Carry Out Computer Tasks with Gesture using Image Pre-Processing and TensorFlow Framework

Swapnil Mishra, Payal Gaikwad, Sanjay Singh, Pratik Magar
Students
Department of Computer Engineering JSPM Narhe Technical Campus, Pune, India

Abstract : **The method for real time Hand Gesture Recognition and feature extraction using a web camera. For humans, hands are used most frequently to communicate and interact with machines. Mouse and Keyboard are the basic input/output to computers and the use of both the devices require the use of hands. Most important and immediate information exchange between man and machine is through visual and aural aid, but this communication is one-sided. To help somewhat mouse remedies this problem, but there are limitations as well. Although hands are most commonly used for day to day physical manipulation tasks, but in some cases they are also used for communication. Hand gestures support us in our daily communications to convey our messages clearly. Hands are most important for mute and deaf people, who depends on their hands and gestures to communicate, so hand gestures are vital for communication in sign language. Hand gesture interaction has been the trending technology for human-computer interaction (HCI). Frequently a number of research works are carried out in this area to expedite and contrive interaction with computers. In this project, we are attempting to create a real-time human-computer interaction system (HCI) using different hand gestures i.e, hand signs. We implemented a system that recognizes the hand gesture performed using the simple web camera of a PC. We used the already established OpenCV library with TensorFlow to implement the various methods of Image Processing. Operations that were carried out were Capturing frames, Background Subtraction using MOG filter, Noise Reduction using Gaussian Blur, converting the captured image to binary image, finding the contours through Convex Hull method which is used for removing convexity defects and segment the image. Using these segmented images we built our own dataset that was used to train the model with TensorFlow. Then we used these methods again for the segmentation of the input image which is then passed to the model to get the output class label.**

Index Terms **- HCI, GOS, CNN, GUI, ROI, GPU.**

I. INTRODUCTION

This paper is about a Real time System which can recognize the human gesture and perform a specific operation on the computer as per the human gesture. We have implemented a simple GUI which will guide the user in understanding how the system works. A CNN model is trained using the dataset we created, which is used to predict the gesture based on the input taken from the web camera of user's PC. Since the advent of the computer, the user has been forced to conform to the interface dictated by the machine. In the 1960s the keyboard of the punch card machine and teletype was considered a big improvement over flipping banks of switches, but the user still had to learn to operate multiple machines to use a computer. When the interactive dumb terminals arrived in the 1970s, all the user had to do was learn to type. However even typing was seen as a burden, and a more efficient interface was developed.

Graphical operating systems of the 1980s, inspired by the "look-and-feel" of a desktop, introduced the mouse-a simple pointing device for the user. In the 1990s, with increases in computational power, decent speech recognition and pen-based computing has become a reality. Some user interfaces have explored the individual modes of communication in a limited sense. All the aforementioned interfaces, with possible exception of the speech recognizer, have been efficient for a trained user, but they are typically inefficient as a human-centric form of communication. For example, when communicating with a friend, we would rather see the person and converse with him or her, rather than staring at a terminal full of characters while typing a message. With recent advances in Human Computer Intelligent Interaction (HCII) it has became more feasible to create interfaces that resemble forms of human communication. Although it is still impossible to create an ubiquitous interface that can handle all forms of human communication, it is possible to create a small multi-modal subset.

Keeping that in mind a system that can take normal hand gestures as input is possible to implement. We need the interaction system to be easy to use so we have used the most common hand gestures that humans use in day to day life. Each of the gestures is mapped with a specific task, which will be performed by the computer when such a gesture is given as input to the system. This system can be used by any person without it being specifically tailored for anyone.

II. OBJECTIVES

1. Creating a system that can segment ROI i.e, hand gesture from the captured frame.
2. Creation of big enough dataset for multiple hand gestures that are commonly used in day to day interactions.
3. Creating a classification model using CNN and TensorFlow framework to classify the input hand gesture into appropriate class label.
4. Creating a simple GUI that guides the user into understanding the working of the gesture recognition system. Mapping each gesture to the specific task that is to be performed by the PC.

III. PROJECT SCOPE

Hand gesture interaction has been the trending technology for human-computer interaction (HCI). Frequently a number of research works are carried out in this area to expedite and contrive interaction with computers. In this project, we are attempting to create a real-time human-computer interaction system (HCI) using different hand gestures. OpenCV library can be used perform the needed task. This library is selected because of the major features and community support available. Tensorflow is used to create the CNN model that is needed for classification of the gestures. It is used because it is free and open-source with lots of features.

IV. LITERATURE REVIEW

1. Ker-Jiun Wang et. al: "Human-Centered, Ergonomic Wearable Device with Computer Vision Augmented Intelligence for VR Multimodal Human-Smart Home Object Interaction" 2019:

In this paper, we will showcase the use of an ergonomic and lightweight wearable device that can identify human's eye/facial gestures with physiological signal measurements.

2. Fang Rong Hsu et. al: "A Study of User Interface with Wearable Devices Based on Computer Vision" 2019:

Smart wearable devices are widely used in the field of healthcare. This article presents three approaches to human–computer interaction (HCI) via computer vision and hand gestures. They are timeline-user interface, virtual keyboard user interface, and handwritten digit user interface, respectively. These user interfaces are achieved by including the image process and machine learning, such as Convolutional Neuron Networks and AdaBoost. To evaluate the approaches, a prototype of the android device with a built-in color camera was employed. The result of the handwritten digit model was significant for high accuracy of 92.9%. In our view, the results emphasize the validity of our models.

3. Jing Xiao, et. al: "An Electrooculogram-Based Interaction Method and Its Music-on-Demand Application in a Virtual Reality Environment" 2019 :

This paper proposes a new nonmanual human-computer interface (HCI) based on a single-channel electrooculogram (EOG) signal and enables real-time interactions with the VR environment. The graphical user interface of the EOG-based HCI in VR includes several buttons flashing in a random order. The user needs to blink in synchrony with the corresponding button's flashes to issue a command, while the algorithm detects the eye blinks from the EOG signal and determines the users' target button. Furthermore, with the EOG-based HCI, we developed a music-on-demand system in the VR environment.

4. Camila Loiola Brito Maia et. al: "An Approach to Analyze User's Emotion in HCI Experiments Using Psychophysiological Measures" 2019:

In this paper, three combined non-invasive psychophysiological measures were used to verify which of them represents the emotion's dimensions. Besides that, an approach to studying the tendency of user's emotion is presented, assisting HCI researchers in HCI experiments. An experiment was conducted using quantitative and qualitative data analysis, and the results show important correlations that were used in the proposed approach.

5. Dharmaraj Ojha et. al: "Histogram based Human Computer Interaction for Gesture Recognition" 2019 :

This paper introduces a technique for human computer interaction using open CV and python. We have first detect, pre-processing and recognize the hand fingers and the count. Then with the help of recognized fingers count, it is act as a mouse to perform the different operations and this hand mouse interface known as a "virtual monitor". The hand mouse is controlled by the virtual monitor provides a virtual space. The accuracy of the proposed algorithm is 80%. This envisioned concept controlling a system by hand has been implemented successfully with effective efforts.

6. Liannan Lin et. al: "Intelligent Human-Computer Interaction: A Perspective on Software Engineering" 2019:

The design of traditional human-computer interaction courses is facing new challenges due to the breakthrough of the third generation of AI technology. New human-computer interaction scenarios, such as smart home and driverless cars, keep on emerging. More natural and efficient intelligent interaction methods are widely used in these scenarios, generating brand- new user experience. Combined with an example on the interactive design of intelligent products and the previous experience on teaching. In this article, an innovative design of human-computer interaction courses is introduced from the perspective on innovative content, cultivation of talents, and practice of software engineering.

7. Jae-Woo Choi et. al: "Short-Range Radar Based Real-Time Hand Gesture Recognition Using LSTM Encoder" 2019:
This paper proposes a hand gesture recognition system for a real-time application of HCI using 60 GHz frequency-modulated continuous wave (FMCW) radar, Soli, developed by Google. The overall system includes signal processing part that generates range-Doppler map (RDM) sequences without clutter and machine learning part including a long short-term memory (LSTM) encoder to learn the temporal characteristics of the RDM sequences. A set of data is collected from 10 participants for the experiment. The proposed hand gesture recognition system successfully distinguishes 10 gestures with a high classification accuracy of 99.10%. It also recognizes the gestures of a new participant with an accuracy of 98.48%.

8. Gang-Joon Yoon et. al: "Three-Dimensional Density Estimation of Flame Captured From Multiple Cameras" 2019:
This paper presents a 3D density flame reconstruction method, captured from the sparse multi-view images, as a constrained optimization problem between the flame and its projected images. For effective estimation of the flame with a complicated structure in an arbitrary viewpoint, we extract the 3D candidate region of the flame and, then, estimate the density field using the compressive sensing.

9. Yanbo Tao et. al: "Human-Computer Interaction Using Fingertip Based on Kinect" 2018:
Controlling the computer through the hand and eye is a new type human-computer interaction, which overcomes the problem of insufficient flexibility in controlling the traditional computer through the mouse and keyboard. This kind of human- computer interaction is more natural and is an inevitable trend of human-computer interaction in the future. The Kinect based hand-eye positioning system is capable of quickly locating the hand-eye position through depth images and color images, and has a strong real-time performance.

10. Sherin Mohammed Sali Shajideen et. al: "Hand Gestures - Virtual Mouse for Human Computer Interaction" 2018:
This research work focuses on the improvement of human computer interaction systems using hand gesture with 3-D space by using two camera in position. The hand pointing gesture is estimated and mapped to the screen coordinate system. Also we use other hand gestures to complete the action of virtual mouse. We use hand pointing to point to the screen and other gestures for other operations such as selection of a folder/an object.

11. Sherin Mohammed Sali Shajideen et. al: "Human-Computer Interaction System Using 2D and 3D Hand Gestures" 2018:
In this paper, we propose a real-time human-computer interaction system (HCI) using two different hand gestures - hand pointing and clenched fist gesture. We propose a single camera hand gesture estimation algorithm for hand gesture tracking in 2D space from a distance and are mapped to the screen coordinates. Moreover, we also propose orthogonal cameras to estimate hand gestures in 3D space from a distance and is mapped to the screen coordinates.

12. Gu Jialu, S. Ramkumar et. al: "Offline Analysis for Designing Electrooculogram Based Human Computer Interface Control for Paralyzed Patients" 2018:
This paper shows an average classification accuracy of 90.72% for convolution features and 91.28% for Plancherel features. Off-line single trail analysis was also performed to analyze the recognition accuracy of the proposed HCI system. The off- line analysis displayed that Plancherel features using LRNN were high compared to convolution features using LRNN.

13. Shravani Belgamwar et. al: "An Arduino Based Gesture Control System for Human-Computer Interface" 2018:
This paper presents a literature survey conducted which provides an insight into the different methods that can be adopted and implemented to achieve hand gesture recognition. It also helps in understanding the advantages and disadvantages associated with the various techniques.

14. Adrian Hoppe et. al: "Multi-user Collaboration on Complex Data in Virtual and Augmented Reality" 2018:
We propose a technique that gives the remote supporter the ability to see a high fidelity point cloud of a real world object in Virtual Reality (VR). The VR user can indicate points of interest via a laser pointer. The local worker sees these indications on top of the real object with an Augmented Reality (AR) headset. A preliminary user study shows that the proposed method is faster and less error-prone regarding the comprehension of the object and the communication between the users.In addition to that, the system has a higher usability.

15. Atsuo Murata et. al: "Automatic Lock of Cursor Movement: Implications for an Efficient Eye-Gaze Input Method for Drag and Menu Selection" 2018:
This study proposed a method-automatic lock of cursor movement (ALCM)-that locks a cursor at the center of a target at the instant the cursor enters the target. The method is intended to suppress irritating subtle cursor movements that

occur when an eye-gaze input system transforms involuntary eye movement (e.g., drift) into cursor coordinates. The effectiveness of the proposed ALCM was verified using pointing performance (speed and accuracy) in two types of HCI tasks. In a drag task, we compared mouse input versus eye-gaze input with use of a backspace (BS) key or voice input.

16. Sourav S. Bhowmick et. al: "VISUAL: Simulation of Visual Subgraph Query Formulation to Enable Automated Performance Benchmarking" 2017 :
Our experimental study demonstrates the effectiveness of VISUAL in accurately simulating visual subgraph queries.

17. Lennart E. Nacke et. al: "Games User Research and Gamification in Human-Computer Interaction" 2017:
Video games have become the focus of attention in the field of human-computer interaction (HCI), a focus that looks beyond the study of video games as mere testbeds for interaction studies, or investigations of a game's user interface. For example, some games have moved to free-to-play business models, where a small number of players pay for premium game content. In these games, user behavior is predicted through the collection of telemetry data, which is also used on mobile phones to provide information about a user's location. This data is then analyzed with machine learning techniques to create personalized experiences.

18. Jinxian Qi et. al: "Intelligent Human-Computer Interaction Based on Surface EMG Gesture Recognition" 2017:
In this paper, linear discriminant analysis (LDA) and extreme learning machine (ELM) are implemented in hand gesture recognition system, which is able to reduce the redundant information in sEMG signals and improve recognition efficiency and accuracy. The characteristic map slope (CMS) is extracted by using the feature re-extraction method because CMS can strengthen the relationship of features cross time domain and enhance the feasibility of cross-time identification. This study is focusing on optimizing the time differences in sEMG pattern recognition, the experimental results are beneficial to reducing the time differences in gesture recognition based on sEMG.

19. Elisabeth Adelia Widjojo et. al: "Virtual Reality-Based Human-Data Interaction" 2017:
In this position paper, we share our views on how VR-based HDI can support exploration of multidimensional large data sets with the aim at providing direction for open research areas that may serve the design of a VR-based HDI system in this emerging field of research.

20. Soumya Priyadarsini Panda et. al: "Automated speech recognition system in advancement of human-computer interaction" 2017:
This paper provides an overview of the automated speech recognition system along with a wide set of possible applications of the technology in advancement of human-computer interactive systems.

21. Nuri Murat Arar et. al: "A Regression-Based User Calibration Framework for Real-Time Gaze Estimation" 2016 :
We propose a novel weighted least squares regression-based user calibration method together with a real-time cross-ratio based gaze estimation framework. The proposed system enables to obtain high estimation accuracy with minimum user effort, which leads to user-friendly HCI applications.

22. Kwang-Ryeol Lee et. al: "Real-Time "Eye-Writing" Recognition Using Electrooculogram" 2016:
In this study, we developed a real-time electrooculogram (EOG)-based eye-writing recognition system, with which users can write predefined symbolic patterns with their volitional eye movements.

23. Daniela Dauria et. al: "Human-Computer Interaction in Healthcare: How to Support Patients during Their Wrist Rehabilitation" 2016:
The increasing use of IT/Informatics within the healthcare context is more and more helpful for both medical doctors and patients in all the surgical specialities. In this paper, we propose a low-cost system exploiting a haptic interface aided by a glove sensorized on the wrist orientation for supporting patients during their wrist rehabilitation allowing the identification of the wrist

24. Regina Jucks et. al: ""I Need to Be Explicit: You're Wrong": Impact of Face Threats on Social Evaluations in Online Instructional Communication" 2016:
This study analyzed the evaluation of face-threatening acts with a 1×3 design. An online forum thread confronted a layperson with an expert who either (a) addressed the layperson's misconceptions directly and frankly, (b) mitigated face threats through explicit hints about the need to be direct or (c) communicated politely and indirectly.

V. ANALYSIS MODEL

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. In agile model we can develop a module and visit it again with the changes required in the future. In this system we developed the image segmentation module first which had to be updated with the addition of further modules.

Steps In Agile Methodology:

1. DEFINE :

a) Problem statement is defined as implementation of a system which can capture and detect gestures from a captured frame and then performing predefined tasks that is mapped with the detected gesture.

b) First module of image segmentation consists of capturing the frame, then reducing the noise while highlighting the features of each type of gesture that the system focuses on.

c) After image segmentation a dataset with at least 1000 images per class is needed for the training model.

d) This dataset is then used to train the model. Hidden layers are needed to be properly created so as to maximize the accuracy while minimizing the loss value.

e) This model is then integrated with the GUI to complete the system that the user can use to interact with using simple hand gestures.

2. DESIGN :

a) For the first module, we need to separate the ROI from the complete frame. The ROI should be cleaned and noise reduction techniques are needed for that.

b) A simple process to capture thousands of images of gestures with proper naming is then required. The process needs functionalities that will allow the developer to choose when the images should be captured.

c) Then CNN model creation requires the function to read the gesture images from the dataset created. It also needs the interface to access the device’s hardware for faster computing. The final module needs the access to device’s web camera and some functions that can make the system usage for user easier.

3. BUILD :

a) We built the first module of image segmentation using MOG2 algorithm for background subtraction. Then we used techniques provided by OpenCV to reduce the gaussian noise in the frames. Then to highlight the features in the gesture image we converted the segmented ROI into binary color format.

b) Then we use the OpenCV methods to save those images labeled as numbers starting from 0. These images are grouped in a folder labeled with the name of that gesture.

c) We used tensorflow to create a CNN model. We had three hidden layers of nodes and then trained the model for 5 epochs having batch size of 32 images. The train-test split ratio was 70-30%.

d) For the GUI we used python’s tkinter library to bind the system with GUI.

VI. PROJECT IMPLEMENTATION AND OVERVIEW OF MODULES

1. Image Segmentation

1.1 Background Subtraction

Background subtraction (BS) is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cameras. It is being used for object segmentation, security enhancement, pedestrian tracking, counting the number of visitors, number of vehicles in traffic etc. It is able to learn and identify the foreground mask. As the name suggests, it is able to subtract or eliminate the background portion in an image. Its output is a binary segmented image which essentially gives information about the non-stationary objects in the image. There lies a problem in this concept of finding non-stationary portion, as the shadow of the moving object can be moving and sometimes being classified in the foreground. Background modeling consists of two main steps: Background Initialization and Background Update. In the first step, an initial model of the background is computed, while in the second step that model is updated in order to adapt to possible changes in the scene.

1.2 Image Enhancement

OpenCV offers various methods to erode or dilate the object. Using these methods we can make the object of interest more clear and reduce the noises. We can also use some Morphological methods to perform this task. This basically means that we want a new version of the image that is more suitable than the original one. For instance, when you scan a document, the output image might have a lower quality than the original input image. We thus need a way to improve the quality of output images so they can be visually more expressive for the viewer, and this is where image enhancement comes into play. When we enhance an image, what we are doing is sharpening the image features such as its contrast and edges.It is important to note that image enhancement does not increase the information content of the image, but rather increases the dynamic range of the chosen features, eventually increasing the image’s quality. So here we actually don’t know what the output image would look like, but we should be able to tell (subjectively) whether there were any improvements or not, like observing more details in the output image, for instance.Image enhancement is usually used as a pre-processing step in the fundamental steps involved in digital image processing (i.e. segmentation, representation).

1.3 Applying Gaussian Blur

Image blurring is achieved by convolving the image with a low-pass filter kernel. It is useful for removing noise. It actually removes high frequency content (e.g: noise, edges) from the image resulting in edges being blurred when this is filter is applied.

1.4 Thresholding Image to Binary

Here, the matter is straight forward. For every pixel, the same threshold value is applied. If the pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value. The function cv.threshold is used to apply the thresholding. The first argument is the source image, which should be a grayscale image. The second argument is the threshold value which is used to classify the pixel values. The third argument is the maximum value which is assigned to pixel values exceeding the threshold. OpenCV provides different types of thresholding which is given by the fourth parameter of the function.

1.5 Hand Detection using Contour Recognition

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. Then Convex Hull is used draw the boundaries around the recognized hand. Convexity of a polygon is measurable trait that is amenable to the analysis of its shape. Shape is crucial in many areas of scientific analysis such as, object classification and identification

2. Dataset Creation

For dataset creation we used the image segmentation module created in the first step. After starting up the system the developer performs some gesture after selecting a proper stable background. Then the system is informed that the developer wants to save the segmented images. Then the system starts capturing the images, then segmenting them as the steps defined in the first module. When the image is segmented properly it is saved to the path defined and the images are labeled according to the number of the image starting from 0. When 1000 images are saved the system informs the developer and the developer can choose to stop the process or increase the number of images in the dataset.

Using this process the dataset is created with 10 classes for 10 different gestures. Each gesture has at least 1000 images. This dataset is then used to train the CNN model created in the third module.

3. Model Creation and Training

3.1 Model Creation

For efficient model training we used TensorFlow’s GPU variant libraries that use the device’s GPU for computations. The TensorFlow version 2.1.0 was used. For the required libraries we need to make sure that the device has GPU drivers installed with the version having cudart64.dll file for interfacing the software with the hardware. Then we created the model with an input layer, 1 output layer and 3 hidden layers. For activation we used Rectified Linear Unit (ReLU) in combination with Softmax function. The the model was compiled using ‘adam’ optimizer and ‘sparse categorical cross entropy’ as loss handling. The metrics used to evaluate the model was set to accuracy.

3.2. Model Training :

The model created in step 1 was trained using the dataset created in module. All of the images were combined together along with the gesture label to which that image belonged to in a list. Then that list was arbitrarily shuffled so as to avoid ambiguous training results. The dataset was split into training and testing parts with the ratio of 70% to 30% respectively. Then the model was trained for 5 epochs (rotations) with a batch size of 32 images per step. The training was complete with accuracy value of 1.00 (100%) with a negligible loss value. The model was then saved.

4. Gesture Prediction

The created model is loaded and used for predicting the output label (gesture name) for the input gesture image. If the gesture image is deemed valid it is captured and saved to a predefined location. Then the model is fed with that image. After its processing the model returns a list of all the possible gesture labels along with the values of confidence that the model has about that gesture belonging to that gesture class. If any of the class label has confidence value over 80% then that image is said to be belonging to that gesture as any lower value could also be a result for false positive image that happened due to unstable background or any disturbances in the captured frame.

When the gesture is properly predicted it gives a call to the mapped function that carries out the task with the computer that was mapped to the gesture.

5. GUI Creation and System Integration

GUI is necessary for simplicity of use for the user. The GUI makes it possible to use the system using simple button presses. The GUI is built using the tkinter library of python.

VII. TOOLS AND TECHNOLOGIES USED

1. OpenCV

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license. OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself. A CUDA-based GPU interface has been also provided since September 2010.

2. TensorFlow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and it is also used for machine learning applications like neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015. TensorFlow can run on multiple CPUs & GPUs. It is available on 64- bit Linux, MacOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allow the easy deployment of computation across a variety of platforms, and from desktops to clusters of servers to mobile and edge devices. It uses data flow graphs to build models and it allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation.

3. Keras

Keras is an API designed for human beings, not machines. It follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides. It is the most used deep learning framework among top-5 winning teams on Kaggle. Because Keras makes it easier to run new experiments, it empowers to try more ideas than your competition, faster.

VIII. SYSTEM REQUIREMENTS

Software Dependencies:

- Python version 3.7.3 to execute the application
- OpenCV version 4.1.0 to deal with web camera and image processing
- TensorFlow version 2.1.0 to load and use the model
- cudart64_101.dll to use the TensorFlow with GPU for faster processing

Hardware Dependencies:

- Web camera
- GPU (minimum Nvidia GTX 1050)

IX. ALGORITHM DETAILS

1. Image Segmentation Algorithm

For image segmentation we used methods provided by OpenCV. The following processes were used :

1. If the camera is started enter an infinite loop ,while capturing the frame per each iteration.
2. Remove the black edges from the frame(for display) by selecting the biggest contour .
3. Apply bilateral filter for applying the smoothing effect on the captured frame.
4. If the user has captured a background then the further part is executed or first three steps are repeated until user captures the background.
   a) The captured frame is used with the background model created with the MOG2 algorithm to create a foreground mask.
   b) Then that mask is used to remove the background elements from the frame by performing ‘bitwise and’ operation with the frame.
   c) The captured frame is converted from BGR model to GRAYSCALE model.
   d) The frame is blurred using GaussianBlur() method provided by the OpenCV library to reduce the Gaussian noise present with the frame.
   e) Then the cleaned frame is converted to binary image format which now has ROI in white pixels on the black

pixels of the masked background.

Thus the resulting binary image is the segmented image that is required.

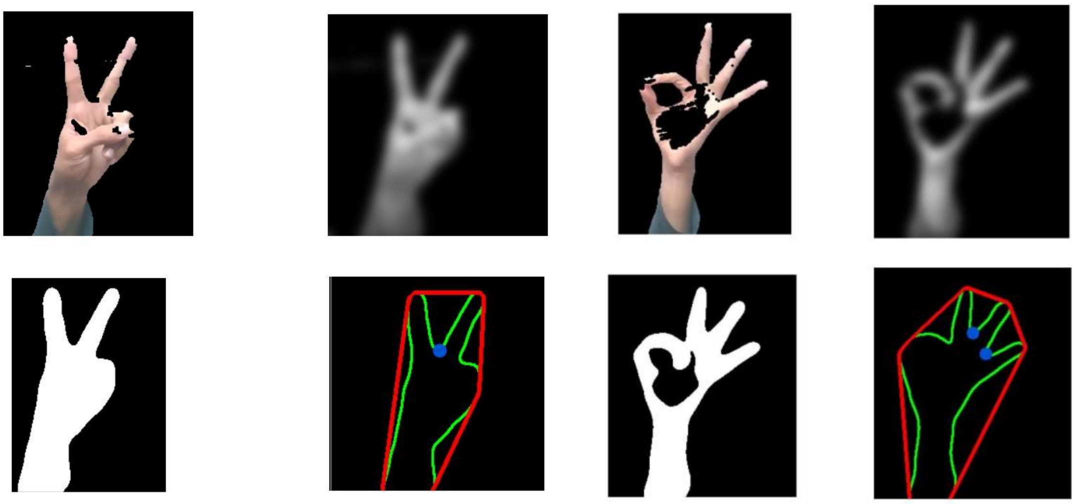
2. Continuous Gesture Prediction Algorithm

For gesture prediction we have used the CNN model that is created using TensorFlow library trained with the dataset created using the images captured by using the first algorithm. The gesture prediction process is as follows :

1. First the image is segmented. Then it is checked if the image lies inside the defined bounding rectangle. If it does the process goes forward or else this step is repeated until the condition becomes true.
2. The image is re-sized to size 50 pixels wide and 50 pixels high for decreasing the computation that is needed to predict the class label related to that gesture.
3. This re-sized image is then saved to a predefined location on the device with a predefined name.
4. The model is the invoked which reads the saved image that is saved in the previous step.
5. The saved image is a binary image thus it only has 2 channels, but the CNN model needs input image with 3 channels so the dimensions of the image loaded are expanded to match the requirement.
6. The trained model is then passed that image and the model returns a list of all possible labels along with the confidence value scored by that label.
7. The label with highest confidence value is then selected for further process.
8. If the label has more confidence value than 80% (to decrease the frequency of false positives), the label is selected else it is discarded.
9. If the same label persists for 3 times (to decrease the frequency of false positives), the gesture is selected as the predicted gesture.
10. The predicted gesture is then is used to perform the function that is mapped with that gesture.

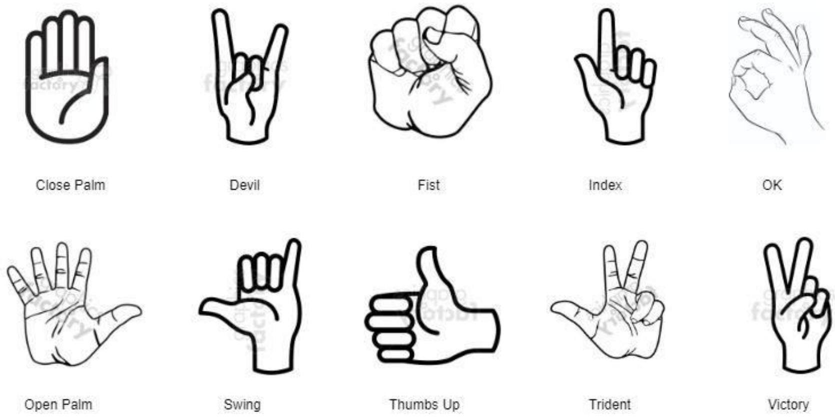
X. RESULTS

1. Phases of segmentation.

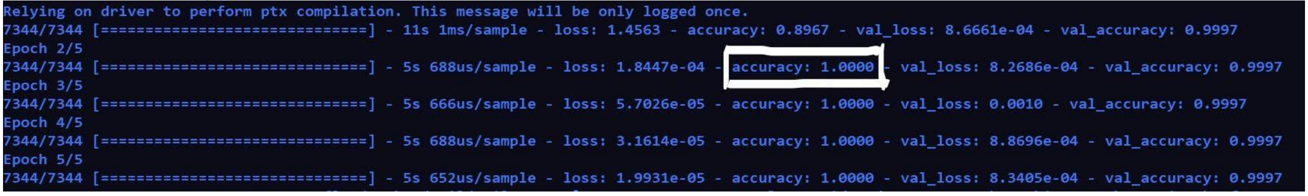


(i) Open Gesture (ii) OK Gesture

2. Supported Gestures



3. Training Accuracy



```
Relying on driver to perform ptx compilation. This message will be only logged once.
7344/7344 [==============================] - 11s 1ms/sample - loss: 1.4563 - accuracy: 0.8967 - val_loss: 8.6661e-04 - val_accuracy: 0.9997
Epoch 2/5
7344/7344 [==============================] - 5s 688us/sample - loss: 1.8447e-04 - accuracy: 1.0000 - val_loss: 8.2686e-04 - val_accuracy: 0.9997
Epoch 3/5
7344/7344 [==============================] - 5s 666us/sample - loss: 5.7026e-05 - accuracy: 1.0000 - val_loss: 0.0010 - val_accuracy: 0.9997
Epoch 4/5
7344/7344 [==============================] - 5s 688us/sample - loss: 3.1614e-05 - accuracy: 1.0000 - val_loss: 8.8696e-04 - val_accuracy: 0.9997
Epoch 5/5
7344/7344 [==============================] - 5s 652us/sample - loss: 1.9931e-05 - accuracy: 1.0000 - val_loss: 8.3405e-04 - val_accuracy: 0.9997
```

The CNN Model for gestures prediction was created using TensorFlow framework and trained with a dataset of 10 gestures each with 1000+ sample images (Total 10000+ samples). The training accuracy achieved was 100.

XI. APPLICATIONS AND LIMITATIONS

Applications

1. This system can be used with computers for controlling applications that mostly work in background (for example music player, operating system functions, etc) while using some other application.
2. This system can be used with a smart TV in households instead of touching the remote or the TV itself.
3. A surgeon can use this system while operating on a patient to look at the X-rays or some other stuff on a monitor without contaminating his hands or equipment.
4. A cook can use this system to look at digital format of a recipe while cooking without touching anything.

Limitations

1. Web camera is a requirement which is not built in with desktop computers.
2. Python 3 and OpenCV library must be installed along with the TensorFlow library with GPU variant.
3. A decent GPU is needed for the system to run smoothly.
4. A stationary background is very important for the system to correctly segment the input hand gestures or the rate of false positives increase drastically.

XII. FUTURE WORK

1. A module can be added which allows the user to set his own gesture along with the already set gestures, which will in turn modify the trained model for increasing the functionalities provided by the system.
2. A module can be added which allows the user to modify the mapped functions as the user wants.
3. We can host the trained model over the internet as a server-client architecture so that the hardware requirements can be countered.
4. We can add functionalities that allows the user to control his smart phone using this system.
5. A module can be added which allows the user to perform hand movement gestures instead of just still gestures that are currently supported.

XIII. CONCLUSION

The proposed method for continuous gesture recognition method based on different algorithms of image processing, which is then assigned several functionalities. Because of the naturalness of the features and the linearity of combination, the derived gesture models have good comprehensibility. Since the proposed method can both recognize the gesture and perform the assigned functions,the system recognizes the image and captures the frame, removes the background using MOG2 algorithm. The system finds the convex hull of the image to identify the convexity defect of the image. Thus proper segmentation is carried out which is then used to create the dataset and to capture the frame with ROI. The CNN model trained with the dataset resulted in 100% accuracy and negligible loss value. The predictions can be derived from the model which takes nearly one second (0.96 second to be exact) to give the result.

REFERENCES

[1] S.Zhou, F. Fei, G. Zhang, J. D. Mai, Y. Liu, J. Y. Liou, and W. J. Li, “2d human gesture tracking and recognition by the fusion of MEMS inertial and vision sensors,” IEEE Sensors J., vol. 14, no. 4, pp. 1160–1170, 2014.

[2] C. Breazeal, “Social interactions in hri: the robot view,” IEEE Trans. Syst., Man, Cybern. C, vol. 34, no. 2, pp. 181–186, 2004.

[3] A. Akl, C. Feng, and S. Valaee, “A novel accelerometer-based gesture recognition system,” IEEE Trans. Signal Process., vol. 59, no. 12, pp. 6197–6205, 2011.

[4] T. Nakamura and Y. Morishita, “Acceptability evaluation in prototype of menu search system by feelings (in Japanese),” IEICE technical report. Data engineering, vol. 112, no. 75, pp. 73–77, 2012.

[5] T. Fan, C. Ma, Z. Gu, Q. Lv, J. Chen, D. Ye, J. Huangfu, Y. Sun, C. Li, and L. Ran, “Wireless hand gesture recognition based on continuous wave doppler radar sensors,”IEEETMICROWTHEORY,vol.64,no.11, pp. 4012–4020, 2016.

[6] Zhang, Z. Tian, and M. Zhou, “Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor,” IEEE Sensors J, vol. 18, no. 8, pp. 3278–3289, 2018.

[7] Lecun. Y. L, Bottou. L, Bengio. Y, and Haffner. P, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol.86, no. 11, pp. 2278–2324, Dec. 1998.

[8] Krizhevsky. A, Sutskever. I, and Hinton, G, “ImageNet Classification with Deep Convolutional Neural Networks,” presented at the NIPS, Lake Tahoe, Nevada, Dec. 03–06, 2012.

[9] Zeiler. M. D, and Fergus. R, Visualizing and understanding convolutional networks. Berlin, Germany: Springer, 2014, pp. 818–833.

[10] Karen. S, and Andrew. Z, “Very Deep Convolutional Networks for Large- Scale Image Recognition,” presented at the ICLR, San Diego, CA, USA, May. 07–09, 2015.

[11] Szegedy. C, Liu. W, Jia. Y, Sermanet. P, Reed. S, Anguelov. D, Erhan. D, Vanhoucke.V, and Rabinovich.A, “Going deeper with convolutions,” presented at the CVPR, Boston, MA, USA, Jun. 07–12, 2015.

[12] X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition," in CVPR, Las Vegas, NV, United States, 2016, pp. 770–778.

[13] T. Fan, C. Ma, Z. Gu, Q. Lv, J. Chen, D. Ye, J. Huangfu, Y. Sun, C. Li, and L. Ran, “Wireless hand gesture recognition based on continuous wave doppler radar sensors,”IEEETMICROWTHEORY,vol.64,no.11, pp. 4012–4020, 2016.

[14] Z. Zhang, Z. Tian, and M. Zhou, “Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor,” IEEE Sensors J, vol. 18, no. 8, pp. 3278–3289, 2018.

[15] A. Just and S. Marcel, “A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition,” COMPUT VIS IMAGE UND, vol. 113, no. 4, pp. 532–543, 2009.

[16] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, “A unified framework for gesture recognition and spatio-temporal gesture segmentation,” IEEE T PATTERN ANAL, vol. 31, no. 9, pp. 1685–1699, 2009.

[17] S. Poularakis and I. Katsavounidis, “Low-complexity hand gesture recognition system for continuous streams of digits and letters,” IEEE T CYBERNETICS, vol. 46, no. 9, pp. 2094–2108, 2016.

[18] S. Iengo, S. Rossi, M. Staffa, and A. Finzi, “Continuous gesture recognition for flexible human-robot interaction,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014, pp. 4863–4868.

[19] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE COMMUN SURV TUT, vol. 15, no. 3, pp. 1192–1209, 2013.

[20] H.-K. Lee and J. H. Kim, “An hmm-based threshold model approach for gesture recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 10, pp. 961–973, 1999.

[21] H.-I. Suk, B.-K. Sin, and S.-W. Lee, “Hand gesture recognition based on dynamic bayesian network framework,” PATTERN RECOGN, vol. 43, no. 9, pp. 3059–3072, 2010.

[22] M. Elmezain, A. Al-Hamadi, and B. Michaelis, “Hand trajectory-based gesture spotting and recognition using hmm,” in Proceedings of the 16th IEEE international conference on Image processing. Cairo, Egypt: IEEE Press, 2009, pp. 3541–3544.

[23] S. Mitra and T. Acharya, “Gesture recognition: A survey,” IEEE Trans. Syst., Man, Cybern. C, vol. 37, no. 3, pp. 311–324, 2007.

[24] R. Srivastava and P. Sinha, “Hand movements and gestures characterization using quaternion dynamic time warping technique,” IEEE SensorsJ., vol. 16, no. 5, pp. 1333–1341, Oct. 2015.

[25] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, “Fusion of inertial and depth sensor data for robust hand gesture recognition,” IEEE Sensors J., vol. 14, no. 6, pp. 1898–1903, 2014.

[26] M. H. Ko, G. West, S. Venkatesh, and M. Kumar, “Using dynamic time warping for online temporal fusion in multisensor systems,” INFORM FUSION, vol. 9, no. 3, pp. 370–388, 2008.

[27] B. Hartmann and N. Link, “Gesture recognition with inertial sensors and optimized dtw prototypes,” in IEEE International Conference on Systems, Man and Cybernetics, 2010, pp. 2102–2109.