International Conference - Innovation-2021-Innovation-2021
International Journal of Scientific Research in Computer Science, Engineering and Information Technology ISSN : 2456-3307 (www.ijsrcseit.com)
Volume 8, Issue 3, May-June-2021

Virtual Painting with OpenCV using Python

Yash Patil[1], Karunesh Singh Bais[1], Deep Paun[1], Mihir Paun[1], Vishal Kisan Borate[2]

[1]Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India

[2]Assistant Professor, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India

ABSTRACT

In modern technologies video tracking and processing the feed has been very essential. This processed data can be used for many research purposes or to express a particular output on a particular system. There are various methods for processing and manipulation of data to get the required output. This paint application is created using OpenCV module and python programming language which is an apex machine learning tool to create an application like this. Given the real time webcam data, this paint-like python application uses OpenCV library to track an object-of-interest (a bottle cap in this case) and allows the user to draw by moving the object, which makes it both awesome and challenging to draw simple things.

Keywords : Machine learning, OpenCV, Morphing Techniques, Human-Computer Interactions, Air Writing.

I. INTRODUCTION

OpenCV was launched in August 1999 at the Computer Vision and Pattern Recognition conference (and so turns 17 years old at the publication of this book). Gary Bradski founded OpenCV at Intel with the intention to accelerate both the research and use of real applications of computer vision in society. OpenCV has nearly 3,000 functions, has had 14 million downloads, is trending well above 200,000 downloads per month, and is used daily in millions of cell phones, recognizing bar codes, stitching panoramas together, and improving images through computational photography.

OpenCV is at work in robotics systems—picking lettuce, recognizing items on conveyor belts, helping self-driving cars see, flying quad-rotors, doing tracking and mapping in virtual and augmented reality systems, helping unload trucks and pallets in distribution centres, and more—and is built into the Robotics Operating System (ROS) [1]. It is used in applications that promote mine safety, prevent swimming pool drownings, process Google Maps and street view imagery, and implement Google X robotics, to name a few examples.

OpenCV has been re-architected from C to modern, modular C++ compatible with STL and Boost. The library has been brought up to modern software development standards with distributed development on Git. Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. OpenCV is a computer vision and machine learning software library that includes many common image analysis algorithms that will help us build custom, intelligent computer vision applications. In this application frequent image feed results in video tracking of our particular object of interest.

Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging, and video editing.

II. RELATED WORK

Machine Learning applications are becoming an integral part of our life and for the surrounding society. Real life applications in the field of machine learning can improve the efficiency and consistency in our day-to-day work, by using computer vision and its associating algorithms and its subsequent modules or libraries can help us develop this type of real-life applications.

OpenCV library is an open source library for building a machine learning architecture that can track and process real life data and manipulate it for a required set of expected output.

Related research on various papers and its summaries are as follows:

Title of the Paper is Hand Gesture Recognition its authors are Rafiqul Zaman Khan, Noor Adnan Ibraheem and the related paper's essential aim of building hand gesture recognition system is to create a natural interaction between human and computer where the recognized gestures can be used for controlling a robot or conveying meaningful information. How to form the resulted hand gestures to be understood and well interpreted by the computer considered as the problem of gesture interaction Human computer interaction (HCI) also named Man-Machine Interaction.

Title of the Paper is Gesture Controlled Computer its authors are S. T. Gandhe, Nikita A. Pawar, Mayuri S. Hingmire, Kalpesh P. Mahajan, Devshri V. Patil. This paper describes the method for humans to interact with digital world and use the computer with just our hand movements. The paper is based on image processing. The camera detects gestures and converts those gestures into equivalent digital algorithms as programmed. This paper deals with the controlling all operations of mouse such as right click, left click and movement of cursor over the desktop, drag and drop, snapshot, Air writing and painting through hand gestures.

III. PROPOSED SYSTEM

The proposed system can be classified into mainly two steps after acquiring the input image from camera, videos or even an Object of Interest. These steps are: Extraction Method image pre-processing and Features estimation and Extraction [1].

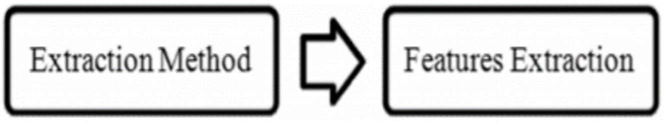

Figure 1: System of architecture.

A. Extraction Method and image pre-processing:

Segmentation process is the first process for recognizing Object of Interest. It is the process of dividing the input image (in this case Object of Interest image) into regions separated by boundaries.

The segmentation process depends on the type of object, if it is dynamic object then the Object need to be located and tracked, if it is static object with no moving parts such as a coloured bottle cap or a pen of particular colour the input image has to be segmented only. The object should be located firstly, generally a bounding box is used to specify the depending on the object colour and secondly, the object have to be tracked, for tracking the object there are two main approaches either the video is divided into frames and each frame have to be processed alone, in this case the object frame is treated as a moving OBJ(Object-of-Interest) and segmented, or using some tracking information such as shape, object colour using some filter. The common helpful cue used for segmenting the object is its colour, since it is easy and invariant to scale, translation, and rotation changes. In this case a coloured markers which provide exact information about the orientation and position.

The colour space used in a specific application plays an essential role in the success of segmentation process, however colour spaces are sensitive to lighting changes.

IV. FEATURES EXTRACTION

Good segmentation process leads to perfect features extraction process and the latter play an important role in a successful recognition process.

The modules of OpenCV for image processing applications are given below:

CORE module contains basic data structures and basic functions used by other modules.

IMGPROC module contains image processing related functions such as linear, non-linear image filtering and geometrical image transformations etc. **VIDEO** module contains motion estimation and object tracking algorithms.

ML module contains machine-learning interfaces. **HighGUI** module contains the basic I/O interfaces and multi-platform windowing capabilities.

Feature's vector of the segmented image can be extracted in different ways according to particular application. Various methods have been applied for representing the features can be extracted. In this method we have used the shape of the object such as object contour [2]. A contour is a closed curve of points or line segments that represents the boundaries of an object in the image. Contours are essentially the shapes of objects in an image. Contours are sometimes called a collection of points or line segments that overall represent the shape of the object in an image.

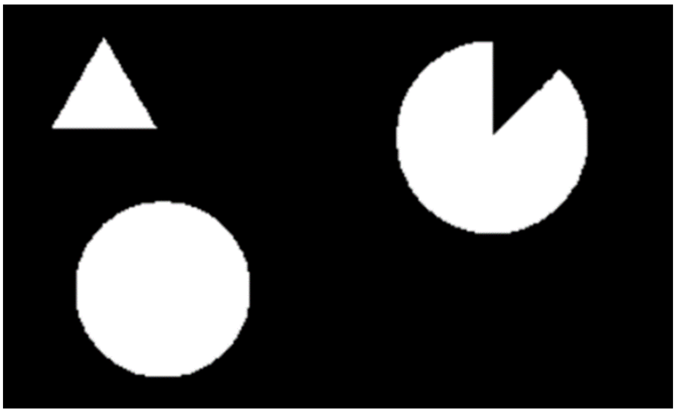

Figure 2: Silhouette using contour selection

V. PROPOSED ALGORITHM

Implementation Method:

In this machine learning application, we have developed code by using Python programming language along with OpenCV library. Main idea behind this algorithm is to use live feed from camera and process each frame. However, the algorithms will be implemented on defined ROI (region of interest).

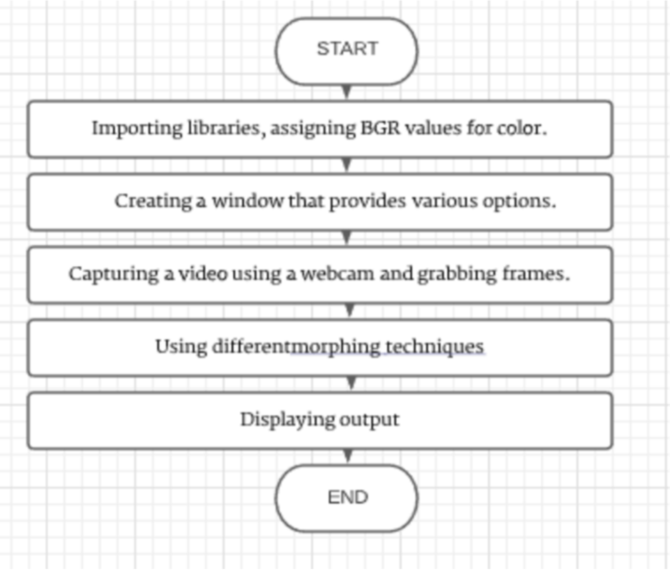
START
Importing libraries, assigning BGR values for color.
Creating a window that provides various options.
Capturing a video using a webcam and grabbing frames.
Using differentmorphing techniques
Displaying output
END

Figure 3: Algorithm.

Step 1: Initialize Some Stuff

Firstly, we import the necessary libraries. Following are mentioned in the image:

```
import numpy as np
import cv2
from collections import deque
```

Then we initialize variables that are used in the following steps. We can take any colour in this case we have taken blue colour to be traced on the Contour-Of-Interest.

The blueLower and the blueUpper NumPy arrays help us in finding the blue coloured cap. The kernal helps in smoothing blue cap once found. The bpoints, gpoints, rpoints and ypoints deques are used to store the points drawn on the screen of colour blue, green, red, and yellow respectively.

Step 2: Setup the Paint Interface

Now we manually set the coordinates of each of the colour boxes on the frame. We use the OpenCV function cv2.rectangle() to draw the boxes.

Step 3: Start Reading the Video (Frame by Frame)

Now we use the OpenCV function cv2.VideoCapture() method to read a video, frame by frame (using a while loop), either from a video file or from a webcam in real-time. In this case, we pass 0 to the method to read from a webcam. We can just add the exact same paint interface for ease of usage.

Step 4: Find the Contour-Of-Interest

Once we start reading the webcam feed, we constantly look for a blue colour object in the frames with the help of cv2.inRange() method and use the blueUpper and blueLower variables initialized in Step 0. Once we find the contour, we do a series of image operations and make it smooth. These operations are specific type of morphing techniques as defined below.

Morphing Techniques:

Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images.

Two basic morphological operators are Erosion and Dilation. Then its variant forms like Opening, Closing, Gradient etc. also comes into play. We have tracked the object of interest by the techniques of dilation and erosion.

Once we find the contour (the if condition passes when a contour is found), we use the centre of the contour (blue cap) to draw on the screen as it moves.

The code finds the contour (the largest one), draws a circle around it using the cv2.minEnclosingCircle() and cv2.circle() methods , gets the centre of the contour found with the help of cv2.moments() method.

Step 5: Start Drawing and Store the Drawings

Now we start tracking coordinates of each point the centre of the contour touches on the screen, along with its colour. We store these set of points of different colours in different deques (bpoints, gpoints etc.). When the centre of the contour touches one of the coloured boxes we put on the screen in Step 1, we store the points in its respective colour deque.

Step 6: Show the Drawings On the Screen

VI. RESULT

We have stored all the points in their respective colour deques. Now we just join them using a line of their own colour. The OpenCV function cv2.line() comes in handy for us to do that.

Once we join all the points in each and every frame with a line and put it on both the windows we created using cv2.imshow() method and it all fits perfectly to work like a paint application. After falling out of the while loop, we entered to read data from the webcam, we release the camera and destroy all the windows hence terminating the application.

Here Colour Detection and tracking is used in order to achieve the objective. The colour marker in detected and a mask is produced. It includes the further steps of morphological operations on the mask produced which are Erosion and Dilation. Erosion reduces the impurities present in the mask and dilation further restores the eroded main mask.

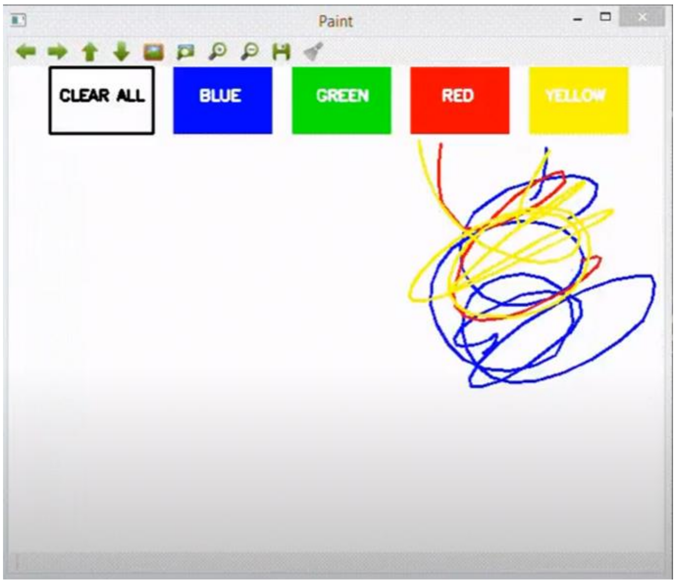

Figure 4: Application GUI.

VII. FUTURE WORK

Some examples that are being implemented in the near future are as follows:

Video Processing using Android Phone:

Mobile devices such as smart phones, iPads and tablet pcs are equipped with cameras, the demand of the image processing applications increased. These applications need to be faster and consumes lower power because the mobile device is only powered by a battery. The hardware technology depends on the semiconductor technology instead we can use an efficient programming language to write an image processing application for the mobile devices [4].

Robot Control:

Controlling the robot using gestures considered as one of the interesting applications in this field proposed a system that uses the numbering to count the five fingers for controlling a robot using hand pose signs. The orders are given to the robot to perform a particular task, where each sign has a specific meaning and represents different function [16].

VIII. CONCLUSION

Demonstration of the image processing capabilities of OpenCV. The ultimate goal is to create a computer vision machine learning application that promotes Human computer interaction (HCI) also named Man-Machine Interaction (MMI)] refers to the relation between the human and the computer or more precisely the machine, and since the machine is insignificant without suitable utilize by the human there are two main characteristics should be deemed when designing a HCI system as mentioned: functionality and usability [5]. System functionality referred to the set of functions or services that the system equips to the users while system usability referred to the level and scope that the system can operate and perform specific user purposes efficiently.

IX. REFERENCES

[1]. https://towardsdatascience.com/Painting-with-opencv

[2]. https://www.geeksforgeeks.org/live-webcam-drawing-using-opencv

[3]. Volume 43, Issue 1, June 2012 ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering: Research Paper -- Hand Data Glove: A Wearable Real-Time Device for Human-Computer Interaction

[4]. International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.4, July 2012, DOI:

10.5121/ijaia.2012.3412 161- HAND GESTURE RECOGNITION: A LITERATURE REVIEW

[5]. OpenCV for Computer Vision Applications, Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 20, 2015

[6]. https://www.geeksforgeeks.org/live-webcam-drawing-using-opencv/

[7]. https://docs.opencv.org/master/d5/d54/group__objdetect.html

[8]. https://circuitdigest.com/tutorial/real-life-object-detection-using-opencv-python-detecting-objects-in-live-video

[9]. Real Time Object Detection and Tracking Using Deep Learning and OpenCV Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018) IEEE Xplore Compliant Part Number: CFP18N67-ART; ISBN:978-1-5386- 2456-2

[10]. Numpy.org, 2017. [Online]. Available: http://www.numpy.org

[11]. (2017, January 17). Object Detection [Online].Available: http://en.m.wikipedia.org/wiki/Object_detection

[12]. Study on Object Detection using Open CV – Python, International Journal of Computer Applications (0975 – 8887) Volume 162 – No 8, March 2017

[13]. Nidhi, "Image Processing and Object Detection", Dept. of Computer Applications, NIT, Kurukshetra, Haryana, 1(9): 396-399, 2015

[14]. OpenCV, Open source Computer Vision library. In http://opencv.willowgarage.com/wiki/, 2009

[15]. http://docs.opencv.org/modules/imgproc/doc/imgproc.html

[16]. http://en.wikipedia.org/wiki/OpenCV

[17]. Javed O, Shafique K, Shah M. A hierarchical approach to robust background subtraction using colour and gradient information. IEEE Workshop on Motion and Video Computing (MOTION' 02); 2002.

[18]. Ammar Anuar, Khairul Muzzammil Saipullah, Nurul Atiqah Ismail, Yewguan Soo "OpenCV Based Real-Time Video Processing Using Android Smartphone", IJCTEE, Volume 1, Issue 3

[19]. Yash Patil, Karunesh Singh Bais, Deep Paun, Mihir Paun, Vishal Kisan Borate, "Virtual Painting with Opencv Using Python", Volume 5, Issue 8, November December 2020 ISSN: 2395 602X, International Journal of Scientific Research in Science and Technology.