Hand Gesture Recognition using CVZONE

Tran Tan Thien

thienttse171043@fpt.edu.vn

FPT University
Ho Chi Minh City, Viet Nam

Nguyen Anh Hao

haonase171792@fpt.edu.vn

FPT University
Ho Chi Minh City, Viet Nam

Ho Quang Hien

hienhqse173123@fpt.edu.vn

FPT University
Ho Chi Minh City, Viet Nam

Ngo Duc Toan

toanndse173680@fpt.edu.vn

FPT University
Ho Chi Minh City, Viet Nam

Vu Ngoc Khanh

khanhvnse170082@fpt.edu.vn

FPT University
Ho Chi Minh City, Viet Nam

Huynh Le Trung Viet

viethltse170565@fpt.edu.vn

FPT University
Ho Chi Minh City, Viet Nam

ABSTRACT

This research investigates the use of the 'cvzone' library for hand gesture recognition, which is an important part of improving human-computer interaction. We create a system capable of interpreting and responding to a range of hand motions using computer vision techniques and machine learning. Our strategy, which real-time image processing, yields encouraging results in terms of accuracy and responsiveness. This technology's potential uses span from virtual reality and gaming to assistive devices, all of which promise a more intuitive and engaging user experience.

KEYWORDS

cvzone, hand gesture recognition, mediapipe hand landmark

1 INTRODUCTION

We are currently living in the Industry 4.0 era. The industrial rev- olution is gaining momentum. Artificial intelligence, automation, big data, and the internet of things (IoT) are always required in in- dustries. Data centers are critical components of the modern digital world, acting as the foundation for the expanding demand for cloud computing and online services. With ever-increasing data volumes and the demand for real-time processing, data center energy usage has become a major concern. The global expense and environmen- tal consequences of high energy consumption require novel ways to increase energy efficiency in these facilities.

In this research, we look at the development and testing of a 'cvzone'-based hand gesture identification system. Our key goals are as follows:

1. Research: 'cvzone' library overview. This section provides an overview of the 'cvzone' library, introducing its features, core capabilities, and integration with 'OpenCV' for computer vision tasks.

2. Construction: Build a model based on the functions of image processing, evaluation and recognition of hand gesture.

In conclusion, the purpose of this research is to investigate the possibilities and challenges of implementing hand motion detection with the 'cvzone' library. We hope to contribute to the advancement of natural and intuitive human-computer interactions by harnessing the power of computer vision and machine learning, ushering in a new era of technology accessibility and user engagement.

2 RELATED WORK

The field of hand gesture recognition has made significant advances through the use of Python libraries such as CVZone, e.g. This study [1] was conducted on a humanoid robot with an upper body im-plementing CVZone for real-time hand gesture recognition. Perfor-mance was evaluated at different distances and brightness levels, with the highest success rate observed at the shortest distance (50 cm) in medium light environments. In another survey [2], hand gesture recognition was discussed in the context of enhancing user interaction in artificial intelligence. This research also uses Python libraries, including OpenCV and cvzone2, for image capture, prepro- cessing, and image detection. These tools are used in conjunction with mapped action pairs to perform specific tasks. A real-time on-device hand gesture recognition (HGR) system was presented in another study [3]. The system is capable of detecting a predefined set of static gestures from an RGB camera, showing potential for real-time applications. Furthermore, this study [4] leveraged a large ASL dataset and advanced techniques to capture the intricate details and movements of ASL gestures accurately. This study highlights the potential of these techniques in understanding complex ges- tures. On the other hand, with a small dataset, there is a comparison of some methods [5]. Wide-DenseNet without transfer learning and Prototipical Networks with transfer learning were figured out to have the best result. In the context of Data-Free Class-Incremental Learning, this study [6] presents BOAT-MI, a Boundary-Aware Pro-totype Model Inversion, which delves deeper into the choice of the best sample for inversion thereby bringing significant improve-ments compared to traditional methods. Another technique for hand recognition problems is using a Temporal Convolution Net- work for Hand Action Recognition Framework [7]. The framework employs a simplified skeleton representation, standardization steps to enhance generalization, and a motion summarization module for efficient per-frame descriptor processing. With low-resolution thermal images, M.Vandersteegen et al. [8] introduced a hand ges- ture recognition model with a lightweight, low-latency algorithm, which combines a 2D Convolutional neural network (CNN) with a 1D Temporal Neural Network (TCN). This study addresses the challenges of low-cost processors. These studies provide a compre- hensive overview of the current state-of-the-art techniques in hand gesture recognition using CVZone. However, there is still room for improvement and exploration which this study aims to achieve.

3 METHOD

Hand Recognition, an intricate field that has undergone decades of extensive research and development, necessitates the application of sophisticated algorithms and advanced techniques. With of the challenges within this problem, we introduce 'cvzone' as a stream- lined solution for hand recognition, meticulously developed on the MediaPipe framework. Cvzone distinguishes itself by delivering exceptional performance, easy for use and rapid data processing, facilitated by its robust hand detection algorithm. It also boasts high practicality, with integrated functions for the preprocessing of raw data sourced from images and videoss. The foundation of cvzone on the Google MediaPipe platform ensures both operational stability and commendable performance. This innovative approach signifies a substantial leap forward in the realm of hand recognition, elevating the precision and efficiency of recognition procedures through the judicious amalgamation of contemporary technology and advanced algorithms.

In the context of Hand Recognition, CVZone has developed a solution based on the MediaPipe Hand Landmarker framework. This is a straightforward approach to detect and identify key points on the hand in images. This method operates on image data using Machine Learning, be it static data or a continuous stream, and sub-sequently provides information such as key landmarks on the hand based on image coordinates and information regarding whether it's the left or right hand for multiple detected hands. The model for identifying these hand landmarks consists of 21 coordinates rep-resenting the positions of hand joints. A connecting line is drawn between these key points, and the angles between them are metic- ulously calculated. This intricate process ensures the precision of hand gesture detection in a realtime setting. To reduce complexity during operation, this method utilizes the region defined by the landmark points in the initial frame to determine the hand's posi- tion in subsequent frames. The hand recognition method is only activated when no hand is detected or when tracking is lost. This helps reduce the frequency of re-invoking the recognition model from the beginning, thereby improving the model's performance.

The cvzone method represents a significant advancement in the field of hand recognition, grounded in the technical prowess of the

MediaPipe framework. Its streamlined and dependable nature positions it as a valuable tool for a wide range of scenarios. The method's continued evolution through focused research and devel-opment will be instrumental in addressing current limitations and expanding its utility.
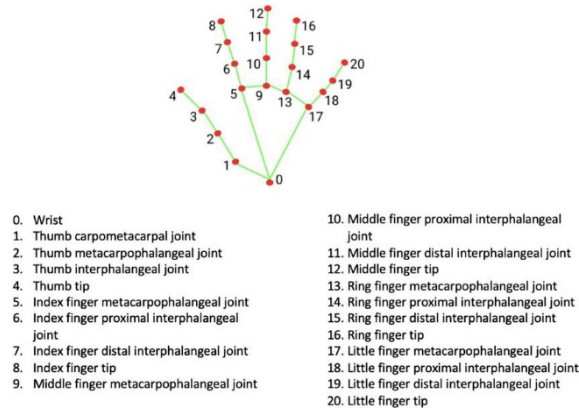

0. Wrist
1. Thumb carpometacarpal joint
2. Thumb metacarpophalangeal joint
3. Thumb interphalangeal joint
4. Thumb tip
5. Index finger metacarpophalangeal joint
6. Index finger proximal interphalangeal joint
7. Index finger distal interphalangeal joint
8. Index finger tip
9. Middle finger metacarpophalangeal joint

10. Middle finger proximal interphalangeal joint
11. Middle finger distal interphalangeal joint
12. Middle finger tip
13. Ring finger metacarpophalangeal joint
14. Ring finger proximal interphalangeal joint
15. Ring finger distal interphalangeal joint
16. Ring finger tip
17. Little finger metacarpophalangeal joint
18. Little finger proximal interphalangeal joint
19. Little finger distal interphalangeal joint
20. Little finger tip

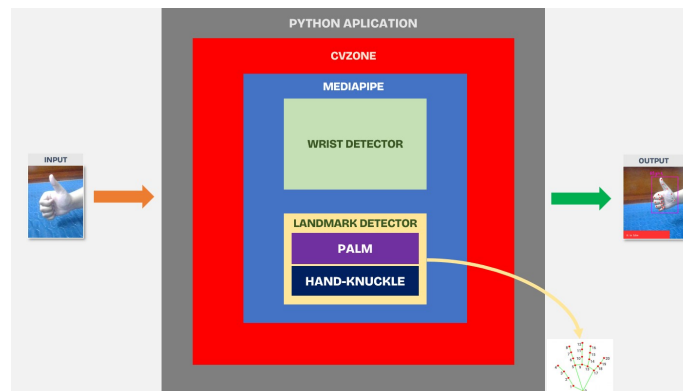**Figure 1: 21 Key Poins in Hand**

Figure 2: Model Work

The model takes in images or videos, including direct images from the camera containing the user's hand. Based on the input data, the model undergoes processing steps to identify the wrist and landmarks on the hand. The objective is to locate the points on the palm. Once these target points are identified, the model begins to match them with 21 pre-defined key points and connects these points to determine hand gestures. After processing as described, the complete output is generated as shown in the sample image.

```
import cv2
from cvzone.HandTrackingModule import HandDetector
detection = HandDetector(detectionCon=0.9, maxHands= 2)
video = cv2.VideoCapture(0)
```

Figure 3: Import and use Cvzone

4 EXPERIMENT

a. Data

We compiled our dataset by inviting multiple individuals to participate in model testing, utilizing a webcam for data collection. This effort resulted in an extensive dataset that comprises over 100 test images. This diverse dataset not only enhances the depth of our resources but also strengthens the model's robustness.

b. Model's hyperparameters

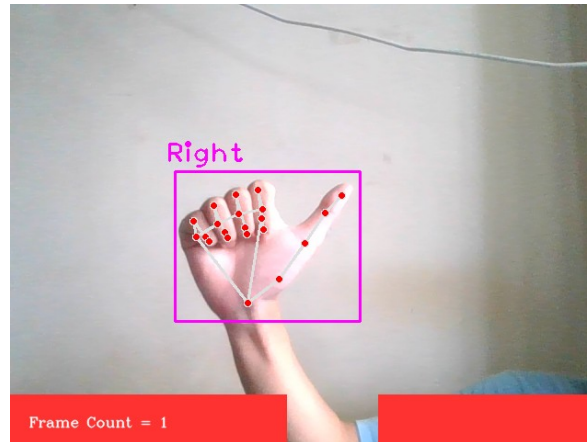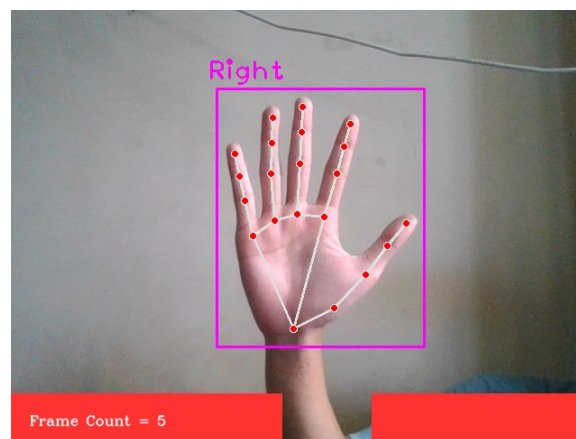HandDetector Model Parameters and their Purpose, Role, Setup:

Figure 4: Model Testing

Figure 5: Example about Data

*staticMode (Static Mode):
+Purpose: StaticMode determines whether the model operates in static mode or not.
+Role: When set to True, the model performs hand detection on each individual image, which may result in slower processing. This is useful for applications where you need to process still images or require high precision in hand detection.

If set to False, the model operates in dynamic mode, allowing it to continuously process a live video stream and detect and track hands in real-time scenarios. This configuration is well-suited for interactive applications, gesture recognition, and use cases that involve tracking hands in motion.
+Setup: You can set staticMode by assigning either True or False when creating the HandDetector object.

*maxHands (Maximum Number of Hands):
+Purpose: MaxHands defines the maximum number of hands that the model will attempt to detect.
+Role: By setting maxHands, you can specify how many hands the model should search for and track simultaneously in the input image. For example, if you set maxHands to 2, the model will focus on detecting and tracking up to two hands in a single frame. This setting provides flexibility and is appropriate for scenarios involving one or two hands. However, if there are more than two hands present, the model will prioritize tracking the most prominent two, potentially ignoring additional hands.

+Setup: MaxHands can be configured by assigning an integer value when creating the HandDetector object.

*modelComplexity (Model Complexity of Hand Landmark Model):
+Purpose: ModelComplexity determines the complexity level of the hand landmark model.
+Role: When set to 0, indicating low complexity, the model runs faster but may provide less detailed information about hand landmarks. This setting is suitable for applications where speed is a priority and fine-grained hand tracking is not critical.

When set to 1, indicating high complexity, the model provides more detailed and precise hand landmarks, which are valuable for applications that require fine-grained hand tracking and ges-ture recognition. However, this higher complexity may require additional computational resources and may result in slower pro- cessing.
+Setup: You can configure modelComplexity by setting it to either 0 or 1 when creating the HandDetector object. The choice between 0 and 1 depends on the specific requirements of your ap- plication, with 0 prioritizing speed and 1 prioritizing accuracy and detail.

*detectionCon (Minimum Detection Confidence Thresh-old):

+Purpose: DetectionCon sets the minimum confidence thresh-old for hand detection.

+Role: The role of this parameter is to establish a confidence threshold for hand detection. Only detection results with confidence scores higher than this threshold are considered valid.

+Setup: DetectionCon can be configured by assigning a numeri- cal value in the range from 0 to 1 when creating the HandDetector object. For example, setting detectionCon to 0.6 means that only hand detections with a confidence score greater than 60% will be considered valid. You can adjust this threshold based on your spe- cific application needs, finding the right balance between reliable detections and the number of detections.

*minTrackCon (Minimum Tracking Confidence Thresh-old):

+Purpose: MinTrackCon establishes the minimum confidence threshold for hand tracking.

+Role: This parameter plays a critical role in setting a confi- dence threshold for tracking hands. Only hands that have been detected and are being tracked with confidence scores higher than this threshold are considered valid.

+Setup: MinTrackCon can be configured by assigning a numeri- cal value within the range of 0 to 1 when creating the HandDetector object. For instance, if you set minTrackCon to 0.6, it means that only hands with tracking confidence scores exceeding 60% will be recognized as valid and continue to be tracked. This parameter allows you to control the quality of tracking results and filter out less reliable tracking data. Adjusting this threshold can help balance precision and recall in the tracking process, with a higher threshold leading to more accurate but potentially fewer tracked hands, and a lower threshold potentially capturing more hands but with some risk of tracking less accurately.

c. Testing process

After collecting a diverse set of 100 image samples, encompass- ing a range of characteristics, sizes, and angles, comprising two categories: Palm and Dorsal side of the hand, we conducted rig-orous testing and analysis. This included statistical assessments, data-driven evaluations, and result reporting.

d. Evaluation metrics

The accuracy metric was chosen for three specific reasons:

-Comprehensive Evaluation: Accuracy is an overarching metric that provides an overall assessment of the model's capability in classifying objects or events. It measures the ratio of correct pre- dictions (true predictions) to the total number of predictions (both true and false predictions).

-Representation of Precision: It offers insight into the model's ability to make both correct and incorrect classifications, making it a versatile metric for overall performance evaluation.

-Model Comparison: Accuracy enables the comparison of per- formance across different models based on the ratio of correct predictions.

In this case, accuracy serves as an appropriate and well-rounded metric to assess the model's performance and compare it to other models.

e Describing the result

The model has been tested on 100 images, including two types: Palm and Dorsal side of the hand. Here are the characteristics of these two types of images and the reasons for better performance with Palm images:

-Palm Images:

Characteristics: Palm images typically show the front side of a hand, where the palm is exposed. These images often have clear and distinguishable features, such as fingers, palm lines, and nails. Better Performance: The model performs better on Palm images due to the well-defined features present in these images. The distinct landmarks on the palm, fingers, and nails provide more salient points for the model to detect, making it easier to recognize and track the hand accurately.

-Dorsal Side of the Hand Images:

Characteristics: Dorsal side images show the backside of the hand, which generally has fewer prominent features compared to the palm. They often lack distinctive landmarks found on the palmside.

Performance Challenge: The model may have a slightly harder time with Dorsal side images because of the relative lack of dis- cernible features. The absence of palm lines, fingernails, and detailed hand contours can make it more challenging for the model to detect and track the hand correctly.

In summary, the better performance of the model on Palm im- ages is primarily attributed to the presence of well-defined and distinguishable features. The rich details and landmarks on the palm side of the hand provide more visual cues for the model to work with, resulting in more accurate hand detection and tracking. The Dorsal side images, in contrast, lack these prominent features, making them a bit more challenging for the model.

4   DISCUSSION

The model demonstrates optimal performance when recognizing the palm side of the hand under well-illuminated conditions, ap- proaching near-perfection. During the testing phase, involving over 100 samples categorized into two primary labels: 'Dorsal side of the hand' and 'palm,' we obtained near-flawless results for palm recognition. However, when working with 'Dorsal side of the hand' samples, the success rate decreased to 79 percent.

Testing Case	No. of Tests	No. of Correct	Accurracy(%)
Palm	70	70	100
Dorsal side of the hand	70	55	79
AVG			89,5

Figure 6: The Accuracy of Data

The results indicate that the model performs effectively and reliably with real-time images from the camera. This is crucial for

applications such as recognizing gestures of individuals with disabilities, controlling robotic arms, or interacting with computer keyboards. These are pressing and practical societal needs.

Figure 7: Demo

Advantages:
- The model exhibits impressive performance in well-lit con-ditions.
- It demonstrates robust recognition of the palm side, achiev-ing significantly high accuracy in most cases.
- The model operates smoothly, lightweight, and stably when all image conditions are met.

Disadvantages:
- Model performance significantly deteriorates in low-light environments.
- The model struggles with the recognition of the 'Dorsal side of the hand.'
- Lack of mobility as the model is not yet implemented on web or smartphone platforms.

5 SUMMARY

We propose a straightforward solution to the intricate challenge of hand recognition. Historically, this problem has posed signifi-cant difficulties. However, our suggested approach introduces an efficient and rapid resolution. CVzone excels in projects requiring real-time image processing and hand gesture recognition, partic- ularly in well-lit conditions. However, it faces notable limitations. CVzone's performance deteriorates significantly in low-light sce- narios, and its lack of mobility restricts its use on web platforms and on resource-constrained devices such as Raspberry Pi comput- ers. These limitations highlight the crucial areas that necessitate focused attention and development. Envisaging the future, CV- zone holds the potential to serve as a proficient tool for a wide spectrum of projects, thanks to its streamlined and dependable nature. In the pursuit of expanding upon this concept, we recog-nize the need for further research and development. Addressing CVzone's low-light performance and optimizing its compatibility with web and resource-constrained platforms will be essential. Ad- ditionally, exploring applications beyond hand recognition, such as gesture-based control systems, and enhancing the user experi- ence for individuals with disabilities are promising directions. This progress in hand recognition not only showcases the evolution of technology but also underscores its capacity to address real-world challenges and societal needs. In the journey ahead, we anticipate witnessing the continued growth and refinement of CVzone as it transforms into an even more versatile and accessible solution, con-tributing to a wide array of practical applications in the realms of technology and social inclusion.

6 ACKNOWLEDGEMENT

REFERENCES

[1] Muhammad Yeza Baihaqi, Vincent, and Joni Welman Simatupang. Real-time hand gesture recognition for humanoid robot control using python cvzone. pages 262–271, 2022.
[2] Chetana D. Patil, Amrita Sonare, Aliasgae Husain, Aniket Jha, and Ajay Phirke. Survey on: Hand gesture controlled using opencv and python. *International Journal of Creative Research Thoughts (IJCRT)*, 2022.
[3] George Sung, Kanstantsin Sokal, Esha Uboweja, Valentin Bazarevsky, Jonathan Baccash, Eduard Gabriel Bazavan, Chuo-Ling Chang, and Matthias Grundmann. On-device real-time hand gesture recognition. *arXiv preprint arXiv:2111.00038*, 2021.
[4] Rupesh Kumar, Ashutosh Bajpai, and Ayush Sinha. Mediapipe and cnns for real-time asl gesture recognition. *arXiv preprint arXiv:2305.05296*, 2023.
[5] Facundo Manuel Quiroga, Franco Ronchetti, Ulises Jeremías Cornejo Fan-dos, Gastón Gustavo Ríos, Pedro Alejandro Dal Bianco, Waldo Hasperué, and Laura Cristina Lanzarini. A comparison of small sample methods for handshape recognition. *Journal of Computer Science & Technology*, 23, 2023.
[6] Shubhra Aich, Jesus Ruiz-Santaquiteria, Zhenyu Lu, Prachi Garg, KJ Joseph, Al- varo Fernandez Garcia, Vineeth N Balasubramanian, Kenrick Kin, Chengde Wan, Necati Cihan Camgoz, et al. Data-free class-incremental hand gesture recognition. pages 20958–20967, 2023.
[7] Alberto Sabater, Iñigo Alonso, Luis Montesano, and Ana C Murillo. Domain and view-point agnostic hand action recognition. *IEEE Robotics and Automation Letters*, 6(4):78237830, 2021.
[8] Maarten Vandersteegen, Wouter Reusen, Kristof Van Beeck, and Toon Goedemé. Low-latency hand gesture recognition with a low-resolution thermal imager. pages 98–99, 2020