International Journal of Engineering Research and Modern Education (IJERME)
Impact Factor: 7.018, ISSN (Online): 2455 - 4200
(www.rdmodernresearch.com) Volume 7, Issue 1, 2022

AIR CANVAS APPLICATION USING OPENCV AND NUMPY IN PYTHON

Dr. M. Parameswari*, L. Priyanka, G. R. Swetha**, T. Uma** & E. Vaishnavi****

* Associate Professor, Department of Electrical and Electronics Engineering, Vivekanandha College of Technology for Women, Tiruchengode, Tamilnadu

** UG Student, Department of Computer Science and Engineering Vivekanandha College of Technology for Women, Tiruchengode, Tamilnadu

Cite This Article: Dr. M. Parameswari, L. Priyanka, G. R. Swetha, T. Uma & E. Vaishnavi, "Air Canvas Application Using Open CV and Numpy in Python", International Journal of Engineering Research and Modern Education, Volume 7, Issue 1, Page Number 10-13, 2022.

Abstract:

On-air writing has become one of the most attractive and challenging areas in the field of image processing and pattern recognition in recent years. It contributes to the development of automation processes and can improve interpersonal and machine interaction across multiple systems. Many research projects focus on new techniques and techniques that can reduce processing time while providing high accuracy of recognition. Tracking an object is considered an important function within the field of Computer Vision. The invention of fast computers, the availability of affordable and high-quality video cameras and the requirements of automated video analysis have provided prominence in tracking techniques. This project focuses on developing a motion-to-text converter that can serve as a software for smart portable writing tools. This project is a touching journalist from time to time. It will use a computer view to track your fingerprint.

Key Words: Pi Camera, Open CV, Computer Vision, Image Processing

1. Introduction:

Air Canvas is a hands-free digital drawing canvas that uses Pi Camera, and Open CV to visualize and map to touch with your fingertips. User "brush" can be adjusted in size and color using the built-in buttons. Here Color Detection and tracking are used to achieve the objective. Color marking is obtained and a mask is produced. It includes additional stages of morphological activity in the mask the effect of erosion and elasticity. Erosion reduces the impurities present in the mask and stretching restores the main eroded mask. We started our project by looking for open-source handwriting recognition software that uses OpenCV in conjunction with Python. In doing so, the design of our project changed as we acquired various image processing algorithms. Our old use wanted to use hand gestures to control our color and size variation. To do so, we first plan to create an image mask that will separate the hand from the back. With some effort and error, we used OpenCV, successfully photographed it, Gaussian blurred it out, and we used a binary mask to significantly compare the shape of the hand from behind. This is a method found in Izane's Fingerprint Study chosen for its use in finding flexibility; in other words, to cut holes between the fingers. However, we found that the sensitivity of the camera to the light of the lab room made this a difficult task and we often encountered external silhouettes in our processed image.

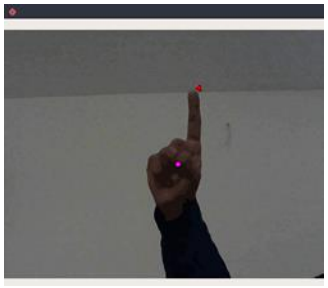
2. Objective:

In this project, we created an Air Canvas that can pull anything out of it by simply capturing the movement of the colored marker with the camera. Here a colored object is used on the fingers as a marker. We used OpenCV computer monitoring techniques to build this project. The language we have used Python in its complete libraries and easy-to-use syntax but basic understanding can be applied to any language supported by OpenCV.

3. Methodology

Two main parts have been discussed under this topic. This system needs a dataset for the Fingertip Detection Model. The Fingertip Model's primary purpose is used to record the motion, i.e., the air character.

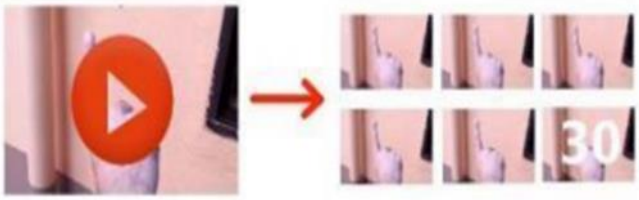
A. Fingertip Detection Model:



Air typing can only be done using a stylus or a different colored air pen. The system, however, uses the finger. We believe that people should be able to write in the air without the pain of carrying a pen. We used the Deep Learning algorithms to obtain fingerprints throughout the framework, generating a list of links. The image below shows a point that is considered a pointing finger. The center is purple and the most remote area is red. And when you have it, get a finger.

B. Techniques of Fingertip Recognition Dataset Creation:

a. Video to Images: In this way, two-second videos of human hand movements are filmed in different places. These videos were then divided into 30 different images. However, since the 30 images produced were from the same video and were in the same location, the data set was toned. Therefore, the model did not work well on different domains from the database.



b. Take Pictures in Distinct Backgrounds: In order to overcome the barrier caused by the lack of diversity in the previous approach, we have created a new website. In this, we realized that we needed to touch something in order to control the system. The idea was to allow the model to clearly see all four fingers. This will allow the user to control the system using the number of fingers they display. Now you can - fast type with one finger pointing, convert this writing movement into two-finger e-text, add space with three fingers, hit backspace with five fingers, predictive mode with four fingers, and show 1.2 fingers, 3. to select the first prediction, 2nd or 3rd respectively. To exit prediction mode, point to five fingers. This database contained 1800 images. With the use of the script, a pre-trained model automatically labels this database. We then edited the images with the wrong labels and introduced another model. 94% accuracy achieved. In contrast to the original, this model worked well in different domains. The image below shows the images in different domains.

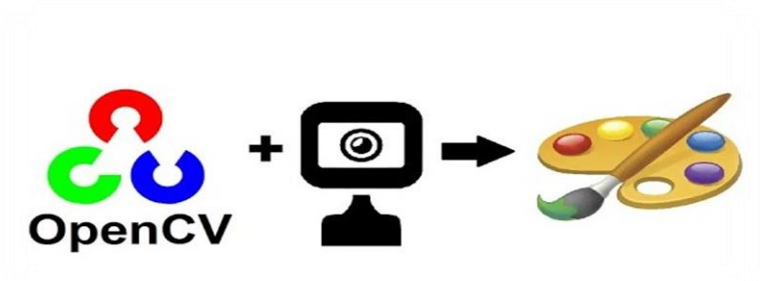
C. Fingertip Recognition Model Training: When the database is ready and the label is divided into train and dev sets (85% -15%). We used the Single Shot Detector (SSD) and RCNN's fast-paced models to train our data. RCNN acceleration was significantly better compared to SSD.SSDs combining two standard access modules - single lifting circuits with each separator. These speeds up performance as items are acquired with a single shot. It is usually used for real-time detection. RCNN Speed uses an outline feature map from Fast RCNN to calculate regional suggestions. They are processed by the Regional Proposal Network and referred to the Regional of Interesting layering. The result is finally given two layers that are fully connected to separate and undo. Fold the fully integrated Faster RCNN layer to see the image fingerprints on the image.

4. Implementation:

We have completed the rest of our project design by polishing the front code, adding brush modification functions, and removing any dependencies on external devices (monitor, mouse, keyboard, etc.). First, we selected easy-to-use screens: one was a measurement screen, and the other was a real drawing screen. Lastly, add the two on-screen buttons and touch function. The "draw" button switches between active and inactive drawing, allowing the user to pause and start drawing as he pleases. The "equalize" button allows the user to return to the original rating screen to start a new drawing or re-resize as desired. Finally, we realized that our visual button embedded in the map was faulty and would only register to click with greater force. So, we decided to choose a screen-changing color function, which allows the user to switch between brush colors by tapping anywhere on the upper part of the screen.

This is the most exciting part of our system. Writing involves many activities. Therefore, the amount of touch used to control the system is equal to the number of actions involved. The basic functions we have included in our system are

- Typing Mode - In this case, the system will track fingerprints and save them.
- Colour Mode - User can change text colour between different available colours.
- Backspace - Says if a user makes a mistake, we need to touch to add a quick backspace.



5. Output:

Figure: Front End of the System

6. Conclusion:

The program has the potential to challenge traditional writing methods. Eliminates the need to carry a cell phone to take notes, providing an easy way to get there to do the same. It will also serve a great purpose in helping especially people who can communicate easily. Even adults or people who find it difficult to use keyboards will be able to use the system easily. To extend the functionality, the system can also be used to control IoT devices soon. Airborne painting can also be made possible. The program will be an excellent software for smart clothing people who can better interact with the digital world. Augmented Reality can bring the text to life. There are certain system limitations that can be upgraded in the future. First, using a handwriting substitute instead of a letter viewer will allow the user to type word for word, making typing faster. Second, a temporary hand-held touch can be used to control the real-time system as is done instead of using a finger number.

7. References:

1. Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016.
2. P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, pp. 1-6, 2016.
3. Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4, 2013
4. Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006
5. Yuan-Hsiang Chang, Chen-Ming Chang, "Automatic Hand-Pose Trajectory Tracking System Using Video Sequences", INTECH, pp. 132- 152, Croatia, 2010
6. Erik B. Sudderth, Michael I. Mandel, William T. Freeman, Alan S. Willsky, "Visual Hand Tracking Using Nonparametric Belief Propagation", MIT Laboratory For Information & Decision Systems Technical Report P2603, Presented at IEEE CVPR Workshop On Generative Model-Based Vision, Pp. 1-9, 2004
7. T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton, "Creating Principal 3D Curves with Digital Tape Drawing," Proc. Conf. Human Factors Computing Systems (CHI' 02), pp. 121- 128, 2002.
8. T. A. C. Bragatto, G. I. S. Ruas, M. V. Lamar, "Real-time Video-Based Finger Spelling Recognition System Using Low Computational Complexity Artificial Neural Networks", IEEE ITS, pp. 393-397, 2006
9. Yusuke Araga, Makoto Shirabayashi, Keishi Kaida, Hiroomi Hikawa, "Real Time Gesture Recognition System Using Posture Classifier and Jordan Recurrent Neural Network", IEEE World Congress on Computational Intelligence, Brisbane, Australia, 2012
10. Ruiduo Yang, Sudeep Sarkar, "Coupled grouping and matching for sign and gesture recognition", Computer Vision and Image Understanding, Elsevier, 2008
11. R. Wang, S. Paris, and J. Popovic, "6D hands: markerless hand-tracking for computer-aided design," in Proc. 24th Ann. ACM Symp. User Interface Softw. Technol., 2011, pp. 549–558.
12. Maryam Khosravi Nahouji, "2D Finger Motion Tracking, Implementation For Android Based Smartphones", Master's Thesis, CHALMERS Applied Information Technology,2012, pp 1-48
13. Eshed Ohn-Bar, Mohan Manubhai Trivedi, "Hand Gesture Recognition In Real-Time For Automotive Interfaces," IEEE Transactions on Intelligent Transportation Systems, VOL. 15, NO. 6, December 2014, pp 2368-2377
14. P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, 2016, pp. 1- 6.
15. Kenji Oka, Yoichi Sato, and Hideki Koike, "Real-Time Fingertip Tracking and Gesture Recognition," IEEE Computer Graphics and Applications, 2002, pp.64-71.