Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
An implementation of virtual white board using open CV for virtual classes

S.V. Aswin Kumer*, P. Kanakaraja, Sheik Areez, Yamini Patnaik, Pamarthi Tarun Kumar
Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP 522502, India

ARTICLE INFO

Article history:
Received 22 December 2020
Accepted 16 February 2021
Available online 10 March 2021

Keywords:
Open CV
Python
Image processing
Feature extraction
Virtual white board
Virtual classroom

ABSTRACT

In the current pandemic situation, we have seen one of the reasons for spreading of corona virus is surface touching, and to avoid this sort of spread in the virus we have come up with a project which is titled as “Drawing on Air”, As the title itself indicates, we are not going to touch the surface but we are going to draw on air and what we have drawn on air will be displayed on the screen / monitor. This will help us reduce the spread of the virus in the pandemic situations. In this pandemic we have seen maximum of the transactions are been made online but for deposit of money or for transfer of money in person we see people use ATMs which is very risky and there are chances of spreading of the virus at such point of times we can use this as a prevention method to a great extent. Not only in ATMs but also in offices for biometrics or in cyber cafes and many more areas we can use this project.

© 2020 Elsevier Ltd. All rights reserved.
Selection and peer-review under responsibility of the scientific committee of the International Conference on Materials, Manufacturing and Mechanical Engineering for Sustainable Developments-2020.

1. Introduction

All we need in this project is a laptop with a web camera installed in it. We will train our laptop or our screen to read whatever the user will be writing infront of the screen. We have used Open CV [2] for object detection that is with which the user will be writing on air and we have used Python language for coding since Python is an easy language and easily understandable and also, we have used Jupyter Notebook latest version since it produces a native and managed code. Firstly, after code is executed, we get a white screen displayed in the monitor and whatever we write on air infront of camera it tracks the object whose properties we have declared in the code. After the tracked points are connected and projected on the screen Fig. 1 Fig. 2 Fig. 3 Fig. 4 Fig. 5 Fig. 6 Fig. 7.

1.1. Open CV

Open CV (Open-Source Computer Library) is a programming feature library specifically targeted at computer vision [2] in real time. This is an open-source platform and free for use. Its primary interface is in c++, it still preserves an older c interface that is less detailed but vast. In the Python interface, all the latest advancements and algorithms appear. It is major open-source computer vision, machine learning and image processing library and now it plays an important part in real time activity in today's systems. We will use this to process photographs and videos to recognize human beings, faces or even handwriting. Python is able to process the OpenCV array structure for review as it is combined with different libraries such as NumPy. We use vector space to recognize the image pattern and its different characteristics and perform arithmetic computations on these traits. It is accessible on Windows, Linux, iOS etc., with Python, C++, C and Java as interfaces.

1.2. Python

Python is a high-level programming language. It is mainly famous for code reusability and simplicity. Even though it is slower it has an important characteristic of python that it can be easily extended within c. Because of this characteristic we can write computationally intrinsic codes in C++/C. Python supports different forms of programming patterns such as procedural programming, object-oriented programming, functional programming etc. It consists of NumPy library. NumPy is highly powerful and optimized for mathematical operations. In order to take advantages of multi-core computing, all items are written in optimized C or C++.
2. Related work

To implement this work, one should first learn object tracking [1] and one should be aware of OpenCV [2] and Jupyter Notebook. Using object tracking we detect the HSV code [3] that is similar like RGB color code [4]. HSV is Hue, Saturation and Value respectively.

Typically tracking algorithms [5] are quicker than detection algorithms. We know a lot about the objects appearance when we are monitoring an object which was observed in the previous frame, we already know the position in the next frame [6], we can use all this data to determine the position of object in the next frame, and to precisely find the object with a limited scan around the estimated location of the object. In the other hand, a successful monitoring algorithm can manage any amount of obstruction. The motion model estimates the objects precise location. To have a more precise assessment based on appearance, the appearance model makes improvements on this assessment.

3. Proposed method

3.1. Work flow

3.1.1. Methodology

So basically, the screen detects the color whose properties we declare in the code. H, S, V are Hue, Saturation and Value respectively. This color code is similar to the RGB color code. You will have to run the object Tracking code e beforehand in order to get the HSV value. All you have to do is slide the taskbars until the only remaining white parts of the “Thresholded Image” windows are correspondent to object [7,8] that you want to detect in real life.

For example, if you want to detect the yellow sharpie, you will have to slide the taskbars so that the white parts of the “Thresholded Image” windows are actually the threshold image of the yellow sharpie [11,12]. After you get HSV value you want, save it for near future references. Then run the webcam code [9,10] remember to change the HSV values into whatever HSV value you found previously. Then hit the “debug” button to draw.

4. Results and discussions

After debug the webcam automatically turns on and we get to draw Infront of the webcam and then the output screen appears and we get the output with the help of the HSV values which we have already declared in code. The HSV adjustment plays major role in this implementation, based on its value, the text that the person drawing on air will be reflected and visible clearly at the output.

5. Conclusion and future work

Finally, this implementation has come up with an idea for this project, how one can reduce the chances of spread of contagious virus or diseases that is beneficial for the society with the help of open-source networks i.e., OpenCV and Jupyter Notebook with the help of object detection. This project can be a very helpful project for the society for the pandemic situations. It was a simple illustration of image processing capabilities of OpenCV.

CRediT authorship contribution statement

S.V. Aswin Kumer: Project administration, Conceptualization, Software, Supervision, Visualization, Writing - review and editing.
P. Kanakaraja: Project administration, Conceptualization, Soft-
Fig. 3. Saturation setup in HSV ADJUSTMENT taskbar.

Fig. 4. Value setup in HSV adjustment Taskbar.

Fig. 5. Output window showing the drawn text ‘smile’ with emoji.

Fig. 6. Output window showing the drawn text ‘Hello’.
ware, Supervision, Visualization, Writing - review and editing. **Sheik Areez:** Project administration, Conceptualization, Software, Supervision, Visualization, Writing - review and editing. **Yamini Patnaik:** Project administration, Conceptualization, Software, Supervision, Visualization, Writing - review and editing. **Pamarthi Tarun Kumar:** Project administration, Conceptualization, Software, Supervision, Visualization, Writing - review and editing.

**Declaration of Competing Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgements**

We humbly acknowledge the valuable advice, guidance and cooperation of Dr. ASWIN KUMER S V, ASSOC.PROF, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Educational Foundation, under whose supervision this work has been carried out. His intellectual advice, encouragement and guidance makes us feel confident and he inspired us to go through different research ideas. Through him, we have discovered that scientific analysis takes a lot of time to understand and implement, and we need to have comprehensive viewpoint on topics from multiple viewpoints. To all the faculty members, administrators and employees of the Department of Electronics and Communication Engineering, we would like to express our sincere gratitude as they have always shared their support to finish this research.

**References**

[1] Ministry of Health, Labour and Welfare. “The survey of difficulties of daily life (survey of nationwide home children with disabilities, and others, 2011),” (in Japanese), Dept. Health Welfare Persons Disabilities, Tokyo, Japan, Tech. Rep., 2013.

[2] S. Tanaka, M. Takuma, Y. Tsukada, Recognition of finger alphabet remote image sensor, (in Japanese), in Proc. 76th Nat. Conv. (IPSJ), 2014, vol. 76. no. 2, pp. 2275–2276.

[3] T. Miyake, D. Wakatuki, I. Naito, “A basic study on recognizing fingerspelling with hand movements by the use of depth image,” (in Japanese), Nat. Univ. Corporation Tsukuba, Univ. Technology, Tsukuba, Japan, Tech. Rep., Dec. 2012, vol. 20, no. 1, pp. 7–13.

[4] D. Takabayashi et al., Training system for learning finger alphabets with feedback functions (in Japanese), IEICE (HIP), Tech. Rep. 112 (483) (2013) 79–84.

[5] M. Maehatake, M. Nishida, Y. Horiuchi, A. Ichikawa, A study on sign language recognition based on gesture components of position and movement, (in Japanese), in Proc. Workshop Interact. Syst. Softw. (WISS), Japan, Dec. 2007, pp. 129–130.

[6] A. Sato, K. Shinozuka, T. Miyake, Y. Horiuchi, A. Ichikawa, A recognition algorithm of numerals written in the air by a finger-tip (in Japanese), J. IPSJ 52 (2) (2011) 910–916.

[7] T. Sonoda, Y. Murakawa, A letter input system of handwriting gesture, (in Japanese), IEICE Trans. Commun., vol. 86-D-II, no. 7, pp. 1015–1025, 2003.

[8] Y. Fujiwara, M. Takezawa, H. Sanada, K. Watanabe, An aerial handwritten character input system (in Japanese), IPSJ (MBL), Tech. Rep. 50 (6) (2009) 1–4.

[9] Y. Tanaka, Training system for learning finger alphabets with feedback functions, M.S. thesis, Faculty Ind. Technol., Tsukuba Univ. Technol., Tsukuba, Japan, 2014.