Webology, Volume 18, Number 4, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I4/123

CHARACTER IDENTIFICATION AND MOUSE CONTROL USING HAND GESTURES: A SURVEY

Surendra Shukla[1], Bhasker Pant[2], Dibyahash Bordoloi[3]

[1]Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand India
[2]Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand India
[3]Head of the Department, Department of Computer Science & Engineering,Graphic Era Hill University, Dehradun, Uttarakhand India

ABSTRACT

Given the rapid advancement of computer technology, this paper suggests a mouse control system that utilises hand motions recorded via a webcam utilising a colour detection algorithm. This system will let the user to pick, click, scroll, and drag objects using a variety of hand gestures. The suggested method tracks the user's hand by using a low-resolution camera as a sensor. Python and OpenCV will be used to implement the system. One of the easiest and most natural forms of communication is the hand gesture.

Keywords: gesture, identification, mouse control, character

INTRODUCTION

The widely recognised art of hand gesture is one of the most effective forms of human communication. Anybody could understand it with ease. The system's experimental design uses a low-cost web camera with high quality recording capabilities mounted on top of the computer monitor or a fixed camera on a laptop to take pictures in the RGB colour space from a defined distance. It can be broken down into four stages, including edge detection, noise reduction, region extraction, and image preparation. An efficient hand gesture detection technology is used in this project. The following steps would be taken after this stage.

An efficient hand has been suggested in this research based on preprocessing, background subtraction, and edge detection techniques. Preprocessing's major goal is to change data into something that can be processed more quickly and easily. Preprocessing can be done in the suggested work by combining the steps of taking an image, removing noise, subtracting the background, and detecting edges. Initially, gloves or wear trackers are used to capture hand gesture photos from the vision-based camera. With this project, we hope to offer webcam-compatible laptops and computers with free hand recognition software. The project focuses on the hand recognition tool, which can be used to scroll, click, select, and recognise character events while moving the mouse pointer.

Webology, Volume 18, Number 4, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I4/123

PROPOSED SYSTEM:

Although there are several quick access techniques for hand and mouse gestures, our solution allows us to use a laptop or webcam and recognise hand gestures to control the mouse and carry out simple operations like selecting, deselecting, clicking left or right, and identifying characters. To determine the hands and hand movements as well as to assign an action to each movement, it uses the straightforward method (CONVEX HULL algorithm). The system we are constructing is written in python code, which is more responsive and readily developed because python is a basic language, is platform independent with a great deal of flexibility, and is also portable with an emphasis on developing a virtual mouse and hand recognition system.

USE OF PROPOSED WORK:

Without the extra hardware needs, this virtual mouse hand recognition application uses a finger. This is accomplished by combining camera input with hand gesture recognition based on eyesight.

METHODS:

This approach will break down each system component individually. The subsections are as follows:

CAMERA SETTING:

The webcam of the linked laptop or desktop controls the runtime actions. We must make a video capture object in order to record a video. Frame by frame image capturing is possible. But don't forget to release the capture at the conclusion. By making a few simple changes to the code, we could also use the colour detection technique on any image.

CAPTURING FRAME:

The usage of an infinite loop ensures that the web camera is always open and taking pictures throughout the duration of the application. We meticulously record the live streaming stream. Then, we translate each frame that was gathered into HSV colour space from RGB colour, which is the default. More than 150 distinct color-space conversion methods are available in OpenCV. But we'll only pay attention to BGR to Gray and BGR to HSV since they are the two that are most usually used.

MASKING TECHNIQUES:

Only the red-colored objects are highlighted as a result of the bitwise AND operations that are carried out during masking on the input image and the threshold image. The res variable stores the outcome of AND operations. while using the imshow() function to display frame, res, and mask on three different windows.

.

DISPLAY THE FRAME:

The wait key must be frequently called in order to use the imshow() method. By using the wait key, the imshow() method completes its loop. Thus, we use a 1ms delay while using the wait key method.

CLICKING:

Applying the close gesture is the next stage. The action is carried out by clicking and dragging the object. It is comparable to the open gesture, but since there is only one object present, we simply

Webology, Volume 18, Number 4, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I4/123

need to determine its centre. And that will be put in the spot where our mouse pointer will be set. We shall employ a mouse press operation in place of a mouse release operation.

DRAG:
We establish a variable called the "pinch flag" in order to accomplish the dragging. If it was previously clicked, it will be set to 1. Therefore, anytime we locate the open gesture, we click and then verify that the pinch flag is set to 1. The drag action is carried out if it is set to one; else, the mouse move operation is carried out.

CHARACTER RECOGNITION:
Step 1:
LOAD DATA: To load the data, we employ the mnist package in Python. Now let's prepare the data so that it may be fed to the model. standardising the photos, dividing the data into training and testing sets, and doing additional basic work

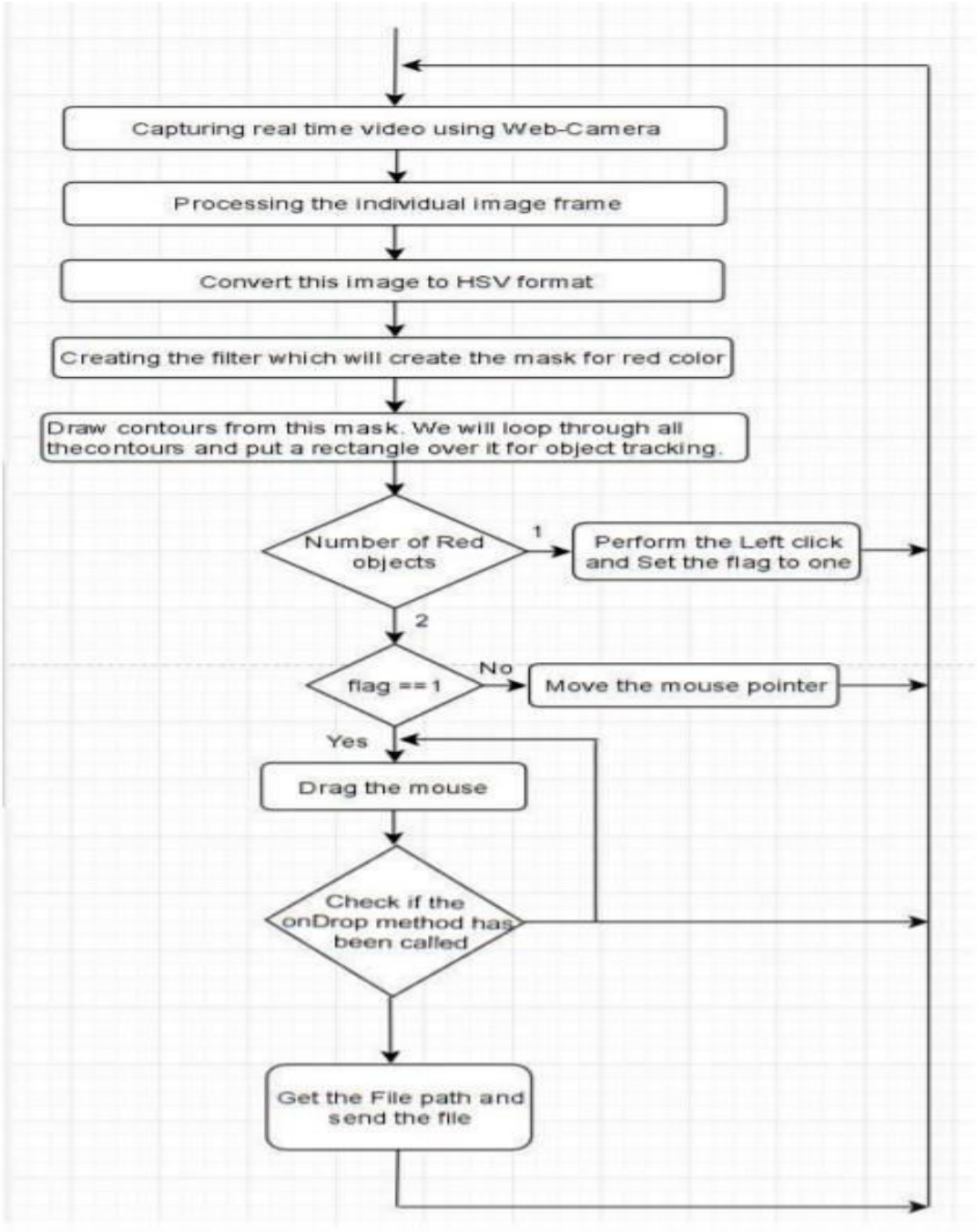
SPECIFY A MODEL:
The definition of a Keras model is a series of layers. The layers with the appropriate neurons are added once we initialise a "Sequential Model"

GENERATING MODEL:
We can compile the model now that it has been defined. The model's backend, or efficient numerical libraries, such Theano or TensorFlow, are used during compilation. Here, we provide a few requirements for the network's training qualities.

FIT MODEL: In this case, we train the model with the aid of a model check pointer, allowing us to retain the best model.

EVALUATE MODEL: On the EMNIST dataset, the model's test accuracy was 91.1%.

Webology, Volume 18, Number 4, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I4/123



RESULT AND EVALUATION:

In this study, we sought to concentrate on enhancing human-machine interaction. Our goals were to develop this technology as affordably as feasible and to do it using a standardised operating system. The primary goal of this system is to minimise the utilisation of computer hardware accessories. Even so, a standard computer with at least a Pentium processor, at least a 2MB front camera, and at least 256MB RAM can execute the application.

Webology, Volume 18, Number 4, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I4/123

CONCLUSION:

The best interaction between a machine and a human is achieved through hand gesture recognition. The development of alternative human computer interactions depends on gesture recognition. It makes it easier for people to interact more naturally with machines. Numerous industries, including augmented reality, computer graphics, video games, prostheses, and biomedical instruments, can benefit from this technology.

REFERENCE

1. Tran, Dinh-Son, et al. "Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network." Applied Sciences 10.2 (2020): 722.
2. Oudah, Munir, Ali Al-Naji, and Javaan Chahl. "Hand gesture recognition based on computer vision: a review of techniques." journal of Imaging 6.8 (2020): 73
3. Esposito, Daniele, et al. "A piezoresistive array armband with reduced number of sensors for hand gesture recognition." Frontiers in neurorobotics 13 (2020): 114.
4. Tsai, Tsung-Han, Chih-Chi Huang, and Kung-Long Zhang. "Design of hand gesture recognition system for human-computer interaction." Multimedia tools and applications 79.9 (2020): 5989-6007.
5. Reddy, Vantukala VishnuTeja, et al. "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition." 2020 IEEE-HYDCON. IEEE, 2020.
6. Lee, A-reum, et al. "Enhancement of surgical hand gesture recognition using a capsule network for a contactless interface in the operating room." Computer Methods and Programs in Biomedicine 190 (2020): 105385.
7. Lee, A-reum, et al. "Enhancement of surgical hand gesture recognition using a capsule network for a contactless interface in the operating room." Computer Methods and Programs in Biomedicine 190 (2020): 105385.
8. Jaramillo-Yánez, Andrés, Marco E. Benalcázar, and Elisa Mena-Maldonado. "Real-time hand gesture recognition using surface electromyography and machine learning: a systematic literature review." Sensors 20.9 (2020): 2467.
9. Abhishek, B., et al. "Hand gesture recognition using machine learning algorithms." Computer Science and Information Technologies 1.3 (2020): 116-120.
10. Foroutan, Yalda, et al. "Control of computer pointer using hand gesture recognition in motion pictures." arXiv preprint arXiv:2012.13188 (2020).
11. Shanthakumar, Vaidyanath Areyur, et al. "Design and evaluation of a hand gesture recognition approach for real-time interactions." Multimedia Tools and Applications 79.25 (2020): 17707-17730.
12. Arozi, Moh, et al. "Pattern recognition of single-channel sEMG signal using PCA and ANN method to classify nine hand movements." Symmetry 12.4 (2020): 541.
13. Waybhase, Nitin, et al. "Towards Controlling Mouse through Hand Gestures: A Novel and Efficient Approach." Mapana Journal of Sciences 21.2 (2022).
14. Mahmud, Saifuddin, Xiangxu Lin, and Jong-Hoon Kim. "Interface for human machine interaction for assistant devices: a review." 2020 10th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2020.
15. Shetty, Monali, et al. "Virtual Mouse Using Object Tracking." 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020.