Hand Gesture Detection and Prediction System aided by Neural Networks

P.Suganya[1], R.Aadith Narayan[2] and L.Shivani[3]

[1]Assistant Professor, Department of Computer Science and Engineering
SRM University, Ramapuram, Chennai, India
[2,3] Department of Computer Science and Engineering,
SRM University, Ramapuram, Chennai, India

---***---

Abstract: *Hand gesture recognition is a futuristic domain in the area of human-computer interaction. Gesture Recognition is a challenging domain and the foundation requires proper working mechanisms that will not only be accurate and fast but which will also open a window for future enhancement. The vision of the proposed system is to recognize static and a few dynamic gestures with accuracy and precision. The proposed system consists of two dimensions. The first dimension involves detection of the hand under testing by implementing background subtraction of the hand using skin pixel rules. The second dimension involves detecting the gesture which is signaled by the hand using a sophisticated machine learning algorithm which can learn as and when the software is put into test. The proposed system eradicates any form of glitches or inaccuracy with respect to the extent of prediction due to the use of machine learning. The detection algorithm is sufficient for hand filtering and does not require the use of any special hardware devices.*

Keywords: RGB,HSB,NumPy,Keras,Theanos.

1. Introduction

Gesture recognition is a domain in computer science and language technology with the goal of interpreting human gestures via image processing algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field include emotion recognition from face and hand gesture recognition. Users can use simple gestures to control or interact with devices without physically touching them. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse. Hence, with the help of a single set of algorithms, it is possible to for the machine to understand and work with a wide-range of users whose behavior is different, under a wide range of environmental conditions. The focal point of this proposed system is to make the system more accurate, faster and stable, applicable under different environmental conditions, understand and learn the wide behavioral pattern of people through machine learning.

1.1 Existing System

A Method of Skin Color Identification Based on Color Classification: [3] The method classifies colors of all pixels in the image into several classes through K-means algorithm. It then segments the image into several parts according to the color class that each pixel belongs. Each class of color is represented by a color feature vector. The class whose feature vector has the minimum distance to the skin color feature vector, previously defined in the color space is taken as human skin color. [3]

Convolutional Networks and Applications in Vision: [1] Convolutional Networks (ConvNets) are a biologically inspired trainable architecture that can learn invariant features. Each stage in a ConvNets is composed of a filter bank, some nonlinearity, and feature pooling layers. With multiple stages, ConvNets can learn multi-level hierarchies of features. While ConvNets have been successfully deployed in many commercial applications from OCR to video surveillance, they require large amounts of labelled training samples. We describe new unsupervised learning algorithms, and new non-linear stages that allow ConvNets to be trained with very few labelled samples. [1]

1.2 Issues in Existing System

Slow and Inaccurate due to lack of Machine Learning: Most of the gesture recognizing devices are often slow and inaccurate. The lack of machine learning algorithms make these devices stick to pre-built gesture recognition which when used under different environment with different orientation leads to slow, inaccurate results.

Limitations on Equipment: Images or video may not be under consistent lighting, or in the same location. Items in the background or distinct features of the users may make recognition more difficult. The variety of implementations for image-based gesture recognition may also cause issue for viability of the technology to general usage. For example, an algorithm calibrated for one camera may not work for a different camera. Furthermore, the distance from the camera, and the camera's resolution and quality, also cause variations in recognition accuracy.

Image Noise: The amount of background noise also causes tracking and recognition difficulties, especially when occlusions (partial and full) occur.

More Sensors and their calibration: In order to capture human gestures by visual sensors, robust computer vision methods are also required, for example for hand tracking and hand posture recognition for capturing movements of the head, facial expressions or gaze direction.

Gorilla Arm Effect: "Gorilla arm" was a side-effect of vertically oriented touch-screen or light-pen use. In periods of prolonged use, users' arms began to feel fatigue and/or discomfort. This effect contributed to the decline of touchscreen input despite initial popularity.

Requirement of more Training Data Sets: The Convolutional Networks used in existing system analyses the whole visual image pixel by pixel, which requires more data and resources to train, this can be reduced by restricting the pixels to be scanned.

2. METHODOLOGY

The proposed system consists of two modules which are the detection and prediction module. The detection module is implemented using Java while the prediction module is achieved through neural networks in Python. The detection module is responsible for detecting the hand with which gestures are shown. The initial step is Background subtraction which is to remove the background such that the hand is focused. The next step is to convert the video feed into a HSB video feed. The video feed is pre-processed such that the areas with skin pixels are converted to white and the other color gradients are converted to black. Now, the hand is represented as white pixels thus detecting it. The module is also capable of saving data set images for the neural networks to analyze for gesture detection. In the prediction module, the convolution neural network is defined as follows. The neural network is written with the Theano library. A NumPy array is used to collect the dataset images. The input feed image is gained from Java. The input image is fed into the neural network and is analyzed with respect to the dataset images. The result is forwarded to the Java module where the resultant values are represented in the java output window in the form of a bar graph.

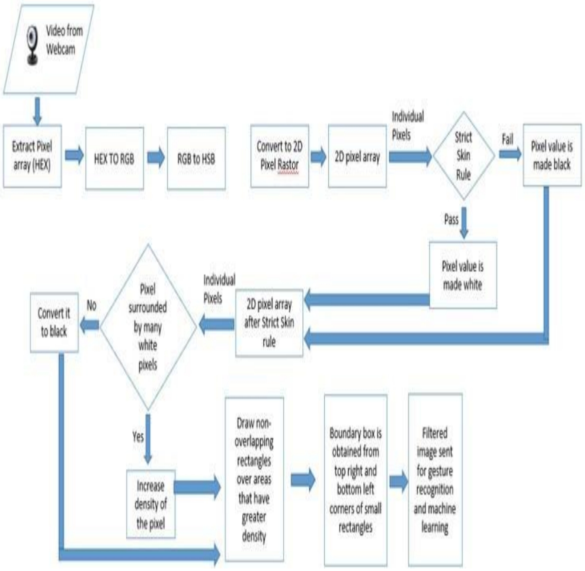

Video from Webcam
Extract Pixel array (HEX)
HEX TO RGB
RGB to HSB
Convert to 2D Pixel Rastor
2D pixel array
Individual Pixels
Strict Skin Rule
Fail
Pixel value is made black
Pass
Pixel value is made white
2D pixel array after Strict Skin rule
Individual Pixels
Pixel surrounded by many white pixels
No
Convert it to black
Yes
Increase density of the pixel
Draw non-overlapping rectangles over areas that have greater density
Boundary box is obtained from top right and bottom left corners of small rectangles
Filtered image sent for gesture recognition and machine learning

Fig -1. System architecture of Java (Filtering) environment

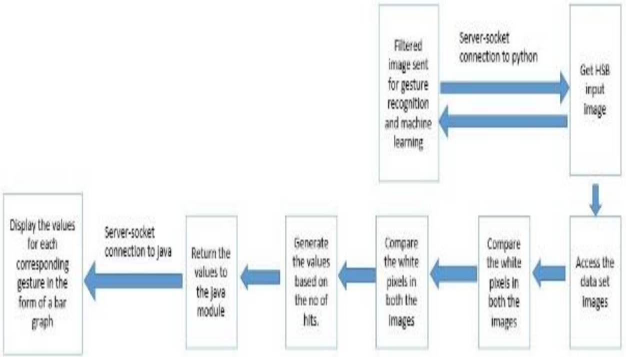

Filtered image sent for gesture recognition and machine learning
Server-socket connection to python
Get HSB input image
Access the data set images
Compare the white pixels in both the images
Compare the white pixels in both the images
Generate the values based on the no of hits.
Return the values to the java module
Server-socket connection to java
Display the values for each corresponding gesture in the form of a bar graph

Fig -2. System architecture of Python (Machine Learning) environment

2.1 Getting live video feed

The default webcam is first accessed using the Webcam library developed by Sarxos. The dimensions of the webcam are then derived. This is to narrow down the focus of the image buffer and pixel raster. The dimensions of the java output window are also initialized with respect to the width and the height derived by multiplying the dimensions with a factor which is optimal with the screen resolution of the PC. The image buffer and pixel raster are initialized with the derived width and height as arguments. The Buffered image or image buffer is used to capture screenshots of the live video feed which is under consideration. The pixel raster contains the gradient value of each pixel in the video feed. The title of the java output window is set and the Java frame window is tweaked and formatted.

2.2 Filtering Process

In order to filter out the hand from the background, Background subtraction has to be performed. The first task is to convert the hex color gradient to RGB color format. The reason for this is to intensify and decode the color gradients in pure Red, Green and Blue format. Then, the RGB color format is converted into HSB format for a clearer distinction of all the colors. The values are stored in a pixel raster which is converted from 1D to 2D using length and breadth values as arguments. The strict skin pixel and loose skin pixel rules are then defined. Based on the HSB color pixels. These rules are used to check if each pixel contains a skin color or not. Thus, when this rule is applied to the video, the hand is analyzed and filtered out as a result. The rule works as follows. If the pixel passes, the skin pixel rule, it is made white and if a pixel fails the rule, it is made black. Any pixel within a certain distance from a white pixel is incremented by one. Overall goal is to fill locations that have been missed by the strict skin rule.

2.3 Region of Interest

The hand thus detected is then overlaid with rectangle boxes which tracks the hand as it moves around in the live video stream. If the density value over a region is ¿=60, a small rectangle is drawn in green color. Any two rectangles must not overlap. In order to focus on just the hand for processing, a hand bound is created. In areas where the rectangle clustering is larger, the intensity of the clusters is increased in order to reflect properly as white pixels. The hand bound is created with respect to the top and bottom rectangles which are present. Thus, the hand is tracked by the green rectangles wherever it moves within the camera frame.

2.4 Creating Dataset Images

In order for the neural network to predict a particular gesture, there should be a reference component with which the algorithm can compare the gesture under test to predict that particular gesture. Therefore, a set of 500 images of the gesture being shown is captured using Buffered image. These images are saved in a real-time path using a writer. The server socket between the python and java platforms is not initialized during the data collection mode. The images are stored in PNG format in the real-time path. Along with the images, there is a raw data txt file which is created which represents the gesture which is taken. That is, the gesture which is considered is set as 1 and all the other gestures are set to 0. The same is repeated for all the 500 images. Thus, when the neural network identifies the gesture with the most hits, it refers to the raw data file for the particular gesture which is then reflected as the result.

2.5 Neural Networks backend

The type of neural network used here is a convolution neural network which is a specifically used algorithm for working with images. The libraries used for the neural networks is Keras with a Theano backend. In order for Theano to be compatible with the dataset images, the images are loaded into a NumPy array which holds all the images. The CNN makes use of the raw data file in order to identify the gestures under consideration. The Neural networks compares the dataset images to the live video feed where each frame of the feed is considered. The gesture which is shown as HSB format is compared to the dataset images in the database. The no of white pixels in both the images are compared. The dataset of the gesture which has the most no of hits is considered as the resultant gesture and hence the prediction is made. The result is reflected on a dynamic bar graph in the java output window which shows the percentage of prediction for each gesture depending on the number of hits it acquired. The connection between the detection module in java and the neural networks module in Python is established using a server socket connection that is socket programming. The connection is localized within the system and is done by handling requests from the server side to the socket side.

2.5.1 Optimization and accuracy

The neural networks develop waits for each gesture whenever a prediction is made. The weights depend upon the number of hits of white pixels it acquires. Therefore, the predicted gesture will receive more weights than the weights received or generated for other gestures. Initially, the predicted gestures weights will be more than the other gestures but will relatively, the magnitude of the weights will have a lesser standard deviation. The magnitude of the weight for the predicted gesture should have a larger standard deviation to emerge distinct from other gestures which will improve the accuracy and the efficiency of the program. Therefore, a neural network is trained for many iterations. The purpose of training is to reduce the margin of error during each iteration therefore making the system more robust and reliable. In each iteration, the prediction value is taken into account and subtracted with the expected outcome. This will provide the range of error which is reimbursed in the next iteration, therefore reducing the extent of inaccuracy. Thus, the program learns as time progresses. The weights are overwritten during each iteration and is stored in a separate .json file. The values from this file is taken as reference during every iteration.

2.6 Server-Socket Connection

In order for the java and python modules to interact with each other, a connection is established between them through socket programming. There exists a server-socket connection where the detection module (java) is the server and the prediction module (Python) is the socket. When the program is executed, a request is sent from the server side to the socket side. The request is accepted by running the python module which accepts the request from the server side which results in a connection being established. During Data collection mode, the server connection is not established because the need for the prediction module is not needed. The host is set as localhost as the program is run locally in one device. The server and the socket is set using the same port no. thus channeling the request properly.

3. RESULTS

There was a need of a proposed system which needed to be robust and reliable without any specialized hardware for hand gesture recognition. THE PROPOSED SYSTEM has satisfied the need for a robust and reliable system for hand gesture recognition and has made use of no specialized hardware for its execution. The usage of neural networks has added value and complexity to the proposed system in terms of optimization and efficiency. The neural networks were also responsible for making the program smarter by learning with each execution thus improving the detection accuracy as time progresses.

3.1 Detecting region of interest

A window that shows how the system detects objects which are of human skin color from user environment. The red boundary box is the area of useful data, commonly known as region of interest. The red boundary box is drawn with the help of the smaller green boxes which are formed in places where there is high density of human skin color.

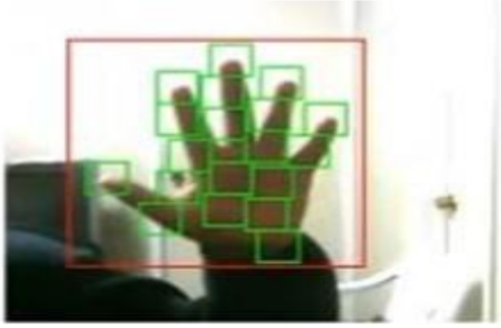


Fig -3. Detection of the hand

3.2 An example of a training image

A training image which is in HSB form. The pixels in the white area are those pixels that have passed the Strict Skin Rule. The pixels in the black area are those pixels that failed the Strict Skin rule.

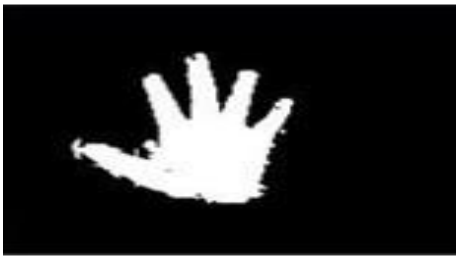
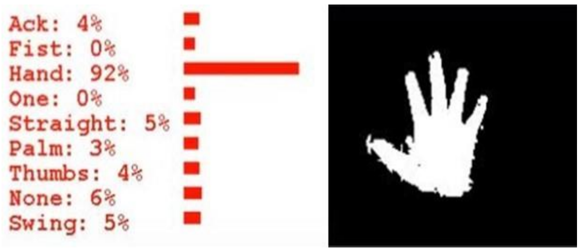


Fig -4. A dataset image of the hand in HSB form

3.3 Percentage of recognition

A bar graph shows how the machine detects and recognizes which gesture is shown by the user. If percentage of the hand as shown above is more, the machine understands that it's a hand and hence, does the corresponding work.



```
Ack: 4%
Fist: 0%
Hand: 92%
One: 0%
Straight: 5%
Palm: 3%
Thumbs: 4%
None: 6%
Swing: 5%
```

Fig -5. The extent of prediction of gestures reflected in the Java output window

4. DISCUSSION AND ANALYSIS

4.1 System Requirements

Hardware requirements: This system requires a computer with minimum 4GB RAM, an i3 Processor, a built-in web camera or external web-camera.

Software requirements: This system requires a computer loaded with Java JRE V8, Anaconda V4.4.0 for Python V3.5.3, Keras library V1.0.6 in Python for machine learning, Theano V0.9.0 which is the backend for Keras.

4.2 Overall Description

4.2.1 System Features

One of the prime features of this proposed system is that it is highly accurate in finding the Region of Interest which is the human hand. It also has the capability to learn and understand the user and with time becomes faster, yet accurate. The proposed system can also be implemented in Augmented Environment and provide user a platform to interact with the system.

4.2.2 User classes and characteristics

There are two classes categorized based on privilege levels. The User class can only provide input to the machine, which is basically gestures. The Machine class is capable of taking in the input, filtering, analyzing and storing in its memory. It then compares the gestures and provides suitable output and functionalities. Hence, the Machine class can store and access the data.

4.2.3 Operating Environment

This proposed system was developed on Windows platform. However, since the programming languages used are platform independent, with suitable JVM and PVM for different operating systems and with the source code, the proposed system can be run in all Operating systems. Design and implementation constraints: This proposed system requires a suitable environment with right amount of brightness and background colors that do not fall in the human skin color range.

4.3 Application

Controlling the cursor (Dynamic Gesture): THE PROPOSED SYSTEM is used to control the mouse using hand gestures which is considered as a valuable application of the proposed system. The Robot class in Java provides functions which can be used to manipulate the mouse and the keyboard. The listeners are first initialized in the program. As far as movement of the hand is concerned, there are two types. One is free movement and the other is fine movement. The hand movement is done by feeding the coordinates to the go () function of the Robot class which directs the mouse to go to those particular coordinates. One of the gestures is set for free movement and another gesture is set for fine movement. The free movement and fine movements are done by multiplying the coordinates with respective factors which will affects the sensitivity of the cursors movement. Another capability which is inherited using the Robot function is single click and double click of the mouse. Each click consists of a pair of Mousepress () and Mouserelease () functions. For single click, this pair is called once and for double click, this pair is called twice. Thus, the functions of the mouse are completely manipulated using hand gestures.

4.4 System Scope

The proposed system has a wide scope of application in several domains as listed below:

Mobile phones: To capture pictures when a smile, peace symbol is detected, to unlock smart phones.

Television: To change channels through left-to-right and vice-versa gestures.

Gaming: Interact with the environment in a natural way if the gaming supports Augmented/ Virtual Reality.
Societal Purpose: To interpret sign language.

5. CONCLUSION

Hand gesture recognition has been a vital and futuristic domain in the fields of image processing and computer vision. There have been several applications and implementations of gesture recognition using various methodologies. THE PROPOSED SYSTEM in particular has proved to predict hand gestures using an optimized detection and prediction mechanism. It has shown its effectiveness without the use of any specialized hardware for boosting its performance and efficiency. It has proved to be robust and reliable by predicting gestures with utmost accuracy and by learning data with progress. THE PROPOSED SYSTEM has laid the foundation and has provided room for various application possibilities in the future. The usage of RRAMS (Resistive Random Access Memory) can add on as a hardware implementation of neural network which can increase and optimize the processing rate. The presence of a larger database can store more dataset images for each gesture. Hence by training the neural network, we can achieve much greater accuracy and reliability. This is will also allow a greater range of gestures to be added for detection.

References

[1] Convolutional Networks and Applications in Vision - Authored by Yann LeCun, Koray Kavukcuoglu and Clement Farabet, published in Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on 30 May-2 June 2010.

[2] Remote Control of a Robotic Platform Based on Hand Gesture Recognition-Authored by Alexandru Pasarica, Casian Miron, Dragos Arotaritei, Gladiola Andruseac, Hariton Costin, Cristian Rotariu, published in E-Health and Bioengineering Conference (EHB), 2017 on 24 June 2017.

[3] A Method of Skin Color Identification Based on Color Classification Authored by Xiaoying Fang, Wenquan Gu, Chang Huang, published in Computer Science and Network Technology (ICCSNT), 2011 International Conference on 26 Dec. 2011

AUTHORS

P.Suganya is currently an assistant professor in the Department of Computer Science and Engineering at SRM University Ramapuram.

R.Aadith Narayan is currently a student pursuing his B.Tech-Computer Science and Engineering at SRM University Ramapuram.

L.Shivani is currently a student pursuing her B.Tech-Computer Science and Engineering at SRM University Ramapuram.