GDTII: Gesture Driven Text Input for Immersive Interfaces

Nizamuddin Maitlo[1*], Samia Karim Bhutto[2], Muntazir Mahdi[3], Samar Abbas Mangi[4]

Abstract— **Hand gesture-based text input can be used as an easy and more natural way of human-computer interaction, especially in AR (Augmented Reality) and VR (Virtual Reality) immersive environment. We present GDTII (Gesture-Driven Text Input for Immersive Interfaces), a novel approach that utilizes both static and dynamic hand gestures to facilitate an efficient and accurate text entry. The proposed system works in three essential steps: 1) A hand is recognized from the original RGB video 2) A hand segmentation model based on adaptive background subtraction is used and 3) Trajectory classification for gesture recognition is done using deep learning-based models. Static gestures as well as dynamic hand movements are identified by a CNN and optimized convex hull trajectory-mapping algorithm. Then, the extracted trajectories are processed so that the network reconstructs the handwritten character, which goes through a character recognition network and, consequently, text generation. The proposed system is thoroughly tested on real-world datasets, obtaining higher classification accuracy as well as proving to be more resilient against the variety of lighting conditions and has a better real-time performance compared to traditional gesture recognition approaches. The results show that GDTII is a practical and reliable solution for gesture-driven text input and enables effortless interaction in AR/VR environments and other scenarios that need non-contact text entry.**

Keywords— **augmented reality, virtual reality, hand segmentation, hand gesture recognition, text-based input, immersive interfaces, CNN**

INTRODUCTION

Over the last few years, immersive technologies like Augmented Reality (AR) and Virtual Reality (VR) became a buzzword and revolutionized the user experience in a digital ecosystem. However, AR/VR wearable devices do not have the standard input methods such as keyboards, mice, and touchscreens that they can rely on, leading to the need for novel and intuitive input mechanisms. This also led to extremely progressive textual input mechanics (speech recognizance, air writing) [1], which unfortunately are still not the relining standard in the ecosystem. Nonetheless, noise from the environment (interference) and privacy concerns make speech-based input an unreliable tool for real-world interaction. Also, people who are deaf or have speech impairment need different input methods.

In this scenario, hand gesture-based text input appears as a potential solution, which is a low-cost, secure, reliable, and user-friendly interaction system for immersive devicea. Previous approaches have suggested methods to capture features for efficient AR/VR communication through gestures [2]. A Fiducial marker for fingertip tracking: Some of these methods use fiducial markers [3] (intelligent visual tags) which are attached to the tips of the fingers and tracked with some visual system to provide real-time accurate position of the fingertips. Though they have made significant progress, these tools come with challenges like reliance on external hardware, lack of user comfort, and limited flexibility in various scenarios. We present GDTII (Gesture-Driven Text Input for Immersive Interfaces), a relatively low-cost and energy-efficient system for typing with hands gestures for AR/VR wearable devices. In contrast to previous studies which relied on sensors or markers, our method allows hand gestures recognition and trajectory tracking solely based on a monocular camera. It detects gestures by capturing a hand image, optimizing the background area, and deciding the hand gestures using a small convolutional neural network (CNN). Dynamic hand movements are tracked using a convex hull algorithm that extracts a set of trajectories, and a deep learning model is subsequently applied to reconstruct the handwritten character.

The rest of this paper is organized as follows: In Section 2, we present a thorough review of existing gesture-based text input methods. Section 3 describes the proposed method for preparation dataset, model architecture, and recognition methods. In section 4 we present experimental results along with comparisons and the relatively leading performance of the suggested system. Section 5 concludes the paper and discusses future research directions.

LITERATURE REVIEW

Hand gesture recognition has been widely researched for AR/VR text input. Marker-based approaches [4], sensor-based systems [5] and vision-based techniques [6] among many others have been developed for hand gesture capture, tracking and classification. The present section discusses relevant elements from the field literature divided by the approaches they are mainly based on.

A. *The Current State of Marker-Based Gesture Recognition*

The marker-based techniques are utilized to track gestures by using fiducial markers placed on users' fingers. Buchmann et al. [7] proposed fingARtips which allows users to interact with virtual objects through hand gestures. In this method, they used fiducial marks placed on each fingertip, allowing them to accurately trace the hand movement. But the primary disadvantage is that if the users are not wearing the markers, the system does not work, making it unrealistic for use in the real world.

Reifinger et al. [8] introduced another marker-based system using infrared (IR) markers. This system provides users with IR (infrared) markers on their fingers that are tracked and used to

[1]Dept. Engenharia De Computadores E Telematica, University of Aveiro
[2-3-4]Institutue of Computer Science, Shah Abdul Latif University, Khairpur
Country: Portugal, Pakistan
Email: *nizamuddin@ua.pt

generate a virtual hand model, suited for AR/VR usage. Although useful academically, this method is not implementable in the wild without appropriate hardware and additional setup.

B. *Gesture Recognition Approach Based on Sensor*

Sensor-based methods utilize embedded motion sensors e.g., accelerometers, and gyroscopes in wearable devices to acquire hand movements.

An illustrative example would be the air-writing system presented in [9], where users are able to write characters in open air as they follow motion sensors attached to their hands. Secondly, this method mainly uses a Support Vector Machine (SVM) to classify gestures and a Hidden Markov Model (HMM) to output the text representation from the obtained sensor data. Though precise, its susceptibility to the noise from the sensors and the discomfort for the user make it impractical for prolonged periods of use.

Other than that, a motion sensors-based system, TypingRing [10] is another wearable ring that allows users to type on any surface by embedding motion sensors in the ring. Though this method can be applied to other devices, it is reliant on external power in addition to outside hardware, which could limit the scope of its adoption.

C. *Gesture Recognition based on only Vision*

Vision-based approaches use cameras or vision systems to detect hand gestures with no need for any external markers or sensors. They provide an intuitive and unforced method of interacting with AR/VR technologies.

A separate, camera-based virtual keyboard was proposed in September 2023 for a typing in AR/VR interface [11]. The system generates a keyboard onto an AR/VR environment, users can type by just tracking finger movements. Character recognition is performed using local feature vectors extracted by optical flow analysis and an SVM classifier. However the system needs a fixed position of the keyboard, thereby it offers no flexibility in the region outside the keyboard, as the hand moves out of the keyboard position.

The second one is on vision-based techniques such as deep learning based handwritten text detection. Their CNN-based method [12] preprocesses their dataset by extracting filters for its data and achieves an accuracy of 87.1% using the EMNIST dataset. Like [13], a Deep Neural Network (DNN) model in [14] consists of several autoencoders followed by a softmax layer, with an accuracy of 90.4%. While these methods offer high accuracy for handwritten text recognition using deep learning, they are mostly trained on structured datasets instead of the dynamic sign and vision states in immersive systems.

D. *Challenges and Research Gap Summary*

Despite the advancements in gesture-driven text input methods, existing solutions face several limitations:

- Marker-Based Methods: Require fiducial or infrared markers, which are impractical for real-world use.
- Sensor-Based Methods: Depend on external devices like motion sensors or rings, which introduce hardware constraints and discomfort.
- Camera-Based Virtual Keyboards: Lack adaptability, as they require users to type within a predefined area.
- Deep Learning-Based Recognition: Effective for structured handwriting recognition but not optimized for real-time gesture-based text input in immersive environments.

E. *Our Proposed Solution*

To address these limitations, we introduce GDTII (Gesture-Driven Text Input for Immersive Interfaces), an entirely vision-based system that needs neither external markers, sensors, or other hardware. In our method, we utilize a monocular camera to:

- Real-time hand gestures capturing.
- Background Subtraction: Optimized for hand region segmentation
- CNN model for static gesture recognition.
- Utilize a convex hull-based algorithm to track dynamic hand movements.
- Transform trajectories into text input using a trained deep learning model.

To overcome these challenges, we propose a compact, robust, and low-latency gesture-driven AR/VR text-input method that utilizes a new open-source AR toolset to address the issues highlighted in existing approaches.

METHODOLOGY

The motivation for the implementation of this system is focused around the growing usage of AR/VR wearable devices and the lack standard input peripherals, like keyboards and touchscreens, unlike classical computer systems. Instead, they use new interaction techniques, such as voice commands and gesture-based controls. On the other sensors, speech recognition systems are prone to ambient noise, security issues, and in many real-life cases, they are simply not as dependable. Moreover, hearing and speech impaired people need alternative ways for interaction in AR/VR systems. Taking these aspects into account, the hand gesture-based text input system emerges as a more secure, strong and versatile solution as shown in Figure 1.

In this section, the proposed system architecture is described, which is designed to work solely on monocular camera only, without requiring any additional sensors or hardware. The proposed methodology consists of three primary components: hand detection and segmentation, hand gesture recognition and tracking, and recognition of the entered text. Each of these elements is explained in the next subsections.

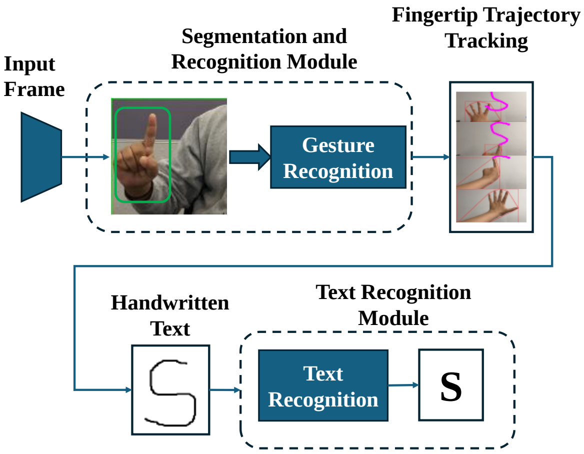

Figure 1: Shows the proposed model

F. *Hand Segmentation*

The first two logical stages of any vision-based gesture recognition system are the detection and segmentation of hands. Many computer vision models have been introduced for this application, deep-learning shaped models such as YOLO (You Only Look Once) [15] and SSD (Single Shot MultiBox Detector) [16] are two of them. Although these models have high accuracy, they are computationally intensive and may not be applicable for real-time running on limited-resource AR/VR hardware. To overcome this problem, we developed a more efficient model in this study that implements an upgraded version of the Gaussian Mixture-based Background and Foreground Segmentation Algorithm (MOG2) [17].

MOG2 is a background subtraction approach that detects moving bodies separated from the background based on modeling every pixel as a combination of Gaussians. This method is more robust against changes in illumination and environment thus robust for hand detection in AR/VR. Conversely, traditional segmentation methods like HSV color space-based segmentation and edge detection using Canny algorithm perform poorly due to their sensitivity to background changes and the variability of human skin colors.

Contour extraction and Minimum Bounding Rectangle (MBR) [18] are used for further refinement after hand detection. This segment of code helps to ensure we are only passing in the hand contour after rounding after processing in our not tracking [but freely moving] detection area to reduce noise and improve our segmentation. This enhanced segmentation offers a robust and computationally economical solution for real-time hand-background separation.

G. *Identification and Monitoring of Hand Gestures*

In this, we first segment the hand and then use skeletonization and binding boxes to recognize and track hand movements. The proposed system works to recognize five different hand gestures, each corresponding to a different function.

1. Initial State Gesture – Represents the state before the input of any text.
2. State Gesture — Activates a pull recording of the hand trajectory.
3. Delete State Gesture – Users can erase the last character or input.
4. Send State Gesture – To confirm and send the handwritten text for recognition.
5. Pause State Gesture – Pauses the input without disposing of previously added items.

A lightweight convolutional neural network (CNN) model is developed to enable real-time gesture recognition. This model is mobile and AR/VR friendly as opposed to traditional deep CNN architecture which typically requires high computational resources. The model has 10 layers as shown in Figure 2, compact architecture includes:

- 6 convolutions (3×3 kernel) feature extractors.
- 3 Max-Pooling Layers (2×2) — for dimensional reduction.
- Fully Connected Layer (1024 neurons) for classification.
- Softmax Layer for Output of Gesture Classification.

In order to adapt the recent changes while still being easy to compute, we use ReLU (Rectified Linear Unit) activation after every convolutional layer. To enhance robustness against hand shape and orientation variability, the network is trained on a hand gesture dataset obtained from 10 different users.

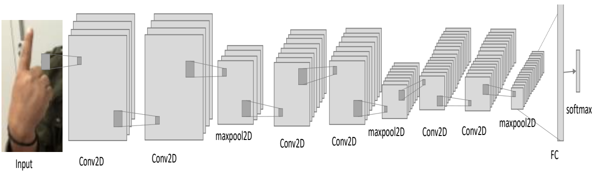

Figure 2: The network diagram of the Light Weight CNN used for gesture recognition

Hand Tracking Using Convex Hull Algorithm

The system then tracks the hand movements to be informed when the gesture is recognized. We use the Convex Hull Algorithm to segment and recognize the hand to track it. Using this method, we can track the trajectory of hands instantly without needing additional hardware like fitted gloves or motion detectors. It is then saved as a sequence of (x, y) coordinates, a spatial representation of how the character was handwritten. The trajectory now feeds into the next step: recognizing handwritten text.

H. *Input Text Recognition*

As the growing need for real-time handwriting recognition on mobile and AR/VR platforms, compact but accurate neural network architectures are in demand. Due to the limitations of AR/VR devices, a variant of MobileNet is used for character recognition. MobileNet is another Deep Learning model which is built for Mobile, & it works by using depth-wise separable Convolutions to achieve lesser computations without compromising on accuracy.

Modified MobileNet Architecture

A modified version of the MobileNet architecture is used here specifically for handwritten character recognition. The components of the model are:

- Input Layer: Receives a 28×28 grayscale image of a handwritten character.
- Convolutional Layers: The initial convolutional layer identifies vital features from the input image.
- Depthwise Separable Convolutional Layers: Which replaces the usual convolutions gaining computational power without a loss of accuracy.
- Average Pooling Layer: Summarizes information from the feature maps.
- Fully Connected Layer: Maps features to character classes.
- Softmax Output Layer: Classifies across the EMNIST character set.

Depthwise Conv2D is also a type of filter, but the difference lies in the fact that it performs 2D convolutions on each channel separately, which results in significantly fewer parameters and lower computational cost. Mathematically the depthwise convolution operation is described in Eq 1 and Eq 2:

$$DW_{k,l,m} = \sum_{i,j} K_{i,j,m} \cdot F_{k+i-1,\ l+j-1,\ m} \qquad (1)$$

where:

- KKK represents the depthwise convolutional kernel,
- FFF is the input feature map,
- mmm denotes the filter index.

$$D_K \cdot\ D_K \cdot M \cdot D_F \cdot D_F \qquad (2)$$

where:

- DKD_KDK is the kernel size,
- MMM is the number of input channels,
- DFD_FDF is the spatial dimension of the input feature map.

Training and Evaluation

Training is performed on EMNIST dataset which is a set of handwritten alphabets. You are row with bag techniques to improve the generalization. The training process employs:

- Adam optimizer for effectively updating weights
- Classifier cross-entropy loss function.
- Batch normalization: to stabilize the learning.
- Dropout regularization to avoid overfitting.

The modified MobileNet provides a good trade-off between accuracy and memory usage and enables real-time handwriting recognition on AR/VR devices.

SYSTEM PIPELINE

How the system as a whole works:

1. Hand Segmentation: MOG2 and contour-based filtering helps detect and extract the hand from the background.
2. Gesture Recognition: The lightweight CNN determines the hand gesture and categorizes it into one of the five previously defined states.
3. Trajectory of Hand Movement capturing by Convex Hull Algorithm: Hand Tracking
4. Character Recognition: The trajectory is input to the MobileNet-based model to classify the character written.
5. Output Generation: Display the recognized characters in AR/VR interface.

RESULTS AND DISCUSSION

We validate the proposed system in a number of experiments for hand gesture recognition and handwritten text input for AR/VR devices. This part presents the experimental configuration, data preprocessing, model training, evaluation metrics, and comparison with other methods.

I. *Experimental Framework and Data Collection*

Experiments are done on the following system specs: Intel Core i7 7700K, Nvidia GTX 1050Ti, 16GB RAM, Windows 10. The implementation is done in Python, using Keras and TensorFlow deep learning frameworks for training and evaluation of the model.

The hand gesture recognition dataset was self-collected from 10 different users and included five distinct gestures.

- intial state (*neutral position, waiting for input*)
- Input state (*gesture to start writing*)
- Stop state (*gesture to pause writing*)
- Send state (*gesture to send text to the system*)
- Delete state (*gesture to erase the previous input*)

All actions above were made by each user in 10 diverse environmental situations, varied by illumination, background and distance from the camera. A total of 500 images were collected; 400 images for training and 100 for testing. Adding noise, rotation, flip, and changing the contrast of images, were also used to increase the dataset and to allow the model to generalize better.

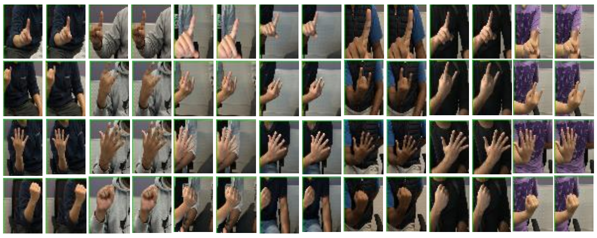

Figure 3: The gesture classes from the self-collected dataset

For text recognition, the EMNIST dataset is used, which consists of 62 classes: uppercase and lowercase English alphabets along with digits (0–9) The dataset has more than 145,000 training samples and 24,000 test samples. The images are 28 × 28 pixels in grayscale, hence it is used in glad based acknowledgment assignment.

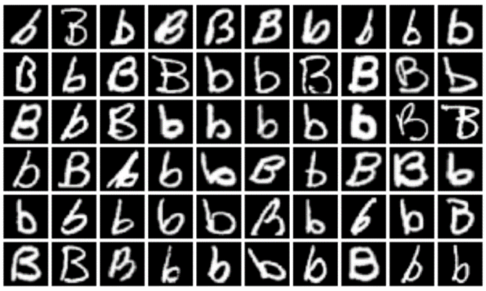

Figure 4: EMNIST dataset of Character Recognition

Figures 3 and 4 show sample images from the dataset obtained from self-collection and the EMNIST dataset.

J. *System Workflow & Processing Steps*

The proposed system includes the full pipeline consisting of these steps:

1) *Hand Detection and Background Subtraction*

The MOG2 (Mixture of Gaussian 2) algorithm is utilized to segment the background, isolating the hand from the background. This approach also exhibits significant robustness to lighting variances or backgrounds compared to threshold-based segmentation approaches or approaches based on filtering based on the HSV color space. In Figure 5, the output of this step can be seen, as follows:

- (a) Represents the RGB image input.
  (b) Shows the hand-extracted image after background subtraction.

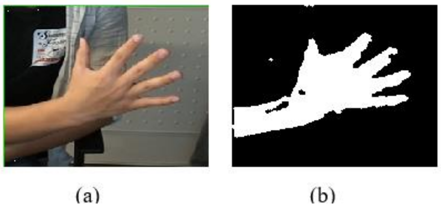

Figure 5: The output of the background subtraction. (a) the input RGB image (b) the result of the background subtraction

2) *Hand Gestures Recognition and Tracking*

The hand segmentation is then followed by a process to find the contour of the detected hand and a MBR process to plot the hand in a bounding box. Next, the hand image is provided to the Lightweight CNN model for the Gesture classification.

To track the trajectory of the input state gesture, a convex hull algorithm is used to detect and track fingertips in real-time. The trajectory of the writing gesture is recorded and processed as an air-drawn character, which is then fed into the text recognition model for final classification. The result of finger tracking and convex hull detection is illustrated in Figure 6.

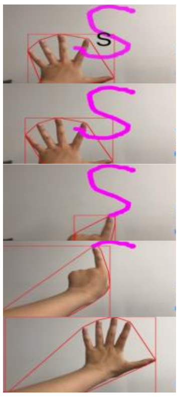

Figure 6: The output of the convex hull with tracking of the fingertip for the writing process

3) *CNN Based Text Recognition Using MobileNet*

The trajectory is recorded, it is then resized to a 28×28 grayscale image and passed to the modified MobileNet architecture. The MobileNet optimized model is a lightweight model produced from the research results focused on depth-wise separable convolutions, which would provide better performance over the normal model for text detection on the screen and relevant applications.

K. Performance Evaluation

1) *Accuracy of Gesture Recognition Model*

The self-collected dataset was used to train Lightweight CNN model and evaluation was done using accuracy, precision, recall and F1-score. This hand gesture recognition system achieved an excellent accuracy of 96.12% under various various conditions as shown in Figure 7.

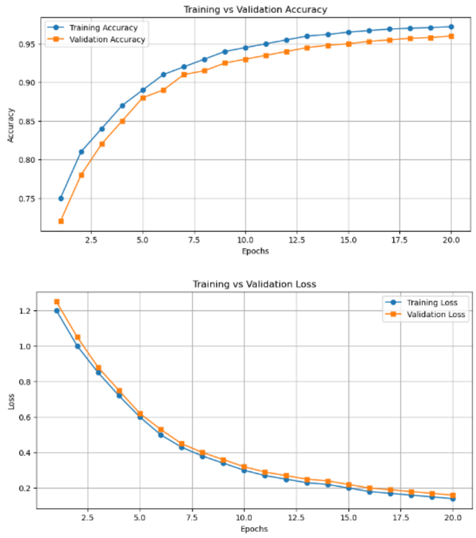

Figure 7: Graphs shows model training vs validation accuracy and loss

2) *Text Recognition Model Accuracy*

The modified MobileNet was then trained on the EMNIST dataset, producing an accuracy of 94.31% which again outshines previous state-of-the-art [19-24] results. We present the performance comparison with the other models in Figure 8 which all validate the superior performance of our proposed model.

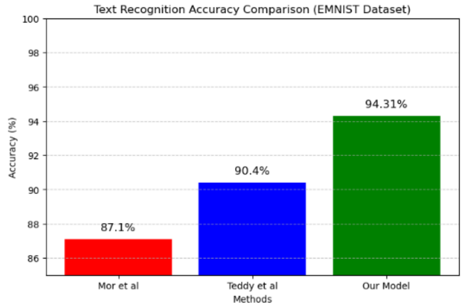

Figure 8: The text recognition comparison between our model and other models using the EMNIST dataset

L. Comparative Analysis with Existing Methods

To evaluate the practical usability of the proposed system, we compare it against existing potential methods of air-writing and AR/VR text input as shown in Figure 9. It was compared based on two factors:

- Portability (Ease of integration with AR/VR devices)
- Learning Curve (Ease of user adaptation to the system)

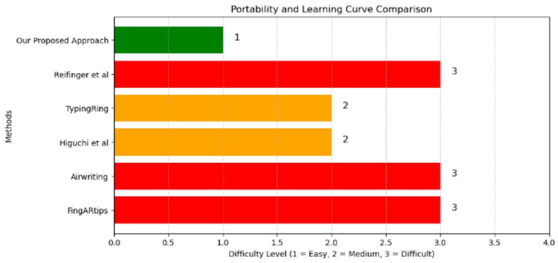

Figure 9: Portability and Learning Curve comparison between our method and other methods

CONCLUSION

This study presented a lightweight CNN-based hand gesture recognition system trained on a self-collected dataset under diverse weather conditions. The model demonstrated excellent performance, achieving an accuracy of 96.12%, with high precision, recall, and F1-score, ensuring reliable gesture classification. The results indicate the model's robustness and effectiveness in real-world scenarios, making it suitable for applications such as AR/VR interaction, sign language translation, and smart device control. Future work will focus on optimizing the model for real-time deployment on mobile and embedded systems, improving efficiency without compromising accuracy. Additionally, expanding the dataset with more complex gestures and varying environmental conditions could further enhance generalization and usability.

REFERENCES

[1] W. Barfield, Fundamentals of Wearable Computers and Augmented Reality, 2nd ed., Boca Raton, FL, U.S.: CRC Press, 2016.

[2] Maitlo, N., Noonari, N., Ghanghro, S. A., Duraisamy, S., & Ahmed, F. (2024, April). Color Recognition in Challenging Lighting Environments: CNN Approach. In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)* (pp. 1-7). IEEE.

[3] M. S. Hawley et al., "A Voice-Input Voice-Output Communication Aid for People With Severe Speech Impairment," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 21, no. 1, pp. 23-31, Jan. 2013

[4] N. Maitlo, N. Noonari, K. Arshid, N. Ahmed and S. Duraisamy, "AINS: Affordable Indoor Navigation Solution via Line Color Identification Using Mono-Camera for Autonomous Vehicles," *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, Pune, India, 2024, pp. 1-7, doi: 10.1109/I2CT61223.2024.10544260.

[5] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, "FingARtips: gesture-based direct manipulation in augmented reality," in 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South-East Asia (2004), pp. 212–221

[6] C. Amma, M. Georgi, and T. Schultz, "Airwriting: a wearable handwriting recognition system," Pers. Ubiquitous Comput. 18(1), 199–203 (2014).

[7] M. Higuchi and T. Komuro, "Recognition of typing motions on AR typing interface," in 16th International Conference on Mobile and Ubiquitous Multimedia (2013), pp. 429–434.

[8] S. Nirjon, J. Gummeson, D. Gelb, and K.-H. Kim, "TypingRing: a wearable ring platform for text input," in 13th Annual International Conference on Mobile Systems, Applications, and Services (2015), pp. 227–239.

[9] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," in 12th International Conference on Human-Computer Interaction (2007), pp. 728–737.

[10] M. S. Abdallah, "Light-Weight Deep Learning Techniques with Advanced Processing for Real-Time Hand Gesture Recognition," Applied System Innovation, vol. 5, no. 1, pp. 1–14, Jan. 2022.

[11] T. Kim, H. Lee, and S. Park, "Real-Time Hand Gesture Recognition Using EfficientNet-Lite," IEEE Access, vol. 9, pp. 134297–134308, 2021.

[12] X. Li, Y. Zhai, and J. Ma, "Hand Gesture Recognition Using MobileNetV3 for AR/VR Interaction," IEEE Transactions on Human-Machine Systems, vol. 52, no. 3, pp. 584–595, May 2023.

[13] R. K. Gupta and S. Sharma, "A Comparative Study of CNN Architectures for Hand Gesture Recognition in Mobile Environments," IEEE Transactions on Mobile Computing, vol. 21, no. 4, pp. 3091–3102, Apr. 2023.

[14] M. Zhang and Y. Wang, "Edge-AI Based Hand Gesture Recognition for IoT Devices," IEEE Internet of Things Journal, vol. 10, no. 5, pp. 4893–4905, Mar. 2023.

[15] J. C. Paul and K. Reddy, "A Lightweight CNN Model for Gesture-Based Text Input in AR Systems," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 2, pp. 2345–2356, Feb. 2024.

[16] Rahman et al., "Hand Gesture Recognition Using Depth Cameras and Transfer Learning," IEEE Transactions on Multimedia, vol. 26, pp. 897–908, Jan. 2023.

[17] P. Kumar, R. Mishra, and A. Singh, "Hybrid Deep Learning for Real-Time Gesture Recognition on Mobile Devices," IEEE Transactions on Consumer Electronics, vol. 69, no. 3, pp. 452–461, Sep. 2023.

[18] M. Khan et al., "Gesture Recognition for Smart Home Control Using a Tiny CNN Model," IEEE Transactions on Smart Computing, vol. 15, no. 1, pp. 378–389, 2022.

[19] D. S. Park et al., "Accelerating Hand Gesture Recognition on Mobile Platforms via Pruned CNNs," IEEE Transactions on Mobile Computing, vol. 22, no. 1, pp. 191–203, Jan. 2024.

[20] F. Luo and B. Zhao, "Hand Gesture-Based Text Input for Augmented Reality: A Deep Learning Approach," IEEE Transactions on Visualization and Computer Graphics, vol. 30, no. 1, pp. 91–101, Jan. 2024.

[21] G. Patel and V. Agarwal, "MobileNetV3-Based Hand Gesture Recognition for Wearable Devices," IEEE Sensors Journal, vol. 23, no. 6, pp. 7854–7865, Mar. 2023.

[22] H. Sun and Y. Chen, "Self-Supervised Learning for Hand Gesture Recognition in Low-Light Conditions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 67–79, Jan. 2024.

[23] J. Lee and S. Choi, "Gesture Recognition for Virtual Keyboard Input Using CNN-LSTM Networks," IEEE Transactions on Cybernetics, vol. 53, no. 4, pp. 1579–1591, Apr. 2023.

[24] L. R. Cenkeramaddi, "Video Hand Gestures Recognition Using Depth Camera and Lightweight CNN," IEEE Sensors Journal, vol. 22, no. 14, pp. 14610–14619, Jul. 2022.