Natural Human-Machine Interaction Using Static Hand Gestures for a Gestural Calculator System with DNN

H. Abdelmoumene[1], L. Meddeber[2], O. Ghali[3], Y. Amellal[4]
Department of Electronics, University of Science and Technology of Oran, Oran, Algeria [1, 2]
Department of Computer Science, University of Naama, Naama, Algeria [3, 4]

***Abstract*—Hand gesture recognition (HGR) represents a real challenge for natural human-computer interaction, which aims to revolutionize the naturalness of traditional interfaces, allowing intuitive control of various devices without using a keyboard or mouse. Despite the availability of frameworks such as MediaPipe, which enable better detection and tracking, the major challenge remains interpreting gestures made with both hands in a natural operational setting. In this regard, this study presents a real-time gesture calculator that combines gestures made with both hands (see using one hand) and aims to address the problem of interpretation in arithmetic operations. By leveraging MediaPipe to classify the 21 hand landmarks, an optimized dense neural network (DNN) was developed capable of recognizing 13 distinct static gestures. The latter includes six gestures for each hand (ranging from 0 to 5) to represent all digits from 0 to 9, five mathematical symbols, and two specialized commands designed explicitly for control management. Even with a standard webcam, this model achieved 91% accuracy on a reduced dataset of gestures from both hands. Beyond gesture recognition, this work demonstrates how these gestures can be integrated into a fluid sequence for arithmetic operations.**

Keywords—Hand gesture recognition; gestural calculator interface; DNN; MediaPipe; natural human-computer interaction

I. Introduction

The field of human-computer interaction (HCI) has seen rapid progress in the development of natural interfaces in recent years, thanks to advances in methods for detecting and tracking body parts and the entire body, as well as deep learning methods such as DNNs [1]. Hand gesture recognition (HGR) stands out as a powerful means of communication with digital systems, given the many degrees of freedom it offers, making it the most expressive organ in the human body. This type of interface offers a level of fluidity and naturalness that traditional input devices, such as keyboards and mice, cannot match. HGR is redefining the way we connect the physical and digital worlds [2]. These systems have evolved into advanced solutions for controlling computers, navigating virtual and augmented reality environments, and providing real-time sign language translation [3]. Modern gesture recognition systems rely on the interaction between computer vision and deep learning [4]. In this field, Google's MediaPipe library, which offers a highly efficient platform for real-time hand tracking and 21-point landmark detection, occupies a central position. This technology offers powerful, accurate processing across a variety of lighting conditions and backgrounds [5]. However, the majority of available HGR systems are limited to simple static classifications and a single hand. There is a notable lack of research on effectively combining two-handed coordination with complex interaction logic in a practical, real-time application.

This research addresses these challenges by developing an intelligent gesture-based arithmetic calculation system. This study explores a pipeline that moves from detecting static gestures from a predefined gesture vocabulary to a dynamic, real-time calculator interface using a dense neural network (DNN) [6]. The system is designed to recognize 13 different static gestures comprising numerical values, arithmetic operators, and system commands optimized for two-handed interaction [7][8]. By combining MediaPipe's efficiency with a customized DNN, this demonstrates how computer vision can be leveraged to create a robust, low-latency human-machine interface [9]. While recent studies have explored gesture-based tools, such as the virtual calculators proposed by Umadevi et al. [7] and Sunanda et al. [8], most of these implementations focus on single-handed input or require significant computing resources. This work builds on these foundations but specifically targets two-handed coordination in an Industry 5.0 context [1], favoring a lightweight architecture that maintains high responsiveness on standard, low-cost hardware.

This study is structured as follows: Section II reviews the existing literature on hand gesture recognition systems. Section III describes the main objective of this work. Section IV addresses the chosen vocabulary and the suggested methodology. Section V presents the implementation, including the DNN architecture and the dataset used. Section VI explores the experimental results, accuracy metrics, and a discussion of system robustness. Finally, Section VII concludes the study by summarizing the contributions and recommending paths for future work in Section VIII.

II. Overview of Hand Gesture Recognition Systems

Gestures are a form of nonverbal communication in which body language conveys specific messages rather than speech [10]. This communication is accomplished through the movement of various body parts, which feels more natural and intuitive to people. Gestures significantly influence verbal messages [11]. Nonverbal communication has evolved to the extent that some languages have developed from a series of gestures. Gesture recognition technology plays a valuable role in enhancing human-machine interaction. Since the emergence of digital video capture technologies, numerous attempts have been made to recognize dynamic gestures for various purposes [12].

These two methods differ in accuracy, calibration complexity, latency, range of motion, ease of use, and cost. Thus, gesture recognition faces two types of challenges: a lower-level challenge that requires accurate posture detection and a higher-level challenge that focuses on the correct interpretation of gestures [13]. The analysis of gestures and postures presents several challenges, making it complex. Many concepts are often associated with the same gesture, and a given concept can be linked to various gestures. Furthermore, the interpretation of gestures varies across language, culture, and context [14]. For example, sign languages vary from one country to another, and even within the same country. Furthermore, gesture meanings likely change depending on the conversational context [15] [16].

III. Objectives

The idea was to create a model that performs basic arithmetic operations using hand gestures. This gesture-controlled calculator operates using both hands to compose numbers from 0 to 10. The numerical representation ranges from both hands closed, indicating 0, up to 10 represented by all ten fingers of both hands, with each hand symbolizing five digits. An advanced real-time model was developed to recognize both hands, determine the number of raised fingers on each hand, and calculate the total number of raised fingers. The calculator supports four arithmetic operations: subtraction (-), addition (+), division (/), multiplication (*), and equals (=). Additionally, two control commands are included: the gesture (e) to clear all input, and the gesture (d) to delete the last symbol entered (see Fig. 1).

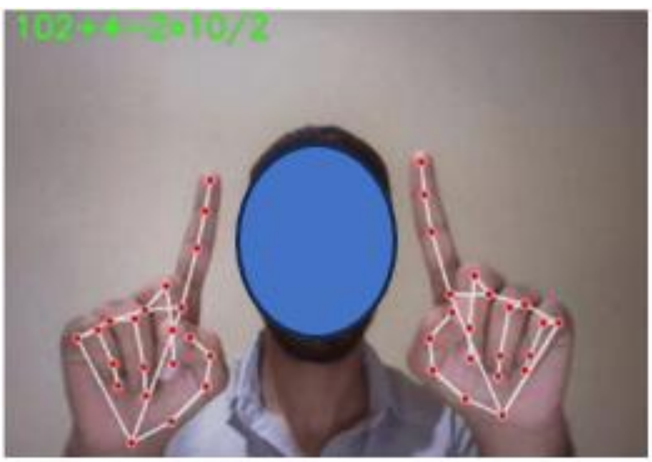

Fig. 1. Example of an arithmetic operation using gestures from both hands.

The impact of this work is manifold. From a scientific standpoint, it demonstrates that lightweight architectures, such as dense neural networks (DNNs), can effectively handle complex tasks that require coordination between two sources of information with optimal latency, especially when coupled with powerful data extraction methods. From an operational standpoint, this system lays the foundation for natural user interfaces (NUIs). It offers a hands-free model in sterile environments such as operating rooms, new accessibility tools for people with reduced mobility, and even intuitive interfaces for early childhood education.

The practical aspects of building an artificial intelligence model to identify hand gestures are detailed.

During this project, an artificial intelligence model was developed, progressing from creating a database of gesture images to real-time gesture recognition. This was achieved using Python, TensorFlow, Keras, and Media.

IV. System Description

The system follows a structured set of steps to predict quick, organized hand gestures and perform mathematical operations. The steps are described in detail in the two algorithms (see Algorithm 1 and Algorithm 2), explained in the following section.

A. Camera Frame Reading

The first step is to keep reading the computer's webcam frame, flip it horizontally, and then convert the color space from BGR to RGB as required by the MediaPipe library.

B. Hand Recognition and Hand Landmark Extraction

The MediaPipe library detects hands and extracts 21 features for each hand (see Fig. 2). It is a pre-trained system that automatically detects the presence of hands in an image. After detection, a set of characteristic points is extracted from each hand. It identifies 21 key points per hand (joints and fingertips) and provides the spatial coordinates for each point. These 21 characteristics constitute a description vector used as input to the proposed gesture recognition system.

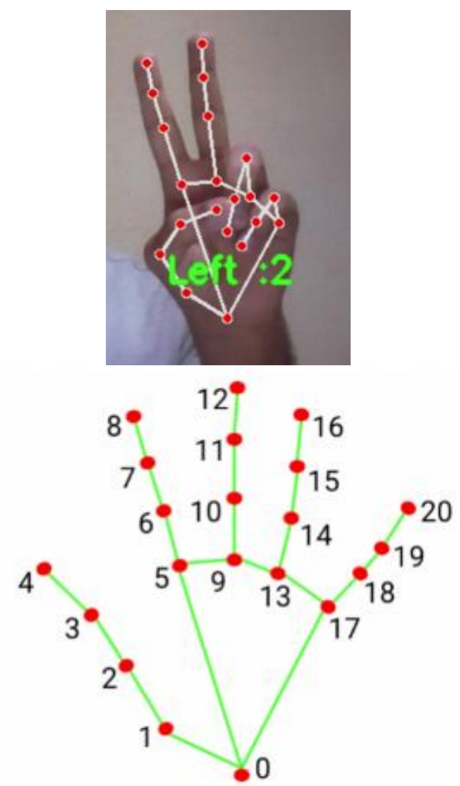

Fig. 2. The 21 hand features using MediaPipe.

C. Data Collection

The database of 13 gestures was created (Fig. 3). Six gestures represent digits 0 to 5, while four gestures correspond to operations (+, -, *, /, and =). The collection was assembled using Python, MediaPipe, and OpenCV. The system captured 21 features from 287 hand gesture samples, which were manually sorted into 0-5 categories and stored in a CSV file. A total of 287 rows were recorded, each containing 42 data points. The 0-5 classification was split with 20% for testing and 80% for training.

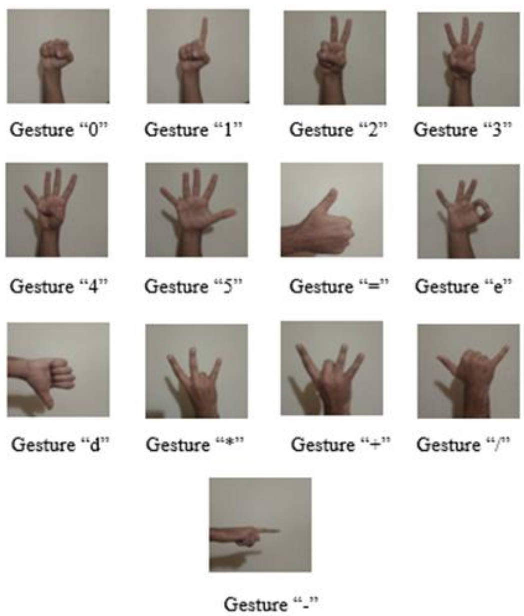

Fig. 3. Dataset of 13 gestures used for the calculating system.

D. Feature Preparation for Prediction

The extracted landmarks are combined into a single feature vector of size 42 (21 x-values and 21 y-values). This vector is reshaped and prepared as input for the trained gesture recognition model.

Algorithm 1: Hand Acquisition and Feature Extraction

```
Initialize
Start Acquisition by Camera
    Initialize MediaPipe Hands module
    Set quit key = ‘q’
Compute
While (pressed key ≠ ‘q’) do
    Get current predicted gesture
    Convert frame to RGB
    Detect and track hands with MediaPipe
        If (at least one hand detected) then
            Extract 42 (x,y) features for each hand
            Send features to classifier
        End
End
```

E. Gesture Prediction

After extracting the feature vector, it makes a prediction and determines which class it belongs to, whether it is a number from 0 to 5, a symbol for arithmetic operations (+, -, *, /, =), or the e command to erase the entire expression, or d to delete the last character. The predicted class is recoded to its corresponding class.

1) Handling numeric gestures (0 to 5): If the hand signal is a number from 0 to 5, the system distinguishes between the right and left hands and stores the value of each. If the two-second cool-down period is completed with the hands together, the system adds the values of the right and left hands and prints the result as a mathematical expression.

Algorithm 2: Gesture-Based Calculator Operation

```
Initialize
Expression ← empty
LeftValue, RightValue ← 0
Compute
While the system running) do
    Acquire current prediction for left hand
    Acquire current prediction for right hand
    If (left hand detected AND right hand detected) then
        If (left gesture is digit 0–5 AND right gesture is digit 0–5) then
            LeftValue ← digit shown by left hand
            RightValue ← digit shown by right hand
            SumValue ← LeftValue + RightValue
            CurrentDigit ← SumValue
        End
    End
    If (CurrentDigit is defined) then
        Append CurrentDigit to Expression
    End
    If (predicted gesture is arithmetic operator '+', '–', '×', '÷') then
        Append operator to Expression
    End
    If (predicted gesture is 'd') then
        Delete last symbol from Expression
    End
    If (predicted gesture is 'e') then
        Clear Expression
    End
    If (predicted gesture is '=') then
        If (Expression is mathematically valid) then
            Evaluate Expression and display result
        Else
            Display error message
        End
    End
End
```

2) Handling symbolic and control gestures: If the sign indicates a mathematical operation symbol (+, -, *, /), the recognized symbol is added to the mathematical expression. If the symbol is "=", the Python eval() function calculates the expression. If the mathematical expression is true, it prints the result. If it is false, it prints an error. If the command symbol is "e," the entire expression is erased.

If the expression symbol is "d", we delete only the last letter of the expression.

3) Display of predictions: The updated expression or result is displayed in real-time on the camera feed using OpenCV's putText function. Users can see their arithmetic operations evolve as they perform gestures.

This model expanded functionality by incorporating symbolic gestures (e.g., arithmetic operators and control commands), turning the system into a real-time gesture-based calculator.

F. Termination Condition

The system continues to run and loop until you press the "q" key to stop the program.

V. System Implementation

A. System Description

The database is a CSV file containing 287 rows, 48 variables, and 13 columns containing numbers from 0 to 5, arithmetic symbols (+, -, *, /, =), and commands (e, d). 80% of the data is dedicated to training and 20% to testing.

B. DNN Model Architecture

The proposed deep neural network (DNN) architecture for gesture classification, shown in Fig. 4, comprises six fully connected (dense) hidden layers, with the number of units decreasing from 1024 to 32.

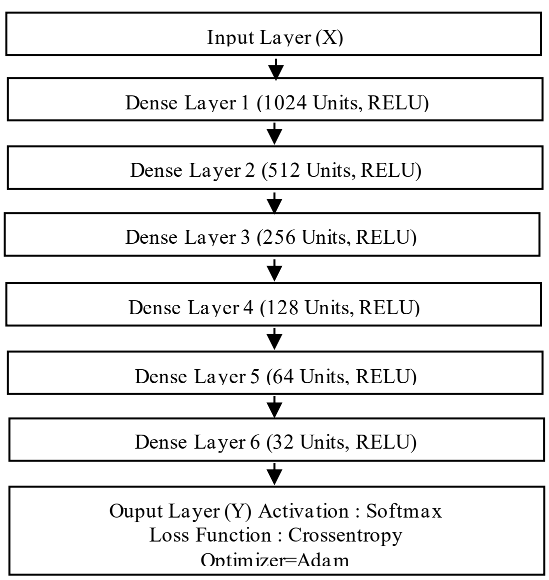

Fig. 4. The proposed deep neural network (DNN) architecture for gesture classification.

Each hidden layer employs the RELU activation function, facilitating hierarchical feature extraction and nonlinear transformations of the input data.

The final output layer consists of 13 units, corresponding to the distinct gesture classes, and utilizes the SoftMax activation function to generate normalized probability distributions for multiclass classification.

The Adam optimizer is chosen for its efficiency in adaptive learning rate adjustment.

This architecture enables robust learning and precise recognition of 13 different hand gestures, rendering the model well-suited for real-time human-computer interaction applications, as evidenced by recent studies employing DNNs in gesture recognition systems.

VI. Results and Discussion

Even with a compact dataset of just 287 samples, our model achieved 91% validation accuracy (Fig. 5), demonstrating its ability to recognize gestures it hasn't seen before reliably. In the world of deep learning, hitting these numbers with such a small dataset is usually a challenge.

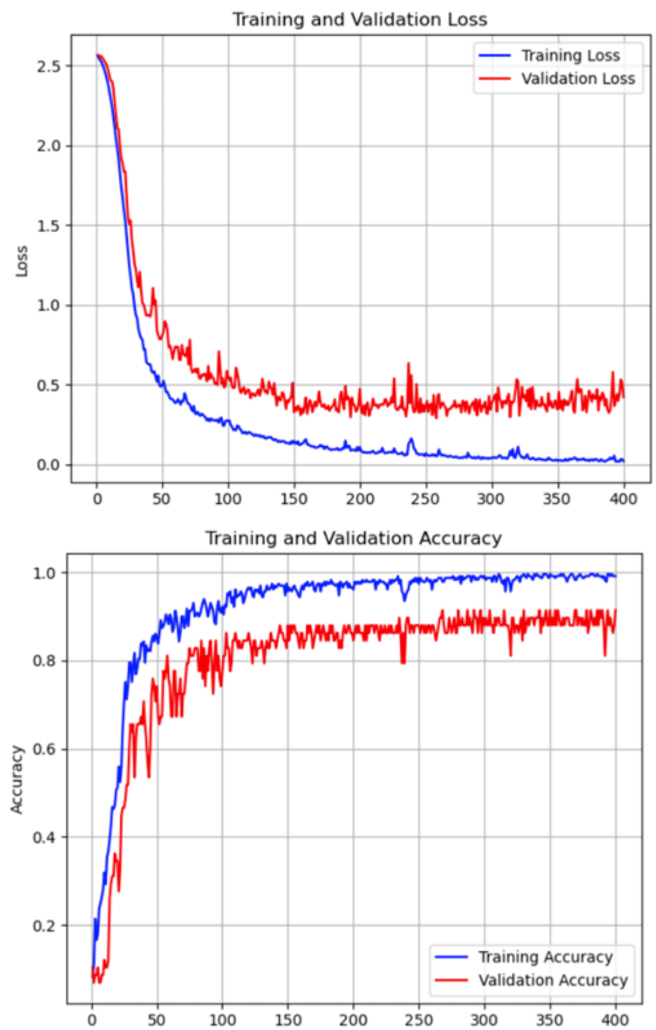

Fig. 5. References Training vs. Validation metrics (loss and accuracy) for the proposed DNN model with MediaPipe.

Using MediaPipe to extract the 21 essential landmarks significantly reduced the size of the features introduced to the learning process compared to the raw input data (image). This significant reduction in data allowed the proposed dense neural network (DNN) to favor hand geometry over environmental features. This also explains why a lightweight model can be just as effective as the complex, data-intensive architectures often encountered in such contexts.

Compared to existing literature, the proposed system strikes a distinctive balance between technical depth and practical accessibility. For example, unlike the massive, highly interconnected CNNs suggested by Tan et al. [6], which require significant GPU capacity, our DNN-based strategy achieves 91% accuracy with significantly lower latency. This makes it much more suitable for real-time clinical or industrial environments [5], [12]. While researchers such as Pathak et al. [2] focus on replacing equipment such as keyboards and mice, this research implements a synchronized logic layer for two-handed arithmetic. This is considered an essential step toward the level of sign language recognition discussed by Long et al. [15]. By relying on landmarks rather than raw pixel processing [4], we achieved competitive results even with a smaller database, effectively bridging the gap between theoretical research and practical tools [13].

Undeniably, the results obtained indicate a slight tendency toward overfitting, which is almost always a challenge when working with restricted data sets. To ensure the system functions properly in real time, this study adopted an approach that aims not only to achieve a higher score but, above all, to guarantee “interaction resilience”. As the system operates at 30 frames per second, feedback is almost instantaneous; in case a gesture is not recognized the first time, the user can naturally adjust the position of their hand without losing rhythm in transparency mode.

During testing, the system remained stable under standard indoor lighting conditions. However, it can still be affected by extremely rapid movements or by a partially obscured hand, thanks to the MediaPipe library, which uses a keypoint inference approach. It is therefore clear that achieving more than these 13 gestures requires much more data. The next steps in our process will involve conducting subject-independent tests to verify that the model is operational for everyone, regardless of hand configuration. Ultimately, this 91% accuracy rate serves as a compelling argument, demonstrating how a MediaPipe DNN pipeline can effectively transform simple detection into a practical, operational calculator.

VII. Conclusion

In this study, a real-time hand gesture recognition system was presented, designed to make interaction with machines more fluid and natural. By combining MediaPipe landmark tracking with an optimized dense neural network (DNN), simple finger counting is exceeded to design a functional two-handed calculator. This work recognizes 13 distinct gestures, allowing users to solve basic mathematical problems with natural movements.

The main scientific goal here is to demonstrate the effectiveness of a lightweight pipeline for two-handed coordination that requires neither high-end hardware nor a large database. It has been clearly proven that reducing the hand image to just 21 key coordinate points combines performance and fluidity, with an accuracy rate of 91%.

This study helps bridge the gap between simple gesture detection and authentic interactive logic, demonstrating that symbolic tasks can be reliably processed using a standard webcam.

In fact, it should be noticed at this level that there is still room for further improvement, considering that the current database has limitations and that the model still perceives gestures as being images rather than continuous movements.

VIII. Future Works

To optimize the system's reliability and efficiency, three main areas are targeted for the next phase of research. Firstly, the database needs to be significantly expanded by building a more diverse panel of examples to ensure the model's accuracy across different hand morphologies and conditions, for better, more accessible technology.

Secondly, although the current configuration is optimal for recognizing static poses, the objective is to explore the “flow” of movements in the context of dynamic gestures. The adoption of more sophisticated architectures, such as LSTMs or Transformers, will allow the system to process each image individually rather than perceiving gestures as continuous movements. This change is crucial for identifying complicated sequences and minimizing disruptions that can occur during real-time tracking.

Finally, the system will undergo much more rigorous testing. We want to observe its behavior in extreme situations, such as rapid gestures, partially obstructed hands, or poor lighting. As things stand, our results have shown that the gesture-controlled calculator is functional. However, to move towards a larger size, it is imperative to improve data and model complexity. This will ultimately pave the way for the application of this technology beyond simple calculator use into more demanding fields such as virtual reality, telesurgery, and robotics.

References

[1] R. Hamdani and I. Chihi, "Adaptive human-computer interaction for industry 5.0: A novel concept, with comprehensive review and empirical validation," *Computers in Industry*, Jun. 2025.

[2] K. Pathak, A. Kalra, S. Patil, A. Joshi, et P. Ahire, "Virtual Mouse and Keyboard Using Hand Gestures," *International Research Journal of Modernization in Engineering Technology and Science*, 15 nov. 2023.

[3] Ponzi, V., Iacobelli, E., Napoli, C., & Starczewski, J. T. (2022). *A Real-time Hand Gesture Recognition System for Human-Computer and Human-Robot Interaction*. In ICYRIME 2022: International Conference of Yearly Reports on Informatics, Mathematics, and Engineering (Catania, August 26-29, 2022). CEUR Workshop Proceedings.

[4] Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. Journal of Imaging, 6(8), 73. https://doi.org/10.3390/jimaging6080073

[5] Amprimo, G., Masi, G., Pettiti, G., Olmo, G., Priano, L., & Ferraris, C. (2023). Hand tracking for clinical applications: validation of the Google MediaPipe Hand (GMH) and the depth-enhanced GMH-D frameworks. ArXiv, published August 2, 2023.

[6] Tan, Y. S., Lim, K., & Lee, C. (2021). Hand gesture recognition via an enhanced densely connected convolutional neural network. Expert

Systems with Applications, 176, 114797. https://doi.org/10.1016/j.eswa.2021.114797

[7] M. Umadevi, P. Nikitha, T. S. Goud, M. Abhinay, and L. Swathi, “Gesture Math AI: Real-Time Math Problem Solving Using Hand Gestures and Computer Vision,” International Research Journal on Advanced Engineering Hub (IRJAEH), May 13, 2025.

[8] D. B. E. Sunanda, V. Bhargavi, V. Yasasri, B. Lahitha, C. S. Sri, and C. Madhuri, “Hand Gesture Based Virtual Mouse and Calculator,” SSRN Electronic Journal, Jan. 1, 2024.

[9] K. Foteinos, J. Cani, M. Linardakis, P. I. Radoglou-Grammatikis, V. Argyriou, P. Sarigiannidis, I. Varlamis, and G. Th. Papadopoulos, “Visual Hand Gesture Recognition with Deep Learning: A Comprehensive Review of Methods, Datasets, Challenges and Future Research Directions,” ArXiv, Jul. 6, 2025.

[10] A. Kendon, *Gesture: Visible Action as Utterance*. Cambridge, UK: Cambridge University Press, 2004.

[11] S. Clough et M. C. Duff, “The Role of Gesture in Communication and Cognition: Implications for Understanding and Treating Neurogenic Communication Disorders,” *Frontiers in Human Neuroscience*, vol. 14, art. 570517, Aug. 2020.

[12] S. Hussain, K. Saeed, A. Baimagambetov, S. Rab, et Md Saad, “Advancements in Gesture Recognition Techniques and Machine Learning for Enhanced Human-Robot Interaction: A Comprehensive Review,” *ArXiv*, 10 Sept. 2024.

[13] Md. Mijanur Rahman, A. Uzzaman, Fatema Khatun, Md Aktaruzzaman, et Nazmul Siddique, “A comparative study of advanced technologies and methods in hand gesture analysis and recognition systems,” Expert Systems with Applications, vol. 244, art. 121964, 1 déc. 2024.

[14] R. S. M. Hirsch, “The Gestalt Hypothesis - On the form and meaning of gestures,” 1993.

[15] Zhiyu Long, Xingyou Liu, Jiaqi Qiao, et Zhi Li, “Sign Language Recognition Based On Facial Expression and Hand Skeleton,” in *2023 38th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 27 août 2023.

[16] S. Clough et M. C. Duff, “The Role of Gesture in Communication and Cognition: Implications for Understanding and Treating Neurogenic Communication Disorders,” *Frontiers in Human Neuroscience*, 11 août 2020.