Sensor signal processing

ImAiR: Airwriting Recognition Framework Using Image Representation of IMU Signals

Ayush Tripathi[1], Arnab Kumar Mondal[2], Lalan Kumar[1,3*], and A. P. Prathosh[4]

[1] *Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi 110016, India*
[2] *Amar Nath and Shashi Khosla School of Information Technology, Indian Institute of Technology Delhi, New Delhi 110016, India*
[3] *Bharti School of Telecommunication, Indian Institute of Technology Delhi, New Delhi 110016, India*
[4] *Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560 012, India*
[*] *Member, IEEE*

Manuscript received 10 May 2022; revised 4 July 2022; accepted 8 September 2022. Date of publication 13 September 2022; date of current version 22 September 2022.

Abstract—The problem of airwriting recognition is focused on identifying letters written by movement of finger in free space. It is a type of gesture recognition where the dictionary corresponds to letters in a specific language. In particular, airwriting recognition using sensor data from wrist-worn devices can be used as a medium of user input for applications in human–computer interaction (HCI). Recognition of in-air trajectories using such wrist-worn devices is limited in literature and forms the basis of the current work. In this letter, we propose an airwriting recognition framework by first encoding the time-series data obtained from a wearable inertial measurement unit (IMU) on the wrist as images and then utilizing deep learning-based models for identifying the written alphabets. The signals recorded from 3-axis accelerometer and gyroscope in IMU are encoded as images using different techniques such as self-similarity matrix (SSM), Gramian angular field (GAF), and Markov transition field (MTF) to form two sets of 3-channel images. These are then fed to two separate classification models, and letter prediction is made based on an average of the class conditional probabilities obtained from the two models. Several standard model architectures for image classification such as variants of ResNet, DenseNet, VGGNet, AlexNet, and GoogleNet have been utilized. Experiments performed on two publicly available datasets demonstrate the efficacy of the proposed strategy. The code for our implementation will be made available at https://github.com/ayushayt/ImAiR.

Index Terms—Sensor signal processing, accelerometer, airwriting, Gramian angular field (GAF), gyroscope, inertial measurement unit (IMU), Markov transition field (MTF), self-similarity matrix (SSM), smart-band, wearables.

I. INTRODUCTION

A. Background

The process of writing letters in open space with unrestrained finger movements is known as airwriting [1]. It can be utilized to provide a user with a rapid and touch-free input alternative that can be used in human–computer interaction applications [2]. Over the last few years, the challenge of recognizing writing from motion sensors has received significant attention, and a variety of techniques have been developed for the task [3], [4], [5], [6]. The studies employing motion sensors can be separated into two categories: dedicated devices, such as a wearable glove [7], Wii remote [8], [9], smartphone [10], and vision-based studies [11]. However, these solutions require the user to carry an additional physical device, which may be inconvenient for the users. To counteract this, the second category of techniques make use of wearable gadgets such as a ring worn on the index finger [12] and smartbands [13], [14], [15] to recognize airwriting.

B. Related Work

Several attempts have been made to recognize palm movements using wrist-worn devices [16], [17]. However, most of the research in this field has focused on the scenario in which a stable and flat surface is employed during the writing process, resulting in hand stabilization. This results in reduced noise because of presence of visual and haptic feedback to the user. In this work, airwriting recognition using a wrist-worn device is addressed. However, it is to be noted that majority of the movement during writing process is contributed from the user's palm, whereas the wearable is present on the wrist. This makes the task of airwriting recognition from wrist-worn devices challenging. The problem of airwriting recognition using a wrist-worn device was first addressed in [18]. However, the work considered only a single subject with dynamic time warping (DTW) as distance measure for classification, thus, making it to be user dependent. A convolutional neural network (CNN)-based user-independent framework was proposed in [13], for detecting airwritten English uppercase alphabets. In [15], the authors explored several techniques for interpolating signals to make them of fix-length in addition to a 2-D-CNN-based classification scheme. All these studies make use of single stage classification with cross-entropy loss minimization. A 2-stage classification scheme based on supervised contrastive loss was suggested in [14] which yielded state-of-the-art airwriting recognition performance on an in-house and a publicly available airwriting dataset. All the studies performed hitherto have been focused on directly using time-series signals. All the studies related to airwriting recognition using motion sensor data have been focused on using the time series directly for classification using different deep learning models. Due to the scarcity of pretrained architectures for time-series, there have not been any attempts in exploring transfer learning techniques for the task. By encoding the time-series to image

Corresponding author: Ayush Tripathi (e-mail: eez208477@ee.iitd.ac.in).
Associate Editor: G. Langfelder.
This work did not involve human subjects or animals in its research.
Digital Object Identifier 10.1109/LSENS.2022.3206307

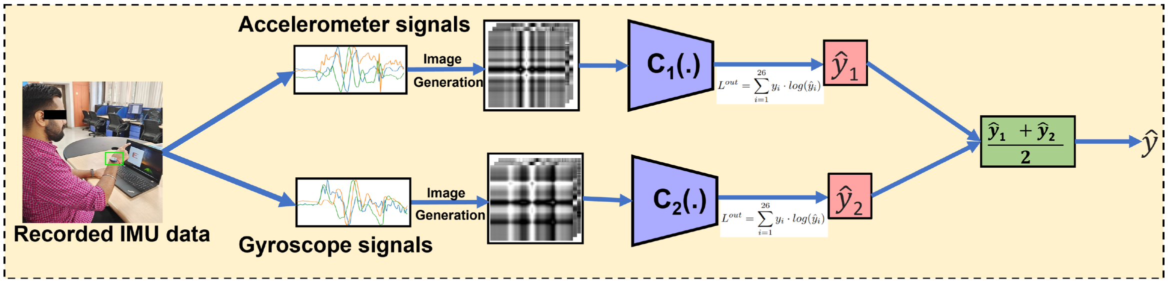
[redacted]
Accelerometer signals
Image Generation
$C_1(.)$
$L^{out} = \sum_{i=1}^{26} y_i \cdot log(\hat{y}_i)$
$\hat{y}_1$
Recorded IMU data
Gyroscope signals
Image Generation
$C_2(.)$
$L^{out} = \sum_{i=1}^{26} y_i \cdot log(\hat{y}_i)$
$\hat{y}_2$
$\frac{\hat{y}_1 + \hat{y}_2}{2}$
$\hat{y}$

Fig. 1. Block diagram depicting the proposed method.

representations, the powerful 2-D-CNN models pretrained on the ImageNet dataset can be utilized. These enable learning the complex attributes present in the encoded image representation and form the core idea of the current work.

C. Objectives and Contributions

In this letter, an airwriting recognition framework using image representation of time-series data is proposed for the first time. The time-series data is collected through 3-axis accelerometer and gyroscope sensors from an inertial measurement unit (IMU). Given that the airwriting recognition task requires the identification of fine-grained movements of the wrist to be identified, the sensitivity of IMU plays a huge role in this task. The required sensitivity is readily available in off-the-shelf smart bands and can therefore be used for HCI applications. By encoding the time-series to image representation, the capabilities of CNN were used to extract more diversified characteristics from complicated local patterns in the images. Given this intuition, the proposed framework consist of encoding time series data from accelerometer and gyroscope into two separate set of images. Then, a 2-D-CNN-based classifier is trained by using individual feature vectors, separately. The posterior probabilities from the two classifiers for each class are averaged out, and the class with maximum probability is chosen as the predicted alphabet. We validate the efficacy of the proposed approach by performing experiments on two different publicly available datasets [13], [14]. For the purpose of image encoding, several techniques—self-similarity matrix, Gramian angular field, and Markov transition field—have been explored, and their suitability for airwriting recognition has been evaluated. Various standard architectures from the image domain—ResNet [19], DenseNet [20], VGGNet [21], GoogleNet [22], and AlexNet [23]— and their variants have been explored for the classifier.

II. PROPOSED METHOD

The ImAiR framework proposed in this article starts with the acquisition of data using IMU sensors during the process of airwriting as shown in Fig. 1. The signals from 3-axis accelerometer and gyroscope are then separated and encoded into images using different image encoding techniques to form two sets of 3-channel images corresponding to the accelerometer and gyroscope respectively. These encoded images are then fed into two separate CNN-based classification models for classifying the alphabet. Finally, the posterior probabilities corresponding to the different classes from both the classifiers are averaged and alphabet prediction is made based on this average probability score.

A. Problem Description

A multivariate time-series recorded using 3-axis accelerometer and gyroscope of an IMU placed on the wrist of the dominant hand of a human subject while writing uppercase English alphabets forms the input for the method. The input can be represented as a $L \times 6$ matrix X, where L is the number of time steps and six is the number of sensor axes. Each individual time-series $x_j (j \in \{1, \ldots, 6\})$ is then encoded into image representation by using the function $f(.) : x_j \rightarrow \mathcal{I}_j$. This results in a $L \times L \times 6$ dimensional representation $\mathcal{I}$ of the input data X. The image representation can be divided into two parts—$\mathcal{I}_A$ and $\mathcal{I}_G$ corresponding to the accelerometer and gyroscope, respectively, each of the two having a dimension of $L \times L \times 3$. The classifier network, $C_1(.)$ maps the input $\mathcal{I}_A$ to a 26-D vector representing the probabilistic distribution $P(\hat{y}_A = c_i|\mathcal{I}_A; \Theta_A)$. Here, $\hat{y}_A$ is the predicted class from classifier C_1, $c_i = \{A, B, \ldots., Z\}$ is the set of classes corresponding to each of the 26 alphabets, and Θ_A is the set of parameters for the classifier. Similarly, classifier network, $C_2(.)$ maps the input from the gyroscope $\mathcal{I}_G$ to a probabilistic distribution $P(\hat{y}_G = c_i|\mathcal{I}_G; \Theta_G)$. Subsequently, we compute the average class probabilities as

$$P(\hat{y} = c_i|\mathcal{I}_A, \mathcal{I}_G) = \frac{P(\hat{y}_A = c_i|\mathcal{I}_A; \Theta_A) + P(\hat{y}_G = c_i|\mathcal{I}_G; \Theta_G)}{2}. \quad (1)$$

Based on the computed average probability, the input X is assigned to class $\hat{y} = \arg\max_{c_i} P(\hat{y} = c_i|\mathcal{I}_A, \mathcal{I}_G)$.

B. Image Encoding Methodologies

There can be various choices of the time-series to image encoding techniques and classifier architectures that can be adopted without any constraints.

1) Self Similarity Matrix: Self-similarity matrix (SSM) is a graphical representation of similar sequences in a time-series. For a time-series $V = \{v_1, v_2, \ldots., v_L\}$ of length L, the entries of SSM are computed as

$$SSM(i, j) = f(v_i, v_j) \quad ; i, j \in \{1, 2, \ldots., L\}. \quad (2)$$

Here, $f(v_i, v_j)$ is a measure of similarity the two points v_i and v_j. In our work, we use Euclidean distance as the similarity measure.

2) Gramian Angular Field: Granular angular field (GAF) [24] is a method to encode time series to an image by using polar coordinate-based matrix, while still preserving the inherent temporal relationships in the time-series. For $V = \{v_1, v_2, \ldots., v_L\}$, a time-series of length L, the first step is to rescale the elements of V so that they fall within the range of $[-1, 1]$. The rescaled time-series is further converted to polar coordinates as $\theta_i = arccos(\tilde{v}_i)$ and $r_i = \frac{t_i}{L}$.

Here, t_i is the timestamp and the division by L is done in order to regularize the span in the polar coordinate system. Rescaling the time-series ensures that there is a proper inverse mapping from (r, θ) to $(\tilde{v}, t)$ system. Subsequently, temporal correlation between samples at different timestamps can be identified by using either trigonometric sum or difference of angles (θ). Gramian angular summation field (GASF) is defined by using the cosine of sum of angles as

$$GASF = \begin{bmatrix} \cos(\theta_1+\theta_1) & \dots & \cos(\theta_1+\theta_L) \\ \cos(\theta_2+\theta_1) & \dots & \cos(\theta_2+\theta_L) \\ \vdots & \ddots & \vdots \\ \cos(\theta_L+\theta_1) & \dots & \cos(\theta_L+\theta_L) \end{bmatrix}. \quad (3)$$

Similarly, Gramian angular difference field (GADF) can defined by using the sine of difference of angles as

$$GADF = \begin{bmatrix} \sin(\theta_1-\theta_1) & \dots & \sin(\theta_1-\theta_L) \\ \sin(\theta_2-\theta_1) & \dots & \sin(\theta_2-\theta_L) \\ \vdots & \ddots & \vdots \\ \sin(\theta_L-\theta_1) & \dots & \sin(\theta_L-\theta_L) \end{bmatrix}. \quad (4)$$

3) Markov Transition Field: Markov transition field (MTF) [24] is another technique to encode the dynamic transition statistics of a time series by representing the Markov transition probabilities in a sequential manner. The first step in generating MTF is to sample elements of a time-series $V = \{v_1, v_2, \dots, v_L\}$ into Q bins. Further, each sample v_i is assigned to its corresponding bin q_j ($j \in [1, Q]$). Using this mapping, a $Q \times Q$ dimensional adjacency matrix W is constructed, the elements w_{ij} of which are given by the frequency with which a point in bin q_j is followed by a point in bin q_i. The elements of W are then used to construct the MTF matrix as

$$MTF = \begin{bmatrix} w_{ij|v_1\in q_i, v_1\in q_j} & \cdots & w_{ij|v_1\in q_i, v_L\in q_j} \\ w_{ij|v_2\in q_i, v_1\in q_j} & \cdots & w_{ij|v_2\in q_i, v_L\in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|v_L\in q_i, v_1\in q_j} & \cdots & w_{ij|v_L\in q_i, v_L\in q_j} \end{bmatrix}. \quad (5)$$

C. Model Architecture

We explored various different standard architectures for the classifier block: ResNet [19] (18, 34, 50, 101, and 152 layer variants), DenseNet [20] (121, 161, 169, and 201 variants), VGGNet [21] (with the number of weight layers equal to 11, 13, 16, and 19; with and without batch normalization), GoogleNet [22], and AlexNet [23]. The parameters of the model were initialized by using pre-trained weights from the ImageNet classification task and are optimized by minimizing the cross-entropy loss using the Adam optimizer.

III. EXPERIMENTS AND RESULTS

A. Datasets

1) Dataset-1: We used the publicly available dataset from [13] that consists of recordings obtained from 55 subjects while writing English uppercase letters (15 repetitions). Data were recorded using a Microsoft band 2 at the maximum sampling rate supported by the device of 62 Hz.
2) Dataset-2: We used another publicly available dataset [14] which comprised recordings obtained from 20 subjects during ten repetitions of writing English uppercase alphabets. Signals were recorded using a Noraxon Ultium EMG sensor (with an internal IMU) [25] placed on the wrist. The sampling rate was 200 Hz and the signals were downsampled to 62 Hz. Performance details of the used IMU are presented in Table 1.

Table 1. Performance Details of the Used IMU in Dataset-2

	Range	Sensitivity
Accelerometer	±16g	2048 LSB/g
Gyroscope	±2000 degree/sec	16.4 LSB/(degree/second)

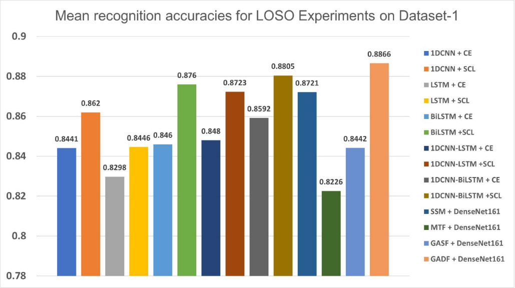

Fig. 2. Recognition accuracies for LOSO experiment on Dataset-1.

B. Experimental Details

Similar to previous studies [13], [14], we applied a two-step preprocessing strategy to the recorded IMU signals. First, the length of each signal was made fixed (to 155) by discarding samples if the signal was of greater length or padding zeros otherwise. Subsequently, Z-normalization was applied to each of the six signals individually.

In the first set of experiments, we evaluated the performance of combination of different image encoding techniques and classification models for the task of airwriting recognition on Dataset-1. For this purpose, we split the data from 55 subjects into two parts—training data comprising of recordings from 40 subjects (40*26*15 = 15 600 samples) and test set from remaining 15 subjects(15*26*15 = 5850 samples). Such a split ensures that the evaluation is performed in a user independent manner. The training data were further divided into a training set and a validation set having 80:20 ratio. A mini-batch training process was employed while keeping the batch size equal to 32 and early stopping with a patience of ten epochs. In the next set of experiments, we performed leave-one-subject-out (LOSO) validation on both Dataset-1 and Dataset-2. We used the DenseNet161 model architecture for classification using all four image encoding techniques: SSM, MTF, GASF, and GADF. The performance of the proposed ImAiR framework in this setting is compared with state-of-the-art performance on both the datasets.

C. Results and Comparison

Table 2 lists the performance of different model architectures and image encoding techniques for the first set of experiments on Dataset-1. The results of the two individual classifiers [using argmax over $P(\hat{y}_A = c_i|\mathcal{I}_A; \Theta_A)$, $P(\hat{y}_G = c_i|\mathcal{I}_G; \Theta_G)$] and their average ($P(\hat{y} = c_i|\mathcal{I}_A, \mathcal{I}_G)$) for accelerometer and gyroscope-based classification using different image encoding techniques are presented. It may be noted that DenseNet161 classifier provides the best recognition accuracy in all the scenarios. Among the different encoding methods, the performance of GADF is superior for all different classifiers. The best performance is obtained on using the GADF+DenseNet161 combination and the achieved accuracy is 88.43%.

In Figs. 2 and 3, the results for LOSO experiments performed on Dataset-1 and Dataset-2 are presented and compared with state-of-the-art techniques that include different deep learning architectures (1DCNN, LSTM, BiLSTM, 1-DCNN-LSTM, and 1-DCNN-BiLSTM) with cross entropy and supervised contrastive loss-based

Table 2. Recognition Accuracies Using Different Image Encoding Techniques and Model Architecture Combinations on Dataset-1

	SSM			MTF			GASF			GADF		
Architecture	**Accelerometer**	**Gyroscope**	**Fusion**	**Accelerometer**	**Gyroscope**	**Fusion**	**Accelerometer**	**Gyroscope**	**Fusion**	**Accelerometer**	**Gyroscope**	**Fusion**
ResNet18	0.7648	0.7933	0.8241	0.7051	0.7308	0.7711	0.7485	0.7315	0.7923	0.8099	0.8537	0.8730
ResNet34	0.8065	0.8032	0.8419	0.6937	0.7417	0.7754	0.7477	0.7626	0.8138	0.7995	0.8561	0.8663
ResNet50	0.7850	0.8118	0.8342	0.6942	0.7328	0.7713	0.7393	0.7521	0.8027	0.8152	0.8609	0.8779
ResNet101	0.8133	0.7995	0.8439	0.7014	0.7552	0.7832	0.7641	0.7309	0.8128	0.8115	0.8653	0.8776
ResNet152	0.8118	0.7974	0.8484	0.6954	0.7685	0.7894	0.7622	0.7480	0.8089	0.8010	0.8547	0.8756
VGG11	0.7421	0.7650	0.8080	0.6458	0.7026	0.7395	0.6969	0.7070	0.7689	0.7769	0.8308	0.8456
VGG11-BN	0.7776	0.7887	0.8299	0.6757	0.7144	0.7576	0.7150	0.7320	0.7856	0.7971	0.8335	0.8569
VGG13	0.7578	0.7756	0.8226	0.6619	0.7205	0.7516	0.7145	0.7070	0.7689	0.7824	0.8263	0.8496
VGG13-BN	0.7651	0.8005	0.8330	0.6802	0.7468	0.7778	0.7190	0.7296	0.7884	0.7779	0.8385	0.8542
VGG16	0.7704	0.7952	0.8326	0.6968	0.7248	0.7703	0.7209	0.7270	0.7868	0.7976	0.8362	0.8597
VGG16-BN	0.8089	0.8048	0.8489	0.7397	0.7692	**0.8046**	0.7526	0.7549	0.8056	0.8106	0.8439	0.8653
VGG19	0.7858	0.7809	0.8315	0.7099	0.7556	0.7959	0.7332	0.7197	0.7858	0.8034	0.8304	0.8586
VGG19-BN	0.7834	0.8080	0.8429	0.7142	0.7617	0.7971	0.7535	0.7484	0.8094	0.8212	0.8289	0.8650
DenseNet121	0.8120	0.7937	0.8441	0.7036	0.7244	0.7723	0.7443	0.7651	0.8092	0.8250	0.8451	0.8711
DenseNet161	0.8238	0.8099	**0.8526**	0.7462	0.7438	0.7903	0.7785	0.7638	**0.8215**	0.8379	0.8621	**0.8843**
DenseNet169	0.7795	0.8287	0.8455	0.7149	0.7362	0.7778	0.7412	0.7465	0.8067	0.8294	0.8617	0.8778
DenseNet201	0.8048	0.8039	0.8479	0.6915	0.7701	0.7892	0.7773	0.7569	0.8138	0.8316	0.8414	0.8725
AlexNet	0.6757	0.7041	0.7521	0.5038	0.5494	0.6044	0.5581	0.6154	0.6631	0.7176	0.7834	0.8072
GoogleNet	0.7791	0.7877	0.8296	0.6472	0.6629	0.7149	0.7342	0.7055	0.7783	0.8034	0.8559	0.8600

Entries in blue denote accuracy greater than 85% and bold entries denote the best achieved accuracy for each image encoding technique.

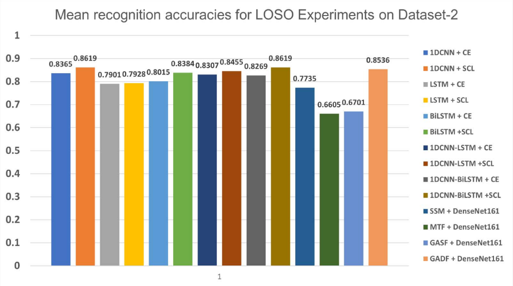

Fig. 3. Recognition accuracies for LOSO experiment on Dataset-2.

techniques. Performing the experiments in a LOSO setting allows us to directly compare our results with those of [13] and [14]. It is observed that on Dataset-1, by using the GADF+DenseNet161 combination, we achieve an accuracy of 88.66%, which is the best reported accuracy on the given dataset. On Dataset-2, the best accuracy achieved is 85.36%, which is comparable to state-of-the-art. The dip in mean recognition accuracy on Dataset-2 in comparison to Dataset-1 may be attributed to the small amount of samples (4940 samples) available for training the model which may not be sufficient to learn the complex attributes present in the encoded image representation.

IV. CONCLUSION

In this letter, we explored an airwriting recognition framework by using encoded image representation of time-series data from accelerometer and gyroscope sensors in a wrist-worn IMU. It is seen that the proposed approach outperforms the state-of the-art accuracy on a publicly available dataset [13], while also performing significantly well on a another dataset [14]. Future work may be focused on exploring techniques that are well-suited to learn the attributes present in the signals from small datasets to increase the utility of the airwriting recognition framework for HCI applications.

ACKNOWLEDGMENT

This work was supported by Prime Minister's Research Fellows (PMRF) Scheme PLN08/PMRF, Ministry of Education, Government of India (GoI).

REFERENCES

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

[2] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man, Cybern., Part C. (Appl. Rev.)*, vol. 37, no. 3, pp. 311–324, May 2007.

[3] C. Amma, M. Georgi, and T. Schultz, "Airwriting: A wearable handwriting recognition system," *Pers. Ubiquitous Comput.*, vol. 18, no. 1, pp. 191–203, 2014.

[4] D. Lu, K. Xu, and D. Huang, "A data driven in-air-handwriting biometric authentication system," in *Proc. IEEE Int. Joint Conf. Biometrics*, 2017, pp. 531–537.

[5] S. Patil, D. Kim, S. Park, and Y. Chai, "Handwriting recognition in free space using WIMU-based hand motion analysis," *J. Sensors*, vol. 2016, pp. 212–221, 2016.

[6] 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.

[7] C. Amma and T. Schultz, "Airwriting: Bringing text entry to wearable computers," *XRDS: Crossroads, ACM Mag. Students*, vol. 20, no. 2, pp. 50–55, Dec. 2013. [Online]. Available: https://doi.org/10.1145/2540048

[8] Y. Li, H. Zheng, H. Zhu, H. Ai, and X. Dong, "Cross-people mobile-phone based airwriting character recognition," in *Proc. IEEE 25th Int. Conf. Pattern Recognit.*, 2021, pp. 3027–3033.

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

[10] C. Li, C. Xie, B. Zhang, C. Chen, and J. Han, "Deep fisher discriminant learning for mobile hand gesture recognition," *Pattern Recognit.*, vol. 77, pp. 276–288, 2018.

[11] P. Roy, S. Ghosh, and U. Pal, "A CNN based framework for unistroke numeral recognition in air-writing," in *Proc. IEEE 16th Int. Conf. Front. Handwriting Recognit.*, 2018, pp. 404–409.

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

[13] T. Yanay and E. Shmueli, "Air-writing recognition using smart-bands," *Pervasive Mobile Comput.*, vol. 66, 2020, Art. no. 101183.

[14] A. Tripathi, A. K. Mondal, L. Kumar, and P. AP, "SCLAiR: Supervised contrastive learning for user and device independent airwriting recognition," *IEEE Sens. Lett.*, vol. 6, no. 2, pp. 1–4, Feb. 2022.

[15] F. A. Abir et al., "Deep learning based air-writing recognition with the choice of proper interpolation technique," *Sensors*, vol. 21, no. 24, 2021, Art. no. 8407.

[16] A. Levy, B. Nassi, Y. Elovici, and E. Shmueli, "Handwritten signature verification using wrist-worn devices," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 1–26, Sep. 2018.

[17] H. Wen, J. Ramos Rojas, and A. K. Dey, "Serendipity: Finger gesture recognition using an off-the-shelf smartwatch," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2016, pp. 3847–3851.

[18] 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.

[19] 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.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[22] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[23] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," 2014, *arXiv:1404.5997*.

[24] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3939–3945.

[25] "Noraxon ultium EMG," Accessed: Sep. 10, 2021. [Online]. Available: https://www.noraxon.com/our-products/ultium-emg/