mmDigit: A Real-Time Digit Recognition Framework in Air-Writing Using FMCW Radar

Jiake Tian, Yi Zou, *Senior Member, IEEE*, Jiale Lai, Fangming Liu, *Senior Member, IEEE*

***Abstract*—Millimeter-wave (mmWave) radar sensors show significant promise in non-contact human-computer interaction. Using air-writing as a substitute for conventional input devices such as keyboards and mice has become a pivotal topic in contemporary research. In response to the absence of a dataset in existing studies about air-writing digits and the insufficient exploration of real-time recognition in edge devices, we propose a real-time air-writing digit recognition framework based on mmWave radar, termed mmDigit. Initially, we use mmWave radar equipped with Frequency-Modulated Continuous Wave (FMCW) technology to collect digital echo data and design a data processing pipeline to track and reconstruct digital trajectory images. These images are subsequently fed into a lightweight neural network, which is only 6.9 K in parameter size, for exploring the images' quality and the recognition and cross-user capabilities of small-scale air-writing datasets. To enhance mmDigit's performance, we implement a transfer learning strategy to accommodate a broader range of digit writing styles and habits, achieving a recognition accuracy of 99.14% and a cross-user capability of 94.13%. Additionally, applying a knowledge distillation strategy enables the lightweight network to extract and learn deep-layer features, thereby improving the cross-user recognition accuracy to 96.22%.**

***Index Terms*—MmWave FMCW Radar, Air-Writing, Digit Recognition, Transfer Learning, Knowledge Distillation.**

I. Introduction

THE advancement of the Internet of Things (IoT) and Artificial Intelligence (AI) have created innovative approaches for humans to transition from conventional keyboard and mouse interactions to non-contact methods for communicating intentions. The significant transformation marks the dawn of a new era in Human-Computer Interaction (HCI) [1], [2]. Hand gestures, complement verbal communication, enhance emphasis, and facilitate emotional expression. Consequently, they have emerged as a significant area of research in the field of HCI [3], significantly expanding the manner of interfacing with machines [4]. The application of hand gestures shows excellent potential in both personal and commercial scenes [5], [6], [7], [8], [9], such as the domain of smart homes [7], where distinct hand gestures can be employed to operate various household devices (e.g., lighting systems, and audio equipment).

Traditional gestures and hand movements, such as simple circular motions or waving, can often open to ambiguous interpretations due to individual variations in execution. Moreover, differing social norms, values, and beliefs across countries and regions can result in identical actions conveying disparate meanings and being utilized in diverse manners [10], [11]. In contrast, air-writing, which conveys specific information such as numbers or letters, offers a standardized method that presents new possibilities for fostering effective cross-cultural communication.

Wearable devices, such as intelligent bracelets and data gloves, exhibit a high degree of accuracy in hand gesture recognition [12], [13], [14], [15]. However, the prohibitive cost and subpar user experience present a significant hurdle to their widespread adoption [16]. Camera-based technology has become the mainstream approach [17], [18], [19], but environmental lighting conditions and obstructions influence its effectiveness. Moreover, camera-based technology has raised concerns about personal privacy [20]. Acoustic sensors are also utilized for hand gesture recognition, demonstrating personal performance within a short-range context [21], [22] [23]. Higher power levels are necessary for long-distance hand gesture recognition, which could potentially harm human auditory health.

In light of the drawbacks inherent above, some studies focus on the application of Wi-Fi signals for air-writing [16], [24]. However, the ambient environment easily influences the longer wavelength of Wi-Fi signals, posing challenges to achieving the high precision of air-writing. In response to the poor user experience with wearable devices, privacy concerns raised by cameras, potential harm to the human body posed by acoustic sensors, and the susceptibility of WiFi to interference, some researchers have begun to explore alternative solutions. Consequently, research efforts have shifted towards radar sensors. Compared to pulse radar, continuous wave radar, and phased array radar, Ultra-WideBand (UWB) radar has significant resolution and penetration capability advantages. In contrast, millimeter-Wave (mmWave) Frequency-Modulated Continuous Wave (FMCW) radar excels in distance and speed measurement. These attributes position them as crucial sensors in the field of air-writing [25]. This paper uses the term FMCW

This work was supported in part by the SCUT Research Startup Fund No.K3200890, in part by the Guangzhou Huangpu District International Research Collaboration Fund No.2022GH13, in part by the Major Key Project of PCL under Grant PCL2024A06, in part by the Shenzhen Science and Technology Program under Grant RCJC20231211085918010, in part by the Major Key Project of PCL under Grant PCL2022A05, and in part by the National Key Research & Development (R&D) Plan under Grant 2022YFB4501703. This article was presented in part at the 21st IEEE International Conference on Machine Learning and Applications (ICMLA2022), Atlantis Hotel, Bahamas (Corresponding authors: Yi Zou). The opinions expressed here are entirely those of the author. No warranty is expressed or implied. The user assumes all risk. (Corresponding author: Yi Zou, Fangming Liu)

Jiake Tian, Yi Zou, and Jiale Lai. are with the School of Microelectronics, South China University of Technology, Guangzhou 511442, China (e-mail: mijiake@mail.scut.edu.cn; zouyi@scut.edu.cn; mijllail@mail.scut.edu.cn).

Fangming Liu is with Pengcheng Laboratory, Shenzhen 518066, China, and also with the Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: fangminghk@gmail.com).

radar interchangeably with mmWave FMCW radar.

The radar-based air-writing digit researches are presented in Table I. Ref.[20], [26], [27] detail the collection of air-writing data using multiple FMCW or UWB radar sensors, which are subsequently analyzed in conjunction with trilateration techniques to detect and localize the trajectories of air-writing digits. The subsequent phase involves the deployment of deep learning networks for the recognition of air-writing digits or letters. Some studies investigate the viability of air-writing digit recognition with either 2 or 1 FMCW radar, which can achieve recognition accuracy surpassing 91% [28], [29]. Ref. [30] reports the particular challenge of tailing in air-writing digits 4 and 5, reach an average accuracy of 97% for digit recognition by removing spurious detection and integrating Convolutional Neural Networks (CNN) classifiers with Hough transform outcomes. In a different approach, some studies abandon the use of trajectories within Cartesian coordinates, focusing instead on the extraction of time-varying patterns of hand movements [25], [31], [32], such as Range-Doppler Map (RDM), Range-Time Map (RTM), Doppler-Time Map (DTM), Amplitude-Time Map (ATM), which also yield impressive recognition accuracy.

Building upon the above research on air-writing, we identify several issues worthy of discussion:

- We note that the data acquisition system for radar sensors is progressing towards reduced complexity and lower power consumption, marked by a significant decrease in sensors and antenna configurations. However, the recognition accuracy of air-writing does experience a decline to some extent.
- The air-writing dataset in existing research is typically limited in scale, making it insufficient to effectively accommodate the diverse range of writing styles and habits, potentially leading to performance fluctuations in more extensive scenarios. However, collecting large-scale datasets entails considerable time and human resource costs that warrant careful consideration.
- Most current studies remain experimental, with only a handful achieving real-time recognition on edge devices. The application value of real-time recognition is substantial, and there are promising development prospects within the realm of HCI.

Inspired by the challenges above, we propose mmDigit, a framework for real-time recognition of air-writing digits using mmWave radar. mmDigit aligns with the trend toward developing low-complexity systems. It delves into and optimizes the challenges posed by small data scales and the dearth of real-time research, thereby augmenting the system's practicality and applicability across a broader range of scenarios. We develop a processing pipeline for millimeter-wave radar echo signals, designed to process echo data of digits 0 to 9 collected by a commercial FMCW radar. This pipeline can track and reconstruct the trajectories of air-writing digits, subsequently generating a corresponding dataset. To prevent the differences in computing power of edge devices and meet the demands of real-time identification in different edge computing environments, we develop a lightweight neural network, the size of its parameters being less than one percent of that of traditional networks, containing a mere 6.9K of parameters. By employing a lightweight network, we explore the quality of small-scale datasets on edge devices and their recognition and cross-user capabilities. We then incorporate a transfer learning strategy, amalgamating a variety of traditional neural networks and exogenous datasets to enhance the scalability of small-scale air-writing digit datasets. This enables the network to assimilate a broader spectrum of writing habits and styles. By applying knowledge distillation, we further augmented the cross-user recognition capabilities for air-writing digits, culminating in practical deployment.

The proposed mmDigit framework's performance is evaluated in three dimensions. We visually demonstrate the framework's effectiveness in processing and reconstructing air-writing digital images, showing that the mmDigit's data processing pipeline can restore visually discernible digit trajectories. Subsequently, using small-scale air-writing datasets, we explore the cross-user recognition capabilities using a diverse range of classical networks and our designed lightweight network. Experimental evidence suggests that these datasets are limited in scalability due to their inability to learn diverse writing styles and habits. Finally, by employing transfer learning and knowledge distillation strategies, the lightweight network can achieve a recognition accuracy of 99.19% and a cross-user capability of 96.22% without collecting additional data.

The following are the main contributions of this paper:

- We **present** a novel framework for real-time digit recognition in air-writing using a single FMCW radar, dubbed mmDigit. The framework integrates data collection, data processing, a lightweight recognition network, and optimization strategies, including the creation of a dataset of air-writing digits[1].
- We **investigate** the recognition accuracy and scalability of a small-scale air-writing dataset, employing transfer learning strategy and knowledge distillation strategy to enhance mmDigit's cross-user adaptability and recognition accuracy by incorporating diverse air-writing digit datasets and various scenarios.
- We **conduct** extensive experimental validation by public datasets and the dataset we created. The results show that mmDigit effectively meets the demands of expanded capabilities and real-time accuracy for no-contact human-computer interaction.

The rest of this paper is organized as follows: Section II describes the principles of FMCW radar, the radar configuration of digit collection, the design of air-writing digit, and the pipeline for digital image generation. Section III details the problem formulation, lightweight network structure, transfer learning and knowledge distillation pipeline, and algorithmic loss. Section IV describes the dataset spiting, the experimental setup, the results of the lightweight network, the transfer learning strategy, the knowledge distillation strategy, and the ablation experiment. Finally, Section V offers conclusions and discussions of this paper.

[1]Air-writing dataset: https://github.com/Tjkjjc/gesture.

TABLE I: The research of air-writing digit recognition based on FMCW and UWB radar.

Paper	Radar	Antenna	Data	Gesture	Input	Model	Accuracy	Real-Time
[26]	UWB×4	Array	3000	0-9	Image	CNN	99.90%	x
[27]	UWB×3	—	—	0-9 —	Image	CNN	99.40%	✓
[25]	UWB×1 FMCW×1	—	2000 10000	0-9	RTM, DTM	CNN	98.50%	x
[20]	FMCW×3	1T1R	1875	A-J, 1-5	Trajectory Image	ConvLSTM DCNN *	98.33% 98.33%	x
[28]	FMCW×2 FMCW×1	1T1R	3750	A-J, 1-5	Trajectory	1D DCNN-LSTM-1D transposed DCNN	97.3±2.67% 90.33±4.44%	x
[29]	FMCW×2 FMCW×1	1T1R	3750	A-J, 1-5	Trajectory	1D TCN *	99.11±0.89% 91.33±4.66%	x
[30]	FMCW×1	Array	3000	0-9	Image	CNN	97%	x
[31]	UWB×1	1T1R	1800	0-9	RDM	3D-CNN-LSTM *	98.5±1.1%	x
[32]	FMCW×1	1T2R	1200	0-9	RTM, DTM, ATM	CNN	95%	x

* LSTM is Long Short-Term Memory, DCNN is Deep Convolutional Neural Network, and TCN represents Temporal Convolutional Network.

II. Data Collection Scheme

A. How to Operate FMCW Radar

Assuming the FMCW radar's transmitted signal takes the form of a cosine wave, it can be mathematically expressed as

$$s_T(t) = \cos(\varphi(t) + \varphi_0), \tag{1}$$

where φ_0 represents the initial phase of the signal, φ represents the offset of the phase, and the sum represents the signal's instantaneous phase. The phase's offset can be expressed as

$$\varphi(t) = 2\pi \int_0^t \left(f_c + \frac{B}{T_m} x \right) \mathrm{d}x = 2\pi \left(f_c t + \frac{B}{2T_m} t^2 \right), \tag{2}$$

where f_c represents the start frequency, B represents the signal bandwidth, and T_m represents the signal period.

Assuming that the relative velocity of the detected target to the FMCW radar is v and the distance is R, the received signal can be expressed as

$$s_R(t) = s_T(t-\tau) = \cos\left(2\pi\left(f_c(t-\tau) + \frac{B}{2T_m}(t-\tau)^2\right)\right). \tag{3}$$

The time delay τ of the received signal is denoted as

$$\tau = \frac{2(R+vt)}{c}, \tag{4}$$

where c represents the speed of light.

Subsequently, the transmitted signal and the received signal are combined and subjected to a mixer and lowpass filter, and the Intermediate Frequency (IF) signal is approximately obtained as

$$s_o(t) = \cos\left(2\pi\left(\frac{2f_c v}{c} + \frac{2BR}{cT_m}\right)t + \frac{4\pi f_c R}{c}\right). \tag{5}$$

Considering the more general case, in a series of successive chirps transmitted by the radar, when the i_{th} chirp transmitted is received by the j_{th} received antenna, the IF signal at the k_{th} sample point of the Analog-to-Digital Converter (ADC) at this point can be expressed as

$$s_o(i,j,k) = \cos\left(2\pi\left(\frac{2f_c v}{c} + \frac{2B(R+vt\cdot i)}{cT_m}\right)\frac{T_m}{N}k + \frac{4\pi f_c(R+vt\cdot i)}{c} + \frac{2\pi f_c j d\sin\theta}{c}\right), \tag{6}$$

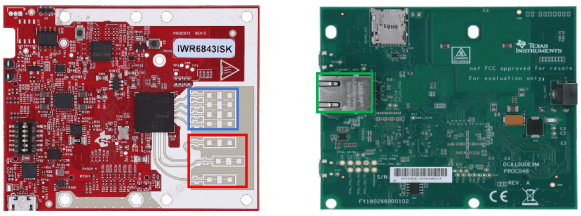
where N represents the total number of sampling points in a chirp, d represents the distance between two neighboring received antennas, and θ represents the angle of arrival. The ADC represents a snapshot of the combined frequency content and phase information, which is vital for subsequent signal processing and digit recognition algorithms.

B. How to Configure FMCW Radar

With an emphasis on cost-effectiveness, we select the TI IWR6843ISK radar to collect air-writing digit data [33]. As shown in Fig. 1(a), the FMCW radar assembly includes three transmitter antennas, four received antennas, an integrator, and an ADC, all coordinated by an external phase-synchronized oscillator imposed to control the range of frequency scanning. The radar's field of view covers a vertical angle of 30° and a horizontal angle of 120°. The radar is configured to operate at the lowest possible transmission power at less than 3 meters. The center frequency of radar is set to 62.00 GHz, with an operating bandwidth of 2.78 GHz; the antenna system operates with one transmitter and two receivers. Each radar frame has a fixed duration of 30 ms and contains 128 chirps, with each chirp having 64 sampling points, resulting in a range resolution of 5.4 cm. The DCA1000 Evaluation Module (EVM) [34] is utilized for the real-time data transmission. As depicted in Fig. 1(b), it transmits the air-writing data to a computer equipped with the mmWave studio tool through a 1 Gbps Ethernet port.

C. How to Collect Digit Data

For air-writing digit data collection, we recruit 11 users, with 6 males and 5 females. The FMCW radar is mounted



(a) IWR6843 ISK (b) DCA1000 EVM.

Fig. 1: The devices of data collection. (a) The red box denotes the transmitted antenna, while the blue represents the received one. (b) The green box indicates the network port.

on the surface of the laboratory platform and configured to transmit and receive signals vertically. Users are instructed to stand approximately 30 cm in front of the experimental platform and write single digits ranging from 0 to 9. Given the mmWave radar's 120° field of view and the average size of a human hand, the hand elevation is limited to within 5 cm and 30 cm. Users are then instructed to repeatedly move their hands across the recognition plane, tracing the shape of the ten digits.

To mitigate the trailing issues inherent in traditional air-writing, as shown in Fig. 2, we instruct users to perform the air-writing of digits 0 through 9 via a continuous writing method. This method aims to streamline the processing of digit trajectories, reducing the interference of transitional writing movements on actual writing actions, thereby enhancing the quality of air-writing data collection. For instance, the digit 5 is composed of three actions resulting in two strokes: the first step involves writing the diagonal stroke and forming the right semicircle, creating stroke 1; the second step involves moving the hand to the position above the diagonal stroke without generating a new stroke; and the third step involves writing the horizontal stroke, forming stroke 2. If adhering to traditional writing methods, the second transitional action trajectory cannot be avoided in the air, resulting in an additional upward trailing stroke for the digit 5, thereby degrading the quality of its formation.

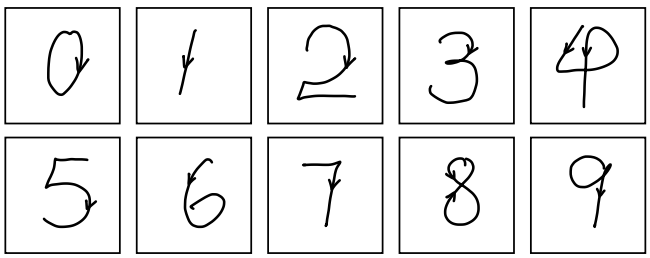


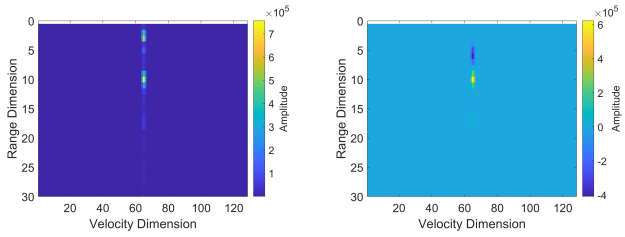
Fig. 2: The guidelines for air-writing digits 0 through 9. Demand a single-stroke approach consistent with the directional arrows indicated in the diagram.

D. How to Obtain Digit Image

Through the FMCW radar principle described in Section II-A, we can capture the form of 3-dimensional Radar Data Cubes (RDC) of air-writing digits, with dimensions corresponding to the number of ADC samples, the number of chirps, and the number of received antennas. We transform the RDC data into digital trajectory images by following these specific steps:

1) 2D FFT: The range information of the hand gesture target is initially obtained through Range FFT on the RDC data. Subsequently, a Doppler FFT is conducted along the slow time axis to derive velocity information. This results in the generation of RDM as illustrated in Fig. 3(a). The RDM constitutes a two-dimensional matrix, with rows and columns denoting range and Doppler, respectively. Each element within the RDM encapsulates the phase and amplitude of the intermediate frequency signal.

2) Clutter Removal and Peak Searching: We remove background clutter from the RDM through 2D FFT processing to mitigate the influence of stationary background clutter. As shown in Fig. 3(b), the transformation of the RDM before and after can be observed. Subsequently, by searching for the highest amplitude value in RDM, we locate the target as a tuple containing 2-dimensional indices. This peak value represents the target, i.e., the scattering center of the target. Robust reflected transmission signals from the target manifest as multiple detection peaks, recognized as the target's scattering centers.

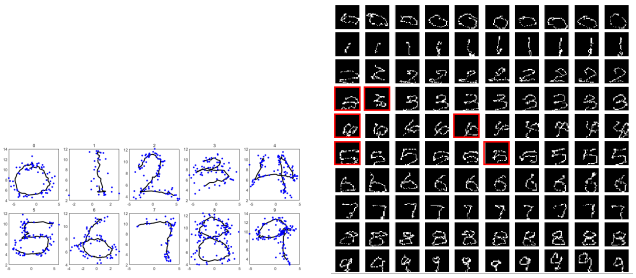


(a) RDM. (b) Clutter filtering.

Fig. 3: The outcome of background clutter elimination within the data processing pipeline. (a) RDM is generated by 2D FFT, and (b) RDM is generated after the background removal.

3) Weighted Average: Following the processing mentioned above, the range, velocity, and angular information of the scattering centers are ascertained. Our design stipulates that a critical point within each frame represents the target hand. Consequently, we determine the centroid of all scattering centers through amplitude-weighted calculations, deriving the centroid's range, velocity, and angle as the representative information for the hand gesture target in a single frame. Radar data frames with a sparse distribution of scattering centers or insufficient cumulative amplitude are classified as invalid. As depicted in Fig. 4(a), the central points of the scattering centers across all frames involved in writing a single digit are presented in a coordinate-based manner, allowing for a precise observation of the primary trajectories of air-writing digits.

4) Image Reconstruction: As mentioned above, different processing flows are carried out once we identify frames as valid versus invalid. In particular, the invalid frame is passed back through a simple recursive filter to update the background



(a) Central points. (b) Digital trajectory images.

Fig. 4: Visualization of image reconstruction within the data processing pipeline. (a) Distribution of central points in multi-frame data, (b) Air-writing digital trajectory images, in which digital 3, 4, and 5 within the red box are difficult to recognize directly by human eyes as they are written in a single stroke.

RDM. If the frame is valid, it is passed to another recursive filter to reconstruct a smoothing trajectory. The recursive filter as below,

$$r_{cur} = r_{prev} * \alpha + r * (1 - \alpha), \tag{7}$$

where r_{cur} is the result used to reconstruct the trajectory, r is the corresponding result of single frame processing and r_{prev} is the value in the previous frame. α is the proportion coefficient set to 0.8 in this paper. The angle is filtered similarly.

5) Angle Estimation: We use the following Eq. (8)-(9) to calculate the angle. Note that the angle information calculation depends on radar antenna distribution.

$$\Delta\Phi = \frac{2\pi\Delta d}{\lambda}, \tag{8}$$

$$\theta = \sin^{-1}(\frac{\lambda\Delta\Phi}{2\pi L}), \tag{9}$$

where Δd is the difference of wave path between two received antennas with L as the physical distance between the two received antennas.

To construct the continuous trajectory of the target, we convert the range r and angle θ into the Cartesian coordinate system. This is achieved by discretizing the continuous trajectory using the following normalization equation,

$$x_{pixel} = round(\frac{x - x_{min}}{x_{max} - x_{min}} * w_{img}), \tag{10}$$

where x is the horizontal position, x_{min}, x_{max} are the boundaries of the detection plane, w_{img} is the width of the trajectory image, and x_{pixel} is the row index of the image. Similarly, the column index y_{pixel} can be obtained from the vertical position y. Next, the location of the target is mapped to the trajectory image by setting the pixel in (x_{pixel}, y_{pixel}) to 1 while setting others to 0.

E. How to Build Digit Dataset

Upon completing the procedures above, this study not only transforms air-writing digits from ADC data into digital trajectory images but also ensures that the clarity and recognizability of these images meet research requirements. Fig. 4(b) presents samples of digital trajectory images for digits 0 through 9 used in this study. Notably, the trajectory images for digits 3, 4, and 5, which are completed in a single and continuous stroke during air-writing, exhibit an elegant form that may lead to confusion with other digits. However, from a global perspective, the digital trajectory images generated through the data processing pipeline maintain a high level of discernibility, enabling essential recognition even through casual observation.

To facilitate management and recognition, we establish a naming convention that creates folders based on digit categories and classifies corresponding digital trajectory images into their folders. The storage method allows researchers to quickly identify the gesture type labels represented by each image through the folder names. Moreover, this study not only focuses on image storage but also thoroughly considers the integrity and traceability of data research. Consequently, we concurrently provide the original ADC data, offering researchers a complete link from data generation to image conversion, enabling them to embark on more in-depth and comprehensive research exploration from the initial state of the data.

III. Proposed Methodology

A. Problem Formulation

The objective is to implement real-time air-writing digit recognition by leveraging a bespoke lightweight deep learning network f. Given input images $I = \{I_1, I_2, \ldots, I_n\}$, yielding a 10-dimensional vector Y that delineates the probability distribution of the image's classification among ten categories. To enhance the recognition accuracy of the proposed lightweight network, we employ a transfer learning strategy. This strategy involves pre-training the network on the MNIST dataset and then fine-tuning it on our proposed dataset. The effectiveness of the transfer learning strategy in boosting recognition performance is validated through extensive experimentation with a diverse array of classification networks, resulting in a compilation of networks $F = \{F_1, F_2, \ldots, F_n\}$. Moreover, we incorporate a knowledge distillation strategy to enhance the cross-user accuracy of air-writing digits. Selecting the few most accurate networks from F as teacher models f_t, we pain them with lightweight student networks f. Both models process image data I, with the teacher model yielding Y_t and the student model yielding Y_s. We want the student network to assimilate more deep-layer features, aiming to minimize the discrepancy between Y_t and Y_s.

B. Network Architecture

To evaluate the recognition and scalability capabilities of small-scale air-writing datasets in resource-constrained edge devices with lightweight networks, We design a lightweight network classifier for real-time air-writing digit recognition. The classifier uses depth-wise separable convolution instead of standard conventional convolutions. The depth-wise separable convolution consists of a depth convolution and a point-wise convolution (1×1 convolution), reducing the computation by a factor of k^2 at slight accuracy loss, where k means

TABLE II: Details of lightweight network. 'DW' is a depth-wise convolution layer, and 'PW' is a point-wise convolution layer.

Module	Layer	Out_channel	Stride	#Param
Conv1	DW(BN+ReLU6)	1	1	9+2
	PW(BN+ReLU6)	16	1	16+32
MaxPool1	MaxPooling	-	2	-
Conv2	DW(BN+ReLU6)	16	1	144+32
	PW(BN+ReLU6)	32	1	288+64
Conv3	DW(BN+ReLU6)	32	1	288+128
	PW(BN+ReLU6)	64	1	2048+256
MaxPool1	MaxPooling	-	2	-
Conv4	DW(BN+ReLU6)	64	1	576+128
	PW(BN)	32	1	2048+64
Conv5	DW(BN+ReLU6)	32	1	288+64
	PW(BN)	10	1	320+20
AvgPool1	AvgPooling	-	7	-
Total		-	-	6847(6.85K)

*The size of each depth-wise convolution filter kernel is 3×3.

the kernel size of the depth-wise convolution. The depth-wise convolution performs lightweight calculation by independently filtering the feature map per input channel, while the point-wise convolution is utilized to extract features between different input channels. A batch normalization layer and an activation function follow each convolution layer. Inspired by MobileNetV2 [35], the activation operations in point-wise convolutions from high dimension to low dimension are removed to prevent non-linear functions from damaging too much information. The classifier consists of five depth-wise separable convolutions, two max-pooling layers, and an average pooling layer. Most of the layers in the MobileNetV2 are removed as the dataset is small and the resources are limited. The details of the model are shown in Table II. The model, which occupies less than 6.9 K of storage, is lightweight enough to execute on an edge device.

C. Pipeline of Transfer Learning

Relying on the small-scale digit dataset from 11 volunteers in Section II-D for air-writing digit recognition, especially in the context of lightweight network applications running on edge devices. The limited sample amount inadequately represents the complex and diverse array of human writing styles and habits, potentially leading to concerns about poor recognition accuracy and scalability. Alternatively, the process of collecting data from a broader user demographic is not only time-intensive but also financially demanding. To address this challenge, we implement a transfer learning strategy using the open-source MNIST dataset comprising 60K training and 10K testing samples, covering digits 0 to 9. Each sample in this dataset is a grayscale pixel image, thus ensuring compatibility with the data types and categories inherent in the proposed air-writing digit dataset. Through this strategic approach, the model's robustness is enhanced, and the limitations of the initial dataset are addressed.

The workflow of the transfer learning strategy is shown in Fig. 5. Initially, we conduct a few pre-training epochs using the proposed lightweight network on the MNIST dataset. This step aims to familiarize the lightweight network with the

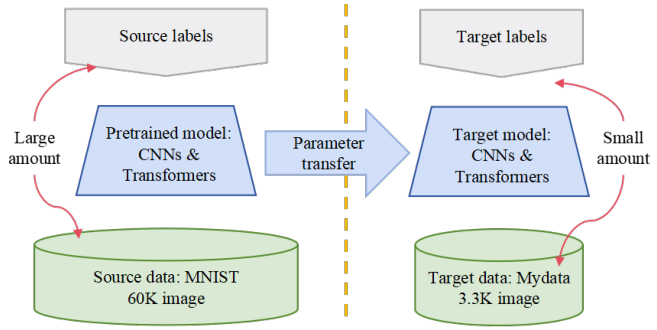


Fig. 5: An illustration of the transfer learning strategy for air-writing digit recognition. We first pre-train model weights on the MNIST [36] dataset and then fine-tune them on the proposed dataset.

fundamental features of digit trajectory images (e.g., edges, corners, textures, etc). Subsequently, we import the pre-trained network weights and transfer the lightweight network to the proposed air-writing dataset for fine-tuning. This fine-tuning process enables the lightweight network to adapt to the unique attributes of the proposed dataset, which may involve dynamic changes and handwriting characteristics specific to air-writing digits. Additionally, we introduce CNN and Transformer classification networks to further explore the feasibility of transfer learning strategies in enhancing the recognition accuracy and scalability of air-writing digit recognition.

D. Pipeline of Knowledge Distillation

To address the challenges posed by traditional classification networks, such as computational load, parameter size, processing speed, memory usage, and deployment difficulties on mobile or edge devices, we propose a lightweight network in Section III-B. However, a significant limitation of lightweight networks is that they often cannot match the performance of larger models, regardless of the training computation involved [37].

Through transfer learning, Section III-C allows us to obtain a multitude of classical CNN and Transformer classification networks. Given this, we introduce a knowledge distillation strategy to transfer the deep-layer feature of air-writing digits from the classical models to the proposed lightweight network. As illustrated in Fig. 6, we select several high-accuracy networks as teacher models. Subsequently, the lightweight network served as the student model. During the training on the air-writing dataset that we created, based on the pre-trained weights from Section III-C, we teach the student model the basic features of the data and enable it to grasp the teacher model's knowledge (soft labels). By employing the knowledge distillation strategy [38], the student model can maintain a smaller parameter size and computational load while achieving higher accuracy. This method ensures the model's efficiency and enhances the lightweight model's performance in real-time air-writing digit recognition.

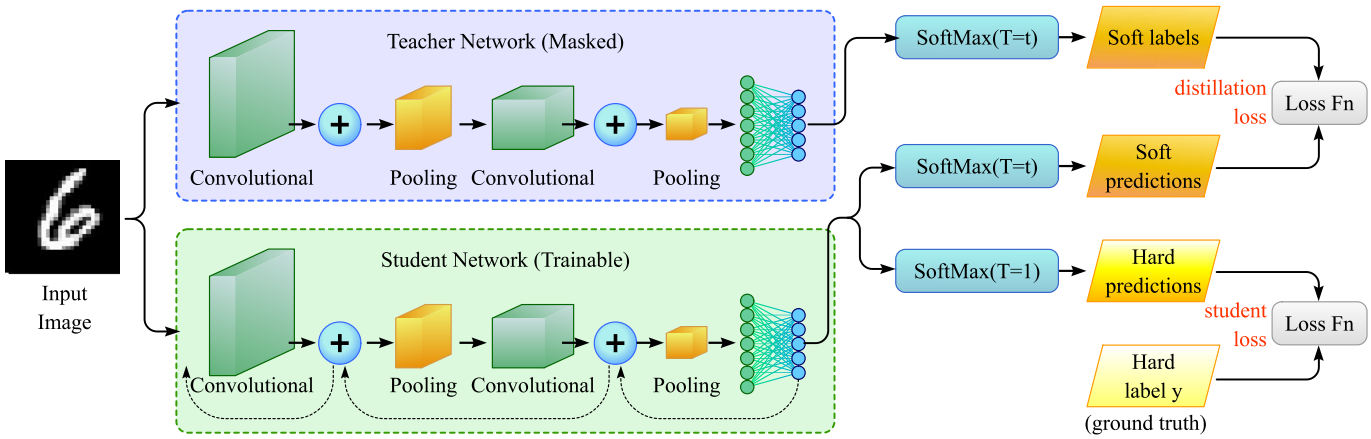


Fig. 6: The strategy of knowledge distillation employs high-precision classification networks within the context of transfer learning to guide the training of our proposed lightweight network on air-writing digit dataset.

E. Loss Evaluation

To validate the recognition and scalability capabilities of small-scale datasets, as well as the feasibility of performance improvement through transfer learning, We focus on training the foundational classification model, designed to classify input data effectively. We utilize the cross-entropy loss function as the optimization criterion to achieve this. The cross-entropy loss function is a standard method for measuring the difference between the model's predicted probability distribution and the true label distribution, and it can be defined as

$$L_{\text{hard}} = -\sum_{i} y_i \log(p_i), \tag{11}$$

where y_i represents the one-hot encoded representation of the true labels, and p_i denotes the probability distribution predicted by the model.

We further enhance the model's cross-user performance in the knowledge distillation phase. This involves transferring the knowledge of a pre-trained, more extensive, high-accuracy teacher network to a smaller student network. To facilitate this, we design a composite loss function that combines the hard loss concerning the actual labels and the soft loss for the teacher network's output. Specifically, the loss function is composed of the following two parts as

$$L_{\text{soft}} = T^2 \cdot KL\left(\sigma\left(\frac{z^S}{T}\right) \| \sigma\left(\frac{z^T}{T}\right)\right), \tag{12}$$

$$L = (1-\beta)L_{\text{hard}} + \beta L_{\text{soft}}, \tag{13}$$

where L_{hard} is the cross-entropy loss between the student network's output and the true labels, L_{soft} is the temperature-scaled cross-entropy loss between the student network's output and the teacher network's output, T is the temperature parameter used to smooth the probability distribution of the teacher model, β is a hyperparameter used to balance the two parts of the loss, z^S and z^T represent the logits output from the student and teacher networks respectively, σ denotes the softmax function, and KL represents the Kullback-Leibler divergence.

IV. Experimental Results and Discussion

A. Experimental Environment and Parameter Setting

We employ the MNIST dataset and a proposed air-writing dataset for experimental purposes. The MNIST dataset is processed following its standard splitting method [36], whereas the air-writing dataset is subjected to two distinct splitting methods. The initial method involves a 7:3 ratio for training and testing samples to evaluate the recognition performance of the mmDigit framework in processing air-writing digit datasets and conduct a comparative analysis with existing studies. The other method entails a 2:9 ratio for training and testing users, aiming to examine the generalization capabilities of the mmDigit framework in the cross-user domain.

All experimental procedures are conducted on a server with the Pytorch framework and NVIDIA GeForce RTX A100 GPU×4. The size of the input images is 28×28, the batch size is set to 32, and the Adam optimizer is utilized for parameter updating. The learning rate is set at 1e-4 to ensure optimal fitting across all algorithms. Regarding the transfer learning strategy, an initial pre-training of 50 epochs is conducted on the MNIST dataset, followed by a fine-tuning phase of 300 epochs on each partitioned dataset. For the knowledge distillation strategy, the model with higher accuracy is selected as the teacher model, with the temperature parameter T in Eq. (12) set to 7 and β in Eq. (13) set to 0.3. The entire training process comprises 300 epochs.

B. Data Augmentation

Data augmentation techniques capitalize on the intrinsic prior knowledge embedded within existing datasets to enhance the richness of the training data and enable the creation of supplementary samples that closely resemble the original dataset. This technique is particularly advantageous in scenarios where the availability of training data is limited, as the performance of deep learning models is intricately dependent on the quantity and quality of the training data. For this study, we employ a data augmentation strategy on the public MNIST dataset in conjunction with our air-writing dataset.

Following a specific sequence, we initiate data augmentation by randomly rotating the images within a range of {-20°, 20°}, enriching the diversity of the dataset. Subsequently, we apply random diffraction transformations to introduce further variations. Furthermore, during the data pre-processing phase, we implement normalization procedures to mitigate scale inconsistencies, thereby enhancing the training effectiveness of the model. In the case of the public MNIST dataset, we employ the normalization formula $x' = (x - 0.1307)/0.3081$, leveraging the mean and variance provided by the official sources [36], where x' represents the normalized value and x denotes the original pixel value. On the proposed dataset, which has deduced a mean of 0.0629 and a standard deviation of 0.2407, we apply the normalization formula $x' = (x - 0.0629)/0.2427$.

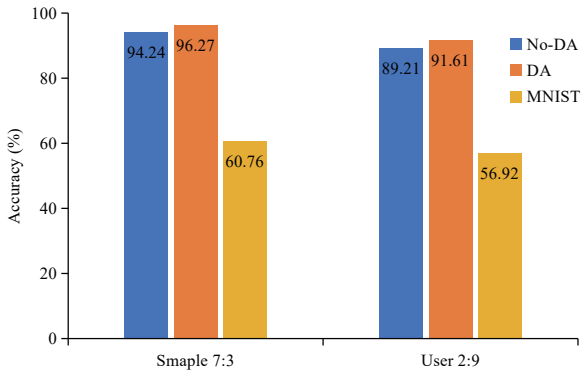
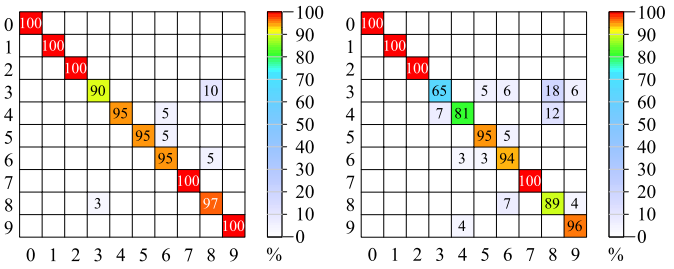


Fig. 7: Feasibility validation of lightweight networks. "No-DA" represents not augmenting the data. At the same time, "DA" indicates data augmentation techniques, and "MNIST" refers to applying the trained lightweight network to the MNIST dataset for performance testing.

C. Experiments on small-scale air-writing datasets

This study first explores the recognition performance and cross-user capabilities of the collected small-scale air-writing dataset. We employ two different data-splitting strategies to train the lightweight network. Fig. 7 displays the experimental results. Without using any data augmentation technology, the proposed lightweight network can achieve a recognition accuracy of 94.24% while using a 7:3 training/testing sample split, a recognition accuracy of 89.21% in the cross-user recognition scenario (training/testing ratio of 2:9). When data augmentation is implemented in the training process, the network's performance is elevated under both training settings, reaching recognition accuracy of 96.27% and 92.44%, respectively. We introduce confusion matrices for specific analysis to assess the recognition accuracy of different digit categories in other training settings. The confusion matrices are displayed in Fig. 8(a) and Fig. 8(b), based on the 7:3 split for training/testing samples and the 2:9 split for training/testing users, respectively. It can be observed from these matrices that the recognition accuracy for the digits 3, 4, and 5 is not as high as it should be.

We further examine the cross-domain recognition capabilities of the small-scale air-writing dataset, i.e., analyzing



(a) Sample 7:3 (b) Users 2:9

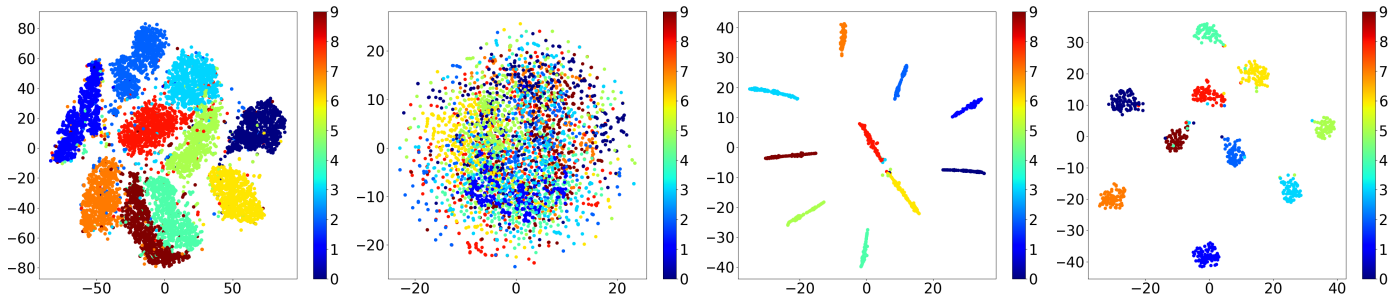
Fig. 8: Confusion matrix of air-writing digits recognition for lightweight networks under two data-splitting methods.

the performance of the lightweight network trained with two different data splitting schemes on the MNIST dataset. The result shows a significant drop in recognition accuracy. When the ratio of training/testing samples is 7:3, the recognition accuracy is achieved at 60.76%, while with a training/testing user ratio of 2:9, the accuracy decreases to 56.92%. To understand the reasons behind this decrement in model performance, we conduct a visual comparative analysis using the t-SNE algorithm on both the MNIST dataset and our air-writing dataset [39]. The distribution of the MNIST dataset is illustrated in Fig. 9(a), whereas the distribution of our air-writing dataset is depicted in Fig. 9(b). Significant differences exist between the two datasets, especially in the more dispersed distribution of air-writing digits. This finding underscores the limitation of our data, collected from 11 volunteers, in fully capturing the rich diversity and complexity of writing habits and styles. This raises concerns regarding the model's deployment capabilities in real-time recognition scenarios.

D. Experimental Results in Transfer Learning

An effective solution to address the significant drop in recognition accuracy and cross-user capabilities of the small-scale air-writing dataset when applied across various domains is to augment the dataset size. However, this raises concerns regarding time and economic costs. In light of this, we propose a transfer learning strategy that allows the lightweight network to be pre-trained on the MNIST dataset without expanding the air-writing dataset, enabling it to capture a broader array of writing styles and habits. Furthermore, we integrate various classic classification algorithms, aiming to comprehensively validate the efficacy of transfer learning strategy in enhancing the capability for recognizing air-writing digits.

1) CNNs: LeNet [40], AlexNet [41], and VGG [42] employ convolutional layers to extract image features, well-suited for early image recognition tasks. With advancements in technology, CNNs with unique structures have been developed, such as GoogLeNet (Inception module) [43], ResNet (Residual blocks) [44], and DenseNet (Dense connections) [45]. These networks allow models to learn image features at greater depth. To address the requirements of mobile devices, lightweight networks like MobileNetV2/V3 [35], [46], and ShuffleNetV2 [47] are proposed to optimize the convolutional



(a) Original MNIST dataset. (b) Air-writing dataset. (c) The output features by VGG16. (d) The output features by mmDigit.

Fig. 9: A comparative analysis of the original data distribution between MNIST and the proposed dataset and the output features of VGG16 and mmDigit was conducted using t-SNE visualization [39].

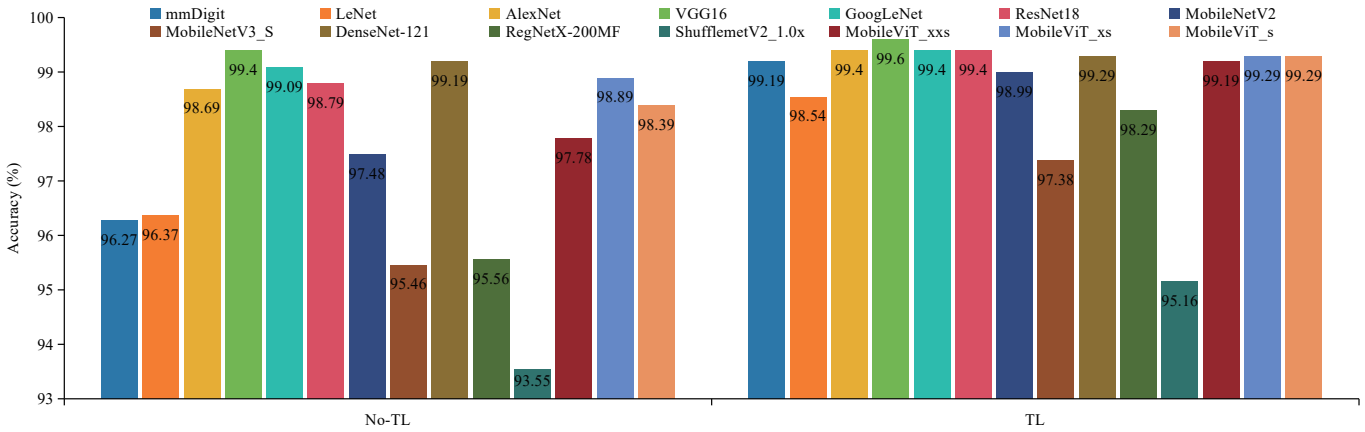


Fig. 10: The experimental results regarding the recognition of air-writing digits under sample 7:3.

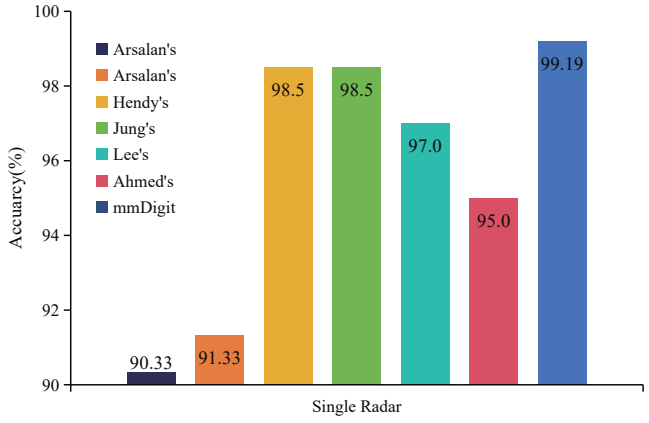


Fig. 11: Comparison of mmDigit's recognition accuracy with existing research under single radar conditions.

structure to reduce computational demands. Additionally, networks like RegNetX [48] can be used to enhance performance through innovative design.

2) Transformers: The Vision Transformer (ViT) marks a groundbreaking incursion from the realms of natural language processing into the vast expanses of computer vision [49], which demands considerable computational resources and parameters. The Mobile-friendly Vision Transformer (MobileViT) [50], a ViT-based lightweight visual model, merges the efficiency and compactness of CNNs with the self-attention mechanism and global view of Transformer [51], designed for use in mobile devices and embedded systems.

The recognition experimental results of the air-writing dataset are shown in Fig. 10. Under the training/testing sample split ratio of 7:3, our proposed lightweight network and traditional network algorithms demonstrate commendable recognition efficiency without employing transfer learning strategies, achieving an accuracy rate exceeding 93.55%. The recognition accuracy significantly improved after integrating transfer learning techniques, with the overall accuracy rate increasing to 95.16%. Notably, the lightweight network can attain a recognition accuracy of 99.19%, approximating the discernment precision of conventional networks with less than one percent of the parameter. This result fully validates the superiority of the air-writing data processing pipeline proposed in mmDigit. Additionally, the study shows that by introducing external datasets, allowing the neural network to learn more helpful information during the pre-training phase, the recognition capability of small-scale air-writing can be effectively enhanced.

We also compare mmDigit with the existing research listed in Table I, and the results are shown in Fig. 11. When using a single radar device, mmDigit demonstrates a sig-

TABLE III: The experimental results regarding the recognition of air-writing digits under User 2:9.

Model		Accuracy (%)		Δ	#Params	#FLOPs
		No-TL*	TL			
CNNs	mmDigit	91.61	94.46	2.85	6.9K	1M
	LeNet [40]	85.98	93.54	7.56	85.8K	1.1M
	AlexNet [41]	92.44	94.37	1.93	3M	60.5M
	VGG16 [42]	91.33	97.69	6.36	14.9M	261.9M
	GoogLeNet [51]	91.24	95.48	4.24	6M	513.6M
	ResNet18 [44]	89.39	97.05	7.66	11.2M	31.9M
	MobileNetv2 [35]	86.25	95.94	9.69	2.2M	6.1M
	MobileNetv3_S [46]	80.17	93.91	13.74	1.5M	2.2M
	DenseNet-121 [45]	93.34	97.32	3.78	7M	347.1M
	RegNetX_200Mf [48]	78.51	94.65	16.14	2.3M	3.9M
	ShuffleNetV2_1.0x [47]	80.23	87.55	7.32	1.3M	3M
Transformers	MobileViT_xxs [50]	88.56	95.48	6.92	1M	5.7M
	MobileViT_xs [50]	91.14	96.49	5.35	1.9M	14.9M
	MobileViT_s [50]	91.51	97.32	5.81	4.9M	31.4M

* It is important to note that we did not incorporate any transfer learning strategy, nor did we make use of the MNIST dataset.

nificant improvement in recognition performance, surpassing other comparative methods [31], [25], [28], [29], [30], [32]. Compared to research schemes that use multiple radar devices [26], [27], [28], [29], mmDigit, which employs a lower complexity solution, achieves recognition accuracy that is on par with these methods. However, it is important to recognize certain limitations in this comparison process. In particular, there are significant differences in the number of radar devices, configurations, dataset selections, and implementation strategies among different methods, which could affect recognition accuracy.

Additionally, as shown in Table III. With a training/testing user split ratio of 2:9, both the lightweight network and the classical network algorithms performed sub-optimally, with the lowest accuracy recorded at 78.51%. However, after introducing transfer learning strategies, the recognition performance improved by at least 1.93%. The accuracy of ShuffleNet reached 87.55%, while the overall accuracy of all other networks exceeded 90%. With only 6.9K network parameters, the lightweight network achieved an impressive recognition accuracy of 94.46%. The experimental results indicate that transfer learning strategies allow the network to be trained on diverse and varied data distributions, significantly improving recognition accuracy, which provides a new solution for expanding small-scale datasets.

We also find that the recognition accuracy of most traditional networks exceeded that of the proposed lightweight network, which was expected due to the significantly smaller number of parameters in the lightweight network, only a fraction of those in the classical networks. We employ the t-SNE algorithm to conduct a visual comparative analysis of the final layer outputs between the VGG16 network and our proposed lightweight network. Fig. 9(c) shows the output feature distribution of the VGG16 network, while Fig. 9(d) displays the output feature distribution of the proposed lightweight network. It suggests that this discrepancy may be attributed to conventional networks' superior clustering of deep features.

E. Experimental Results in Knowledge Distillation

Although the proposed lightweight network already demonstrates its ability to maintain satisfactory recognition accuracy while reducing parameters and computational complexity, we continue to seek further improvements to enhance the network's practical recognition capabilities post-deployment. Given that the classical classification networks described in Section III-C have a more significant number of parameters and achieve higher accuracy, a knowledge distillation strategy is employed to enhance the accuracy of recognition. Specifically, we select VGG16 [42], DenseNet-121 [45], and MobileViT_s networks [50], which show superior performance in Table III, as teacher models. Utilizing their powerful feature extraction capabilities, we successfully unearth potential knowledge within digital trajectory features and integrate it into the lightweight network.

As detailed in Fig. 12 of the experimental results, the recognition accuracy can be enhanced by the lightweight network without changing the model's parameter size of 6.9K through the use of the knowledge distillation strategy. Employing VGG16 as the teacher model yields the pinnacle of recognition accuracy at 96.22%. Utilizing MobileViT_s as

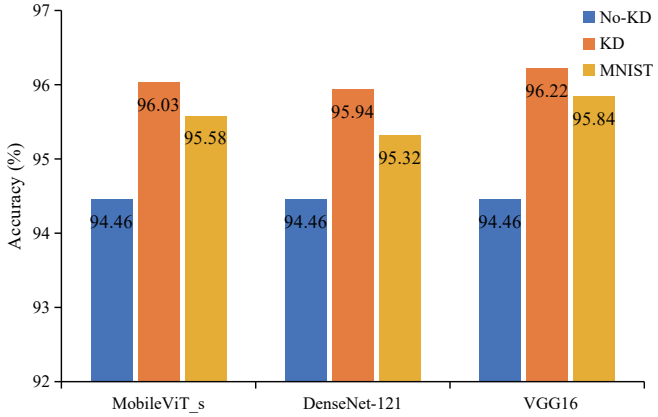
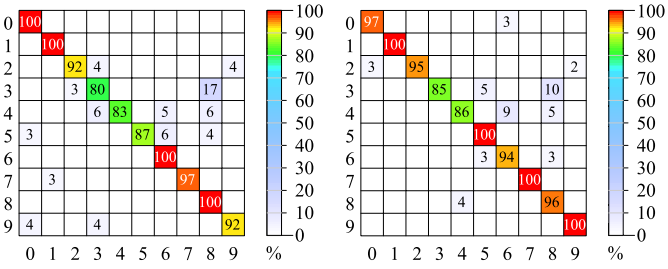


Fig. 12: Comparison of lightweight network performance after knowledge distillation of three teacher models, "No-KD" represents the use of the TL strategy without the KD strategy, while "KD" indicates the use of the KD strategy, and "MNIST" refers to applying the trained lightweight network to the MNIST dataset for performance testing.

the teacher model, we can achieve an accuracy of 96.03%. Moreover, adopting DenseNet-121 as the teacher model precipitates a 1.38% augmentation in recognition accuracy. The lightweight network achieves a cross-user recognition accuracy of 96.22%, a mere one percent of the parameters results in only a one percent loss in performance. We also conduct tests using the MNIST dataset, where the user splitting ratio was set at 2:9. The lightweight network's recognition capability increased from 56.92% to 95.84%. The lightweight network can acquire deeper numerical features by assimilating a greater variety of writing habits and styles through transfer learning and knowledge distillation strategies. Small-scale air-writing datasets with this enhancement see a significant improvement in recognition and scalability.



(a) Transfer learning strategy. (b) Knowledge distillation strategy.

Fig. 13: Confusion matrix of air-writing recognition for lightweight networks under two optimization strategies.

We employ confusion matrices to display the improvement in recognition accuracy of air-writing digit cross-users through knowledge distillation. The confusion matrix for digital categories before applying knowledge distillation is detailed in Fig. 13(a). Our proposed lightweight network demonstrates inferior accuracy in recognizing digits 3, 4, and 5, with digit three particularly susceptible to misclassification, registering an accuracy of only 80% and frequently mistaken for digit 8. Post-knowledge distillation, there is a discernible enhancement in the accuracy across all digital categories, with the accuracy of digit 3 increasing to 85%. Nevertheless, it remains highly prone to being identified as 8. This phenomenon can be attributed to the inherent difficulty in air-writing digits providing a standardized writing trajectory, thereby confusing.

In the early stages of algorithm development, we leverage the powerful computational and analytical tools of MATLAB® 2021 to validate the data processing workflows and lightweight networks thoroughly. This crucial process ensures the algorithm's integrity and robustness. Following successful validation, we move to the implementation phase, translating the processing algorithm into C++ and integrating this codebase into a computer equipped with an Intel® Core™ i7-11700K CPU and 100GB of memory, which serves as the edge computing unit for real-time recognition of air-writing digit. As depicted in Fig. 14, we separate the front-end data acquisition devices from the computing hardware. The front-end device is dedicated to real-time data acquisition and transmits data directly to the edge computing unit via the DCA1000EVM.

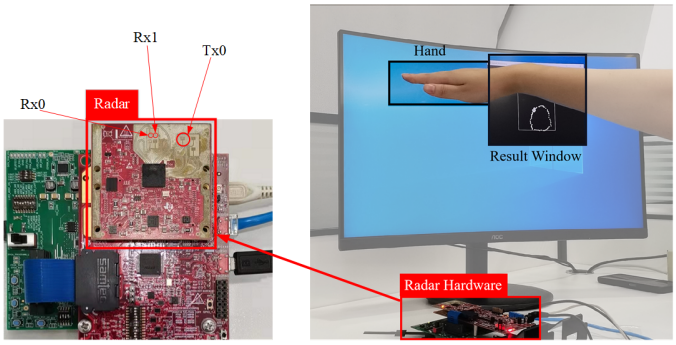


Fig. 14: A real-time recognition system, where the left shows the radar RDC data directly to a PC for processing. The right shows how the testing system is mounted relative to the hand motion and the auxiliary display screen.

To ensure the mmWave sensor provides a seamless user experience post-deployment, mmDigit must accurately identify the start and end moments of air-writing gestures, which we refer to as *gesture sequence cutting*. To address complex scenarios such as multi-stroke words or character recognition, gesture sequence cutting is crucial for enabling real-time continuous recognition. To enhance gesture sequence cutting for digit recognition, we introduce a mechanism to allow the system to switch between two different modes, namely the *open mode* and the *wait mode*. Two counters, c_o and c_w, to track the number of mode transitions for each, with mode switches governed by threshold values established through environmental experimentation. In open mode, the system processes data for gesture recognition; conversely, in wait mode, the system halts data processing, signaling that the user may be at the inception or conclusion of a gesture. Inspired by touch writing, we incorporated a pause mechanism, requiring users to maintain hand stillness for a brief interval at the onset

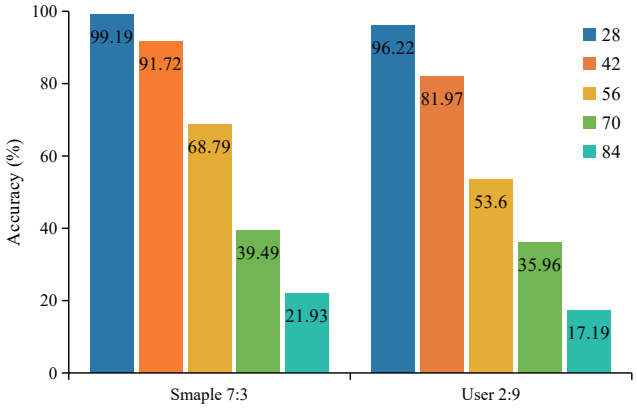


Fig. 15: The impact of different data input sizes. Notably, input sizes are 28, 42, 56, 70, and 84, and the network exhibits a parameter of 6.9K, with respective FLOPS measurements of 1M, 2.1M, 3.8M, 5.9M, and 8.6M.

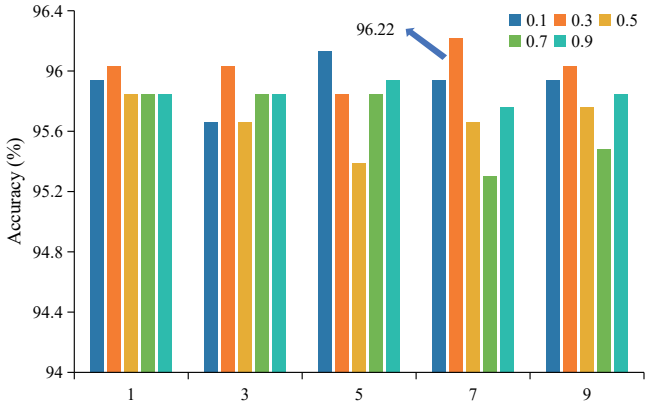


Fig. 16: The impact of different temperature parameters T and weight parameters β, where T is 1, 3, 5, 7, 9, and β is 0.1, 0.3, 0.5, 0.7, 0.9.

or conclusion of a gesture, thereby serving as a signal for gesture sequence cutting.

F. Ablation Experiment

Additional evaluation is required to determine how the input image size affects the network's ability to recognize and the computational burden. The experimental results across five scales of 28, 42, 56, 70, and 84 are shown in Fig. 15. The computational load for the lightweight network tends to grow close to N^2 when the image input size is increased, with N being multiples of 24. However, recognition accuracy and cross-user ability have also seen a noticeable decline. We suggest that input data distribution can change depending on the input size, resulting in more complex spatial relationships and structures. The intricacy of the task may surpass the model's capacity for effective learning, and a situation can be exacerbated when the training data fails to encompass the necessary variability to address potential changes at an expanded scale. Consequently, the research opts for a standard input size of 28×28 pixels. By choosing this wisely, we achieve a balance between recognition capability and computational efficiency.

As part of the knowledge distillation phase of model deployment, we extensively investigate how hyperparameter selection affects the network's performance. With the goal achieved by adjusting the temperature parameter T in Eq. (12) and the loss weight parameter β in Eq. (13). Notably, the temperature parameter T determines the smoothness of the probability distribution of the teacher model's outputs, thereby influencing the ease with which the student model learns the knowledge from the teacher model. A higher temperature value makes learning easier for the student model, especially for features of low-probability classes. However, an excessively high temperature may cause the learning process of the student model to become less focused, thereby affecting the convergence speed and final performance. Conversely, a temperature that is too low may cause the student model to overlook some important class information, thereby reducing the model's generalization ability. Therefore, the temperature parameter T needs to be determined through extensive experiments or empirical methods. The experimental results depicted in Fig. 16 indicate that the model achieves peak accuracy when the temperature and loss weight parameters are set to 7 and 0.3, respectively. These findings confirm the effectiveness of the knowledge distillation strategy in enhancing the performance of lightweight networks and offer empirical guidance on optimizing the critical hyperparameters within the knowledge distillation process.

V. Conclusion

We propose a real-time air-writing digit recognition framework, termed mmDigit, which encompasses the configuration of FMCW radar, the design of digit handwriting actions, the construction of the data processing pipeline, the setup of a lightweight network, and the gesture segmentation method for real-time recognition. Leveraging transfer learning and knowledge distillation strategies, mmDigit achieves a recognition capability of 99.19% on small-scale air-writing datasets, with cross-user performance reaching 96.46% while maintaining network parameters as low as 6.9K. This guides subsequent real-time air-writing recognition and expansion. Additionally, we have provided an air-writing dataset, including radar data cubes and processed digit trajectory images, which are publicly available to promote the development of the air-writing field.

Looking forward, we plan to delve deeper into the following domains:

- Expanding from numerals to English alphabets and special characters. At present, our air-writing dataset is only focused on digits. We aim to expand the range of air-writing maneuvers to incorporate English alphabets and special characters. Our objective is to achieve air-writing recognition of a diverse array of keyboard characters and their combinations, enhancing the semantic diversity of gesture communication in human-computer interaction.
- Implementing human writing actions to tackle the problem of trailing trajectories. The proposed continuous

writing method prevents data trailing caused by transitional movements during air-writing digits, enhancing data quality. As the complexity of data categories increases, the differentiation in writing styles will continue to grow, limiting individuals from interacting according to specific writing methods and potentially degrading user experience. Our next research focus will be to design a more sophisticated and effective data processing pipeline that adheres to human writing habits, aiming to address the trailing issue.

- Continually improving the performance of the air-writing system. Our current research area may not have enough external datasets due to the diversification of data types. Therefore, we must investigate other methods, like efficient network structure and quantization compression, to continuously enhance air-writing's recognition performance and scalability on edge devices.

In essence, we aspire for the air-writing system to tackle system-level challenges in dynamic environments, fully utilizing the potential of human-computer interaction in IoT scenarios and providing a more intuitive and seamless user experience.

REFERENCES

[1] I. S. MacKenzie, *Human-computer interaction: An empirical research perspective*. Elsevier, 2024.

[2] T. Kosch, J. Karolus, J. Zagermann, H. Reiterer, A. Schmidt, and P. W. Woźniak, "A survey on measuring cognitive workload in human-computer interaction," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–39, 2023.

[3] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, pp. 1–54, 2015.

[4] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak, "Hand gesture recognition for human computer interaction," *Procedia Comput. Sci.*, vol. 115, pp. 367–374, 2017.

[5] W. Xu, "From automation to autonomy and autonomous vehicles: Challenges and opportunities for human-computer interaction," *Inter.*, vol. 28, no. 1, pp. 48–53, 2020.

[6] H. Detjen, S. Faltaous, B. Pfleging, S. Geisler, and S. Schneegass, "How to increase automated vehicles' acceptance through in-vehicle interaction design: A review," *Int. J. Hum.-Comput. Interact.*, vol. 37, no. 4, pp. 308–330, 2021.

[7] A. Bissoli, D. Lavino-Junior, M. Sime, L. Encarnação, and T. Bastos-Filho, "A human–machine interface based on eye tracking for controlling and monitoring a smart home using the internet of things," *Sensors*, vol. 19, no. 4, p. 859, 2019.

[8] D. Marikyan, S. Papagiannidis, and E. Alamanos, "Cognitive dissonance in technology adoption: A study of smart home users," *Inf. Syst. Front.*, vol. 25, no. 3, pp. 1101–1123, 2023.

[9] P. Zhao, C. X. Lu, B. Wang, N. Trigoni, and A. Markham, "Cubelearn: End-to-end learning for human motion recognition from raw mmwave radar signals," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10 236–10 249, 2023.

[10] F. Galvano, "Cultural variances analysis through gestures and hands," *no. February*, 2024.

[11] H. Wu, J. Gai, Y. Wang, J. Liu, J. Qiu, J. Wang, and X. L. Zhang, "Influence of cultural factors on freehand gesture design," *Int. J. Hum. Comput. Stud.*, vol. 143, p. 102502, 2020.

[12] J.-S. Wang and F.-C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2998–3007, 2011.

[13] C. Amma, M. Georgi, and T. Schultz, "Airwriting: a wearable handwriting recognition system," *Pers. Ubiquitous Comput.*, vol. 18, pp. 191–203, 2014.

[14] T.-Y. Pan, C.-H. Kuo, H.-T. Liu, and M.-C. Hu, "Handwriting trajectory reconstruction using low-cost imu," *IEEE Trans. Emerging Top. Comput.*, vol. 3, no. 3, pp. 261–270, 2018.

[15] S. K. Singh and A. Chaturvedi, "Leveraging deep feature learning for wearable sensors based handwritten character recognition," *Biomed. Signal Process. Control*, vol. 80, p. 104198, 2023.

[16] D. Wu, R. Gao, Y. Zeng, J. Liu, L. Wang, T. Gu, and D. Zhang, "Fingerdraw: Sub-wavelength level finger motion tracking with wifi signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–27, 2020.

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

[18] M. S. Alam, K.-C. Kwon, S. Md Imtiaz, M. B. Hossain, B.-G. Kang, and N. Kim, "Tarnet: An efficient and lightweight trajectory-based air-writing recognition model using a cnn and lstm network," *Hum. Behav. Emerg. Technol.*, vol. 2022, no. 1, p. 6063779, 2022.

[19] A. Choudhury and K. K. Sarma, "Trajectory-based recognition of in-air handwritten assamese words using a hybrid classifier network," *Int. J. Doc. Anal. Recogn.*, vol. 26, no. 4, pp. 375–400, 2023.

[20] M. Arsalan and A. Santra, "Character recognition in air-writing based on network of radars for human-machine interface," *IEEE Sens. J.*, vol. 19, no. 19, pp. 8855–8864, 2019.

[21] C. Zhang, Q. Xue, A. Waghmare, S. Jain, Y. Pu, S. Hersek, K. Lyons, K. A. Cunefare, O. T. Inan, and G. D. Abowd, "Soundtrak: Continuous 3d tracking of a finger using active acoustics," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–25, 2017.

[22] Y. Wang, J. Shen, and Y. Zheng, "Push the limit of acoustic gesture recognition," *IEEE Trans. Mob. Comput.*, vol. 21, no. 5, pp. 1798–1811, 2020.

[23] P. Wang, R. Jiang, and C. Liu, "Amaging: Acoustic hand imaging for self-adaptive gesture recognition," in *2022 IEEE Conf. Comput. Commun.* IEEE, 2022, pp. 80–89.

[24] Z. Han, Z. Lu, X. Wen, J. Zhao, L. Guo, and Y. Liu, "In-air handwriting by passive gesture tracking using commodity wifi," *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2652–2656, 2020.

[25] J. Jung, S. Lim, J. Kim, and S.-C. Kim, "Digit recognition using fmcw and uwb radar sensors: A transfer learning approach," *IEEE Sens. J.*, 2023.

[26] F. Khan, S. K. Leem, and S. H. Cho, "In-air continuous writing using uwb impulse radar sensors," *IEEE Access*, vol. 8, pp. 99 302–99 311, 2020.

[27] S. K. Leem, F. Khan, and S. H. Cho, "Detecting mid-air gestures for digit writing with radio sensors and a cnn," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1066–1081, 2019.

[28] M. Arsalan, A. Santra, K. Bierzynski, and V. Issakov, "Air-writing with sparse network of radars using spatio-temporal learning," in *2020 Proc. 25th Int. Conf. Pattern Recognit.* IEEE, 2021, pp. 8877–8884.

[29] M. Arsalan, A. Santra, and V. Issakov, "Radar trajectory-based air-writing recognition using temporal convolutional network," in *2020 Proc. 19th IEEE Int. Conf. Mach. Learn. Appl.* IEEE, 2020, pp. 1454–1459.

[30] H. Lee, Y. Lee, H. Choi, and S. Lee, "Digit recognition in air-writing using single millimeter-wave band radar system," *IEEE Sens. J.*, vol. 22, no. 10, pp. 9387–9396, 2022.

[31] N. Hendy, H. M. Fayek, and A. Al-Hourani, "Deep learning approaches for air-writing using single uwb radar," *IEEE Sens. J.*, vol. 22, no. 12, pp. 11 989–12 001, 2022.

[32] S. Ahmed, W. Kim, J. Park, and S. H. Cho, "Radar-based air-writing gesture recognition using a novel multistream cnn approach," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23 869–23 880, 2022.

[33] Texas Instruments, "MmWave FMCW Radar IWR6843ISK," 2017. [Online]. Available: http://www.ti.com/tool/IWR6843ISK.

[34] Texas Instruments, "DCA1000EVM for Real-Time Data Capture and Streaming," 2018. [Online]. Available: https://www.ti.com/tool/DCA1000EVM.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[36] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, 2012.

[37] Y. S. Taspinar, "Light weight convolutional neural network and low-dimensional images transformation approach for classification of thermal images," *Case Studies in Thermal Engineering*, vol. 41, p. 102670, 2023.

[38] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[39] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[40] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

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

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

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

[46] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for mobilenetv3," in *2019 Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[47] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient CNN architecture design," in *14th Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[48] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *2020 Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2020, pp. 10 428–10 436.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[50] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *31st Int. Conf. Neural Inf. Process. Syst.*, 2017, p. 6000–6010.

Jiake Tian received the B.S. and M.S. degrees from the School of Information Science and Technology, Henan University of Technology, Zhengzhou, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the School of Microelectronics, South China University of Technology, Guangzhou, China.

His research interests include object detection, radar perception, and sensor fusion.

Yi Zou (Senior Member, IEEE) is currently a Chair Professor at the South China University of Technology (SCUT), China. He received the M.Eng. degree from Nanyang Technological University, Singapore in 2002 and Ph. D. in Computer Engineering from Duke University, USA in 2004. Before joining SCUT, he was a postdoctoral fellow at Duke University, USA and a Senior Staff Research Scientist at Research Labs, Intel Corp., USA. He has published more than 70 publications including book chapters, technical papers, and patents. He serves as a frequent program committee member and reviewer for IEEE transactions and conferences. Most recently, he is a TPC member at IEEE NAS'21, ACM SEC'20/21, IEEE DataComp'19, etc. His current research areas of interest are intelligent compute architectures and systems, scale-out memory and storage, data and sensor fusion, edge computing and AI, etc.

Jiale Lai received the B.S. degree from the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, in 2022. He is currently pursuing the M.S. degree with the School of Microelectronics, South China University of Technology, Guangzhou, China.

His research interests include radar signal processing and radar perception.

Fangming Liu (S' 08, M' 11, SM' 16) received the B.Eng. degree from the Tsinghua University, Beijing, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong. He is currently a Full Professor at the Huazhong University of Science and Technology, Wuhan, China. His research interests include cloud computing and edge computing, data center and green computing, SDN/NFV/5G, and applied ML/AI. He received the National Natural Science Fund (NSFC) for Excellent Young Scholars and the National Program Special Support for Top-Notch Young Professionals. He is a recipient of the Best Paper Award of IEEE/ACM IWQoS 2019, ACM e-Energy 2018 and IEEE GLOBECOM 2011, the First Class Prize of Natural Science of the Ministry of Education in China, as well as the Second Class Prize of National Natural Science Award in China.