and testing, we split the collected dataset into three sets with an approximate ratio of 8:1:1. We divide the data by person to ensure robustness of the developed model toward different individuals.

*2) Metrics:* We evaluate performance with two metrics: average decoding FPS and character error rate (CER). On the one hand, we include FPS as a performance metric since ensuring a real-time operation is crucial for decoders. We measure FPS by averaging the total number of frames decoded in a second. On the other hand, CER represents the decoding accuracy which is defined as

$$\text{CER} = \frac{\text{MCD}(S, P)}{\text{length}_c(P)} \times 100\,(\%) \tag{7}$$

where $\text{MCD}(S, P)$ is the minimum character distance (the Levenshtein measure) between the decoded phrase $S$ and the ground-truth phrase $P$, and $\text{length}_c(P)$ is the number of characters in $P$. The Levenshtein measure counts the number of insertions, deletions, and substitutions of characters or words to transform $S$ into $P$.

### B. Implementation Details

We trained the WiTA models using NVIDIA RTX2080Ti (14.2 teraFLOPs) and Intel i9-10980 XE with the learning rate warm-up scheme [67] and the Adam optimizer [68] after resizing images to $112\times112$. We set the learning rate as $1e$-3. We set the batch size as 4 for 18-layered models, 8 for 10-layered models, and 1 for measuring FPS. For model selection and stopping condition of training procedures, we followed the early stopping scheme [69]. All models converged within 175 epochs of training.

To investigate the effect of each design choice, we trained WiTA models using different schemes. We controlled the following conditions: the number of layers (10 or 18), the type of pooling layers (max-pooling [70] or average-pooling [54]), data augmentation (random rotation[8] and photometric distortions including brightness, contrast, saturation and hue), the loading of pretrained weights (trained on the Kinetics-400 dataset) for 18-layer models and the composition of training data.

[8]Random rotation accounts for shaky environments.

### C. Results and Analysis

*1) Search of Optimal Learning Configuration:* In order to identify the best learning configuration, we fixed the architecture as ST-R3D and varied the learning conditions. Most of the better performing configurations, including the best one, came from the 10-layered models for English, while the best configuration for Korean came from the 18-layered models (see Table VI). We assume the reason Korean requires a deeper model is due to higher complexity in writing. For English, the performance improved with augmentation and the pretrained weights with a few exceptions (the 18-layered models with max-pooling). For Korean, pretrained weights and augmentation had a different effect on the model performance; generally, the pretrained weights boosted the performance, while augmentation did not. We presume this phenomenon occurred since some Hanguls have similarities in shape, causing ambiguity and confusion when rotated. Moreover, it is likely that the last letter was mistakenly considered as the first letter since the first and the last letters of Hangul are consonants. There were some exceptions to this pattern: augmentation along with max-pooling and without the pretrained weights enhances the performance. Ultimately, the best configurations for Korean and English mismatched. This suggests that it is important to carefully select the design choices based on the characteristics of the language.

*2) Effect of Model Architecture:* Table VII investigates the architectural effects using the best learning configurations (the Korean ST-R(2+1)D ran into the out-of-memory error due to the long lengths of Korean sequences and ST-MC consistently failed to converge in the English dataset). For Korean, we used the pretrained weights, 18-layers, and max-pooling for all four networks, whereas for English, 10-layers, average pooling, and augmentation for all four networks. For both languages, ST-R3D displayed the lowest CER, and ST-rMC outperformed ST-MC—indicating that extracting temporal information in the later layers leads to a better performance. However, none of the network architectures using 2-D convolution could beat the performance of the ST-R3D architecture (only using 3-D convolution). This implies that capturing both temporal and spatial information simultaneously throughout the entire network is crucial for the WiTA task. In both languages, FPS ensures real-time operations: 435.27 and 697.39 for Korean and English, respectively.

*3) Impact of Training Data Configuration:* Table VIII summarizes the effect of training data configuration on the performance. It is worth noting that the total number of the lexical data is approximately five times larger than that of the nonlexical data. The experiment results indicate that the increase of data prompts performance gain, which is apparent. Next, the role of the nonlexical data was more significant for English than for Korean (group 2 and 4). The nonlexical English data is well-distributed (see Fig. 5), thus, contribute to a better generalization. On the other hand, for Korean, removing the lexical data to account for the addition of the nonlexical data led the models to get trained on less-likely-to-appear data—degrading the performance.

TABLE VI
RESULTS OF THE ABLATION STUDY FOR SEARCHING THE OPTIMAL LEARNING CONDITION ON THE VALIDATION DATASET

| Training Condition | | | | Korean (CER) | | | English (CER) | | |
|---|---|---|---|---|---|---|---|---|---|
| #Layers | Pooling | Augment | PreTr | Lexical | N-Lexical | Overall | Lexical | N-Lexical | Overall |
| 10 | Max | - | - | 49.85 | 64.24 | 51.71 | 29.77 | 42.75 | 31.47 |
| 10 | Max | ✓ | - | 44.55 | 58.66 | 46.37 | 29.07 | 43.40 | 30.95 |
| 10 | Avg | - | - | 45.34 | 62.05 | 47.50 | 27.24 | **42.21** | 29.20 |
| 10 | Avg | ✓ | - | 39.00 | 54.13 | 40.96 | **27.12** | 42.32 | **29.12** |
| 18 | Max | - | - | 67.28 | 79.08 | 68.81 | 33.10 | 49.35 | 35.24 |
| 18 | Max | ✓ | - | 29.72 | 40.35 | 31.09 | 82.03 | 87.99 | 82.81 |
| 18 | Max | - | ✓ | **28.02** | **39.79** | **29.54** | 84.93 | 90.91 | 85.72 |
| 18 | Max | ✓ | ✓ | 64.75 | 76.04 | 66.21 | 33.10 | 49.35 | 35.24 |
| 18 | Avg | - | - | 65.84 | 73.92 | 66.88 | 76.85 | 91.45 | 78.76 |
| 18 | Avg | ✓ | - | 68.94 | 77.74 | 70.07 | 41.29 | 60.71 | 43.84 |
| 18 | Avg | - | ✓ | 52.90 | 69.68 | 55.06 | 63.81 | 78.14 | 65.69 |
| 18 | Avg | ✓ | ✓ | 69.44 | 76.33 | 70.33 | 29.44 | 40.26 | 30.80 |

We controlled the following four factors: the number of layers, the type of pooling, the application of augmentation, and the usage of pretrained weights.
The bold entities indicate the best performing architecture.

TABLE VII
ARCHITECTURAL IMPACT ON THE PERFORMANCE

| Model | Korean (CER) | | | | English (CER) | | | |
|---|---|---|---|---|---|---|---|---|
| | Lexical | N-Lexical | Overall | FPS | Lexical | N-Lexical | Overall | FPS |
| **ST-MC** | 60.42 | 69.21 | 61.48 | 704.26 | - | - | - | - |
| **ST-rMC** | 54.18 | 67.47 | 55.78 | 791.28 | 92.78 | 93.96 | 92.94 | 1046.67 |
| **ST-R3D** | **31.62** | **44.37** | **33.16** | 435.27 | **28.10** | **36.46** | **29.24** | 697.39 |
| **ST-R(2+1)D** | - | - | - | - | 86.80 | 91.98 | 87.51 | 588.13 |

We measured the performance on the test dataset.
The bold entities indicate the best performing architecture.

TABLE VIII
EFFECT OF TRAINING DATA CONFIGURATION ON THE PERFORMANCE

| Training Data Configuration | | Korean (CER) | | | English (CER) | | |
|---|---|---|---|---|---|---|---|
| Lexical | N-Lexical | Lexical | N-Lexical | Overall | Lexical | N-Lexical | Overall |
| 100% | 100% | 31.62 | 44.37 | 33.16 | 28.10 | 36.46 | 29.24 |
| 100% | 50% | 34.49 | 47.60 | 36.07 | 28.20 | 40.83 | 29.93 |
| 100% | 0% | 38.77 | 54.06 | 40.61 | 32.14 | 51.98 | 34.85 |
| 90% | 50% | 59.16 | 72.43 | 60.76 | 28.43 | 40.94 | 30.14 |
| 80% | 100% | 64.66 | 74.25 | 65.82 | 30.99 | 39.90 | 32.20 |
| 50% | 100% | 41.50 | 54.14 | 43.02 | 36.71 | 42.71 | 37.53 |
| 50% | 50% | 53.03 | 64.65 | 54.43 | 47.32 | 58.23 | 48.81 |
| 50% | 0% | 53.22 | 63.58 | 54.46 | 83.06 | 91.15 | 84.16 |
| 40% | 50% | 69.42 | 79.80 | 70.67 | 62.30 | 71.46 | 63.55 |
| 30% | 100% | 70.64 | 79.14 | 71.66 | 48.79 | 48.54 | 48.75 |
| 20% | 0% | 66.07 | 74.75 | 67.11 | 88.18 | 92.40 | 88.76 |
| 0% | 100% | 79.72 | 78.39 | 79.56 | 92.95 | 94.58 | 93.17 |

Each row represents a training dataset configuration and the performance on the test dataset. The numbers below the "training data configuration" column indicate the amount of data composed for each experiment. We designed five groups of experiments and the double lines separate each group below.

## VI. DISCUSSION

### A. Why WiTA?

We address some of the limitations of the existing communication methods between human and technology and explain why WiTA is a fair alternative in a few human–computer-interaction (HCI) settings.

1) *Touchscreen keyboard* demands user's constant monitoring to make sure there is no typo in their writing due to lack of tactile feedback. It also occupies a big portion

of the screen, intervening interaction between user and device [1].
2) *Gesture-based recognition* requires users to memorize which commands are available and how to trigger them only to result in restricted communication due to a limited set of gestures [71].
3) *Speech recognition* does not guarantee privacy [72].

WiTA overcoming these restrictions would offer natural user experience in various HCI settings:

1) turning contact-based services (e.g., ATM) into contact-free services which is particularly essential in the COVID19 era;
2) providing private text-entry methods in password entering scenarios;
3) offering device-free text entry method for virtual reality applications [7];
4) enabling remote signature [2];
5) serving a fun way to teach young students how to write [2].

*1) Future Directions:* First of all, we can investigate more efficient and effective model architectures in future studies. The need for a study on model architectures that achieve higher accuracy through less computational complexity remains. We can hardly train the current baseline models with a larger batch size because of the high computational complexity. If future research results in a lighter and faster model architecture, we expect that the training efficiency will improve as well. In addition, the fast and accurate model architectures will maximize the usability of WiTA systems. This will foster the active utilization of WiTA in various fields.

Next, we can diversify the data collection settings in the following study. In the following studies, we can make WiTA performance more robust by collecting data using various devices from more diverse and dynamic environments. With the introduction of new devices, the data collection conditions, including FPS, image resolution, color space, and the background, will vary. Moreover, we would collect fingertip annotation for all data instances. Fingertip annotations can bring great value and facilitate further research using multimodalities. As these factors diversify, the WiTA system developed using these more diverse data will become more reliable.

Furthermore, we can improve accuracy by integrating the WiTA system with NLP language models (LMs). We would not be able to reduce ambiguity between some characters, no matter how much data is available. Thus, there may exist limitations in driving performance improvement with data alone. We expect that using the character LM in WiTA systems can reduce the model's apparent inaccurate prediction by employing semantic context. We can utilize LM in WiTA systems in an end-to-end manner or a modular manner.

Moreover, exploring attention-based models [73] would offer a way to enhance the performance dramatically. The attention mechanism has displayed its effectiveness over a variety of fields. At the same time, it is a challenging research topic since such models demand intensive memory usage for handling a sequence of images; the input dimension becomes $3 \times 112 \times 112$ for a sequence of images compared to that of 512 or 1024 for NLP. Thus, improving the accuracy with the attention mechanism might require thorough technical investigation.

Last but not least, we can extend the current WiTA proposed in this work to various languages. Currently, the dataset contains only Korean and English; however, the WiTA dataset can be expanded for other languages using the data collection tool disclosed in this study. In the process of supporting various languages, it is necessary to consider the unique features of the language, such as designing a specific encoding method for each language. In addition, when multiple language data is collected, a single integrated WiTA system can support multiple languages at once. Then, the WiTA system can handle various types of user inputs and become versatile.

## VII. Conclusion

In this work, we collected a benchmark dataset for WiTA systems. To the best of authors' knowledge, our benchmark dataset is the most comprehensive and the only dataset enabling real-world implementation. The dataset consists of five subdatasets in two languages including both lexical and nonlexical text to ensure universality. We captured the finger movement with RGB cameras in a third-person view from 122 participants—resulting in 209 926 videos. This data collection setting guarantees accessibility, cost-efficiency, and generality. Next, we proposed baseline models for the WiTA task. In developing the baseline models, we designed four ST residual networks inspired by 3-D ResNet. The proposed ST residual networks effectively handle both spatial and temporal contexts within the input sequence. The proposed models exhibited 33.16% and 29.24% of CER in Korean and English datasets, respectively, with the processing speed of 435 and 697 FPS securing a real-time operation. We expect that our dataset and proposed baseline models would activate the research on WiTA; we make our dataset and the source codes public.

## References

[1] U.-H. Kim, S.-M. Yoo, and J.-H. Kim, "I-keyboard: Fully imaginary keyboard on touch devices empowered by deep neural decoder," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4528–4539, Sep. 2021.
[2] X. Zhang, Z. Ye, L. Jin, Z. Feng, and S. Xu, "A new writing experience: Finger writing in the air using a kinect sensor," *IEEE MultiMedia*, vol. 20, no. 4, pp. 85–93, Oct.–Dec. 2013.
[3] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognition–Part I: Modeling and recognition of characters, words, and connecting motions," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 3, pp. 403–413, Jun. 2016.
[4] C. Amma, M. Georgi, and T. Schultz, "Airwriting: A wearable handwriting recognition system," *Pers. Ubiquitous Comput.*, vol. 18, no. 1, pp. 191–203, 2014.
[5] 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, 2019.
[6] M. Arsalan, A. Santra, K. Bierzynski, and V. Issakov, "Air-writing with sparse network of radars using spatio-temporal learning," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 8877–8884.
[7] Z. Fu, J. Xu, Z. Zhu, A. X. Liu, and X. Sun, "Writing in the air with WiFi signals for virtual reality devices," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 473–484, Feb. 2019.
[8] A. Schick, D. Morlock, C. Amma, T. Schultz, and R. Stiefelhagen, "Vision-based handwriting recognition for unrestricted text input in mid-air," in *Proc. 14th ACM Int. Conf. Multimodal Interact.*, 2012, pp. 217–220.

[9] Y. Huang, X. Liu, X. Zhang, and L. Jin, "A pointing gesture based egocentric interaction system: Dataset, approach and application," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 16–23.

[10] S. Mukherjee, S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Fingertip detection and tracking for recognition of air-writing in videos," *Expert Syst. Appl.*, vol. 136, pp. 217–229, 2019.

[11] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6450–6459.

[12] Y. Chherawala, P. P. Roy, and M. Cheriet, "Feature set evaluation for offline handwriting recognition systems: Application to the recurrent neural network model," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2825–2836, Dec. 2016.

[13] M. Sepahvand, F. Abdali-Mohammadi, and F. Mardukhi, "Evolutionary metric-learning-based recognition algorithm for online isolated Persian/Arabic characters, reconstructed using inertial pen signals," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2872–2884, Sep. 2017.

[14] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing Chinese characters with recurrent neural network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 849–862, Apr. 2018.

[15] V. Carbune et al., "Fast multi-language LSTM-based online handwriting recognition," *Int. J. Document Anal. Recognit.*, vol. 23, pp. 89–102, 2020.

[16] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 767–779, Apr. 2011.

[17] D. Tao, X. Lin, L. Jin, and X. Li, "Principal component 2-D long short-term memory for font recognition on single Chinese characters," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 756–765, Mar. 2016.

[18] I. Z. Yalniz and R. Manmatha, "Dependence models for searching text in document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 49–63, Jan. 2019.

[19] H. Ding et al., "A compact CNN-DBLSTM based character model for offline handwriting recognition with Tucker decomposition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit.*, 2017, vol. 1, pp. 507–512.

[20] H. Ding, K. Chen, and Q. Huo, "Compressing CNN-DBLSTM models for OCR with teacher-student learning and Tucker decomposition," *Pattern Recognit.*, vol. 96, 2019, Art. no. 106957.

[21] K. M. Sayre, "Machine recognition of handwritten words: A project report," *Pattern Recognit.*, vol. 5, no. 3, pp. 213–228, 1973.

[22] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[23] R. Messina and J. Louradour, "Segmentation-free handwritten Chinese text recognition with LSTM-RNN," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, 2015, pp. 171–175.

[24] Z. Xie, Z. Sun, L. Jin, H. Ni, and T. Lyons, "Learning spatial-semantic context with fully convolutional recurrent network for online handwritten Chinese text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1903–1917, Aug. 2018.

[25] M. Yousef and T. E. Bishop, "OrigamiNet: Weakly-supervised, segmentation-free, one-step, full page text recognition by learning to unfold," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14710–14719.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[27] G. Retsinas, G. Louloudis, N. Stamatopoulos, and B. Gatos, "Efficient learning-free keyword spotting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1587–1600, Jul. 2019.

[28] D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbune, "Multi-language online handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1180–1194, Jun. 2017.

[29] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proc. 16th Int. Workshop Mobile Comput. Syst. Appl.*, 2015, pp. 9–14.

[30] D. Moazen, S. A. Sajjadi, and A. Nahapetian, "AirDraw: Leveraging smart watch motion sensors for mobile human computer interactions," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf.*, 2016, pp. 442–446.

[31] Y. Yin, L. Xie, T. Gu, Y. Lu, and S. Lu, "AirContour: Building contour-based model for in-air writing gesture recognition," *ACM Trans. Sensor Netw.*, vol. 15, no. 4, pp. 1–25, 2019.

[32] L. Jing, Z. Dai, and Y. Zhou, "Wearable handwriting recognition with an inertial sensor on a finger nail," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit.*, 2017, vol. 1, pp. 1330–1337.

[33] K. Sakuma, G. Blumrosen, J. J. Rice, J. Rogers, and J. Knickerbocker, "Turning the finger into a writing tool," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2019, pp. 1239–1242.

[34] A. Markussen, M. R. Jakobsen, and K. Hornbæk, "Vulture: A mid-air word-gesture keyboard," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2014, pp. 1073–1082.

[35] H. Ding et al., "RFIPad: Enabling cost-efficient and device-free in-air handwriting using passive tags," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 447–457.

[36] X. Yi, C. Yu, M. Zhang, S. Gao, K. Sun, and Y. Shi, "ATK: Enabling ten-finger freehand typing in air based on 3D hand tracking data," in *Proc. 28th Annu. ACM Symp. User Interface Softw. Technol.*, 2015, pp. 539–548.

[37] H. J. Chang, G. Garcia-Hernando, D. Tang, and T.-K. Kim, "Spatio-temporal hough forest for efficient detection–localisation-recognition of fingerwriting in egocentric camera," *Comput. Vis. Image Understanding*, vol. 148, pp. 87–96, 2016.

[38] P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra, "Study of text segmentation and recognition using leap motion sensor," *IEEE Sensors J.*, vol. 17, no. 5, pp. 1293–1301, Mar. 2017.

[39] G. Bastas, K. Kritsis, and V. Katsouros, "Air-writing recognition using deep convolutional and recurrent neural network architectures," in *Proc. 17th Int. Conf. Front. Handwriting Recognit.*, 2020, pp. 7–12.

[40] M. Alam et al., "Trajectory-based air-writing recognition using deep neural network and depth sensor," *Sensors*, vol. 20, no. 2, 2020, Art. no. 376.

[41] J. Gan, W. Wang, and K. Lu, "A unified CNN-RNN approach for in-air handwritten english word recognition," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2018, pp. 1–6.

[42] J. Gan and W. Wang, "In-air handwritten english word recognition using attention recurrent translator," *Neural Comput. Appl.*, vol. 31, pp. 3155–3172, 2019.

[43] C. Cao, Y. Zhang, C. Zhang, and H. Lu, "Body joint guided 3-D deep convolutional descriptors for action recognition," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 1095–1108, Mar. 2018.

[44] W. Wu, D. He, T. Lin, F. Li, C. Gan, and E. Ding, "MVFNet: Multi-view fusion network for efficient video recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 4, pp. 2943–2951.

[45] T. Ahmad, L. Jin, X. Zhang, S. Lai, G. Tang, and L. Lin, "Graph convolutional neural network for human action recognition: A comprehensive survey," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 128–145, Apr. 2021.

[46] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5533–5541.

[47] S. Li, F. He, B. Du, L. Zhang, Y. Xu, and D. Tao, "Fast spatio-temporal residual network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10522–10531.

[48] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[49] Y.-H. Yoo, U.-H. Kim, and J.-H. Kim, "Convolutional recurrent reconstructive network for spatiotemporal anomaly detection in solder paste inspection," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 4688–4700, Jun. 2022.

[50] J. Xue et al., "Cascaded MultiTask 3-D fully convolutional networks for pancreas segmentation," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 2153–2165, Apr. 2021.

[51] Z. Liu et al., "TEINet: Towards an efficient architecture for video recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11669–11676.

[52] J. Zhang, Y. Han, J. Tang, Q. Hu, and J. Jiang, "Semi-supervised image-to-video adaptation for video action recognition," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 960–973, Apr. 2017.

[53] J. Weng, M. Liu, X. Jiang, and J. Yuan, "Deformable pose traversal convolution for 3D action and gesture recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 136–152.

[54] L. Wang et al., "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2740–2755, Nov. 2019.

[55] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6202–6211.

[56] T. Wang et al., "Decoupled attention network for text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 12216–12224.
[57] F. Zhan, "Hand gesture recognition with convolution neural networks," in *Proc. IEEE 20th Int. Conf. Inf. Reuse Integration Data Sci.*, 2019, pp. 295–298.
[58] H. Hu, W. Zhou, and H. Li, "Hand-model-Aware sign language recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, pp. 1558–1566.
[59] Y. Zou and L. Cheng, "A transfer learning model for gesture recognition based on the deep features extracted by CNN," *IEEE Trans. Artif. Intell.*, vol. 2, no. 5, pp. 447–458, Oct. 2021.
[60] C. Chelba et al., "One billion word benchmark for measuring progress in statistical language modeling," 2013.
[61] L. Jin, D. Yang, L.-X. Zhen, and J.-C. Huang, "A novel vision-based finger-writing character recognition system," *J. Circuits, Syst., Comput.*, vol. 16, no. 3, pp. 421–436, 2007.
[62] M. M. Alam and S. M. M. Rahman, "Detection and tracking of fingertips for geometric transformation of objects in virtual environment," in *Proc. IEEE/ACS 16th Int. Conf. Comput. Syst. Appl.*, 2019, pp. 1–8.
[63] M. M. Alam, M. T. Islam, and S. M. Rahman, "Unified learning approach for egocentric hand gesture recognition and fingertip detection," *Pattern Recognit.*, vol. 121, 2022, Art. no. 108200.
[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[65] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[66] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 11, pp. 2180–2193, Nov. 2018.
[67] P. Goyal et al., "Accurate, large minibatch SGD: Training imagenet in 1 hour," 2017. [Online]. Available: http://arxiv.org/abs/1706.02677
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[69] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 402–408.
[70] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2559–2566.
[71] W. Delamare, C. Coutrix, and L. Nigay, "Designing guiding systems for gesture-based interaction," in *Proc. 7th ACM SIGCHI Symp. Eng. Interactive Comput. Syst.*, 2015, pp. 44–53.
[72] P. Wu, P. P. Liang, J. Shi, R. Salakhutdinov, S. Watanabe, and L.-P. Morency, "Understanding the tradeoffs in client-side privacy for downstream speech tasks," *APSIPA ASC*, 2021.
[73] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6836–6846.

**Ue-Hwan Kim** received the B.S., M.S., and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020, 2015, and 2013, respectively, all in electrical engineering.

Since 2021, he has been with the AI Graduate School, GIST, Gwangju, South Korea, where he is leading the Autonomous Computing Systems Lab as an Assistant Professor. His current research interests include visual perception, service robots, intelligent transportation systems, cognitive IoT, computational memory systems, and learning algorithms.

**Yewon Hwang** received the B.S. degree in mechanical engineering from The Pennsylvania State University, State College, PA, USA, in 2018 and the M.S. degree in electrical engineering in 2021 from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, where she is currently working toward the Ph.D. degree.

Her current research interests include natural language processing, multimodal learning, and human–robot interaction.

**Sun-Kyung Lee** received the M.S. degree in electrical engineering and the B.S. degree in mechanical engineering in 2018 and 2020, respectively, from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, where he is currently working toward the Ph.D. degree.

His current research interests include system engineering, learning systems, and hand gesture recognition.

**Jong-Hwan Kim** (Fellow, IEEE) received the Ph.D. degree in electronics engineering from Seoul National University, Seoul, South Korea, in 1987.

Since 1988, he has been with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, where he is leading the Robot Intelligence Technology Laboratory as a KT Endowed Chair Professor. He is the Director for both of KoYoung-KAIST AI Joint Research Center and Machine Intelligence and Robotics Multisponsored Research and Education Platform. He has authored five books and five edited books, two journal special issues, and around 400 refereed papers in technical journals and conference proceedings. His research interests include intelligence technology, machine intelligence learning, and AI robots.