Fingertip Air-Writing with Ambient Light

Hao Liu, Hanting Ye, Xiangxie Zhang, Jie Yang, and Qing Wang(✉)

Delft University of Technology, Delft, The Netherlands
{h.liu-8,h.ye-1,j.yang-3,qing.wang}@tudelft.nl

Abstract. Contact-free interaction with public devices could become popular. To achieve this purpose, state-of-the-art methods mainly use cameras to capture mid-air hand gestures, which are power-hungry and can raise privacy issues. In this work, we design **LightDigit**, a system for fingertip air-writing of digits with ambient light and photodiodes, to enable contact-free interactions with public devices. The key enabler is detecting and interpreting dynamic shadows on photodiodes introduced by fingertip movements. We design an embedded deep learning model **LightConvRNN** –customized ConvRNN with attention pooling– to capture spatial and temporal patterns in the dynamic shadows. We evaluate LightDigit through extensive experiments under different light conditions. Evaluation results show that our model can achieve an accuracy of up to 98%. Through model compression, the model size is reduced by 92% with less than a 5% drop in the performance. LightDigit is robust to ambient light positions (60°) and ambient light intensity (5000 lx).

Keywords: Air-writing · Ambient light · Embedded AI

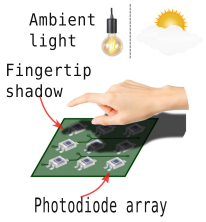
1 Introduction

Touch-free interaction with public devices might play an important role in our future lives, because it provides a means for people to avoid touching public devices shared among many people. This could reduce the potential spread of virus. Recently, a mid-air virtual touchscreen has been created to help people avoid touching public screens [1]. This is done by deploying a *camera* to capture and recognize mid-air hand gestures such as sliding up and sliding down. Although promising, using cameras leads to high energy consumption and raises privacy issues. Also, the Analog Devices company introduces a new generation of infrared-based dynamic optical sensors to sense a broader range of gestures [16]. This sensor can recognize simple gestures such as swipes and mid-air click by modulating several infrared LEDs and detecting the reflected infrared light.

Meanwhile, the ubiquitous visible light is gaining significant interest as a medium to sense/connect things (objects). Besides *active* visible light systems where LEDs are modulated to send information and objects leverage light sensors to receive that information, researchers are also studying *passive* visible light systems for various objects sensing [12,13,19,20].

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 594, pp. 179–191, 2024.
https://doi.org/10.1007/978-3-031-63992-0_11

(a) Using toothpicks during the pandemic



(b) The LightDigit system

Fig. 1. The motivation (a) and illustration (b) of our LightDigit system, a device-free fingertip air-writing system with ambient light and simple photodiodes.

Motivated by these rising fields, in this paper, we design *LightDigit*, an embedded and real-time system to enable touch-free fingertip writing of the digits 0–9 with *ambient light*. As illustrated in Fig. 1, with LightDigit people can use a finger to write digits in front of but not touching the photodiodes. The key enabler is the detection and embedded-learning-based decoding of the dynamic shadow cast by the fingertip movement.

Challenges. To achieve fingertip air-writing of digits with only ambient light and simple photodiode, several challenges must be tackled.

- *Human impact: Different people usually write certain digits differently.* Taking the digit '7' as an example, it is commonly written without an additional horizontal line crossing the middle, as depicted in Fig. 2. However, many people have alternative writing styles. Another example is '4': people typically stop the horizontal line at the vertical line, whilst others make them cross. Such systematical features of handwritten digits can also be observed from popular handwritten digit datasets collected from different regions, e.g., MNIST/EMNIST [6,7] (US), [3] (Europe), CASIA/HIT-OR3C [14,21] (Asia), etc. These regional variations in handwritten digits, as well as *writing speed* and *sizes of handwritten digits*, bring us challenges in recognizing air-writing digits with ambient light.
- *Environment impact: The shadow area –detected and interpreted for digit recognition– is affected by the intensity levels of ambient light, varying angles of light incidence, and passersby.* With different ambient light like the illumination and sunlight, the position of the ambient light source can be different, leading to different shadow angles. Even with the same shadow angles, the various heights and sizes of fingertips will also generate different shadow areas. Besides, the system's stability may be weakened by the fact that many light sources cast different shadows; passersby also affect the shadows.
- *Device impacts: Embedded devices are resource-constrained.* To handle the complex digit recognition task, deep learning models usually have a significant amount of parameters, which do not fit in embedded devices that have limited computing and storage resources. It is challenging to design light-weighted deep learning models that can run in real-time on resource-limited embedded devices with reduced inference time and memory usage.

Fig. 2. Regional digit writing styles: *left)* mostly adopted in western countries; *right)* mostly adopted in Asian countries.

Our Contributions. To the best of our knowledge, LightDigit is the first embedded system that exploits deep learning models in resource-constrained devices for *real-time* recognition of air-writing digits with light. Our work builds the first-of-its-kind ambient light dataset of air-writing digits and investigates state-of-the-art model compression techniques for efficient model inference on embedded devices. We summarize our contributions as follows:

- We propose, design, and prototype the LightDigit system. It is the first embedded system that enables fingertip air-writing of digits with ubiquitous ambient light and simple photodiodes. To improve the system's robustness, we also propose a calibration method to actively adapt to the shadow area variations caused by changes in ambient light and fingertip height.
- We analyze the attributes of air-written digits and build the first *ambient light dataset of air-written digits*, a dataset including time-series information. Traditional data-gathering methods need to recruit hundreds or thousands of volunteers. To avoid such time-consuming and labor-intensive processes, we analyze the properties of air-written digits to create our own training dataset and evaluate it by existing datasets. *Our LightDigit dataset has 20880 samples of air-writing digits, which represent 174 different types of the ten digits 0–9.* Due to the space limit, please refer to [10] for the dataset details.
- We thoroughly evaluate our system in different experimental scenarios and progressively construct the LightDigit system with the best performance in terms of accuracy, inference time, and the number of parameters. The testbed evaluation results show that the proposed embedded deep learning models are effective, with LightConvRNN achieving 98% accuracy in the Intra-Subjects scenario. With knowledge distillation, we can reduce its model size by more than 92% while keeping less than a 5% reduction in Accuracy. We also demonstrate that LightDigit is robust to ambient light positions (e.g., 60°) and different ambient light intensity levels (up to 5000 lx).

2 System Design

2.1 System Architecture

We design LightDigit as an embedded sensing system to enable touch-free fingertip writing of the digits 0–9 with ambient light (unmodulated), such as indoor illuminating light and sunlight. The overall system mainly has three components (cf. Fig. 3): *moving fingertip*, *photodiode (PD) array*, and *embedded deep learning algorithms*. The key enabler is the detection and interpretation of the dynamic shadow brought by the fingertip movements when a person uses one of his/her fingers to write digits in the air (on the top of the PD array).

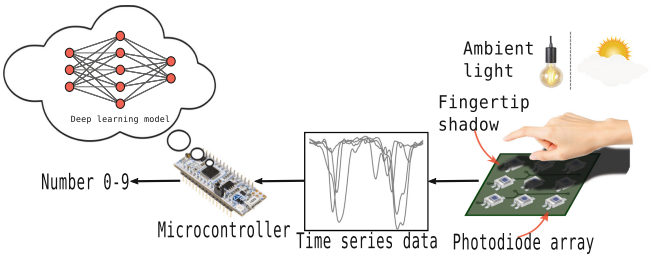


Fig. 3. System architecture.

Moving fingertip. It is the object whose trajectory will be sensed directly. When drawing a digit with finger in the front of the PD array, the movement of the finger will reflect the surrounding ambient light towards the PD array. The corresponding dynamic shadow, implicitly carrying the shape of the digit, is cast on the PD array.

Photodiode (PD) array. We use a group of PDs to detect dynamic shadows. The PDs are sampled sequentially with a fixed frequency, and the sampled data is sent to the embedded DL algorithm as the input for recognizing the digit.

Embedded deep learning algorithm. In this paper, we design an embedded deep learning algorithm LightConvRNN –customized ConvRNN with Attention– to recognize the air-written digits. LightConvRNN can be *trained and run on embedded devices* such as Raspberry Pi or BeagleBone AI and Black. We also propose a method to pre-process the captured data. To train the model, we built our own dataset LightDigit. Due to the space limit, please refer to our technical report [10] for the details of the LightDigit dataset.

2.2 Robustness Design

Complicated location distribution of ambient light sources and dynamic changes in handwriting height greatly impact the distribution of shadow areas, which also raises great challenges in recognizing handwritten digits. There is an intuitive method which we can calculate the angle from the light source to the sensing PD array by comparing the output values of the PDs from different columns or rows of PDs. However, the influence from the different heights of fingertips is not taken into account. For example, when there is a 45° illumination light for the sensing area, the shadow area on the sensing area varies according to the different heights of the fingertip to the sensing area. Thus, we propose an adaptive shadow recognition method to expand the original sensing area and automatically select a new sensing area.

We define two areas: (i) *writing area*, perpendicularly to which a person moves his/her finger to write the digits, and the M PDs contained in the writing area are marked as PD_1, PD_2, $\cdots$, PD_M. (ii) *sensing area,* that host M PDs from where the system will capture the shadow for further processing (Refer to Fig. 4 for details). The available writing area was also considered to be an important factor in this regard.

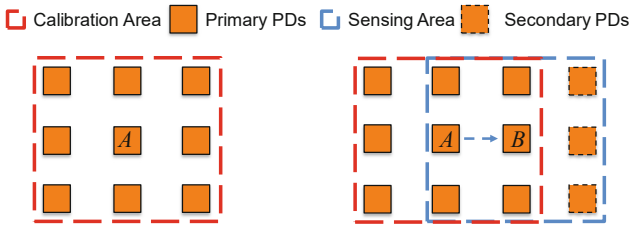


Fig. 4. Proposed calibration for better robustness.

The adaptive recognition method consists of four steps:

(1) *Reset calibration area.* We first need to set an area as the calibration area. As shown in Fig. 4, we mark an area as the calibration area, which contains M primary PDs that can output M channel values. Note that we only selected M PDs as the primary PDs. If more primary PDs are picked for the calibration area, calibration mode can provide a higher resolution in shadow projection.

(2) *Ambient light measurement.* Since the indoor light intensity does not change much in a short time, we first measure the ambient light intensity through primary PD i under the current environment, denoted as $p_i, i \in \{1, 2, \cdots, M\}$.

(3) *Handwriting position measurement.* Users need to put a fingertip on the center PD A of the calibration area. According to the actual light source distribution and the position distribution of the user's handwriting, the new output values of the M PDs are different, denoted as $q_i, i \in \{1, 2, \cdots, M\}$.

(4) *Shadow area determination.* Through the operation of step 1 to step 3, we can calculate the percentage change of the output value of each PD caused by the user operation, denoted as $\alpha_i = \frac{|p_i - q_i|}{p_i}, i \in \{1, 2, \cdots, M\}$. The largest α_i means that the shadow of the finger almost falls in this area, and the shadow of subsequent writing of numbers falls in the direction from the center PD of the calibration area to PD i. As shown in Fig. 4, we can obtain shadow destination $B = \arg\max_{i \in \{1,2,\cdots,M\}} \alpha_i$, and then activate the secondary PDs in the corresponding sensing area on the extension line of the shadow direction $\overrightarrow{AB}$ to obtain the correct handwritten digital shadow data. If $A = B$, the shadow is completely projected on the calibration area. Thus, we regard the calibration area as the sensing area.

3 Embedded Deep Learning Algorithm

We continue to introduce the embedded Deep Learning (DL) algorithm designed for the recognition of the air-writing digits. Our embedded DL algorithm takes as input the pre-processed data and then classifies the input into one of the ten digits 0–9. The key requirement for an effective embedded DL model is

the ability to capture both the spatial pattern among the channels and the temporal pattern across different time steps while accounting for the variations of writing styles within and between subjects. To this end, we design a customized architecture LightConvRNN –ConvRNN with Attention– for better control of the learning process for embedded machine learning. Next, we first present the data preprocessing and then describe the proposed embedded DL algorithm.

3.1 Data Preprocessing

We denote the raw data stream from channel/PD m as a vector $\boldsymbol{x}_m = [x_{m,1}, x_{m,2}, \cdots, x_{m,L}]$. Due to the noise interference and the limited on-device computing capability, $\boldsymbol{x}_m$ cannot be directly fed to the embedded deep learning models. Thus, we propose a procedure to preprocess the raw data.

Step 1) Outliers reset. The sampled abnormal value appears in $\boldsymbol{x}_m$ when the user unconsciously touches the sensing area with fingertips and causes the PDs array to move violently. Thus, we take the values at 99^{th} and 1^{st} percentile of the data $\boldsymbol{x}_m$ as the upper limit x_U and lower limit x_L, respectively. The filtered data $x_{m,l}$, $\forall l \in \{1, 2, \cdots, L\}$ can then be calculated as

$$x_{m,l} = \begin{cases} x_U, & x_{m,l} > x_U \\ x_L, & x_{m,l} < x_L \end{cases}. \tag{1}$$

Step 2) Data smoothing. There are many methods to reduce the output fluctuates caused by noise interference, including low-pass filters, high-pass filters, and Kalman filters [11] [5]. It is not suitable to use high-pass and low-pass filters in our case since the characteristics of the visible light signal we collect are distributed in various frequency bands. Also, Kalman filters can filter and smooth the data and predict the next data value. However, Kalman filters are overpowerful. To solve this issue, we deploy the lightweight Savitzk-Golay filter [17] that is designed for smoothing the data without distorting the signal tendency in our system. The filtered data $\tilde{\boldsymbol{x}}_{m,l}$ can be calculated as $\tilde{x}_{m,l} = \sum_{i=-w}^{+w} k_i x_{l+i}$, where w represents the window size, and k_i is the smoothing coefficient.

Step 3) Signal synchronization and time series compression. To reduce the computational overhead in embedded devices, we use energy detection to synchronously extract the effective shadow signal segment from the data stream $\tilde{\boldsymbol{x}}_{m,l}$. Firstly, we use *minmaxscaler* tool to obtain normalized $\hat{\boldsymbol{x}}_{m,l}$. Next, we set a threshold to extract the action signal $\bar{\boldsymbol{x}}_m = [\hat{x}_{m,a}, \hat{x}_{m,a+1}, \cdots, \hat{x}_{m,b}]$, where the starting and ending index of action signal is calculated as $a = \arg\min_l (x_{m,l} < \gamma)$ and $b = \arg\max_l (x_{m,l} < \gamma)$, respectively. The threshold is set as $\gamma = 0.7 x_{max}$, where $x_{max} = \max \hat{\boldsymbol{x}}_{m,l}$. The difference $(b - a)$ may vary as channel m changes. Thus, the length of processed data is shortened to the fixed number of time steps N by using interpolation to make the data better adapt to our deep learning models, denoted as $\bar{\boldsymbol{x}}_m = [\hat{x}_{m,1}, \hat{x}_{m,2}, \cdots, \hat{x}_{m,N}]$.

Finally, denoting the number of channels as M and integrating the data of channel $m \in \{1, 2, \cdots, M\}$, the input data format to the learning models is $D_{M \times N} \triangleq [\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \cdots, \bar{\boldsymbol{x}}_M]$.

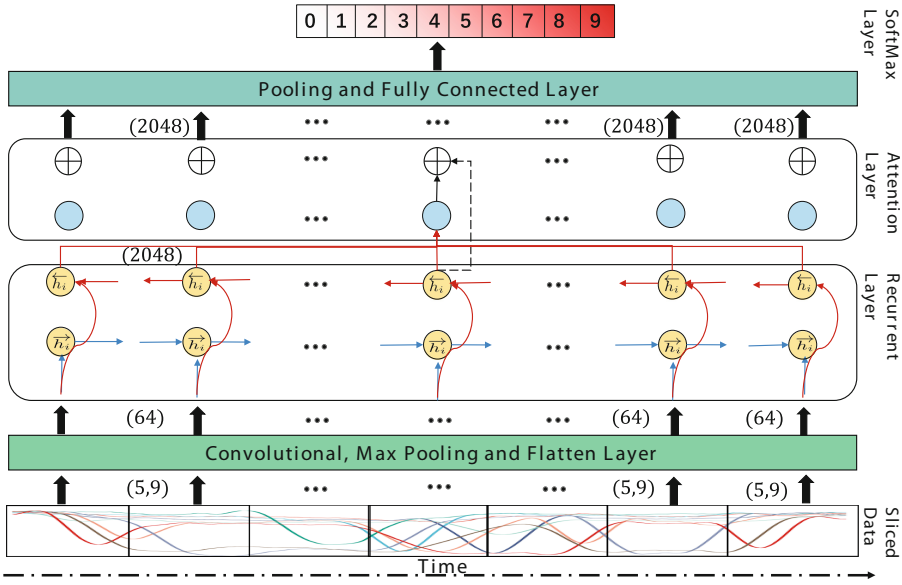


Fig. 5. The architecture of the proposed LightConvRNN: a customized *ConvRNN with Attention* model.

3.2 LightConvRNN: Customized ConvRNN with Attention

We explicitly model the spatial and temporal patterns to better control the learning process. To do so, we first segment the multi-dimensional time series data $D_{M\times N}$ into k smaller chunks $k\times D_{M\times \frac{N}{k}}$. Our network starts with a convolutional layer on each of the chunks, so as to explicitly model the spatial pattern among channels. Note that it also helps to capture short-term temporal patterns. Afterwards, we apply the pooling operator for information aggregation and obtain a new data representation of size $D_{M'\times k}$. We then build bi-directional recurrent layers, i.e., bi-directional GRU [4], to capture longer-term temporal patterns. GRU [9] uses the gating mechanism to balance the information flow from previous time steps in encoding the time series, and the bi-directional GRU [18] contains a forward and a backward layer to encode information flow from both directions of the time series for better information preservation. Even with the gating mechanism, it is still challenging for GRU to deal with long range dependencies in the times series. We therefore, add the attention pooling mechanism to help the model preserve relevant information from all time steps: the output is a weighted sum, i.e., a selective summary, of the bi-directional GRU's hidden units at all time steps. Output of the attention pooling layer is then fed to the fully-connected layers and finally to the softmax layer to generate the classification result, a vector of size ten, representing the predicted probability of the digit being 0–9. The overall model structure is depicted in Fig. 5.

To reduce the complexity of LightConvRNN, we compress it using knowledge distillation [2], a technique that is also used by state-of-the-art studies on visible light sensing such as [15]. Knowledge distillation trains a large model (i.e., the teacher model) and then transfers the learned patterns to a smaller model (i.e., the student model), by training the smaller model to learn from both the original labels and the output from the larger model (referred to as pseudo labels, as in contrast to the original labels).

4 Performance Evaluation

4.1 Testbed Implementation

The prototype of LightDigit is shown in Fig. 6. We use Raspberry Pi 4 as our controller due to its compatibility with deep learning platforms like TensorflowLite, Tensorflow, Keras, and Pytorch. In addition, OPT101 is selected since it is an integrated combination of PDs and trans-impedance amplifier on a single chip, which is very stable to sensing light. The sensing board consists of 16 OPT101 PDs with fixed equal intervals, which is shown in Fig. 6. Note that we only use M=9 out of the 16 PDs for detecting the shadow. The 9 PDs are dynamically selected during the calibration, cf. Sect. 2.2 for the details. The size of the square sensing area is set to 41 mm × 41 mm by the tradeoff between a convenient writing area for people and fewer PDs. We use two Analog-to-Digital Converters (ADCs) to sample the 16 PDs. In our implementation, the sampling frequency of the PD is fixed to 100 Hz because we find that it is already enough for handling the fastest digit writing (<0.5 s) in our test.

4.2 Experimental Setup

Testing Environments. The data is collected in two typical indoor environments: a student apartment and a university lab. We use a reading lamp and fluorescent lamps as the ambient light source (the light is not modulated). The average ambient light intensities at the sensing board can be changed between

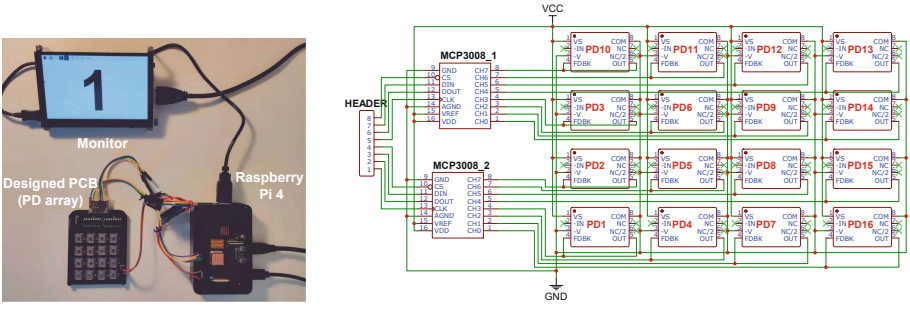


Fig. 6. LightDigit prototype: *left)* hardware, *right)* schematic.

Table 1. Performance of our LightConvRNN model (including comparisons with state-of-the-art works). Results are averaged over ten runs in each configuration.

Scenarios	RNN [1,8]			ConvRNN [15]			LightConvRNN		
	ACC	#Param	Itime(s)	ACC	#Param	Itime(s)	ACC	#Param	Itime(s)
Intra-Subjects	85.9%	808458	0.5531	91.4%	1369994	0.4494	97.9%	1754059	0.5147
Inter-Subjects	77.2%	808458	0.5569	79.2%	1369994	0.4406	90.9%	1754059	0.5189

143 lx and 5000 lx. The incident angle of the light emitted by the lamps to the sensing board can be adjusted between 0° and 60°.

Hyperparameter Setting. Optimal hyperparameters are searched for LightConvRNN in each scenario. We empirically set optimal parameters based on a hold-out validation set that contains 10% of the test data on Intra-Subjects and Inter-Subjects scenarios. For the learning rate, we apply a grid search in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The number of filters in CNN and the number of hidden units in GRU are explored from the set {8, 16, 32, 64, 128, 256} and the set {32, 64, 128, 256, 512, 1024}, respectively. Model training is performed using Adagrad with mini-batches of size 32. The dropout rate is selected from the set {0.0, 0.1, 0.2, 0.3}. Model compression is performed as follows. After the optimal hyperparameters are found, we reduce the hidden units in the GRU layer.

We consider two *test scenarios* to evaluate our system: (1) **Intra-Subjects**: When LightDigit is only used by a fixed population, such as in schools, the capacity to recognize similar data is crucial. Thus, we shuffle each participant's data in the real dataset and divide it into training (80%) and test (20%). The training set is enriched with our simulated dataset. The training set is enriched with our simulated dataset. (2) **Inter-Subjects**: If LightDigit is deployed in a public place, anyone from various geographic regions has the ability to use it. In this instance, the system's ability to recognize data from unknown domains, i.e. its ability to generalize, is crucial. Using cross-validations, we utilized 2/3 of the real data, or 18 individuals, for training and the remaining 6 for testing.

Metrics. We use the metrics Accuracy (ACC), number of parameters (Param), and inference time (Itime). ACC quantifies the proportion of correct categorization across all instances of data. Param measures the model size, a key component in deciding the amount of storage space in an embedded device. The model's inference time running on the embedded device is impacted by the number of parameters, input size, and CPU operating modes (serial/parallel) (Table 1).

4.3 Evaluation Results

Proposed Model vs. State of the Art. We first compare LightConvRNN with the state-of-the-art RNN [1,8] and ConvRNN [15]. We evaluate these models on our LightDigit dataset. Table 2 demonstrates their performance and model size in both scenarios. Our LightConvRNN surpassed RNN and ConvRNN by 5% to 15% in terms of accuracy. In terms of model size, LightConvRNN is

Table 2. Performance under different light angles.

Scenarios	Angle	Accuracy	
		With calibration	Without calibration
Intra-Subjects	0°	94.67%	95.91%
	30°	85.12%	15.38%
	60°	75.14%	8.65%
Inter-Subjects	0°	90.63%	90.92%
	30°	84.09%	24.79%
	60°	69.89%	10.58%

approximately 54% larger than RNN and 22% larger than ConvRNN. As for inference time, ConvRNN and LightConvRNN have comparable inference times since they require a similar amount of serial processing. We can observe that the inference time of RNN is longer than ConvRNN and LightConvRNN because the architecture of ConvRNN separates input data into smaller chunks, hence reducing the number of serial operations. LightConvRNN requires more time for inference than ConvRNN since it has an additional attention layer.

We can observe that the optimal performance is under the Intra-Subjects scenario, where LightConvRNN achieves almost 98% accuracy. Performance in the Inter-Subjects scenario is lower: 91%. The result confirms the larger variation of air-writing styles across subjects as compared to that within the same subject, thus posing a bigger challenge for digit recognition. In the Inter-Subjects scenario, our system achieves over 90% accuracy with LightConvRNN. This result demonstrates the effectiveness of our system in recognizing air-writing digits, especially by subjects who have been historically seen by the system.

Shadow Shifting. We compare the performance of LightConvRNN on calibrated and non-calibrated data with a fixed fingertip height between 0.5 cm and 1.5 cm. Results are given in Table 2 on the three incident light angles (0°, 30°, and 60°). We observe that LightDigit performance under 0° incident angle is equally well in calibrated and non-calibrated modes across all scenarios (less than 2% difference) due to the overlapping drawing and sensing areas. When it comes to the 30° incident angle, the performance in non-calibration mode shows a heavy drop from over 90% accuracy to less than 30% in all scenarios. By contrast, LightDigit performance in the calibration mode remains high: 85% and 84% in Intra-Subjects and Inter-Subjects scenarios, respectively. When the incident angle is 60°, the system performs poorly in non-calibration mode with less than 20% accuracy in all scenarios, while keeping around or over 70% accuracy in the different scenarios. Those results demonstrate the strong effectiveness of the proposed calibration for robustness improvement. We note that there is still a performance drop in high-incident angle situations. It happens for several reasons: fingertip shadow under high incident angles can cover multiple PDs, as compared to an individual PD in normal situations; slight changes of the ver-

Table 3. Accuracy under different ambient light intensities.

Light intensities (lux)	500	1000	5000
Accuracy	91%	91.5%	99.1%

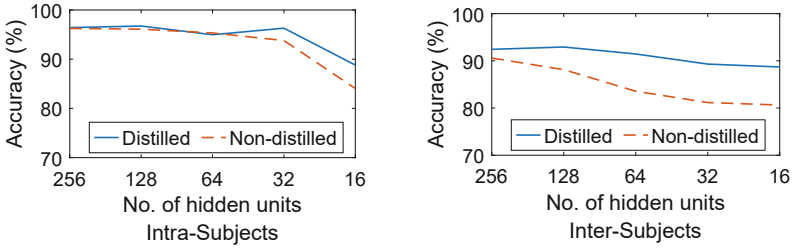


Fig. 7. Distilled vs. non-distilled LightConvRNN. Number of parameters under different number of hidden units are 1754k, 484k, 144k, 48k, 18k, 8k, respectively.

tical distance between the fingertip and the sensor – which frequently happens for complex digits like '6' and '8' – can easily move the shadow from one PD to another. We leave it to future work to tackle those issues.

Different Ambient Light Intensities. To test the robustness of our system under different ambient light intensities, we place an indoor light source directly above our sensing board and vary the light source's height to achieve different light intensities perceived by the PDs. The evaluation results are given in Table 3. As the light intensity increases, we can see that the accuracy rate remains above 90%. This demonstrates the stability of our system up to an ambient light intensity of 5000 lx. In addition, it can be observed that the accuracy is higher at 5000 lx than under other light intensities. This is primarily because, at 5000 lx, the light source is very close to the system. As a result, the projection of the finger is deeper and more compact, resulting in cleaner signals, and thus, a slightly higher accuracy.

Model Compression. To make our model run on embedded devices, we use knowledge distillation to reduce the model size and inference time. To evaluate the performance, we compare the distilled model (i.e., student model) with the non-distilled model that has the same number of parameters trained from scratch. Figure 7 shows the result for LightConvRNN, with the number of hidden units decreasing 256 (Intra- and Inter-Subjects) to 16, halved in each step; note that the teacher model has 512 hidden units in the Intra- and Inter-Subjects scenarios. We observe a similar result that distilled LightConvRNN consistently outperforms non-distilled LightConvRNN. In the Intra- and Inter-Subjects scenarios, the number of hidden units can be reduced to 128 with performance kept almost unchanged, amounting to 92.27% reduction of parameters. In all scenarios, the number of hidden units can be reduced to 32, while keeping less than 5% drop in performance. This amounts to 98.95% (Intra-Subjects and

Table 4. Inference time for different numbers of hidden units in LightConvRNN.

Number of units	512	256	128	64	32	16
Inference time (s)	0.518	0.508	0.497	0.495	0.495	0.494

Inter-Subjects) parameter reduction, respectively. Table 4 shows the changes in inference time that accompany changes in the model parameters. In summary, we have observed a strong effect of knowledge distillation in compressing the model. This result indicates that hidden units in the recurrent layers of LightConvRNN contain a large amount of redundant information that can be removed without affecting the performance.

5 Conclusion

We have designed LightDigit, an air-writing digit recognition system by exploiting the dynamic shadow caused by the fingertip movement with a low-cost PD array and ambient light. We offer a calibration method and utilize data processing techniques to enhance the system's robustness. To execute the recognition, we designed an embedded deep-learning model LightConvRNN. We created a reliable and trustworthy LightDigit dataset for model training. We prototyped LightDigit and validated that it can achieve superior performance. We envision it can motivate more follow-up research on passive sensing with ambient light.

References

1. Alam, M.S., et al: Trajectory-based air-writing recognition using deep neural network and depth sensor. Sensors (2020)
2. Ba, L.J., Caruana, R.: Do deep nets really need to be deep? arXiv:1312.6184 (2013)
3. van Breukelen, M., et al: Handwritten digit recognition by combined classifiers. Kybernetika (1998)
4. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555 (2014)
5. Cipra, T., Romera, R.: Robust kalman filter and its application in time series analysis. Kybernetika (1991)
6. Cohen, G., Afshar, S., Tapson, J., Van Schaik, A.: EMNIST: extending MNIST to handwritten letters. In: IEEE IJCNN (2017)
7. Deng, L.: The MNIST database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. (2012)
8. Duan, H., Huang, M., Yang, Y., Hao, J., Chen, L.: Ambient light based hand gesture recognition enabled by recurrent neural network. IEEE Access (2020)
9. Graves, A., et al: Speech recognition with deep RNN. In: IEEE ICASSP (2013)
10. Hao, L.: LightDigit Dataset (2021). https://www.dropbox.com/s/lbllt66mnunj22g
11. Li, Q., Li, R., Ji, K., Dai, W.: Kalman filter and its application. In: ICINIS (2015)
12. Li, T., An, C., Tian, Z., Campbell, A.T., Zhou, X.: Human sensing using visible light communication. In: ACM MobiCom (2015)

13. Li, T., Zhou, X.: Battery-free eye tracker on glasses. In: ACM MobiCom (2018)
14. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: CASIA online and offline Chinese handwriting databases. In: IEEE ICDAR (2011)
15. Liu, H., Ye, H., Yang, J., Wang, Q.: Through-screen visible light sensing empowered by embedded deep learning. In: AIChallengeIoT Workshop (2021)
16. Maxim integrated: Max25405, April 2021. https://datasheets.maximintegrated.com/en/ds/MAX25405EVKIT.pdf
17. Menon, S.V., Seelamantula, C.S.: Robust Savitzky-Golay filters. In: ICDSP (2014)
18. Schuster, M., et al: Bidirectional recurrent neural networks. IEEE TSP (1997)
19. Yang, Y., Hao, J., Luo, J., Pan, S.J.: CeilingSee: device-free occupancy inference through lighting infrastructure based led sensing. In: IEEE PerCom (2017)
20. Zhang, C., Tabor, J., Zhang, J., Zhang, X.: Extending mobile interaction through near-field visible light sensing. In: ACM MobiCom (2015)
21. Zhou, S., et al: An empirical evaluation on HIT-OR3C database. In: ICDAR (2011)