HAND GESTURE RECOGNITION FOR USER INTERACTION IN AUGMENTED REALITY GAMES

Nguyễn Thị Thanh Tâm*, Duong Doan Tung+
* Posts and Telecommunications Institute of Technology
+ Faculty of Electrical and Electronic Engineering, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi 12116, Vietnam

Abstract: This paper presents a hand gesture recognition system for interactive augmented reality games, utilizing skeletal and image data to improve accuracy. We collected a comprehensive dataset of hand gestures comprising RGB images and skeletal coordinates for five distinct gestures. A Late Fusion model, which combines skeletal data with RGB image information, was proposed and achieved a test accuracy of 88.20%. This model was successfully integrated into a Unity 3D game, allowing players to control in-game actions through intuitive hand gestures. Experimental results demonstrate the effec tiveness of the proposed approach in enhancing user interaction and delivering a highly responsive gaming experience in AR environments.

Keywords: Hand Gesture Recognition, Human Computer Interaction, Augmented Reality, Data Fusion, Transfer Learning, Deep Learning.

I. INTRODUCTION

In the digital era, gesture recognition technology has become a powerful and intuitive medium for human computer interaction, particularly in interactive games. As a natural form of human communication, Gestures convey a wide range of meanings and emotions, mak ing them a rich channel for immersive user experiences. Using hand gestures for interaction in gaming offers an engaging and unique experience for players and contributes to the enhancement of motor skills and cognitive abilities.

Augmented Reality (AR) provides a compelling platform where gesture recognition can be effectively applied, transforming user interaction by allowing players to manipulate virtual objects in a natural and immersive environment seamlessly.

In this study, we focus on applying hand gesture recognition to AR action games, which particularly involve a variety of gestures, ranging from simple and distinct to complex and very similar. AR action games overlay virtual elements onto the real world, allowing players to interact with digital objects in their physical environment. Hand gesture recognition in this context can significantly enhance player immersion and control, enabling more intuitive interactions such as aiming, shooting, and manipulating virtual items without the need for physical controllers. However, developing reliable and accurate gesture recognition systems for AR poses significant challenges due to factors such as gesture variability, lighting conditions, and system latency.

To address these challenges, we have collected a comprehensive dataset of RGB images and skeletal hand frames for various hand gestures. Building on this dataset, we conducted extensive experiments to develop a gesture recognition model. Based on a late fusion of skeletal and RGB data using deep learning architectures, the resulting model has been integrated into an interactive AR-based game developed using Unity 3D, allowing players to control in-game objects using natural hand gestures.

In this paper, we make the following contributions:

1) We introduce a novel dataset comprising RGB images and skeletal coordinates for five distinct hand gestures.

2) We conduct and present a comprehensive study on different classification methods to highlight their characteristics.

3) We propose a late fusion gesture recognition model that combines skeleton and RGB data.

4) We demonstrate possible methods to integrate the model into a game engine, specifically the Unity engine.

II. RELATED WORKS

Hand gesture recognition (HGR) has emerged as a pivotal technology in enhancing user interaction within AR systems. The integration of HGR into AR facilitates a more immersive experience and allows for intuitive control mechanisms that align closely with natural human communication methods. The foundation of effective HGR lies in its ability to interpret human gestures as commands within a digital environment. As noted by Yousefi and Li, the analysis of 3D hand gestures is crucial for developing a robust human device interaction system, particularly in AR applications where gestures can serve as natural inputs for controlling virtual elements [1]. For instance, the family of grab gestures has been identified as particularly effective in 3D scenarios, providing a basis for user interactions that feel intuitive and seamless [1]. This is echoed by Piumsomboon et al., who highlight the importance of user-defined gestures in AR, allowing players to interact with virtual content in a

Tác giả liên hệ: Nguyễn Thị Thanh Tâm,
Email: ntttam@ptit.edu.vn
Đến tòa soạn: 8/2024, chỉnh sửa: 9/2024, chấp nhận đăng:10/2024.

manner that feels both natural and engaging [2]. Deep learning methodologies have significantly advanced the capabilities of HGR systems. Recent studies, such as those by Li et al., demonstrate the effectiveness of neural networks in recognizing dynamic gestures, which are essential for interactive gaming environments [3]. The Attentive 3DGhost Module proposed by Li et al. enhances gesture recognition by leveraging positive knowledge transfer, thereby improving the accuracy and responsiveness of gesture-based interactions [3]. This technological advancement is crucial for AR games, where real-time feedback is essential for maintaining user engagement and immersion. Moreover, the application of HGR in AR games extends beyond mere gesture recognition; it encompasses creating an interactive experience that blends physical and virtual worlds. For instance, the work by Xu emphasizes the market potential for HGR technologies, projecting significant growth as these systems become integral to user interfaces in AR and VR environments [4]. This growth is supported by the increasing demand for intuitive control systems that do not rely on traditional input devices, enhancing the overall user experience. The role of hand gestures in ARis further underscored by the findings of Sch¨afer et al., who discuss the importance of arbitrary one-handed gestures in AR applications [5]. This flexibility allows users to engage with virtual environments without the constraints of physical controllers, thereby fostering a more immersive experience. The ability to perform gestures freely in a 3D space aligns with users' natural movements, making interactions feel more organic and less mechanical. In addition to enhancing user experience, HGR also plays a critical role in accessibility. For individuals with disabilities, gesture-based interfaces can provide alternative means of interaction without traditional input methods. Research by Kang et al. highlights the potential of sEMG-based recognition systems, which can interpret gestures through muscle signals, thus offering new avenues for interaction in AR environments [6]. This capability is particularly relevant in gaming, where inclusivity can significantly broaden the audience and enhance engagement. Furthermore, integrating HGR in AR games can lead to innovative gameplay mechanics that leverage the unique capabilities of gesture recognition. For example, the use of advanced models like Vision Transformer (ViT) for static hand gesture recognition can enhance the expressiveness of player interactions, allowing for more complex and nuanced gameplay experiences[7]. By accurately interpreting a wider range of gestures in realtime, players can engage with the game environment in more dynamic and immersive ways, ultimately enriching their overall gaming experience. This is particularly relevant in multiplayer settings, where gestures can serve as non-verbal communication tools, enriching the social dynamics of gameplay. However, the challenges associated with HGR in AR cannot be overlooked. Issues such as occlusion, varying lighting conditions, and the need for real-time processing pose significant hurdles. As noted by Zhao, traditional camera systems often struggle with depth perception and field of view limitations, which can hinder gesture recognition accu racy [8]. Addressing these challenges requires ongoing research and development, particularly in the areas of computer vision and machine learning.

Image-based methods use processing techniques to extract features from hand gestures captured in images or videos. While they do not require specialized equipment, they are highly susceptible to noise and image resolution issues. Notable works in this area include [9], [10], and [11], which explore various image processing algorithms for gesture recognition.

Skeleton-Based methods leverage sensors like Leap Motion or Kinect to capture the position and orientation of finger joints. They offer higher accuracy and stability but require expensive and complex hardware. Studies such as [12], [13], and [14] have demonstrated the effectiveness of skeleton-based approaches in gesture recognition.

In conclusion, integrating HGR into AR systems significantly advances human-computer interaction. By leveraging natural gestures, developers can create immersive experiences that resonate with users on a deeper level. As technology continues to evolve, the potential for HGR in AR will likely expand, paving the way for more intuitive, engaging, and accessible experiences. Future research should focus on enhancing the robustness of gesture recognition systems, exploring new interaction paradigms, and ensuring that these technologies are inclusive and user-friendly.

III. METHODOLOGY

In this section, we outline the comprehensive approach adopted to achieve the objectives of our study. The methodology is designed to ensure the reliability and validity of the results, encompassing various stages from initial planning to final analysis.

To develop a robust gesture recognition model, we collected a comprehensive dataset comprising hand gesture images and skeletal data. The gestures were selected with two key considerations in mind: first, the initial three gestures are easily distinguishable, serving to confirm the effectiveness of the model; second, the last two gestures are very similar to each other, mimicking scenarios where gestures are not always distinct. These considerations ensure their relevance and applicability in interactive game scenarios.

Figure 1. Examples of hand gestures used in the dataset for gesture recognition model development, demonstrating both distinct (Idle, Use item, Pickup item) and similar (Aim, Shoot) gestures.

The game scenario we choose involves a shooting, level-based mechanism where players navigate through

various zones, collecting items and engaging with enemies. The gestures include: Pickup Item, Use Item, Aim, Shoot, and Idle. These gestures are described in Figure 1.

A. Features

First, a camera is placed in front of the user, capturing an RGB image stream of the person performing the gesture. Then, MediaPipe [15] is used to extract skeleton data and the bounding box of the hands. From the bounding box, we can further crop the image to get only images of the hands. After these steps, we end up with three types of data: raw images, cropped images, and skeleton data.

• Raw images: These are the original images captured by the camera, containing the entire scene, including the person and their surroundings.

• Cropped images: These images are cropped to include only the hands, based on the bounding box provided by MediaPipe.

• Skeleton data: This consists of the landmarks information of both hands extracted by MediaPipe, which includes the positions of key points on the hands.

B. Hand Gesture Recognition Model

We propose a Late Fusion model comprising a Convolutional Neural Network (CNN) for skeletal data feature extraction and a transfer learning network with a MobileNetV2 base pre-trained on ImageNet for image data extraction, as illustrated in Figure 2.

We chose the Late Fusion model for several key reasons. First, this approach effectively combines the strengths of both skeletal data and RGB image data, significantly enhancing gesture recognition accuracy. By leveraging the unique characteristics of each data type, the model achieves a more comprehensive understanding of gestures.

Moreover, the Late Fusion model is particularly adept at handling variability, which is a common challenge in AR environments. The integration of skeletal and image data allows the model to adapt to variations in gesture execution and lighting conditions, thereby improving its robustness in real-world scenarios.

Furthermore, the fusion approach enables the model to capture high-level features from images while simultaneously analyzing detailed hand movements from skeletal data. This dual capability leads to more accurate and reliable gesture recognition outcomes.

To prepare the skeletal data for processing, it was normalized to ensure consistency, with the thumb positioned on the left and the pinky on the right. The angles between finger joints were calculated using the *atan2* function, resulting in 15 distinct features that serve as input for the model.

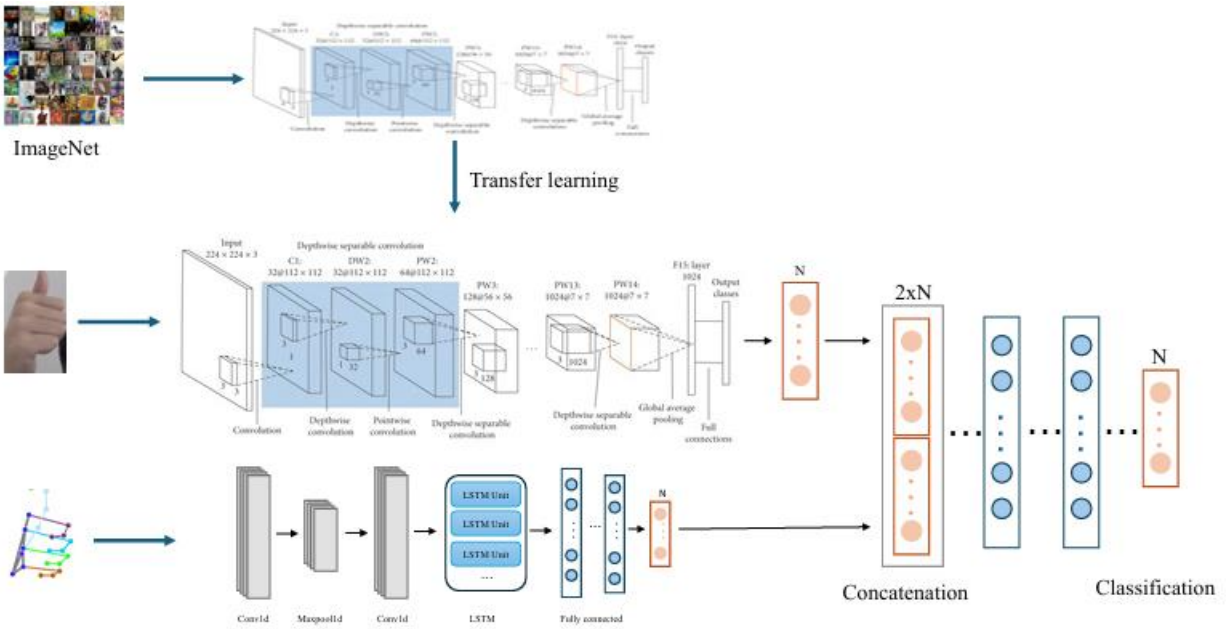

Figure 2. Proposed architecture for late fusion hand gesture recognition.

The CNN architecture consists of several layers designed to process the skeletal data. Initially, the input data is passed through a series of convolutional layers. The first convolutional layer has 64 filters with a kernel size of 3, followed by a max-pooling layer with a kernel size of 2. This is succeeded by a second convolutional layer with 128 filters and the same kernel size, followed by another max-pooling layer. The output from the convolutional layers is then reshaped and fed into an LSTM layer with 100 hidden units. Finally, the LSTM output is passed through two fully connected layers, with the first layer having 50 units and the second layer producing the final output corresponding to the number of gesture classes.

The transfer learning network utilizes a MobileNetV2 architecture, which is pre-trained on the ImageNet dataset. This network is employed to extract high-level features from the image data. The pretrained MobileNetV2 model is fine-tuned on our HGR dataset to adapt it to the specific task of gesture classification. The extracted features from both the DNN and the transfer

learning network are then fused at a later stage to make the final prediction.

C. Game Intergration

The HGR model can be integrated into a Unity 3D game in two different ways, allowing players to control in-game actions using their gestures. The model processes the gestures in real-time, transmitting the recognized commands to the game via the User Datagram Protocol (UDP). This setup ensures minimal latency and a responsive gaming experience.

The first integration method involves using a separate Python script that performs inference in the background. This script processes the gesture data and sends the recognized commands to the game through localhost using UDP. This approach allows for efficient communication between the HGR model and the game, ensuring that the player's gestures are accurately reflected in real-time within the game environment.

The second integration method involves exporting the HGR model to the Open Neural Network Exchange (ONNX) format and performing inference using Unity's Barracuda library. This approach allows the model to be directly embedded within the Unity game, leveraging Barracuda for efficient on-device inference. This method reduces the dependency on external scripts and can potentially improve the overall performance and responsiveness of the game.

In our experimental game scenario, we also implemented head movement detection for character navigation. This was achieved using the Haar cascades classifier, which detects the user's head movements and translates them into navigation commands. The character's pathfinding was managed using the A* algorithm, ensuring efficient and accurate movement within the game environment.

IV. EXPERIMENT

A. Data Collection

Data was collected on 14 participants using a custom-built data acquisition tool and a set of sample gesture images. Participants were instructed to perform each gesture in front of a camera. Each participant contributed approximately 1,200 images per gesture. After filtering out low-quality images and duplicates, we obtained a total of 62,680 images and corresponding skeletal data[1].

The collected data was automatically labeled and organized into separate folders for each gesture. The dataset was then divided into training, validation, and testing sets by participants to ensure a balanced and comprehensive model evaluation. Specifically, data from 8 participants was used for training, data from 2 participants for validation, and data from 4 participants for testing.

Table I. data distribution across training, validation, and test sets.

	Number of Participants	Data Points
Train	8	35716
Validation	2	10990
Test	4	15974

B. Ethical Considerations

All data collection and experiments were conducted in accordance with institutional ethical guidelines. Participant consent was obtained prior to collecting hand gesture data by signing a consent form, which explained the purpose of the study, the procedures involved, and the measures taken to ensure confidentiality. We ensured personal identifying information was anonymized, with each participant assigned a unique 4-digit number for identification in the dataset.

C. Implementation Details and Hyperparameter Tuning

The model was implemented using the PyTorch framework, which offers the flexibility and efficiency necessary for our deep learning tasks. To optimize the performance of our handgesture recognition system, we conducted a series of experiments with various architectures and hyperparameters.

Hyperparameter tuning was a crucial aspect of our model optimization process. We systematically explored different configurations to identify the best values for key hyperparameters. The final selected hyperparameters for our model are as follows:

- **Learning Rate:** 1×10^{-2} (initial learning rate)
- **Batch Size:** 64
- **Total Epochs:** 10

Additionally, we implemented a learning rate scheduler to gradually reduce the learning rate to 10% of the original rate every 3 epochs. This strategy was essential in ensuring stable training and improved convergence of the model, which is proven effective by Smith et al. in Cyclical Learning Rates for Training Neural Networks [16], who demonstrated that using a learning rate scheduler can significantly enhance the training process and improvemodel performance in neural networks.

D. Models Evaluation

We conducted experiments on both image-based and skeleton-based models to obtain comprehensive results. As described in Section III, our proposed model comprises a CNN for skeletal data and a transfer learning network for image data.

For the skeletal data, we experimented with two different scenarios: using only normalized landmark positions and using calculated angles between adjacent joints. For the image data, we tested different basemodels, including ResNet50, MobileNetV2, and DenseNet121, using both raw images and cropped images to include only the hands.

E. Evaluation Metrics

We evaluated the modelp erformance using accuracy, precision, recall, and F1 score metrics. These metrics provide a comprehensive assessment of the model's ability to correctly recognize gestures. Additionally, the game performance was assessed, focusing on the responsiveness, delay, and accuracy of gesture-based controls. This evaluation ensures that themodel not only performs well in a controlled environment but also

[1] Code and dataset https://github.com/dtungpka/HGR-AR

provide sasatisfactory user experience in real-time applications.

V. RESULTS

In this section, we present the results of our experiments.

A. Skeleton Model

We evaluated the performance of the skeleton-based models using two different feature sets: landmark positions and calculated angles between adjacent joints. The results are summarized in Table II.

Table II. data comparison of key performance metrics between the two models.

Metric	Landmark Model	Joint Angle Model
Test Accuracy (%)	78.50	80.54
Precision	0.78	0.81
Recall	0.78	0.81
F1-Score	0.78	0.80

Both models demonstrate satisfactory performance for most gesture classes. However, when comparing the two, we observe that the joint angle-based model consistently achieves better results, especially for the more challenging gestures in Class 4 and Class 5. These classes had poor recall in the landmark position-based model, indicating a higher rate of false negatives. The joint angle approach helps mitigate this issue by providing a more informative representation of hand pose.

1) Results Using Landmark Positions: The model using landmark positions achieved average test accuracy of 78.50%. The average precision, recall, and F1 score were all 0.78. The detailed class-wise performance is shown in Table III.

Table III. performance of the model using landmark positions.

Class	Precision	Recall	F1-Score
1	0.99	0.98	0.99
2	0.90	0.97	0.93
3	0.94	0.87	0.91
4	0.60	0.77	0.67
5	0.51	0.32	0.39

The model already demonstrates strong performance for most gesture classes, especially Class1 and Class2, which achieve near-perfect precision and recall scores.

2) Results Using Calculated Angles: The second model, trained using calculated joint angles, slightly outperforms the first model, with an average test accuracy of 80.54% and a reduced test loss of 0.0343. The model achieves an overall precision of 0.81, recall of 0.81, and F1-score of 0.80. Table IV provides the detailed class-wise performance metrics.

This model improves classification for the poorly performing classes in the previous model. For Class 4, the F1-score remains modest at 0.64, but this represents a notable improvement over the previous model. Similarly, Class5 improves to an F1-score of 0.60, significantly higher than the performance using landmark positions.

Table IV. performance of the model using joint angles.

Class	Precision	Recall	F1-Score
1	0.97	0.99	0.98
2	0.89	0.97	0.93
3	0.96	0.87	0.91
4	0.65	0.64	0.64
5	0.60	0.60	0.60

B. Image Model

On the other hand, using RGB data for HGR produced varying results across different models. Table V summarizes the performance metrics for all image-based models.

Table V. performance of image models on hgr

Model	Accuracy (%)
ResNet50 (Raw)	61.78
MobileNetV2 (Raw)	75.02
DenseNet121 (Raw)	74.54
ResNet50 (Cropped)	76.80
MobileNetV2 (Cropped)	**85.48**
DenseNet121 (Cropped)	80.82

The highest accuracy was achieved by using MobileNetV2 as a base model on cropped images, with a test accuracy of 85.48%, precision, and F1 score of 0.85. Cropping the images to focus solely on the hand region proved to be a critical factor in improving the model's performance across the board. In contrast, models trained on raw images demonstrated significantly lower accuracy and precision, with ResNet50 (raw) performing the worst, achieving only 61.78% accuracy and an F1score of 0.58.

1*) Model Comparison and Analysis:* Among the cropped image-based models, MobileNetV2 stands out as the best performer. Its architecture, which balances the depth and width of the network, appears to be well-suited to the HGR task, especially when combined with image cropping. DenseNet121 also performed well on cropped images, yielding an accuracy of 80.82%, but fell short compared to MobileNetV2.

The ResNet50 model, while showing significant improvement on cropped images (76.80%), performed notably worse than the other models, indicating that deeper networks may not necessarily perform better for this task.

2) Comparison with Skeleton Model: When comparing the image-based models to the skeleton model (Section 5.1), the best image-based models (MobileNetV2 and DenseNet121) out performed the skeleton models in terms of accuracy and precision. The skeleton model's highest accuracy was 80.54%, achieved with joint angle data. In contrast, MobileNetV2 achieved 85.48%, indicating that the image data, particularly when cropped, captures essential gesture information more effectively. However, for the more challenging gestures (Class 4 and Class 5), the skeleton model showed relatively better

performance in recall, reducing false negatives, especially when using joint angle representations.

C. Proposed Late Fusion Model

Table VI. performance comparison of rgb-based, skeleton based, and late fusion models on hgr

Model	Accuracy (%)
ResNet50 (Raw)	61.78
MobileNetV2 (Raw)	75.02
DenseNet121 (Raw)	74.54
ResNet50 (Cropped)	76.80
MobileNetV2 (Cropped)	85.48
DenseNet121 (Cropped)	80.82
Skeleton (Landmark)	78.50
Skeleton (Joint Angle)	80.54
Late Fusion (Our proposed method)	**88.20**

As shown in Table VI, the *Late Fusion model* achieved a test accuracy of 88.20% with a test loss of 0.0339, indicating a strong generalization capacity on the test set. The model's precision, recall, and F1 score were balanced across all classes, with values of 0.87, 0.88, and 0.88, respectively. These metrics demonstrate that the fusion approach mitigates the shortcomings observed in the separate models.

Table VII. precision, recall, and f1-scores for the late fusion model across all gesture classes.

Class	Precision	Recall	F1-Score
1	0.76	0.71	0.74
2	0.98	0.98	0.98
3	0.88	0.97	0.92
4	0.83	0.77	0.80
5	0.92	0.98	0.95

For example, the *image-only model* achieved a test accuracy of 85.48%, while the *skeleton-only model* reported a maximum accuracy of 80.54% (refer to Skeleton Model section). This shows that combining both data types improves recognition accuracy, particularly in gestures where image or skeleton data alone are insufficient.

When compared to the *skeleton-only model*, the Late Fusion model shows a marked improvement across all classes, especially for similar gestures like Class 4 (*Aim*) and Class 5 (*Shoot*). The skeleton model struggled with these classes, achieving F1-scores of 0.67 and 0.60, respectively, indicating a higher rate of false negatives. The fusion approach significantly reduced this error rate by incorporating contextual information from RGB images.

Similarly, the *image-only model* performed well on simple gestures but failed to capture the finer details of hand positioning required for more complex gestures like Class 4. The image-only model's F1-score for Class 4 was 0.64, compared to 0.80 achieved by the fusion model. This demonstrates that the addition of skeletal data helps to capture nuances in hand orientation and joint angles, which are critical for recognizing subtle movements.

D. Game Integration Performance

In this section, we presents the performance metrics related to game integration for the models utilized in our study.

1) Using UDP to Send Inference Results: One advantage of utilizing the UDP protocol for transmitting inference results is that the model is run on PyTorch, which efficiently handles complex model architectures. The average inference times for the various models are summarized in Table VIII.

Table VIII. average inference times

Model	Average Inference Time (ms)
ResNet	29.5
MobileNetV2	**14.5**
DenseNet121	33.6

However, a downside to this method is the additional time required for sending data to the Unity game using UDP, which averages around 7-10 ms. When combined with the average 38 ms taken to process skeleton data using MediaPipe, the total processing time amounts to approximately 76 to 79 ms per frame.

2) ONNX: The ONNX model can be run directly in Unity using Barracuda, which makes it more compact and easier to deploy. Despite these advantages, the ONNX model consistently takes between 60.734 ms and 73.131 ms to produce results for each frame. Furthermore, preprocessing skeleton data using MediaPipe plugin adds an additional 45-50 ms. Consequently, the total time for the ONNX model, including preprocessing, ranges from approximately 105 ms to 123 ms per frame.

E. Limitations

Although our proposed hand gesture recognition model demonstrates promising results, several limitations should be noted:

• The dataset size was relatively small, comprising only from 14 participants, which may affect the model's generalizability.

• Our testing was limited regarding subjects with hand mobility issues, which may restrict the applicability of our model for all users.

• Additionally, the computational requirements of our model may limit its real-time applications on resource-constrained devices, potentially hindering broader implementation.

To enhance the robustness and applicability of our system, future work should focus on addressing these limitations through larger-scale data collection and optimization strategies tailored for edge devices.

VI. CONCLUSIONS

In this study, we explored the application of gesture recognition technology in an interactive AR game. We developed a dataset comprising both hand gesture images and skeletal data and trained a deep learning model to recognize five distinct gestures. Our proposed Late

Fusion model, which combines skeletal and image data, achieved a test accuracy of 88.20%, demonstrating its effectiveness in recognizing complex gestures. By integrating this model into a Unity 3D game, we highlighted the potential of gesture recognition to enhance interactive gaming experiences. Future work will focus on improving model accuracy and expanding the set of recognized gestures to enrich user interactions further.

ACCKNOWLEGDE

This research is funded by Posts and Telecommu nications Institute of Technology under grant number 02-2024-HV-ĐPT.

REFERENCES

[1] S. Yousefi and H. Li, "3d hand gesture analysis through a real time gesture search engine," International Journal of Advanced Robotic Systems, 2015.

[2] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn, "User-defined gestures for augmented reality," 2013.

[3] J. Li, R. Liu, D. Kong, S. Wang, L. Wang, B. Yin, and R. Gao, "Attentive 3d-ghost module for dynamic hand gesture recognition with positive knowledge transfer," Computational Intelligence and Neuroscience, 2021.

[4] X. Xu, "Training-free acoustic-based hand gesture tracking on smart speakers," Applied Sciences, 2023.

[5] A. Sch¨ afer, G. Reis, and D. Stricker, "Anygesture: Arbitrary one-handed gestures for augmented, virtual, and mixed reality applications," Applied Sciences, 2022.

[6] S. Kang, H. Kim, C. Park, Y. Sim, S. Lee, and Y. Jung, "semg-based hand gesture recognition using binarized neural network," Sensors, 2023.

[7] C. K. Tan, K. M. Lim, R. K. Y. Chang, C. P. Lee, and A. Alqahtani, "Hgr-vit: Hand gesture recognition with vision transformer," Sensors, vol. 23, no. 12, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/12/5555.

[8] R. Zhao, "Large-field gesture tracking and recognition for augmented reality interaction," Journal of Physics Conference Series, 2023.

[9] J.-H. Sun, T.-T. Ji, S.-B. Zhang, J.-K. Yang, and G.-R. Ji, "Research on the hand gesture recognition based on deep learning," in 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), 2018, pp. 1–4.

[10] N. Mohamed, M. B. Mustafa, and N. Jomhari, "A review of the hand gesture recognition system: Current progress and future directions," IEEE Access, vol. 9, pp. 157422–157436, 2021.

[11] D. Sarma and M. Bhuyan, "Methods, databases and recent advancement of vision-based hand gesture recognition for hci systems: A review," SN Computer Science, vol. 2, 11 2021.

[12] C. Li, X. Zhang, L. Liao, L. Jin, and W. Yang, "Skeleton-based gesture recognition using several fully connected layers with path signature features and temporal transformer module," 2018. [Online]. Available: https://arxiv.org/abs/1811.07081.

[13] J. Liu, Y. Wang, S. Xiang, and C. Pan, "Han: An efficient hierarchical self-attention network for skeleton based gesture recognition," 2021. [Online]. Available: https://arxiv.org/abs/2106.13391

[14] X. Nguyen, L. Brun, O. Lezoray, and S. Bougleux, "Skeleton-based hand gesture recognition by learning spd matrices with neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1905.07917

[15] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[16] L. N. Smith, "Cyclical learning rates for training neural net works," in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 464–472.

[17] and speech impaired people," in *2022 IEEE Students Conference on Engineering and Systems (SCES)*. IEEE, 2022, pp. 1–6.

NHẬN DẠNG CỬ CHỈ TAY CHO TƯƠNG TÁC NGƯỜI DÙNG TRONG TRÒ CHƠI THỰC TẾ TĂNG CƯỜNG

Tóm tắt: Bài báo này giới thiệu hệ thống nhận dạng cử chỉ tay dành cho trò chơi thực tế tăng cường có tính tương tác, tận dụng dữ liệu xương và hình ảnh để cải thiện độ chính xác. Một tập dữ liệu cử chỉ tay được thu thập bao gồm hình ảnh RGB và tọa độ khung xương cho năm cử chỉ khác nhau. Mô hình Late Fusion, kết hợp dữ liệu xương với thông tin hình ảnh RGB, đã được đề xuất và đạt độ chính xác kiểm thử 88,20%. Mô hình này đã được tích hợp thành công vào trò chơi Unity 3D, cho phép người chơi điều khiển các hành động trong trò chơi thông qua các cử chỉ tay trực quan. Kết quả thực nghiệm cho thấy hiệu quả của phương pháp đề xuất trong việc tăng cường tương tác người dùng và mang lại trải nghiệm trò chơi có độ phản hồi cao trong môi trường thực tế tăng cường.

Từ khóa: Nhận dạng cử chỉ tay, Tương tác Người-Máy, Thực tế tăng cường, Kết hợp dữ liệu, Học chuyển giao, Học sâu.

Nguyen Thi Thanh Tam received her bachelor's degree in Information Technol ogy from Thai Nguyen University in 2004. In 2017, she received her degree of master of science in Information systems from the University of Engineering and Technology, Ha Noi National University. Tam has 15 years of teaching experience. Currently, she is a lecturer at the Faculty of Multimedia- The Posts and Telecommunications Insti tute of Technologies (PTIT), Hanoi, Viet nam. Her research interests include Machine Learning, Multimedia Application Development.

Email: ntttam@ptit.edu.vn

Duong Doan Tung is a third-year student at the Faculty of Electrical and Electronic Engineering, Phenikaa University, Viet nam, studying toward a Bachelor of Engi neering with a specialization in Robotics and Artificial Intelligence. His research interests include Computer Vision, Rein forcement Learning, and Human-Robot In teraction.

Email: 21010294@st.phenikaa-uni.edu.vn