Proceedings of IYSC, (2022), vol. 11

Journal homepage: http://journals.sdu.edu.kz/index.php/iysw

RECOGNITION OF BASIC HAND GESTURES ON A HORIZONTAL SURFACE USING A SINGLE CAMERA

O. Sarybay

Suleyman Demirel University, Kaskelen, Kazakhstan

Proceedings of IYSC, (2022), vol. 11

Journal homepage: http://journals.sdu.edu.kz/index.php/iysw

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. A key element of this work is the research of a recognition algorithm using only single camera. In the presented framework, the hand detection works as a converting BGR image to RGB before processing. Then, the palm and fingers are segmented so as to detect and recognize the fingers. There are handedness and hand landmarks on the image as a result of a hand detection. Each point of landmark has coordination x, y, z values. There is comparing algorithm of points to recognize hand gesture by fingers. The model has been implemented getting landmark values on a data set of hand images, which collected from video frames. 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 performance of the method highly depends on the result of hand detection on horizontal surface and collected dataset.

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

Introduction

It is well-known that CV is implemented in a daily life, and vision-based technology as hand gesture recognition is one of the most important part of human and computer interaction. Nowadays, there are some projects 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, because not everyone have these kind of opportunities. That's why in this project take attention to this problem and try to help people with disabilities, because here is some research of working only single camera on computer and recognition algorithms.

Gesture is a symbol of physical behavior or emotional expression. It includes body gesture and hand gesture. 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 works with sensors, ovation, raspberry PI, etc. methods. These methods claim another devices, gadgets, but there is not comfortable for using them in casual life.

First, the hand region is detected from the original images from the input devices camera. Then, some kinds of features are extracted to describe hand gestures, the recognition of hand gestures is accomplished by measuring the similarity of the feature data. The input devices providing the original image information includes normal camera. The skin color sensitive to the lighting condition and feature points are attended to detect and segment the hand region. One of the first steps were recognizing a hand by finding finger indices and angles. There is MediaPipe algorithm to help detect hand and find landmarks, drawing handedness. The best solution is the best trained algorithm.

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[1–3], augmented reality (virtual reality) [4–7], sign language interpreters for the disabled[8] , and robot control [9, 10].

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.

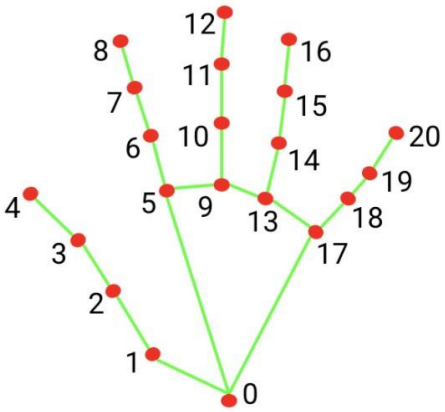
The main difference of this project from other hand recognition project is working with only two fingers, finding the gesture by these fingers commands and not only one command, which can be solved with only one frame. The project "Gesture Hand Controller" of Luiz Henrique da Silva Santos and Matheus Vyctor Aranda Espíndola recognizes the hand command by all fingers.

They show one command with hand, the program detects it and makes according command [11]. 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 [12]. 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 [13].

Methods and Materials

Detecting hands is a decidedly complex task. First, there is trained a palm detector instead of a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. In addition, as palms are smaller objects, the non-maximum suppression algorithm works well even for two-hand self-occlusion cases, like handshakes [16]. Moreover, palms can be modelled using square bounding boxes (anchors in ML terminology) ignoring other aspect ratios, and therefore reducing the number of anchors by a factor of 3-5. Second, an encoder-decoder feature extractor is used for bigger scene context awareness even for small objects (similar to the RetinaNet approach) [16].

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 by MediaPipe.



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.

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 1).

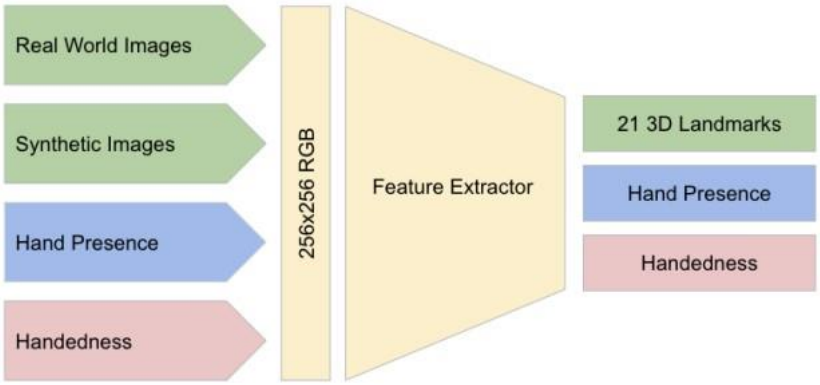


Figure 2. Architecture of research hand landmark model. The model has three outputs sharing a feature extractor. Each head is trained by correspondent datasets marked in the same color [14].

Hand landmark model operates on bounding boxes to provide key-point localization of 21 3D hand coordinates via regression algorithms that pass to coordinate prediction (Figure 3). The model learns a consistent hand pose representation and is still good even to partially visible hands[15].

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 key-points.

A gesture recognizer for classification [15].

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" the program will need more x, y, values from 2 points and collect them into a one dimensional array and classify them. Also, no need to keep 252 points in the dataset, the researchers try to make model simple and fast using only 4 x 4 points, and classification integers (0 - zoom in, 1 - zoom out).

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 are different. In general the frame numbers of video was 4. The program divided it into 4 and read each frame of gesture to get landmarks of hand. Each x, y position of each gesture is written in key-points CSV file only with "x"-position, "y"-position.

For this experiment there are 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 3 (Figure 3).

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 3. Example of 0-zoom in and 1-zoom out type gesture dataset in x positions dataset file.

The another type of collecting data was getting 2 points from fingers, for train and future prediction, which firstly gave the type of gesture then show the gesture.

Results and Discussion

```
Testing SVC with 9 tuples...

              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


0.444444444444444
```

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

Figure 4. SVM and KNN algorithms result.

In the beginning there were tested KNN, SVM algorithm. The first SVM had an accuracy

of 0.44 and KNN of 0.22. The decision to dive deeper and modify algorithm to make the best classification. First part was to collect 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, decision was to keep only one type of target classification: zooming in and out. 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%.

The results obtained from the training NN model was pretty good from the 30 epochs:

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

...

The data used for testing model are 2 points from fingers, for train and future prediction (Figure 5).

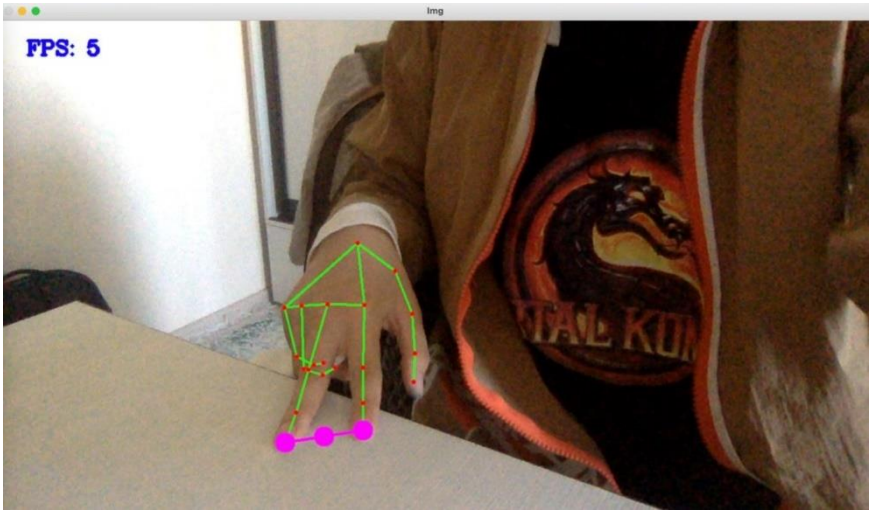


Figure 5. Zoom in hand gesture detection with the two fingers.

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

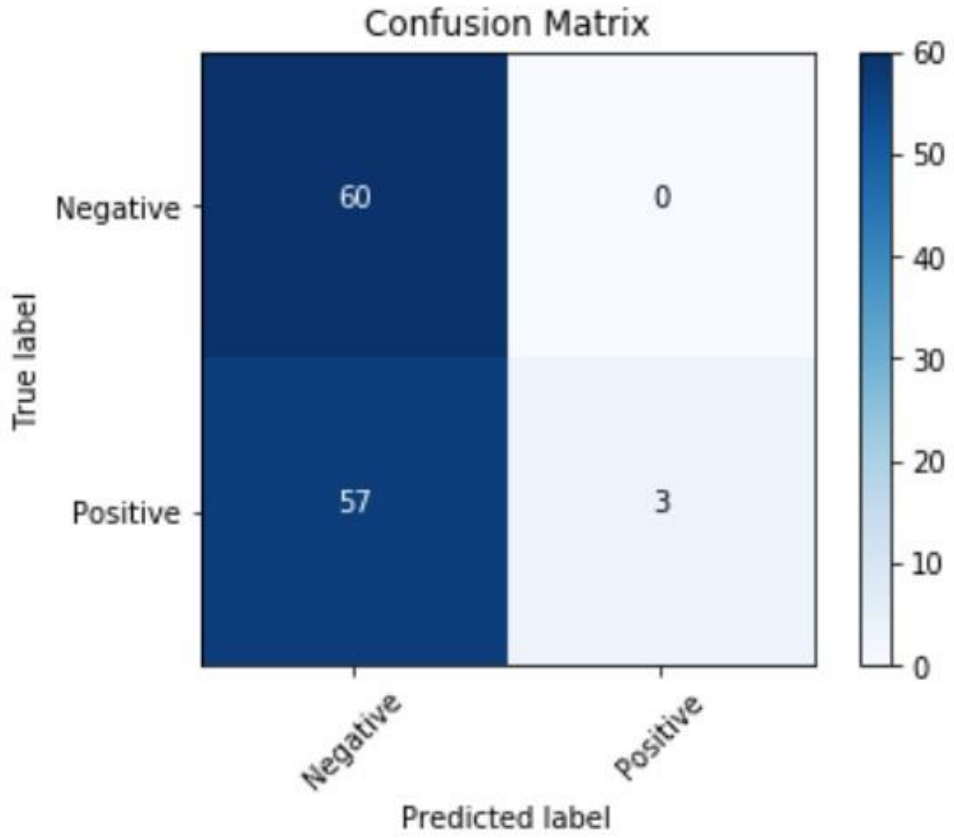


Figure 6. Confusion matrix for NN model.

As a process conclusion, there is finger indices and positions which can help to recognize hand gestures from the animation (list of key-points) and perform some commands. Firstly, to make the better model, data collection plays the main role, so developed simple collection script in the project will help easily collect classification data, while the data is collected in real time by pressing space button. The second big impact from script is to prevent False Positive positive results retraining the model adding new dataset on fly. The recognition of hand gestures is found using well-known classification algorithms in machine learning. The effectiveness of method is evaluated on a data set of hand images. The experimental results will be shown after researches' approach.

The performance of the their method will highly depend on the result of hand detection on horizontal surface and collected dataset. If there are moving objects with the color similar to that of the skin, the objects exist in the result of the hand detection and then degrade the performance of the hand gesture recognition as in previous works. That's why it's better to write wrapper function to separate hand, and then to pass the copy of the new image to MediaPipe. The researchers hope in future works, machine learning methods and 2d cameras may be used to address the complex background problem and improve the problem of hand detection.

For the future research: should add logic recognition hand gesture in horizontal and adding some algorithms, by training, to make it smarter.

References

1. Matthias 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. 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.
4. D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time sign language letter and word recognition from depth data," in Proceedings of the IEEE International

Conference on Computer Vision Workshops (ICCV '11), pp. 383–390, November 2011.

5. N. Pugeault and R. Bowden, "Spelling it out: real-time ASL fingerspelling recognition," in Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11), pp. 1114–1119, November 2011.
6. 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.
7. V. Frati and D. Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics," in Proceedings of the IEEE World Haptics Conference (WHC '11), pp. 317–321, June 2011.
8. 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.
9. T.-D. Tan and Z.-M. Guo, "Research of hand positioning and gesture recognition based on binocular vision," in Proceedings of the IEEE International Symposium on Virtual Reality Innovations (ISVRI '11), pp. 311–315, March 2011.
10. 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.
11. L. Silva Santos and M. ArandaEspíndol, Gesture Hand Controller, URL:https://github.com/luizhss/Gesture_Hand_Controller, 2021
12. L. Silva Santos and M. Aranda Espíndol, Gesture Hand Controller Video Example, URL: https://www.youtube.com/watch?v=OKRuiNP62Qc, 2021
13. L. Silva Santos and M. Aranda Espíndol, Gesture Hand Controller Dataset, URL:
14. https://github.com/luizhss/Gesture_Hand_Controller/blob/master/dataset_train.csv, 2021
15. 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.
16. 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 16. V. Bazarevsky and F. Zhang, A.Vakunov, A.Tkachenka, G. Sung, C. Chang, M.

Grundmann,

17. MediaPipe Hands: On-device Real-time Hand Tracking, URL: https://arxiv.org/pdf/2006.10214.pdf, 2020