Air Handwritten Recognition using Movement Detection and Hand-Gesture Recognition

line 1: 1st Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 2nd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 3rd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 4th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCI

***Abstract* — Air writing detection has garnered extensive attention due to its potential applications in intelligent systems. Despite this widespread interest, fundamental challenges in air writing have not been adequately addressed. This paper investigates the feasibility of air handwritten character recognition using a combination of movement detection, hand-gesture recognition, and machine learning. Leveraging the MediaPipe framework and the EMNIST dataset, we explore the model's performance on synthetic and real-world handwritten data. Our experiments reveal promising results for the EMNIST-based model, achieving accurate predictions for air-written characters and demonstrating its generalizability beyond the training dataset. However, limitations arise when encountering real-world data with significant variations in image quality and writing tools. Skewed images and different pen types negatively impact accuracy, highlighting the need for robust preprocessing techniques and diverse training datasets. This study emphasizes the importance of thorough experimentation and adaptation when transitioning machine learning models from controlled environments to real-world applications. Our findings pave the way for further research in addressing image quality issues, diversifying training data, and refining preprocessing pipelines to build more robust and versatile handwriting recognition models.**

Keywords— Air Writing Detection, Movement Detection, Convolutional Neural Network (CNN), Hand Tracking

I. INTRODUCTION

Detecting hand movements is a crucial component of Human-Computer Interaction (HCI) that has garnered significant attention in recent years, especially in applications such as air writing recognition. Air writing detection involves capturing hand or finger movements in the air to draw letters or numbers and has various practical applications. Among these applications is the hands-free input of text into computer systems, offering a contactless interaction with hardware. While there are numerous advantages to air writing detection, one notable benefit is its potential in preventing the spread of viruses, such as COVID-19(figure1).

In contrast to traditional writing, where the movement of a pen transfers information, air writing relies on the path of hand or finger movement to express desired letters or numbers [1]. Considering the limitations of conventional input methods such as keyboards and touchscreens, Human-Computer Interaction (HCI) plays an increasingly significant role in our digital world every day. This is particularly true for individuals with visual impairments and challenging conditions such as low light, where real-time digital character input in HCI environments is essential. However, it requires overcoming challenges in converting handwritten content into digital formats. Eliminating

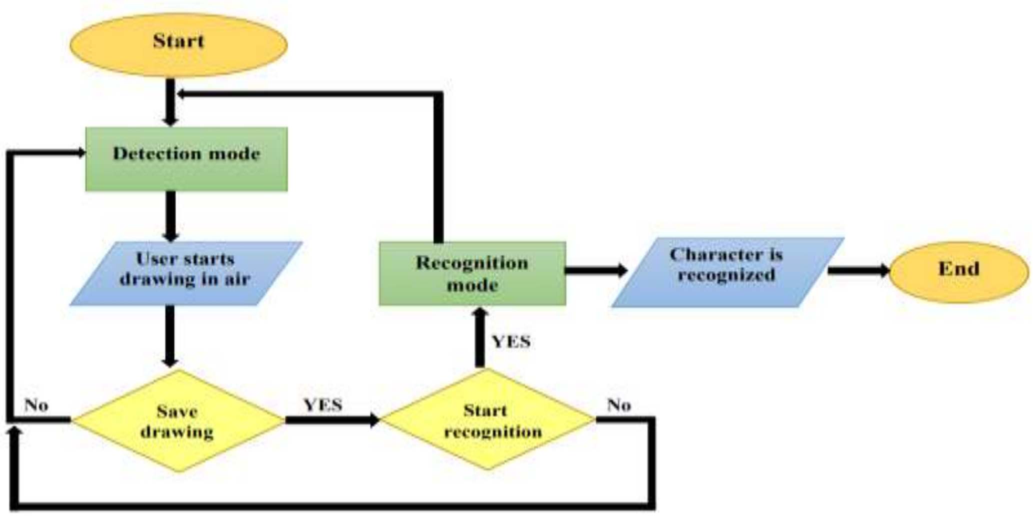

Figure 1: The flowchart of air handwritten recognition

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

the need for paper and pen can be achieved through more accessible and faster accessibility methods, making air writing appear as a simpler solution [2].

Air writing has emerged as a dynamic gesture for human-computer interaction, enabling natural communication with artificial intelligence systems, including smart TVs and robots. Unlike device-based methods that rely on handheld or wearable devices to capture hand movements, device-free approaches like vision and radio-based methods provide greater comfort and accessibility for users. This allows for a more seamless interaction with AI systems, fostering a connection that extends beyond traditional device-dependent approaches[3]. Air writing, as a subset of motion detection, has found applications in areas such as music interaction, robotics, and sign language translation, providing a suitable and natural means for information transfer in the realm of Human-Computer Interaction (HCI). It involves creating letters or words in a three-dimensional space through hand or finger movements, challenging traditional writing by eliminating the need for discrete strokes (hand or finger pressure) and dependencies between movements [4].

Various challenges exist in machine vision methods for hand or finger tracking, distinguishing between 2D and 3D image sensors[3]. In the world of 2D camera-based systems, enhancing tracking performance often involves the use of color markers on fingers since marker less finger tracking poses a significant challenge. Conversely, 3D camera-based systems effectively address hand and finger tracking issues by utilizing depth information provided by 3D image sensors such as Kinect, Leap Motion Controller (LMC), or Intel RealSense cameras. Writing in the air, as a unique gesture-based input method, presents its own set of challenges due to the absence of a reference framework on the writing surface, leading to the lack of clear starting and ending points for the lines drawn. This challenge introduces a segmentation problem or what is termed as "pressure to write," where the system must automatically identify the coordinates of the starting and ending points of the written letters in the air. Some proposed solutions include using specific hand gestures to represent the brief conclusion of the writing action, such as closing the hand into a fist. However, this introduces complexity and requires users to maintain specific positions[5]. The process of air handwritten recognition shown in figure 1.

In contrast, when depth information is available, segmenting writing actions using a depth threshold becomes easily achievable. In summary, 3D camera-based systems offer more elegant solutions to the first two challenges compared to their 2D counterparts, albeit with increased complexity and cost. Additionally, the use of a virtual box helps limit the writing area and reduces variability in letter input, simplifying subsequent processing but making the task more challenging for users[4].

In this article, we introduce a new approach to handwriting recognition utilizing deep Convolutional Neural Networks (CNN) with the aid of a cost-effective 2D webcam. This method effectively addresses existing challenges and can support versatile applications, including smart TV interfaces.

Our key contributions include:

(a) A robust algorithm for capturing air-written paths using a 2D webcam. The model employs hand and finger detection through the MediaPipe [1] library, accurately identifying and locating 21 distinct points, including the three-dimensional palm coordinates. This approach overcomes the complexities of finger tracking, resolving the pressure-to-write issue without dependencies on constraints or imaginary boxes, thus granting users unrestricted freedom in air writing.

(b) A novel data preprocessing scheme normalizing the x and y coordinates of the writing path and combining them into one-dimensional and two-dimensional arrays. These data arrays aid in training 1D-CNN and 2D-CNN models. We leverage EMNIST dataset in our Convolutional Neural Networks (CNNs), designed to automatically learn hierarchical spatial features from input images using convolutional filters, simplifying CNN architectures and enhancing their performance on complex images[3].

The remaining sections of this article are as follows: Section two reviews related previous works. Section three elaborates on the proposed method. Experimental results are presented in Section four. Finally, conclusions are drawn in Section five.

II. Related Work

Handwriting Recognition, also known as Air-Writing Recognition or Air-Painting Recognition, involves the act of writing or drawing in free space using hand movements, followed by the recognition of written or drawn letters or words. Previous research in the field of handwriting recognition has laid the foundation for the approach used in this study. Early advances in handwritten character recognition, such as the work conducted by SRI in the late 1950s, paved the way for recognizing handwritten letters and ultimately contributed to the evolution of handwriting recognition systems[5]. Challenges in handwriting recognition, including the detection and tracking of hand movements, as well as variations in posture, position, scale, and rotation, have been recognized in formal texts[6]. Various approaches, such as computer vision, object tracking, and machine learning techniques for letter recognition in handwriting systems, have been explored. Furthermore, research in the modeling and recognition of characters, words, and connecting movements based on hand motion data has been conducted, indicating the potential of machine learning and convolutional neural networks in handwriting recognition[5,7,8].

Recent research has also delved into the domain of 2D-camera-based airborne handwriting recognition using hand gesture estimation and hybrid deep learning models. This has led to further advancements in the human-computer interaction domain through airborne handwriting technology. The conclusion drawn by Fouad Alabir and his colleagues is that the proposed airborne handwriting recognition method significantly outperforms all existing advanced methods for both user-dependent and user-independent learning principles across all datasets. Additionally, in comparison to existing methods, the proposed model provides an accuracy improvement ranging

[1] https://github.com/google/mediapipe

from 0.52% to 3.55%, depending on the dataset and learning principle [9].

Challenges in handwriting recognition, including the detection and tracking of hand movements, as well as variations in posture, position, scale, and rotation, have been formally acknowledged in linguistics. Furthermore, the use of deep learning techniques and the selection of appropriate interpolation methods to enhance the accuracy and generalization of airborne handwriting recognition systems have been explored, paving the way for robust and multipurpose implementations. The study by Taiki Watanabe and his colleagues focused on character recognition using two datasets: one containing letters and another containing digits. Features were extracted from image data using CNN and from time series data using BiLSTM. The combined CNN-BiLSTM model was trained using five-fold cross-validation. The proposed system achieved high accuracy, with 99.3% for letter recognition and 99.5% for digit recognition. Comparison with existing studies using webcams demonstrated that the proposed system outperforms others, attributing its success to the use of gravity and the combination of CNN and BiLSTM. Validation on the 6DMG dataset confirmed the superior performance of the system, reaching an accuracy of 99.48% for letter recognition and 99.17% for digit recognition, surpassing existing methods. Overall, the conclusion strongly asserts that the proposed method establishes a new standard for accuracy and adaptability in airborne handwriting recognition [10].

Several other studies have been presented that demonstrate the effectiveness of marker-based techniques. Oka et al. [11] utilized a sophisticated sensor device for fingertip tracking, achieving a detection rate of 97.7%. Roi et al. [12] employed a fixed-color marker for tracking and segmentation, achieving a detection rate of 97.7% for simple English numerals. Rahman et al [13] improved marker tracking with calibration and dual RNN-LSTM networks, reaching high detection rates of over 98.75% for single digits and 85.27% for multi-digit recognition. Masra and colleagues [14] developed a hand motion detection scheme using a red fingertip marker, achieving a detection rate of 96.95% for 58 movements. However, it is worth noting criticisms: marker-based methods impose limitations and restrict their applicability in real-world scenarios.

Building upon the insights gained from the aforementioned studies, this research aims to advance the field of handwriting recognition by integrating the MediaPipe library for detecting hand movements and utilizing EMNIST data for recognizing letters in shapes written in the air. Further details about this endeavor will be discussed in the following sections.

III. PROPOSED METHOD

The handwriting recognition method depicted in Figure 2 comprises three key stages: Path Acquisition, Data Processing, and Network. Using a webcam, a sequence of images is captured, and a new finger tracking algorithm is employed to calculate the finger's movement path in the air.

Subsequently, the path data undergoes processing and is transformed into two-dimensional arrays. These data types are integrated into the path datasets for offline training of Convolutional Neural Network (CNN) models.

In the prediction stage, the system receives real-time data from the webcam, attempts to visualize the received image data, and then predicts the digit (or symbol) that the user has written using previously trained models. The initial three stages of the proposed system are outlined as follows.

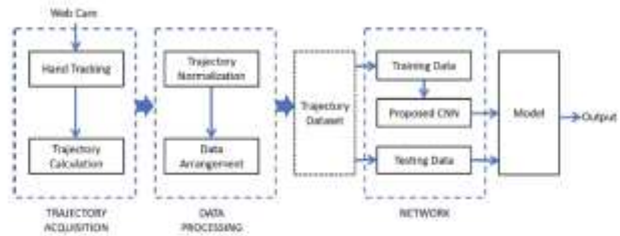

Figure 2: Handwriting Recognition Stages

A. Path Acquisition or Routing

This unit pursues two main objectives: firstly, capturing the two-dimensional page created by the user while writing in the air, and secondly, documenting the sequence of positions of a line, known as the writing direction. The direction is defined by the movement of the pointing finger. Therefore, identifying and tracking the moving pointing finger within the sequence of the two-dimensional page are vital functions of this unit. The history of finger detection and tracking has a long-standing background, yet achieving robustness and accuracy remains a persistent challenge. In this article, we employ the MediaPipe library for finger movement detection and effectively track the moving hand by storing the X and Y coordinate values. The proposed library is characterized by its robustness and real-time performance.

Hand detection using MediaPipe is a simple and fast method. However, it is susceptible to errors due to variations in light conditions and different user behaviors. To address issues such as light condition interference, hand complexity, background complexity, camera quality, and computational resources, a dynamic feature is incorporated into the MediaPipe hand detection algorithm[15].

The MediaPipe library utilizes a combination of models and algorithms for finger pose detection. The first model is a palm detection model that identifies the presence of hands and several key points in the image. The second model is a hand landmark model that predicts the three-dimensional coordinates of 21 specific hand points for each identified hand. The third component is a gesture recognition algorithm that estimates the direction and rotation of each finger based on hand signs. The gesture recognition algorithm compares the result with a set of predefined gesture descriptions[16].

The MediaPipe library manages interference issues such as occlusion, self-occlusion, and motion blur using a lightweight hand tracking algorithm. This algorithm determines the hand locations for detecting subsequent signs by employing a limited Intersection over Union (IoU) threshold between hands in the current frame and the last frame for tracking hand movements. If the confidence score of hand presence from the hand landmark model is below a specific threshold, the palm detection model is reactivated[17].

Furthermore, MediaPipe primarily utilizes a combination of computer vision techniques, including deep learning, for hand tracking and finger pose estimation to adapt to hand features. The algorithms and models used by MediaPipe may become more accurate over time as the library evolves and improves. MediaPipe often employs a deep neural network architecture for hand tracking, capable of detecting the position and direction of the hand in a specific video frame. Additionally, for estimating finger positions, Convolutional Neural Networks (CNNs) are typically used to predict the positions or angles of finger joints. To reduce interference issues and enhance the robustness of hand tracking and finger position estimation, techniques such as data augmentation, model optimization, and post-processing algorithms may be applied. MediaPipe may also use large and diverse datasets to train neural networks for better generalization to different hand shapes, sizes, and orientations[18-20]. The process of Capturing Handwritten Digits or Characters in the Air:

a) The user positions themselves in front of a webcam. As the camera system prepares to detect the hand, the process of writing in the air begins (Figure 3).

b) By raising the hand and drawing letters or numbers with finger movements, the system identifies the moving hand. Immediately after detecting the moving hand, it starts drawing a line between the X and Y coordinates of the index finger.

c) After the disappearance of the hand (moving the hand out of the frame) or using a shortcut key, the stages of drawing the desired letter or number come to an end, and the image of the written symbol is recorded.

Figure 3: Preparing for complete hand detection and filtering it to the index finger

Figure 4: Coordinates of the index finger's center are recorded

In all frames between the start and end of hand movement in the frame, the coordinates of the center of the index finger are recorded, forming the path of the letter or digit (Figure 3). This method eliminates the pressure challenge for writing without the need for a separator.

B. Data Preprocessing

As mentioned earlier, we store handwritten data in a 2D image (with the original size recorded). Users likely write commands in various positions in the air. To cope with positional changes, we convert the recorded image to a 360x360 image centered in a window. Additionally, for better readability, we increase the thickness of the lines drawn by the user. The resulting image is shown in Figure 4.

The data transformation is performed using the following relationships:

$$
\begin{aligned}
x_i &= \frac{x_i - x_{avg}}{1.4_r} * 360 + 180 \\
y_i &= \frac{y_i - y_{avg}}{1.4_r} * 360 + 180 \\
x_{avg} &= (x_{max} + x_{min})/2 \\
y_{avg} &= (y_{max} + y_{min})/2 \\
r &= \begin{cases} \Delta x = x_{max} - x_{min}, if\ \Delta x \geq \Delta y \\ \Delta y = y_{max} - y_{min}, if\ \Delta x < \Delta y \end{cases}
\end{aligned} \tag{1}
$$

In the above equations, x_i and y_i are the original coordinates on the x and y axes, and x_i and y_i are the transformed coordinates. x_{max} and y_{max} represent the maximum values of coordinates on the x and y axes, respectively. The purpose of the coefficient 1.4·r in the equations is to create margins of 0.2·r on the left and right edges of the final image after plotting the coordinates. To reduce the computational load during the training phase, the normalized image with a size of 360×360 is resized back to 36×36 using linear interpolation. The resized set of images is used for implementation and comparison with common existing approaches.

C. Dataset

This research utilizes the EMNIST (Extended Modified National Institute of Standards and Technology) dataset for character recognition. EMNIST is an extension of the MNIST dataset, which is a well-known database for handwritten digits. The EMNIST-Balanced dataset comprises sets of characters with an equal number of samples in each class. Additionally, the EMNIST Letters dataset integrates uppercase and lowercase letters in a balanced 26-class categorization. It's worth noting that the EMNIST Digits and EMNIST MNIST datasets include balanced sets of handwritten digits, maintaining compatibility with the original MNIST dataset.

The EMNIST Letters dataset, by merging all classes of uppercase and lowercase letters into a balanced 26-class categorization, aims to reduce errors associated with distinguishing between small and capital letters. Similarly, the

EMNIST Digits class includes a balanced subset of handwritten digit data, with 28,000 samples for each digit, maintaining compatibility with the original MNIST dataset[20].

D. Model and Evaluation

In this stage, the images are flattened, meaning each 28x28 image is transformed into a 784-element array (Figure 5). This transformation is necessary as fully connected layers are used instead of convolutional layers, which can process two-dimensional inputs (Figure 6).

To aid the training process, the data is normalized by dividing it by the maximum pixel value (255).

One-hot encoding is applied to the labels, transforming them into a binary matrix essential for multi-class classification.

Finally, a Keras model is created to stack the layers in the network. Two dense (fully connected) layers with 500 units/neurons each, both using ReLU activation functions, are added. The final dense layer with a number of units equal to num_classes employs a softmax activation function for output, suitable for multi-class classification. The model is compiled using categorical cross-entropy as the loss function, 'rmsprop' as the optimizer, and accuracy as the metric (Figure 7). The model is trained for 100 epochs with a batch size of 128, and ultimately evaluated on the test dataset.

The model has been trained with an accuracy of 95% and achieved an accuracy of 91% on the test set (Figure 8).

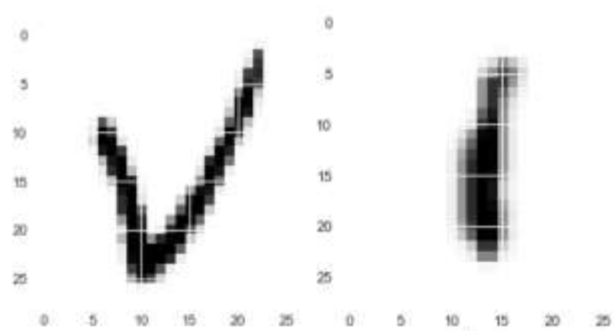

Figure 5: Display of Characters in the Frame

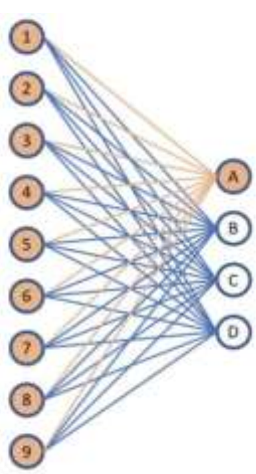

Figure 6: Visualization of Dense (Fully Connected) Layers

```
# Compile model
model.compile(loss='categorical_crossentropy', optimizer='rmsprop' , metrics=['accuracy'])
```

Figure 7: The Utilized Model

```
model.fit(X_train, y_train, batch_size=128, epochs=10, shuffle=True, verbose=2)
Epoch 10/10 1240000/1240000 - 19s - loss: 0.1345 - acc: 0.9512

model_loss, model_accuracy = model.evaluate(X_test, y_test, verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
20800/20800 - 2s - loss: 0.4131 - acc: 0.9130
Loss: 0.41310153188356846, Accuracy: 0.9129807949066162
```

Figure 8: Accuracy of the Utilized Model

IV. Experimental Results

During the experiments, an intriguing pattern emerged as the first image was mispredicted, while the same character in a different font was accurately identified. Notably, the second entered image exhibited higher similarity to the EMNIST dataset, suggesting a font-dependent influence on predictions.

The model underwent training on newly generated handwritten data by the author to assess its performance on real handwriting, leveraging the EMNIST model. Surprisingly, the outcomes did not align with expectations. A diverse set of alphabet letters (limited to uppercase) was captured using various writing tools, including pencil, pen, marker, and highlighter. The primary objective was to identify the writing tool yielding the highest accuracy with the EMNIST dataset.

In contrast to existing studies, where airborne handwriting recognition methods have showcased significant accuracy improvements ranging from 0.52% to 3.55%, depending on the dataset and learning principle, the current experiments revealed a nuanced scenario. While the EMNIST model demonstrated robust performance in recognizing handwritten characters, the challenges encountered in capturing skewed images, particularly with various writing tools, underscore the complexity of real-world scenarios. Unlike marker-based techniques that have achieved impressive detection rates, the unpredictability introduced by diverse writing tools adds an additional layer of complexity. The need for further experiments and refinement in data collection becomes evident, emphasizing the dynamic nature of achieving accurate airborne handwriting recognition across different writing mediums.

However, challenges arose as all entered images appeared skewed. Attempts to rectify this by rotation resulted in excessively bright and unreadable images, yet the model still produced successful predictions. To enhance accuracy, it is imperative to capture more images and optimize the conversion of pixels from 4D to 2D effectively.

Despite these challenges, the EMNIST model demonstrated commendable performance. On the other hand, the model trained on images written in the air requires further experimentation, albeit producing correct predictions in the initial tests.

This highlights the need for additional refinement in data collection and preprocessing techniques to ensure the model's robustness across diverse real-world scenarios.

Table 1: Different approaches accuracy for air handwritten

Models	Accuracy	Runtime	Trainable Parameters
CNN	98.5%	0.72 s	94,344
CNN-LSTM	97.5%	0.94 s	726,310
LSTM	99.5%	0.82 s	367,810
BLSTM	99%	0.85 s	367,810
TCN-Dynamic	97.5%	0.07 s	49,410
TCN-Static	97%	0.09 s	66,910
LSTM&CNN	98%	1.21 s	827,374
MediaPipe & EMNIST	%٧٧	1.12 s	814,255

Conclusion

The experimental results underscore the necessity of conducting extensive and meticulous experiments when applying machine learning models to real-world problems. The current model performs well on the EMNIST dataset but exhibits limitations when faced with data that significantly deviates from the features of the training dataset. For practical deployment, this model requires a more comprehensive preprocessing pipeline to address image quality issues, and it should be trained or at least validated on a dataset that better reflects the observed diversity in real-world handwriting, including handwritten samples created with various writing tools under different conditions. Moving from the laboratory to real-world applications introduces various complexities to the handwriting recognition task, with one of the least being variations in human handwriting and the conditions under which it is recorded. Our experiments demonstrate that while the baseline model shows good performance on standard datasets, substantial effort is required to prepare it for a broad spectrum of inputs beyond controlled datasets like EMNIST. To address these challenges, a systematic approach, including redefining preprocessing strategies, augmenting data, model complexity, and collecting real-world data, is essential for the model to truly excel in practical applications. This study emphasizes the importance of thorough and comprehensive experimentation with machine learning models before deploying them in the real world. Given that human handwriting can deviate significantly from standard training data, machine learning models need stronger preprocessing pipelines and more comprehensive training datasets for effective performance in real-world conditions. The results of this study also highlight the importance of focusing on specific challenges in the real world, such as image quality and the diversity of writing tools, to develop stronger and more reliable handwriting recognition models.

References

[1] Shahzad Ahmed, Wancheol Kim, Junbyung Park, and Sung Ho Cho. Radar-based air-writing gesture recognition using a novel multistream cnn approach. IEEE Internet of Things Journal, 9(23):23869–23880, 2022.

[2] Hongyu Zhang, Lichang Chen, Yunhao Zhang, Renjie Hu, Chunjuan He,Yaqing Tan, Jiajin Zhang, et al. A wearable real-time character recognition system based on edge computing-enabled deep learning for air-writing. Journal of Sensors, 2022, 2022.

[3] Chaur-Heh Hsieh, You-Shen Lo, Jen-Yang Chen, and Sheng-Kai Tang. Air-writing recognition based on deep convolutional neural networks. IEEE Access, 9:142827–142836, 2021.

[4] Grigoris Bastas, Kosmas Kritsis, and Vassilis Katsouros. Air-writing recognition using deep convolutional and recurrent neural network architectures. In 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 7–12. IEEE, 2020.

[5] M. Chen, G. AlRegib and B. -H. Juang, "Air-Writing Recognition—Part I: Modeling and Recognition of Characters, Words, and Connecting Motions," in IEEE Transactions on Human-Machine Systems, vol. 46, no. 3, pp. 403-413, June 2016, doi: 10.1109/THMS.2015.2492598.

[6] Mukherjee, Sohom & Sekh, Arif Ahmed & Dogra, Debi & Kar, Samarjit & Roy, Partha. (2019). Fingertip Detection and Tracking for Recognition of Air-Writing in Videos. Expert Systems with Applications. 136. 10.1016/j.eswa.2019.06.034.

[7] Nahar KMO, Alsmadi I, Al Mamlook RE, Nasayreh A, Gharaibeh H, Almuflih AS, Alasim F. Recognition of Arabic Air-Written Letters: Machine Learning, Convolutional Neural Networks, and Optical Character Recognition (OCR) Techniques. Sensors (Basel). 2023 Nov 28;23(23):9475. doi: 10.3390/s23239475. PMID: 38067848; PMCID: PMC10708688.

[8] Zare, M., Jampour, M., Arezoomand, A. S., & Sabouri, M. (2019, March). Handwritten recognition based on hand gesture recognition using deterministic finite automata and fuzzy logic. In 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA) (pp. 93-99). IEEE.

[9] Watanabe, T.; Maniruzzaman, M.; Hasan, M.A.M.; Lee, H.-S.; Jang, S.-W.; Shin, J. 2D Camera-Based Air-Writing Recognition Using Hand Pose Estimation and Hybrid Deep Learning Model. Electronics 2023, 12, 995. https://doi.org/10.3390/electronics12040995

[10] Abir FA, Siam MA, Sayeed A, Hasan MAM, Shin J. Deep Learning Based Air-Writing Recognition with the Choice of Proper Interpolation Technique. Sensors (Basel). 2021 Dec 16;21(24):8407. doi: 10.3390/s21248407. PMID: 34960499; PMCID: PMC8705512.

[11] Oka, Kenji & Sato, Yoichi & Koike, Hideki. (2002). Real-time fingertip tracking and gesture recognition. Computer Graphics and Applications, IEEE. 22. 64- 71. 10.1109/MCG.2002.1046630.

[12] Roy, Prasun & Ghosh, Subhankar & Pal, Umapada. (2018). A CNN Based Framework for Unistroke Numeral Recognition in Air-Writing. 404-409. 10.1109/ICFHR-2018.2018.00077.

[13] Rahman, Adil & Roy, Prasun & Pal, Umapada. (2021). Air Writing: Recognizing Multi-Digit Numeral String Traced in Air Using RNN-LSTM Architecture. SN Computer Science. 2. 10.1007/s42979-020-00384-9.

[14] Misra, Songhita & Singha, Joyeeta & Laskar, Rabul. (2018). Vision based hand gesture recognition of alphabets, numbers, arithmetic operators and ASCII characters in order to develop a virtual text-entry interface system. Neural Computing and Applications. 29. 10.1007/s00521-017-2838-6.

[15] C. Gunda, M. Maddelabanda and H. Shanmugasundaram, "Free Hand Text Displaying Through Hand Gestures Using MediaPipe," 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), Kannur, India, 2022, pp. 996-1000, doi: 10.1109/ICICICT54557.2022.9917991.

[16] B Sundar, T Bagyammal, American Sign Language Recognition for Alphabets Using MediaPipe and LSTM, Procedia Computer Science, Volume 215, 2022, Pages 642-651, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2022.12.066.

[17] S. Salim, M. M. A. Jamil, R. Ambar, R. Roslan and M. G. Kamardan, "Sign Language Digit Detection with MediaPipe and Machine Learning Algorithm," 2022 IEEE 12th International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2022, pp. 180-184, doi: 10.1109/ICCSCE54767.2022.9935659.

[18] Kavana, K. M., and N. R. Suma. "Recognization of Hand Gestures Using Mediapipe Hands." International Research Journal of Modernization in Engineering Technology and Science 4.06 (2022).

[19] Zhang, Fan, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang and Matthias Grundmann. “MediaPipe Hands: On-device Real-time Hand Tracking.” ArXiv abs/2006.10214 (2020): n. pa

[20] V. Jayasundara, S. Jayasekara, H. Jayasekara, J. Rajasegaran, S. Seneviratne and R. Rodrigo, "TextCaps: Handwritten Character Recognition With Very Small Datasets," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2019, pp. 254-262, doi: 10.1109/WACV.2019.0003