IRSTI 50.10.31

O. Sarybay
Suleyman Demirel University, Kaskelen, Kazakhstan

DATA COLLECTION OF HAND GESTURES ON A HORIZONTAL SURFACE USING MEDIAPIPE LIBRARY

Abstract. The horizontal hand gesture recognition is an innovative, cheaper way for human-computer interaction. Currently, most researchers work with sensors, devices for hand gesture recognition, which require more resources. Instead, the presented horizontal method for hand gesture signal recognition by frames, with trained model algorithms. A key element of this work is the research of a recognition algorithm using only a single camera and collecting dataset to train a hand recognition model. In the presented framework, the hand detection works with computer vision (CV) algorithms, in general MediaPipe as a converting blue, green, red (BGR) image to red, green, blue (RGB) before processing. There are handedness and hand landmarks on the image as a result of a hand detection. Each point of the landmark has coordination x, y, z values. The collected dataset will be used to train a model with machine learning (ML) or neural network algorithms to develop this project as a hand gesture recognition project.

Keywords: CV, ML, MediaPipe, neural networks, hand gesture, BGR, RGB, human-computer interaction.

***

Аңдатпа. Көлденең қол қимылын тану адам мен компьютер әрекеттесуінің инновациялық, арзан әдісі болып табылады. Қазіргі уақытта бұл бағыт бойынша зерттеушілердің көпшілігі сенсорлармен, қол қимылын тану құрылғыларымен жұмыс істейді. Мұндай құрылғылар көбірек ресурстарды қажет етеді. Бұл ғылыми деректе құрылғылардың орнына көлденең жазықтықта бір ғана камера көмегімен қолдың қимылдарын тану модель алгоритмдері ұсынылды. Бұл жұмыстың негізгі элементі қолды тану моделін үйрету үшін тек бір камера көмегімен көлденең жазықтықта қолдың қимылдарын тану алгоритмін зерттеу және қолдың тану алгоритмі бойынша деректерді жинау болып табылады. Ұсынылған алгоритмде қолды анықтау компьютерлік көру, оның ішінде MediaPipe алгоритмдерімен өңдеу алдында көк, жасыл, қызыл (BGR) кескінді қызыл, жасыл, көк (RGB) түрлендіру ретінде жұмыс істейді. Қолды анықтау нәтижесінде қолдың қимылын кесін нәтижесінде ала аламыз және қол белгілерінің нүктелерін аламыз. Қол белгілерінің әрбір нүктесінде x, y, z координациялық мәндері бар. Қолды танып-білу деректер

жинағы алдағы уақытта қол қимылын тану жобасы ретінде әзірлеу үшін, алдағы уақытта машиналық оқыту немесе нейрондық желі алгоритмдері бар үлгіні үйрету үшін пайдаланылады.

Түйін сөздер: : адам мен компьютердің әрекеттесуі, машиналық оқыту, нейрондық желі алгоритмдер, MediaPipe, қол қимылын тану, BGR, RGB.

***

Аннотация. Горизонтальное распознавание жестов рук — это инновационный и доступный способ взаимодействия человека с компьютером. В настоящее время большинство исследователей работают с датчиками, устройствами для распознавания жестов рук, которые требуют больших затрат. Для решения данной проблемы представлен горизонтальный метод распознавания сигналов жестов рук по кадрам, с использованием алгоритмов обученной модели. Для разработки и создания модели потребуется сбор данных. Ключевым элементом работы является исследование алгоритма распознавания с использованием одной камеры и сбор набора данных для обучения модели распознавания рук. В представленной структуре обнаружение рук, используются алгоритмы компьютерного зрения, в целом MediaPipe как преобразование синего, зеленого, красного (BGR) изображения в красное, зеленое, синее (RGB) перед обработкой. В результате обнаружения рук на изображении присутствуют кисть и специальные точки рук. Каждая точка руки представлена по оси x, y, z. Собранный набор данных будет использоваться для обучения модели с помощью алгоритмов машинного обучения или нейронных сетей для разработки проекта распознавания жестов рук.

Ключевые слова: распознавание жестов рук, машинное обучение, нейронные сети, компьютерное зрение, MediaPipe, BGR, RGB.

I. Introduction

It is well-known that computer vision (CV) is implemented in daily life, and vision-based technology such as hand gesture recognition is one of the most important parts of human and computer interaction [5]. Nowadays, there are some projects that project basic interactions between human and computer using sensor display, touch screen, keyboard and mouse, but in other cases, quick development of hardware and software, new types of human-computer interaction methods have been required [4][6].

Gesture is a symbol of physical behavior or emotional expression. It includes body gestures and hand gestures. It falls into two categories: static gesture and dynamic gesture[1]. For the person, the posture of the body or the gesture of the hand denotes a kind of signal. Gesture can be used as a tool of communication between computer and human [2]. It is greatly different from the traditional hardware based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user

action through the recognition of the gesture or movement of the body parts. Nowadays hand gesture recognition projects work with sensors, ovation, raspberry PI, etc. methods. These methods claim other devices, gadgets, but they are not comfortable for using them in casual life.

There are two types of one dataset. First dataset includes all x, y, z parameters sequentially. In the second dataset each value is in different files. For example, all x are in one file. The x, y, z values are the coordinates of the palm, each finger, etc. In total, there are 63 points. To detect the hand and get a point there used the MediaPipe library.

MediaPipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame. This article is the beginning part of another project, which recognises hand gestures on a horizontal surface by a single camera in real time. So, that project needs to be learned by ML or neural networks algorithms, that's why this article is about data collection and its methods.

First, there are collected videos by right hand gestures: down, up, left, right, zoom in, zoom out. Each video is divided into frames to get some landmark gesture coordinations. Then the hand region is detected from the original images from the input devices. MediaPipe finds landmarks of hand in x, y, z position. So, each landmark consists of 21 dots,with 3D landmarks it will be 63 dots in general. There are 3 landmarks for each finger and 6 landmarks around the palm. In general, there are about 3200 coordinates in x, y, z directions of the gestures, to learn a model.

There are many types to write an algorithm to recognise hand gestures: it can be trained by sequence images dataset - which got from video frames, just programming algorithms - with calculating distance between fingers, trained dataset by getting landmark positions [11][12].

The purpose of the article is to find the best dataset type to get more accuracy results, which will be trained and can be tested. The main difference of this project from other hand recognition projects is working with only two fingers, finding the gesture by these fingers commands and not only one command, which can be solved with only one frame - train with frames, which got form video. If there was only one frame the algorithm cannot recognise it, because "down" command is opposite of "up", "left" command is opposite of "right", "zoom in" is opposite of "zoom out". If there is the video for example for "down command" - if turn the video back, it will be "up command".

II. Literature review

In the past decades, many researchers have strived to improve the hand gesture recognition technology. Hand gesture recognition has great value in many applications such as sign language recognition[13][14], augmented reality (virtual reality) [15], sign language interpreters for the disabled[16] , and robot control [17].

In [1, 2], the authors detect the hand region from input images and then track and analyze the moving path to recognize America sign language. In [10], Shimada et al. propose a TV control interface using hand gesture recognition. Keskin et al. [10] divide the hand into 21 different regions and train a SVM classifier to model the joint distribution of these regions for various hand gestures so as to classify the gestures.

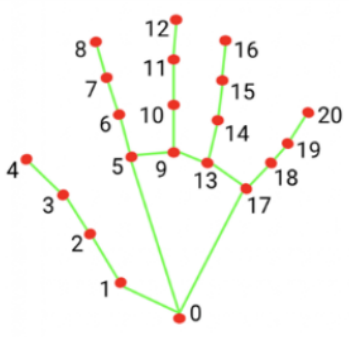
Zeng et al. [8] improve the medical service through the hand gesture recognition. The HCI recognition system of the intelligent wheelchair includes five hand gestures and three compound states. Their system performs reliably in the environment of indoor and outdoor and in the condition of lighting change.

The main difference of this project from other hand recognition project is working with only two fingers, finding the gesture by these fingers commands and not only one command, which can be solved with only one frame. The project "Gesture Hand Controller" of Luiz Henrique da Silva Santos and Matheus Vyctor Aranda Espíndola recognises the hand command by all fingers. They show one command with hand, the program detects it and makes according command [8]. For example, the hand gesture "like" the big finger up - means "click", "zoom out", "zoom in" command are the detect by the first and second fingers. When the distance between fingers are big then - zoom out, if - less, then zoom in command will be done and there is no any animation [9]. The dataset of this project is collected by MediaPipe library, also. It works with the dataset, which contains x, y, z values in one file. That's why it can detect in real-time frame [10].

III. Method and Materials

Data

There are many lists of frames collected manually to classify the gesture signs with the following signs: zoom in, zoom out, right, left. Each type of hand gesture is found using only 2 points from the hand fingers as shown in the figure 12,16 (Figure 1). There are also some faster data collection methods like to cut video by 4 frames, and collect key points and keep them in the classified folders. There are 4 frames are accessible for reading and setting landmarks.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 1. Hand Landmarks.

There is a requirement that the hand recognition model should be read on a horizontal surface and the distance between hand and camera should be 1m. The hand tracking model first finds a palm and then draws landmarks of fingers where MediaPipe recognition algorithms best fits for. A palm detector that works on a full input image and makes bounding boxes for palms for localization.

1.2 About Hand Landmark Model (MediaPipe)

There are many algorithms used to detect the hand and find landmarks of it: firstly it takes a frame from a video. Then synthesize images, make hand presence, find handedness dots and lines (Figure 2). The main algorithm firstly finds the palm, then by palm finds finger landmarks. There are 5 points of the palm, which are in every finger position and one is in the bottom part of the palm. Every finger has 3 landmarks: upper side, medium, bottom side (Figure 5).

Hand landmark model operates on bounding boxes to provide keypoint localization of 21 3D hand coordinates via regression algorithms that pass to coordinate prediction (Figure 4). The model learns a consistent hand pose representation and is still good even to partially visible hands[3].

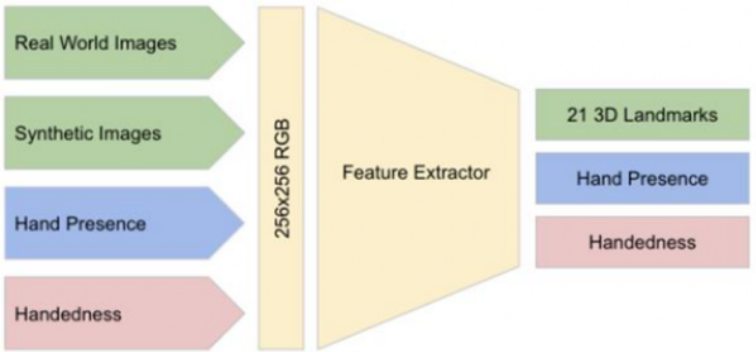


Figure 2. Architecture of hand landmark model. The model has three outputs sharing a feature extractor. Each head is trained by correspondent datasets marked in the same color [6].

Then there is a 3D model of the hand (Figure 5). In the video with tracking scenario, there is deriven a bounding box from the landmark prediction of the previous frame as input for the current frame, this feature helps us to avoid applying the detector on every frame. Instead, the detector is only applied on the first frame or when the hand prediction indicates that the hand is lost. These landmarks and poses are necessary for the hand detection model and gesture detection recognition model [3].

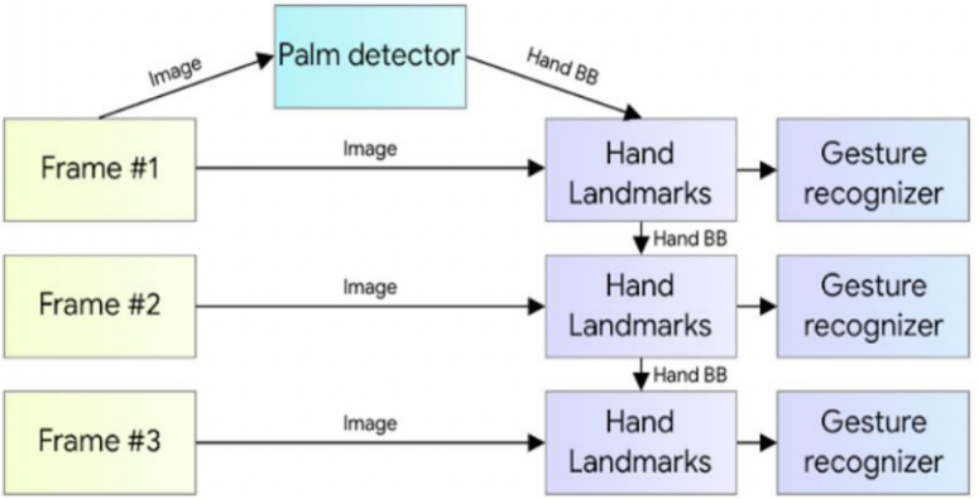


Figure 3. Hand detection pipeline

Overall there are several trained models perform together :

- A palm detector model (called BlazePalm) that operates on the full image and returns an oriented hand bounding box.
- A hand landmark model that runs on the cropped image region defined by the palm detector and returns high fidelity 2.5-3D hand keypoints.
- A gesture recognizer for classification [3].
-

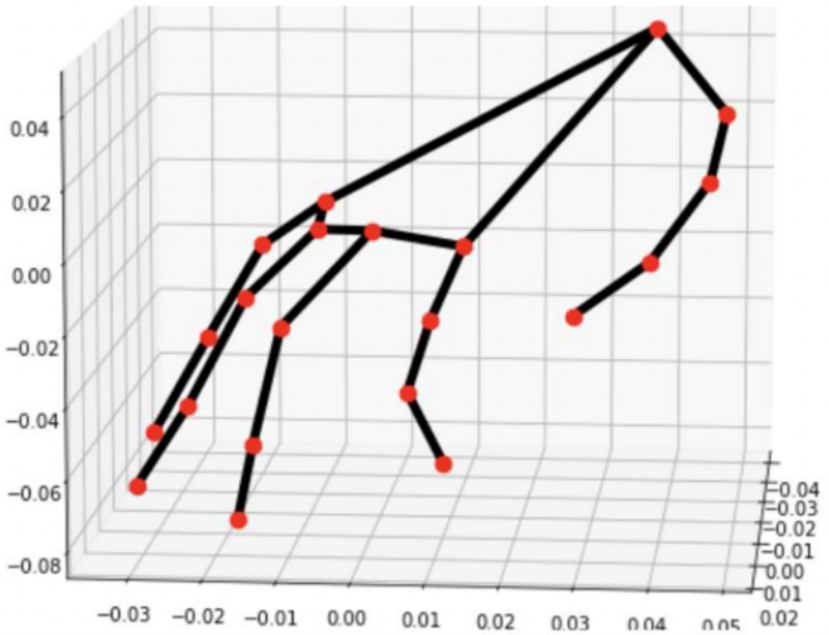


Figure 4. 3D model of hand

1.3 Train datasets and compare models.

In this paper to classify the gestures used the NN algorithm to build the Hand Sign recognition model. As the model optimizer there is used the Adam algorithm and as the loss function used Sparse Categorical Entropy .

Epoch 1/1000
1/27 [>.....] - ETA: 0s - loss: 1.1295 - accuracy: 0.3203
Epoch 00001: saving model to model/classifier/classifier.hdf5
27/27 [==========] - 0s 11ms/step - loss: 1.1004 - accuracy: 0.3602 - val_loss: 1.0431 - val_accuracy: 0.5220
Epoch 2/1000
1/27 [>.....] - ETA: 0s - loss: 1.0440 - accuracy: 0.4844
Epoch 00002: saving model to model/classifier/classifier.hdf5
27/27 [==========] - 0s 3ms/step - loss: 1.0503 - accuracy: 0.4297 - val_loss: 0.9953 - val_accuracy: 0.6397
Epoch 3/1000
1/27 [>.....] - ETA: 0s - loss: 1.0043 - accuracy: 0.5312
Epoch 00003: saving model to model/classifier/classifier.hdf5
27/27 [==========] - 0s 4ms/step - loss: 1.0210 - accuracy: 0.4582 - val_loss: 0.9545 - val_accuracy: 0.6523
Epoch 4/1000
1/27 [>.....] - ETA: 0s - loss: 0.9503 - accuracy: 0.5625
Epoch 00004: saving model to model/classifier/classifier.hdf5
...

1. All x, y landmarks in one file - this is the first dataset.
2. Every point is in different files - this is the second dataset.
3. Write an algorithm to collect data from camera simply.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2
1	0.50	0.67	0.57	3
2	1.00	0.50	0.67	4
accuracy			0.44	9
macro avg	0.50	0.39	0.41	9
weighted avg	0.61	0.44	0.49	9

0.4444444444444444

accuracy			0.22	9
macro avg	0.13	0.22	0.17	9
weighted avg	0.13	0.22	0.17	9

Figure 5. SVM and KNN algorithms result.

In the beginning there is tested KNN, SVM algorithm. The first SVM had an accuracy of 0.44 and KNN of 0.22. So, the decision was to dive deeper and modify our algorithm to make the best classification. The first part was to collect a dataset using simple commands and write them automatically to a CSV file by dividing the program into Train and Test modes. To increase the probability, the decision was to keep only one type of target classification: zooming in and out.

You can see the process of pointing 2 points from fingers, for train and future prediction (Figure 6).

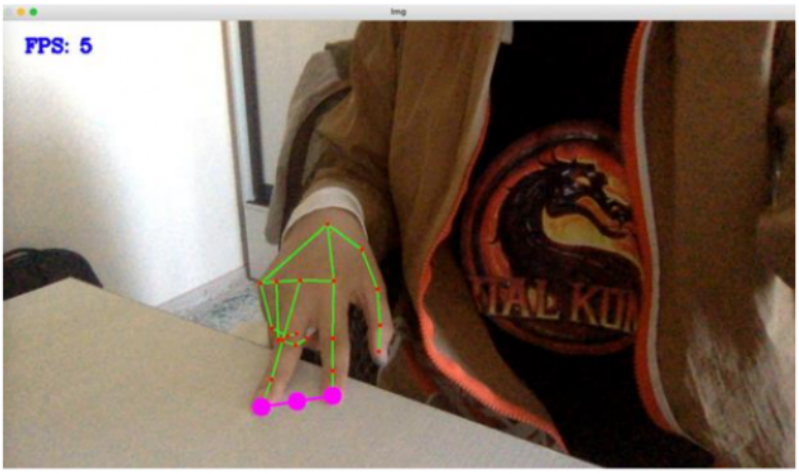


Figure 6. Zoom in hand gesture detection with the two fingers.

The best results are obtained from the NN algorithm. As the prediction was almost perfect for 2 classes, zooming in and out. Accuracy was good in the ideal environment, showing more than 90% (Figure 7).

IV. Results

There are many videos of hand gestures: zoom in, zoom out, left, right, down, up. That helps us to collect frames automatically, but the number of frames were different. In general the frame numbers of video was 4. The programm divided video into 4 and read each frame of gesture to get landmarks of hand. Each x, y position of each gesture is written in keypoints CSV file only with "x"-position, "y"-position.

For this experiment using only 2 types of classes: 1st type - "zoom in" gesture;

2nd type - "zoom out" gesture. The representation of the dataset is in the following figure 6.

0	0.2399062961	0.2463133335	0.2429064214	0.2340668887
0	0.4205651879	0.3967759013	0.3858606815	0.3879730999
1	0.4462751746	0.4238047004	0.409937799	0.4092714489
1	0.4412176013	0.418343842	0.404360652	0.4055868089

Figure 6. Example of 0-down and 1-up type gesture dataset in x positions dataset file.

Each frame has 21 numbers of landmarks in x, y, z position. So, in general there are 63 points. There are 252 point data for one gesture and only one ended video motion frames, it has got from 63*4 points. All these data points help to train a model for hand gesture recognition. For example, for "zoom in gesture" will need more x, y, values from 2 points and collect them into a one dimentional array and classify them. Also, no need to keep 252 points in the dataset, the reseachers try to make our model simple and fast using only 4 x 4 points, and

classification integers(0 - zoom in, 1 - zoom out). So as a result, there will be 2 predicted classes. For example, models return a list and they can be compared just getting the highest score.

There is a test result for zoom in: [9.8105639e-01 1.8674169e-02 2.2328216e-04 4.6191799e-05], 0 - zoom in

IV. Conclusion

This project is about collecting hand gesture dataset for hand recognition algorithms as a model. So, firstly there are video and manually collected datasets, which are divided into 4 frames and have landmark points of hand gesture. Each finger has 3 landmarks and a palm has 6 landmarks behind it. This algorithm processes the image from BGR to RGB and draws handedness, gets height and width of hand, then gets the overall 63 landmarks of the hand. There are almost 3000 frames and 65 types of data for x position, y position, z position in separate dataset files. The second type of collecting data set was to give the name of the gesture and show that gesture in real time. The second type is comfortable, because it works only with two points of two fingers - second and third finger as needed for the project, collect them in one-dimensional array and classify them.To detect the hand and find landmarks of the hand, the MediaPipe library's "Hands" algorithm is used.

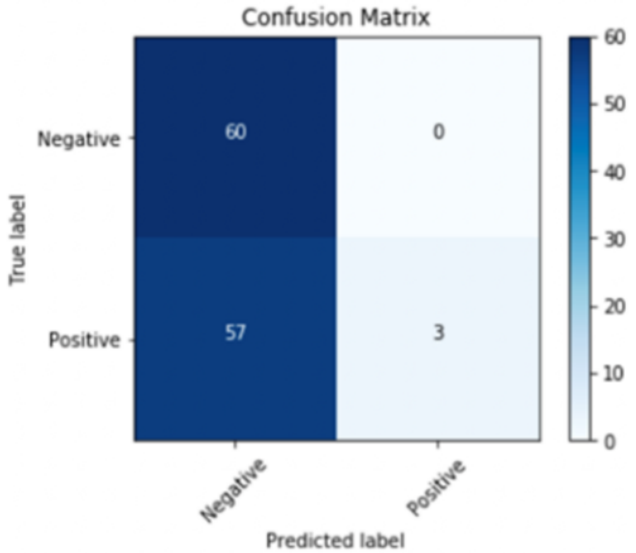


Figure 7. Confusion matrix for NN model.

For gesture detection, there are points to make a prediction for gesture recognition systems. The best dataset type was all values included dataset. Because the MediaPipe library works in such a way. So, from the first frame array it can almost detect the frames. There are other projects that work with the MediaPipe library. In this project there were experiments with two type dataset and two training algorithms. The best solution for the training algorithm was NN

with the all included dataset with the result accuracy > 0.9 (Figure 7). The NN was a better solution compared to simpler SVM, KNN algorithms. For continuing the following work the algorithms can be changed and added more data to the dataset.

In the near future, collected datasets will help to build multiple more accurate models, for the project of hand and gesture recognition systems. The project is willing to encourage more researchers in this field to build powerful algorithms to continuously increase the efficiency and accuracy.

References

1 M. Rehm, in Human-Centric Interfaces for Ambient Intelligence, 2010

2 R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 3, pp. 462–477, 2010.

3 V. Bazarevsky and F. Zhang, On-Device, Real-Time Hand Tracking with MediaPipe, URL: https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html, 2019

4 C. Wan, T. Probst, L. Van Gool, and A. Yao. Self-supervised 3d hand pose estimation through training by fitting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 10853–10862, 2019.

5 Facebook. Oculus Quest Hand Tracking. https://www.oculus.com/blog/oculusconnect-6-introducing-hand-tracking-onoculus-quest-facebook-horizon-and-more/.

6 L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. 3d hand shape and pose estimation from a single rgb image. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 10833–10842, 2019.

7 V. Bazarevsky and F. Zhang, A.Vakunov, A.Tkachenka, G. Sung, C. Chang, M. Grundmann, MediaPipe Hands: On-device Real-time Hand Tracking, URL: https://arxiv.org/pdf/2006.10214.pdf, 2020

8 L. Silva Santos and M. Aranda Espíndol, Gesture Hand Controller, URL:https://github.com/luizhss/Gesture_Hand_Controller, 2021

9 L. Silva Santos and M. Aranda Espíndol, Gesture Hand Controller Video Example , URL: https://www.youtube.com/watch?v=OKRuiNP62Qc, 2021

10 L. Silva Santos and M. Aranda Espíndol, Gesture Hand Controller Dataset, URL: https://github.com/luizhss/Gesture_Hand_Controller/blob/master/dataset_train.csv, 2021

11 S. Paul, Video Classification with a CNN-RNN Architecture, URL: https://keras.io/examples/vision/video_classification/, 2021
12 U. Tiofack, Simple Hand Mouvement Recognition Code: Hand tracking. Mediapipe, URL: https://gist.github.com/TheJLifeX/99cdf4823e2b7867c0e94fabc660c58b, 2022
13 R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 3, pp. 462–477, 2010.
14 Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," in Proceedings of the 13th ACM International Conference on Multimodal Interfaces (ICMI '11), pp. 279–286, November 2011.
15 D. Wickeroth, P. Benölken, and U. Lang, "Markerless gesture based interaction for design review scenarios," in Proceedings of the 2nd International Conference on the Applications of Digital Information and Web Technologies (ICADIWT '09), pp. 682–687, August 2009.
16 J. Choi, H. Park, and J.-I. Park, "Hand shape recognition using distance transform and shape decomposition," in Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11), pp. 3605–3608, September 2011.
17 J. Zeng, Y. Sun, and F. Wang, "A natural hand gesture system for intelligent human-computer interaction and medical assistance," in Proceedings of the 3rd Global Congress on Intelligent Systems (GCIS '12), pp. 382–385, November 2012.