Through-Screen Visible Light Sensing Empowered by Embedded Deep Learning

Liu, Hao; Ye, Hanting; Yang, Jie; Wang, Qing

DOI
10.1145/3485730.3493454

Publication date
2021

Document Version
Final published version

Published in
SenSys 2021 - Proceedings of the 2021 19th ACM Conference on Embedded Networked Sensor Systems

Through-Screen Visible Light Sensing Empowered by Embedded Deep Learning

Hao Liu
Delft University of Technology
Delft, The Netherlands
h.liu-7@student.tudelft.nl

Hanting Ye
Delft University of Technology
Delft, The Netherlands
h.ye-1@tudelft.nl

Jie Yang
Delft University of Technology
Delft, The Netherlands
j.yang-3@tudelft.nl

Qing Wang
Delft University of Technology
Delft, The Netherlands
qing.wang@tudelft.nl

ABSTRACT

Motivated by the trend of realizing full screens on devices such as smartphones, in this work we propose through-screen sensing with visible light for the application of fingertip air-writing. The system can recognize handwritten digits with under-screen photodiodes as the receiver. The key idea is to recognize the weak light reflected by the finger when the finger writes the digits on top of a screen. The proposed air-writing system has immunity to scene changes because it has a fixed screen light source. However, the screen is a double-edged sword as both a signal source and a noise source. We propose a data preprocessing method to reduce the interference of the screen as a noise source. We design an embedded deep learning model, a customized model ConvRNN, to model the spatial and temporal patterns in the dynamic and weak reflected signal for air-writing digits recognition. The evaluation results show that our through-screen fingertip air-writing system with visible light can achieve accuracy up to 91%. Results further show that the size of the customized ConvRNN model can be reduced by 94% with less than a 10% drop in performance.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Computer systems organization** → **Embedded systems**.

KEYWORDS

Through-screen sensing, embedded AI, embedded deep learning

ACM Reference Format:
Hao Liu, Hanting Ye, Jie Yang, and Qing Wang. 2021. Through-Screen Visible Light Sensing Empowered by Embedded Deep Learning. In *The 3rd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT 21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3485730.3493454

SenSys '21, November 15–17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11.
https://doi.org/10.1145/3485730.3493454

1 INTRODUCTION

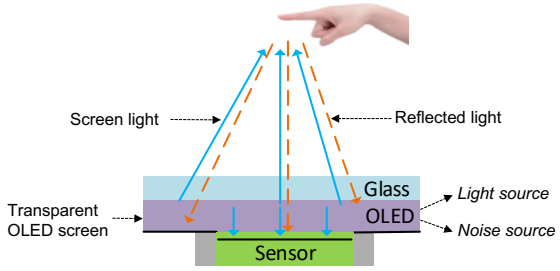

Figure 1: Proposed through-screen visible light sensing (targeted application in this work: air-writing digit recognition).

The COVID-19 pandemic has changed our lifestyles. We are trying to avoid physically touch objects in our daily lives, such as avoiding touching the buttons in the elevator and ATM with our fingers. In this case, touchless mid-air controllable devices are becoming more and more popular. Researchers have studied deploying cameras to realize touchless interactions such as air-writing digit recognition [1]. Although promising, using cameras brings high energy consumption and privacy safety issues. Other researchers combine the ubiquitous ambient light to realize the preliminary air-writing digit recognition [6, 13, 14]. The principle of these systems comes from identifying the shadow caused by the moving object that blocks the light source. However, this application scenario faces a big challenge: *the light sources in real life is complex and changeable, making these systems not that reliable in many practical scenarios.* Although some lightweight algorithms in [13, 14] and machine learning methods in [6] have been proposed to improve the robustness of the recognition system based on ambient light, but still, they cannot cover all ambient light scenes.

Recently, the new trend toward narrow-bezel and even no-bezel (i.e., full-screen) smartphone designs to deploy more sensors (ambient light sensor, camera) under the screen. *This motivates us to leverage the screen as the light source and the under-screen sensors as the receiver to enable through-screen sensing such as air-writing recognition by using the reflection of the finger on the screen light, as depicted in Figure 1.* The benefit of this design is that the screen light source is *fixed* and that *no additional external light source* is required. Nevertheless, we still face several challenges as described below.

Firstly, people from different regions of the world write digits with different shapes, which significantly increases the types of digits to be recognized. Currently, there is a dataset *LightDigit* [8] for air-writing digit recognition based on blocked light. However, the recognition based on blocked light and reflected light is different. In the blocked light scenario, the receiver detects the shadow changes caused by the finger's movement that blocks the light source. In the reflected light scenario, the receiver recognizes handwritten digits by detecting the light emitted from the light source to be reflected by the finger. In this work, we study through-screen air-writing digit recognition where we exploit the reflection of light by our fingers. We propose a signal flip method for the data collection and this could allow us to extend and reuse the LightDigit dataset.

The second challenge is that the screen is not only a fixed light source for the through-screen air-writing system but also a *strong noise source*. Theoretically, there should be no change in the brightness of the screen at full brightness. However, from our experiments, this is not always true. We often observe that the brightness of the screen fluctuates sharply even at full brightness. Thus, we propose a data preprocessing method to reduce the interference of screen noise source. The process is performed by detecting and removing useless abnormal values in the signal, smoothing and filtering the residual interference signal to recover the weak signal characteristics of the light reflected by the finger.

The third challenge we encounter is recognizing handwritten digits from weak reflected signal characteristics on resource-constrained embedded devices. In this work, we design an embedded deep learning model *ConvRNN*, which is a customized convolution-recurrent neural architecture for better control of pattern recognition. Our model can learn the spatial and temporal patterns in the dynamic and weak reflected signal for air-writing digits recognition. Further, we apply knowledge distillation for model compression to meet with the limited resources on embedded devices.

2 BACKGROUND

2.1 Transparent OLED screen

The OLED screen has a sandwich structure, where is an organic light-emitting layer between the anode and cathode. The organic light-emitting layer emits light on both sides, and the anode usually uses transparent material to display the contents of the screen. When the cathode is also made of a transparent material such as transparent indium tin oxide, a *transparent screen* can be formed. Meanwhile, external light can pass through the entire OLED screen.

2.2 Air-writing digit with reflected light

When deploying simple photodiodes to recognize air-writing digits, the relative position of fingers and light source is paramount. It can be divided into two cases according to blocked light and reflected light (cf. Figure 2). When it comes to blocked light, the light source is generally placed above the finger (ceiling, high wall area), and then the photodiode at the bottom place (table, low wall area) will be utilized to sense the shadow changes caused by the movement of the finger blocking the light source; When it comes to reflected light, the light source is usually placed between the finger and the receiver, one application scenario is to surround a circle of LEDs on the sensing area as light sources [11]. Thus, the photodiode of the sensing area can detect the light emitted from the light source to be reflected by the finger. However, there is a special scene that uses sensors under the newly emerging and popular full-screen and transparent OLED screen as the receiver and the screen as the light source for the reflected light in the sensing scene. As shown in Figure 1, the screen not only serves as a signal source of reflected light from the finger but also as a fixed strong interference source. It also brings us great challenges in air-writing digit recognition.

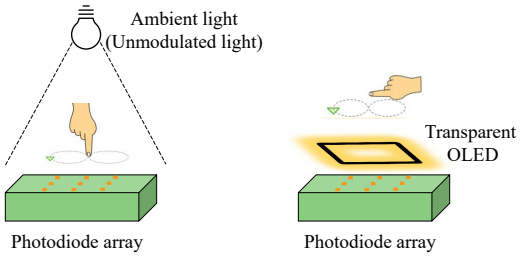

Figure 2: Air-writing digit based on blocked light (left) and reflected light (right).

2.3 The LightDigit dataset

Recently, there is a new air-writing digits dataset *LightDigit* [8] by an individual going through 70000 images in the MINTS dataset [5] and replicating them with air-writing and ambient light to obtain the time-series information. Especially, the air-writing of each type is simulated 120 times, and each time lasts for 5 seconds. As a result, a dataset with 20880 instances of air-writing digits is built in *LightDigit*. Each instance consists of $9 \times 100 \times 5 = 4500$ points (number of channels $\times$ sampling frequency $\times$ seconds). Also, five feature dimensions for air-writing digits are defined as: *Height (small and large), Width (small and large), Inclination (vertical and inclined), Shape (27 shapes for digits 0-9),* and *Time Sequence*. Thus, there are a total of 174 different types of air-writing digits for the digits 0–9 in *LightDigit* by combining the first four features and removing some impossible types of handwritten numbers.

2.4 Embedded deep learning model

Machine learning provides a variety of methods to automatically capture feature patterns from massive data. In particular, Deep learning is expert at extracting feature representations from low-level features (e.g., pixels) through multi-layer neural networks. Specialized neural architectures have been designed for representation learning from different types of data, e.g., convolutional networks for learning spatial features from image data [12] and recurrent networks for learning temporal features from time series data [16] [4] [7]. The latest development in further optimization of network architectures includes, e.g., the residual connection [10] to improve the effectiveness of model training, the model compression technologies [9] [2] to lightweight the model size and the attention mechanism [3] to focus selectively on certain relevant parts of the input. Our work presents a working embedded sensing system for air-writing digits recognition, with a dataset for model training, one deep learning models, that are compressed by the state-of-the-art knowledge distillation method.

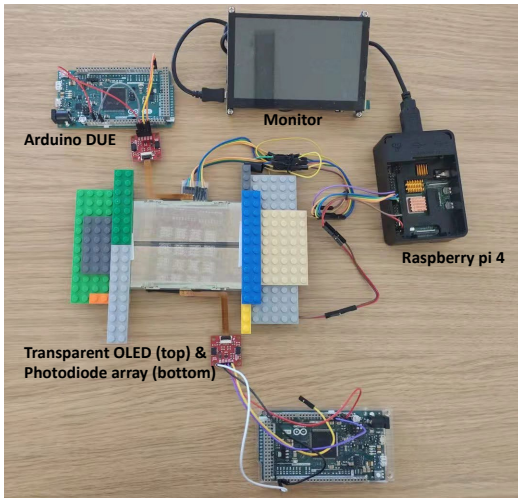

Figure 3: The hardware prototype of the proposed through-screen visible light sensing of air-writing digits.

3 SYSTEM DESIGN

3.1 System architecture

We design an embedded sensing system to recognize mid-air handwritten digits by utilizing the screen light without any additional light source. The key enabler of through-light sensing is simple: *the reflected screen light can pass through the transparent OLED screen and could be detected by the photodiodes.* The user moves his/her finger above the screen to draw digits, reflecting the light emitted by the screen. Then the reflected light will go through the transparent screen and can be detected by the under-screen photodiode array with the existing screen light interference. Thus, the whole system consists of four parts: *transparent OLED screen, moving fingers, photodiode (PD) array, and sensing algorithm*, as shown in Figure 3. The received superimposed light signal is translated by our designed embedded deep learning model and output the recognized number on the monitor.

- *Transparent OLED screen.* The screen emits light to the fingers and generates corresponding reflected light, which can be cast to the sensor through this transparent screen.
- *Moving fingertips.* It is the object whose trajectory will be sensed. When a person draws a digit with fingers in the front of the transparent OLED screen, the movement of the finger will reflect the light emitted from the OLED screen towards the photodiode array. The corresponding dynamic reflection light, implicitly carrying the shape of the digit, will be cast on the photodiode array.
- *Photodiode array.* We leverage M photodiodes, that are placed in a $\sqrt{M} \times \sqrt{M}$ array, to detect the dynamic shadows. In each sampling slot, the M photodiodes are sampled sequentially, and the sampled data is sent to the sensing algorithm as the input for recognizing the air-writing digit.
- *Sensing algorithm.* We design an embedded deep learning model to recognize the air-writing digits. This part includes the method we design to pre-process the captured data.

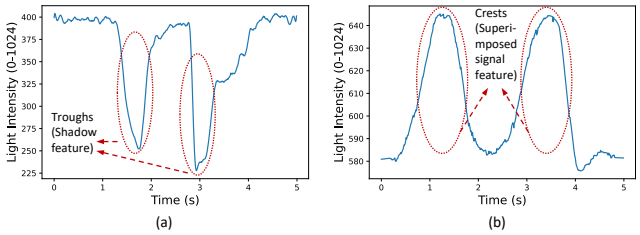

Figure 4: The comparison of datasets: (a) blocked light dataset; (b) reflected light dataset.

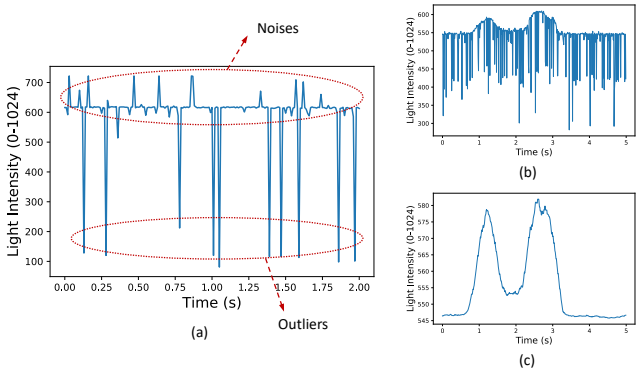

Figure 5: The captured visible light signal at the one of the photodiodes: (a) When the screen is full brightness (no finger at the top of the screen); (b) When the screen is full brightness and digit '0' is handwritten in the air; (c) When the screen is full brightness and digit '0' is handwritten in the air (processed data).

3.2 Data collection and processing

The dataset is mainly composed of two parts: blocked light dataset and reflected light dataset. In the blocked light dataset, the dataset mainly contains the characteristics of shadow (Troughs in the received light signal) changes caused by users' finger movement to block the light source due to the light source is above the finger. However, in the reflected light dataset, since the light source is between the finger and the receiver, the received superimposed light signal (Crests in the received light signal) is enhanced due to the composed of the reflected light of the finger and the light emitted by the screen. The comparison of two kinds of datasets is shown in Figure 4. Currently, there is a *LightDigit* dataset for mid-air handwritten digit recognition. It contains a total of 20880 data.

As for the reflected light dataset, we collected from 3 recruited participants, and each digit gesture of participant in the dataset takes a fixed time of 5 seconds to be collected. In order to generalize and reuse the existing blocked dataset, we have made the following transformation to collected data. We denote the raw data stream from channel m as a vector $\boldsymbol{x}_m = [x_{m,1}, x_{m,2}, \cdots, x_{m,n}]$ and create a zero vector with the same time step as $\boldsymbol{O}_m = [O_{m,1}, O_{m,2}, \cdots, O_{m,n}]$. Next, we get a flipped vector $(\boldsymbol{O}_m - \boldsymbol{x}_m)$ and we use *minmaxscaler* tool to obtain normalized flipped vector $\boldsymbol{y}_{m,n} = [y_{m,1}, y_{m,2}, \cdots, y_{m,n}]$ to set the same light intensity range for subsequent signal processing: i) *Outliers reset* to remove significantly abnormal values due to improper operation; ii) *Data smoothing* to reduce the output

fluctuates caused by noise interference; iii) *Action signal synchronization and extraction* to extract the effective signal segment from the data stream. Especially when the screen is at full brightness, the received signal does not remain stable but fluctuates sharply due to the instability of the control circuit current as shown in Figure 5(a), which cannot be caught by the naked eye but can be detected by our designed photodiode array. Compared with the stable ambient light signal, the screen signal can be regarded as a severe noise interference signal for the reflected weak light signal. Also, the screen light is used as both a noise source and a light signal source, the intensity of which is several orders of magnitude of the intensity of the reflected light from finger movement. Thus, we first consider removing the abnormally downward peak point of the screen (The lower part of the sampled signal in Figure 5(b)). Note that because the screen light is also used as the light signal source, the reflected light signal from the finger is also very weak at this time, and it is not enough to support the detection of the characteristics of the finger movement. Similarly, when the screen light signal fluctuates at the average intensity, the stronger/weaker of the screen noise signal will also make the reflected light brought by the finger stronger/weaker. Therefore we need to smoothly filter the interference of the screen noise signal (The upper part of the sampled signal in Figure 5(b)). We then consider using a Savitzk-Golay filter [17] is shown as follows:

$$\tilde{y}_{m,l} = \sum_{i=-w}^{+w} k_i y_{l+i} \,, \tag{1}$$

where w represents the window size, and k_i represents the smoothing coefficient. Here, to remove the interference noise signal of the screen as much as possible and obtain more stable filtered data, we set a large smoothing coefficient k_i. As shown in Figure 5(c), we can observe some weak reflected light characteristics caused by mid-air digital handwriting from the processed data. The next step is to integrate these processed data containing feature values into a data format as $D_{M\times N} \triangleq [\bar{\boldsymbol{y}}_1 \bar{\boldsymbol{y}}_2 \cdots \bar{\boldsymbol{y}}_M]$ and input into our customized neural network.

To evaluate our system, we consider three realistic scenarios:

1) *Cold-Start:* The training set is the *LightDigit* dataset [8], and the test set contains is the newly collected reflected light dataset. Without any prior knowledge of the reflecting light within the training set, this scenario is extremely challenging and can help test the utility of the reflected light dataset.

2) *Within-Subjects:* Considering that the designed system may be used in some large organizations with fixed user groups, the system has the authority to access the handwritten digital database of users. Thus, we shuffle the newly-collected reflected data from each participant and split it into training (80%) and test (20%) in our experiment. In addition, the training set is augmented with the *LightDigit* dataset.

3) *Between-Subjects:* Here, a typical and tougher scenario where the system has got to recognize digits from participants never seen historically is considered to evaluate system performance in some places that often receive external visitors. In this scenario, we split the reflecting light data by participants, using data from 67% people for training and therefore the rest 33% for the test dataset. We build three versions of the training-test data, each with a random participant within the test set, which enables us to use cross-validation to get stable leads to the experiment. Similarly, the training set is additionally augmented with the *LightDigit* dataset.

3.3 Embedded deep learning model

Next, we will introduce our customized deep learning model. It takes the previously processed data $D_{M\times N}$ as input to output the recognition results of air-writing digits. The model is mainly designed according to the characteristics of data (e.g., time, spatial, local, global characteristics, etc.). In addition to model design, we also consider model compression technology to make our model more lightweight.

3.3.1 Customized ConvRNN model. Overall, our data is a long time series with temporal dependencies among the time points. Recurrent neural layers, e.g., GRU [4], can capture long-term temporal features; they apply the gating mechanism to balance the importance of past time step information and current time step information. Such layers are therefore suitable for our scenario. Note that in theory, a bidirectional recurrent neural layer is more suitable for our recognition task. For example, many people write the digit '5' in two directions, i.e., from the top to the bottom and vice versa, or write the digit '0' clockwise and counterclockwise. Yet, our empirical results show that the bidirectional GRU does not give higher performance than a uni-directional one. This is likely due to the additional information in data that contains strict one-directional patterns, e.g., the information from the period when the finger enters the sensor board to the beginning of writing the digit and the information from the period when finger ends writing to moving out the sensor board. Apart from GRU, we further consider convolutional layers to capture the spatial pattern of data across the M channels. We divide the multi-channel data $D_{M\times N}$ into k chunks with $D_{M\times \frac{N}{k}}$, and each chunk is connected with a convolution neural layer. We then use a pooling layer to synthesize the data information, and connect it to the recurrent layer mentioned earlier. The output of the recurrent layer will be fed to the full connection layer; and then through the softmax layer, we get the recognition result, the probability prediction of digits 0-9. The overall model is shown in Figure 6.

3.3.2 Model Compression. We compress the customized ConvRNN model using knowledge distillation. The key idea is to transfer the knowledge learned from complex models to lightweight models, while improving the performance of the lightweight models as much as possible without increasing the number of parameters. The original complex model is commonly known as the teacher model and the lightweight model is commonly known as the student model. The transfer of knowledge from the teacher model to the student model is done by leveraging the output of the teacher model as additional supervision signals to train the student model. The learning objective of the student model becomes

$$Loss = CE(y, p) + \alpha CE(q, p) \,, \tag{2}$$

where CE denotes cross-entropy; $CE(y, p)$ measures the loss between the true label y and the predicted label p from the student model, and $CE(q, p)$ measures the loss between the pseudo label

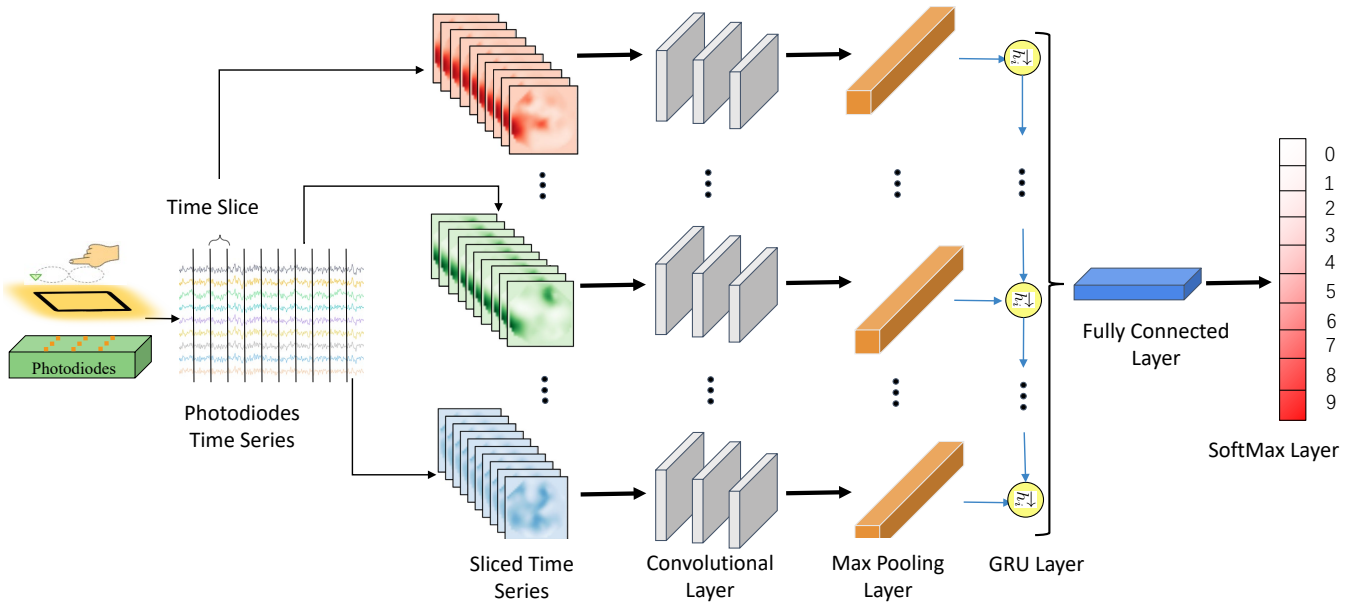

Figure 6: Architecture of the customized ConvRNN model.

q from the teacher model and the predicted label p from the student model; α is a hyperparameter that controls the strength of knowledge distillation.

4 PERFORMANCE EVALUATION

4.1 Experiment setup

Preparation of collecting the reflecting dataset. The *LightDigit* dataset [8], a blocked light dataset, is regarded as the primary training dataset to recognize the digits data drawn under reflected light in our system. In this case, we need to adjust the finger posture. The reason is that when we write digits with our fingers under blocked light, in most cases, the full shadow of the fingers will be projected to the photodiode array by the light source. To emulate this effect, when we write digits under reflected light, we will spread our fingers as horizontally as possible, as shown in Figure 2.

Hyperparameter setting. Optimal hyperparameters are searched separately for each scenario. We empirically set optimal parameters based on a head-out validation set containing 10% of the test data on Within-Subjects and Between-Subjects scenarios. For the Cold-Start scenario, we use 10% of the training set (i.e., the LightDigit set) as the validation set to strictly simulate the situation where no real data is accessible by the system. 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 that of hidden units in GRU are explored from {8, 16, 32, 64, 128, 256} and {32, 64, 128, 256, 512, 1024}, respectively. Model training is performed using Adagrad [15] with mini-batches of size 128. The dropout rate is selected from the set {0.0, 0.1, 0.2, 0.3}.

Metrics. Accuracy and Macro F1-score (MF1s) [18] are chosen to evaluate the system performance. Accuracy illustrates the percentage of correct classifications over all the data instances. MF1s specifically considers the performance concerning individual digit classes and takes into account class imbalance by weighting equivalently the results from different classes before aggregation.

4.2 Impact of the finger height

Because the light emitted from the transparent OLED screen to the finger will be diffusely reflected to the photodiode array by the finger, the height of the finger will determine the range of diffuse light. In order to explore the best diffuse reflection range, we experimented with the height of the finger. In order to find the best finger height, we increase 5 mm from close to the screen (0 mm) until we find the finger height with the highest performance. An implementation detail is that we find a fixed participant to test the performance under Within-Subjects and Cold-Start scenarios with different heights of fingers. Note that we didn't use the Between-Subjects scenario because there are too few people to do cross validation. After taking the models under these two scenarios to train and adjust parameters respectively, the final results are shown in Table 1. We can clearly observe that the accuracy of Cold-Start and Within-Subjects scenarios increases with the rising of the finger height before 10 mm; After 10 mm, the accuracy will drop suddenly. This is mainly because if the finger height becomes too high, the range of light reflected by the finger will increase, which will lead to the main features of air-writing digits becoming more and more blurred. In this case, we use 10 mm as our default finger height.

4.3 Performance in all scenarios

We train the model and adjust the hyperparameters for each scenario, and the final results are shown in the Table 2. First, by observing the performance of the model under the Cold-Start scenario, it has achieved about 61% accuracy and MF1 score. Although this value is not particularly high, considering the huge difference between our main training dataset (blocked light dataset) and test

Table 1: Performance of ConvRNN under different finger's heights (averaged over 10 runs in each configuration result).

Height	Cold Start		Within Sub.	
	Accuracy	MF1s	Accuracy	MF1s
0 mm	37.91%	0.3252	81.67%	0.7658
5 mm	56.61%	0.529	80.36%	0.7974
10 mm	60.92%	0.6037	91.41%	0.9123
15 mm	28.36%	0.2664	58.21%	0.5859

Table 2: Performance of ConvRNN in three scenarios (averaged over 10 runs in each configuration result).

Scenarios	ConvRNN		
	Accuracy	MF1s	#Parameters
Cold Start	60.92%	0.6037	1257482
Within Sub.	91.41%	0.9123	1157386
Between Sub.	72.68%	0.7030	1257482

dataset (reflected light dataset), we can prove the utility of the blocked light dataset. Then taking a look at the model in the Within-Subjects scenario, and the accuracy has reached more than 91%, which verifies the effectiveness of our model in processing this kind of time series data. The performance of the Between-Subjects scenario is between the former two scenarios, about 72%. We did error analysis by generating a confusion matrix and dynamic graphs of data and found that although we chose the best height of fingers as much as possible, the light emitted from the transparent OLED screen would still be diffused to many photodiodes, resulting in an increase in the misclassification rate. We leave this problem to be solved in the future.

As for parameter quantity, Cold-Start and Between-Subjects scenarios share the same set of hyperparameters, and the Within-Subjects scenario has fewer CNN filters. This proves that the huge variation between datasets leads to more filters to learn more abstract features.

4.4 Effectiveness of model compression

To evaluate the effectiveness of knowledge distillation skill on the customized deep learning model, we gradually reduce the number of parameters of the layer which utilize plenty of hidden units. In the model, since the GRU layer usually applies most neural units to store previous information, we mainly reduce the weight of the model by adjusting the number of neural units of the GRU layer.

Figure 7 shows the comparison of the accuracy of the model after distillation and that of the non-distillation under the same amount of parameters. Note that in the performance of the previous Subsection 4.3, the well-trained models have 512 hidden units, so the number of hidden units in this subsection is reduced from 256 to 16, halved in each step. The corresponding parameter quantities are shown in Table 3. By looking at the distillation performance of Cold-Start and Between-Subjects scenarios, their performance is maintained in a stable state when the number of hidden units is bigger or equal to 128. This can be well explained by observing the data

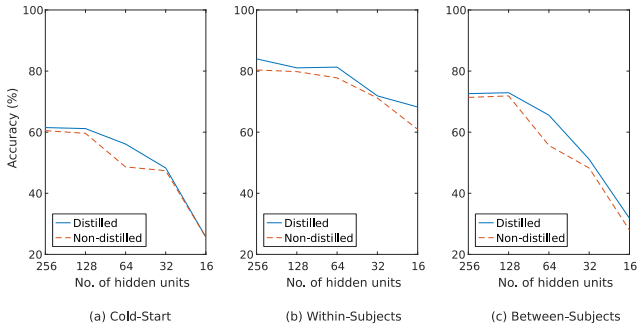

Figure 7: Performance of the customized ConvRNN: distilled vs. non-distilled. The number of parameters under different numbers of hidden units is shown in Table 3.

Table 3: The number of parameters (No. params) for different number of hidden units (No. units) in ConvRNN.

No. units	256	128	64	32	16
Cold Start	436746	173834	79242	41162	24426
Within Sub.	385802	147466	65162	33226	19562
Between Sub.	436746	173834	79242	41162	24426

compositions in three scenarios. There is bare or no reflected light data in the training datasets of Cold-Start and Between-Subjects scenarios. In this case, almost all the information obtained by the models are from the blocked light dataset. This can easily cause the model to learn a lot of less valuable information, so the performance of the model will not fluctuate after removing this less valuable information. If we sacrifice 10% performance, we find that the number of hidden units will be reduced to 64, and about 94% of the parameters will be reduced in three scenarios. The reduction for the Within-Subjects scenario is completely acceptable because of its high accuracy. For the other two scenarios, we prefer to select 128 hidden neural units, which will reduce the number of parameters by 86%.

5 CONCLUSION

In this work, we designed a through-screen air-writing digit recognition system based on reflected light. The main working principle is that the transparent OLED screen emits light as the light source, and then part of the reflected light from the near-range finger will be perceived by the under-screen photodiode array as input to the ConvRNN deep learning model to do the digit recognition. We utilize the existing blocked light dataset *LightDigit* to train the proposed ConvRNN model and follow our proposed data preprocessing procedure to make collected data suitable for our test set. Finally, we validated the system performance by carrying out experiments and performing analyses in different scenarios. Also, the effectiveness of knowledge distillation in model compression is verified.

ACKNOWLEDGMENTS

This work has been funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement ENLIGHTEM No. 814215.

REFERENCES

[1] Md. Shahinur Alam, Ki-Chul Kwon, Md. Ashraful Alam, Mohammed Y. Abbass, Shariar Md Imtiaz, and Nam Kim. 2020. Trajectory-Based Air-Writing Recognition Using Deep Neural Network and Depth Sensor. *Sensors* (2020).

[2] Jimmy Ba and Rich Caruana. 2014. Do Deep Nets Really Need to be Deep?. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[4] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[5] Li Deng. 2012. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine* (2012), 141–142.

[6] Haihan Duan, Miao Huang, Yanbing Yang, Jie Hao, and Liangyin Chen. 2020. Ambient light based hand gesture recognition enabled by recurrent neural network. *IEEE Access* (2020), 7303–7312.

[7] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2016. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems* 28, 10 (2016), 2222–2232.

[8] TU Delft ENS Group. 2021. Technical Report: LightDigit Dataset. (2021). https://www.dropbox.com/s/lbllt66mnunj22g/TR_LightDigit_dataset.pdf?dl=0

[9] Song Han, Huizi Mao, and William J. Dally. 2016. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *In Proceeding of the IEEE CVPR*. 770–778.

[11] S. Kitamura, K. Oka, K. Ikutomo, Y. Kimura, and Y. Taniguchi. 2008. A distinction method for fruit of sweet pepper using reflection of LED light. In *2008 SICE Annual Conference*. 491–494.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (2012), 1097–1105.

[13] Tianxing Li and Xia Zhou. 2018. Battery-free eye tracker on glasses. In *Proceedings of ACM MobiCom*.

[14] Yichen Li, Tianxing Li, Ruchir A Patel, Xing-Dong Yang, and Xia Zhou. 2018. Self-powered gesture recognition with ambient light. In *Proceedings of ACM UIST*. 595–608.

[15] Agnes Lydia and Sagayaraj Francis. 2019. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci* 6, 5 (2019).

[16] Larry R Medsker and LC Jain. 2001. Recurrent neural networks. *Design and Applications* (2001), 64–67.

[17] Sreeram V Menon and Chandra Sekhar Seelamantula. 2014. Robust Savitzky-Golay filters. In *Proceedings of IEEE ICDSP*. 688–693.

[18] Yiming Yang Yiming. 1999. An evaluation of statistical approaches to text categorization. In *Journal of information Retrieval*. Citeseer.