International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Smart Home Automation using Hand Gesture Recognition System

Vignesh Selvaraj Nadar, Vaishnavi Shubhra Sinha, Sushila Umesh Ratre

Abstract: ***Visual interpretation of hand gestures is a natural method of achieving Human-Computer Interaction (HCI). In this paper, we present an approach to setting up of a smart home where the appliances can be controlled by an implementation of a Hand Gesture Recognition System. More specifically, this recognition system uses Transfer learning, which is a technique of Machine Learning, to successfully distinguish between pre-trained gestures and identify them properly to control the appliances. The gestures are sequentially identified as commands which are used to actuate the appliances. The proof of concept is demonstrated by controlling a set of LEDs that represent the appliances, which are connected to an Arduino Uno Microcontroller, which in turn is connected to the personal computer where the actual gesture recognition is implemented.***

Index Terms: ***Arduino, Gesture Recognition, Home Automation, Human Computer Interaction, Machine Learning.***

I. INTRODUCTION

With the advent of modern, superior tools for processing data and high performance computers, what was once considered impossible to achieve, is now just a few clicks apart. Computer Vision, Object Recognition, Automation are not just fancy terms, but an actual thing of the present. This revolution has been brought in by breakthrough research work done in the field of Machine Learning, Artificial Intelligence and other related fields. What this means is, there are now newer avenues of exciting world problems and use cases that can be tackled through with the help of these wonderful tools.

Human Computer Interaction is that field of study, which focuses on how a human interacts with a computer[1][2]. The goal of this study is to devise such an idealistic method of interaction, wherein, the communication between a human user and a computer completely mimics a human to human communication. It should feel just as natural and just as intuitive. Lots of theories, projects, demos and research work have been provided which helps us further understand this field of work.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Vignesh Nadar*, Student, Department of Computer Science, Amity University Mumbai, India.
Vaishnavi Sinha, Student, Department of Computer Science, Amity University Mumbai, India.
Sushila Ratre, Professor, Department of Computer Science, Amity University Mumbai, India.

Our idea proposed is a cross concept between Human Computer Interaction and Computer Vision, where the computer vision is in turn implemented by the use of Machine Learning and Artificial Intelligence. We have developed a system, that can distinguish and identify multiple gestures, fed through the web-cam of a personal computer, and then automate certain actions, based on the gesture detected.

This concept of ours can be broken down into 3 basic parts: Training the machine learning model via the input provided to the system, Classifying the incoming gestures properly and identifying it, Automating the action sequence that is specific to each valid gesture.

These segments together constitute a Gesture recognition system based on Computer Vision, which we felt is a natural and a very intuitive way of interacting with the computer. We are implementing this system to function as a control unit for automating the appliances present in our home, thus making this prototype, a smart home enabler.

To demonstrate the working of the system, we have used an Arduino Uno Microcontroller[3][4], with a ATmega 328 microprocessor on board, which is connected to the personal computer. The Arduino is also connected to a set of LEDs to represent the appliances present in a house. We are controlling the LEDs via gesture inputs provided to the webcam of the personal computer. This is achieved by defining and training gestures for the commands – "START", "STOP", and some gestures to identify the appliances. So, to glow the LED numbered #1, the following gesture sequence has to be provided - "START" "LED1".

This demonstration can be further scaled to include many other commands and accommodate multiple appliances. We selected these basic commands as we felt that these are the most widely used, and that it sufficiently demonstrates the working of our concept. The following is a detailed explanation of the work done.

II. LITERATURE SURVEY

A1. Maria Eugenia Cabrera, Juan Manuel Bogado, Leonardo Fermin, Raul Acuna and Dimitar Ralev[5] proposed a system wherein they perfected a Glove based Hand Gesture Recognition System. In this paper, they have presented a unique Glove, which has accelerometer sensors attached on it. These sensors calculate the degree of flexions on each finger, along with the orientation of the hand along the three axes. They then use the data generated by the sensors and feed it into neural networks to identify the gestures.

A2. Christopher Lee and Yangsheng Xu[6] developed another glove based gesture recognition system. The glove has embedded sensors on it that records the orientation data.

This provides them with enough information to develop a gesture recognition system that uses neural networks to classify up to 14 letters from the Hand motioned alphabets, after providing only one or two examples of each.
The following points were noted in following research papers:
1.Usage of this mechanical apparel, that is the Glove, is a very tedious task and does not feel natural at all.
2.The mechanical Glove can recognize only up to a fixed total of varying gestures.
A3. Etsuko Ueda and Yoshio Matsumoto[7] proposed a hand pose estimation technique. This technique is a subset of Vision based Gesture Recognition System wherein, multiple images of the Hand region are taken using a multiple viewpoint camera system, and then constructing a "Voxel".
Then a 3-dimensional model is fitted using the Hand model and the Voxel model.
This 3-dimensional model is then analyzed to extract the gesture information.
The following points were noted in the research paper:
1. Creation of a new model using inputs provided as an image via a camera setup.
2. Creation of the hand model follows a very complex method of fitting.

III. SYSTEM OVERVIEW

As introduced earlier, the concept work can be broken down into definitive steps, which include – training the model based on the input provided by the user, identifying the inputs during runtime, and finally, automating the chain or sequence of actions based on the input identified by the system.
A. TRAINING THE MACHINE LEARNING MODEL:
For the purpose of actually recognizing the input provided, we decided to create our own Machine Learning model. There is a vast availability of the models that have been developed with the purpose of recognizing and detecting images and other gestures. But we have still decided to train our own model because using a general machine learning model would mean accommodating and scaling our own algorithm theory to match with that of the generic model. Using our own model provides us with the flexibility to build the entire system exactly to our liking, mold and architect the flow entirely suiting to the use of the system.
So, to train our own model, we use what is known as MobileNet Model. MobileNet[8][9] is a pre trained model that is available to use in a web browser, with the help of TensorFlow.js. This MobileNet model can directly classify images, if the images are present in the ImageNet repository, but since we are providing custom gesture input, we are not going to use MobileNet directly. Instead, we are going to fine tune our own model, keeping the MobileNet model as the base model. Therefore, by fine-tuning the MobileNet model, we have created our very own image recognition model that is customized to interpret and identify the gestures provided.
MobileNet is a light weight, deep Neural Network. Due to its light weight, it is best suitable for mobile and embedded vision applications just like ours, where we need such a neural network to run on the browser itself.
MobileNet uses depth wise separable convolutions, which basically means that, there are multiple layers of convolutions, where in each layer, the neuron is only connected to a portion of the input image and all the neurons have the same weights of connection.
MobileNet provides us with the flexibility to fine tune our own machine learning model, instead of having to code and create an entire machine learning model from scratch. The parameters of the model can be shared, thus reducing the number of calculations needed to identify an image in this scenario.
We use MobileNet. This is because with the availability of TensorFlow.js, it is now possible to run and execute machine learning algorithms right on the computer's browser itself, instead of downloading external applications to work on it. Keras uses the TensorFlow engine in the background, which makes it very easy to use and accessible across various platforms.
So this is how the image recognition system is setup, in the background. This phase automatically begins when the index file is run when the server is started. As the page loads in the browser, there are multiple options present on the webpage, once it has successfully loaded.

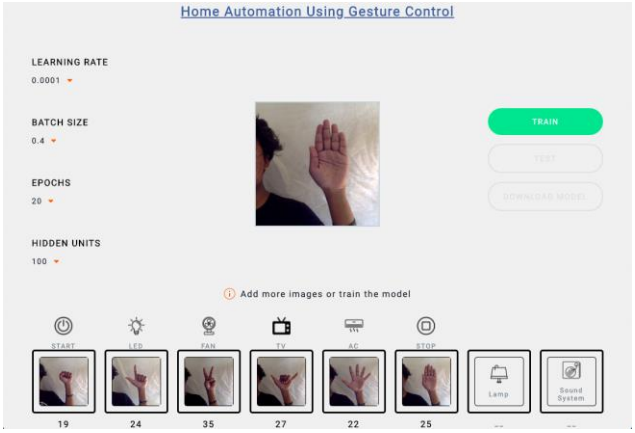

Fig.1 Training of Model by providing training images

There are fine tuning options present on the left side of the home screen-
LEARNING RATE: Generally, while training in deep neural networks, the optimization algorithms calculate the error in the current model's state and the examples given during as training dataset. The algorithm then updates the weights via backpropagation. The amount with which the weights are updated, also called step size, is known as the Learning Rate. This is usually a number between 0.0 and 1.0
BATCH SIZE: This is a term that is used in Machine Learning, which basically refers to the number of training examples the model utilizes in each iteration.
EPOCHS: This is a term that is used in Machine Learning, which refers to how many times the entire training dataset is provided as a whole to the algorithm.
HIDDEN UNITS: There are multiple number of layers present between the input and the output layers of a Machine Learning model. Each neuron present in these layers, hidden from the outside system, are called as Hidden Units.
On the bottom of the page, there are options to add images. This is a very important and crucial part of the system. It is this part where the inputs of gestures are provided to the MobileNet model to train our very own model. Once the input gesture images are provided for training, we can then start training our model by pressing on the Train Model button present on the right side of the page.

On pressing that button, the images are sent for training and within some time, our model is created. Once the model is created, the cost function is calculated and the resultant Loss of the model is displayed accordingly. We can then either Test or Download the model.

The next phase, identifying the input given to the webcam uses this model that has been trained and tuned by MobileNet.

B. IDENTIFYING INPUTS DURING RUNTIME

After the model has been trained, we can start giving gestures during runtime to the webcam. This new gesture will be sent to the model, and it will then try to classify it into one of the inputs provided during training, also called as classes.

Once the input has been recognized, it will then highlight the training class to which the input corresponds to or matches. What happens behind the scenes is that, once a class has been identified, a list is generated. This list contains the ClassId that identifies the class which is selected. This list, which now contains a list of multiple ClassIds, is then checked for a pattern that simulates a command. And if there is a match in the pattern, it is then sent for Automation.

Its Working is as follows :

When a class is identified, the ClassId is pushed onto the list. In the proposed system, we have identified two main commands – "START" and "STOP". Gestures are already provided for these commands along with gestures for the appliances while training.

When the class "START" or "STOP" is encountered, it clears the list and then pushes the corresponding ClassId in the zeroth index position. Also, whenever an Appliance is encountered, the corresponding ClassId is pushed onto the list only when the list contains either a START or STOP ClassId at the first position.

So after the ClassId of any appliance is pushed onto the list, this list is now sent as a POST method to the server, which has been setup using Express Framework[10] of NodeJS.

Express.js is basically a Node JS web application framework, which has been specifically designed for either single page, multiple page or hybrid web-applications.

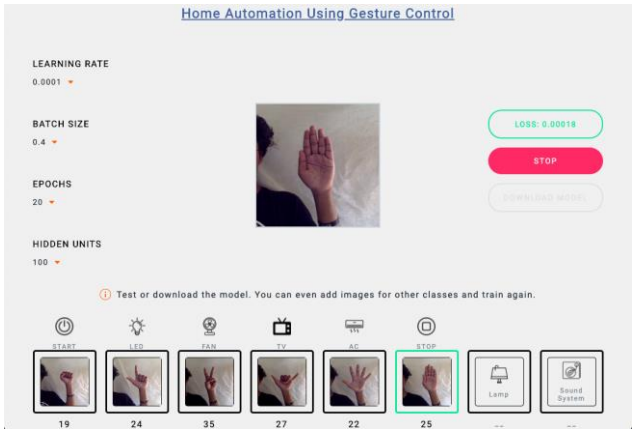

Fig.2 Recognition of "STOP" gesture during runtime

This method of sending the list by POST has been used because it is very difficult to send any data during runtime from a web-browser to another physical device without the use of a local server.

C. AUTOMATION OF THE GESTURE SEQUENCE

Once the list has been sent to the server, it has to be now communicated to the Arduino Board connected to the computer system.

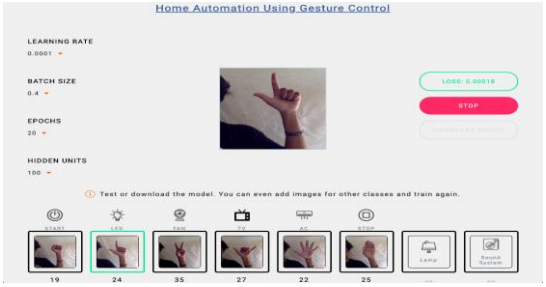

Fig.3 Recognition of the Gesture during runtime

Arduino Uno has been used to demonstrate and replicate the home environment system. LEDs have been connected to the microcontroller, which mimic the appliances present in a home system.

To achieve the automation, that is, controlling the lights in the current scenario, we have created a JavaScript code file, which contains the code to control the LED's based on the input provided by the list present on the server.

This JS file also resides in the server. To send the code written in JS to Arduino, we used a framework called as JOHNNY FIVE[11]. This framework allows the user to control an Arduino via JavaScript. This also requires the setup and running of a local server, which we have already implemented using the Express Framework

Johnny Five has classes and methods to control the Arduino Board. These methods are used to first define the Arduino Uno as an Object, then using prebuilt functions for defining the pins as LEDs, we can control the toggling on it. All of this work is done in the JavaScript file itself. To initialize the Arduino though, we have to upload the "Standard Firmata"[12] code onto the microcontroller. This code is found in the example section of the Arduino IDE. Once the code has been uploaded to the Arduino Uno, we just have to connect the Arduino to the computer and run the JavaScript file present on the server to control it.

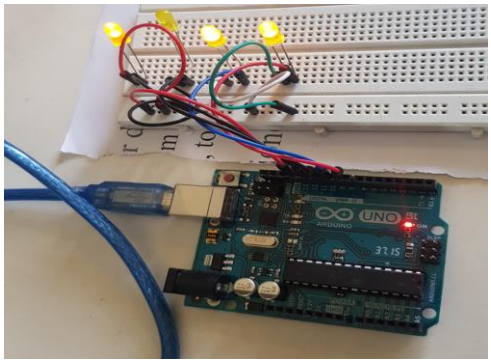

Fig.4 Toggling of LEDs based on Input Gesture Sequence

JS is a scripting language, and since we have used it to run continuously throughout the runtime, we have to setup the server locally and run all these files on it. So, the Index file of the system, which is basically the most important file and also the file which contains the home page, has to be run once the server is started. This page will load up the model and train the new inputs and send the recognized gesture sequence to the JS file that controls the Arduino.

This JS file, which uses Johnny Five framework, will then parse the List and then automate the toggling of the LEDs accordingly.

IV. CURRENT LIMITATIONS AND FUTURE SCOPE

Even though this proof of concept worked with an accuracy that is more than sufficient, it is still not feasible to replicate the same scenario at a live product stage. We used the web-camera present in the personal computer to take an image input, but in a real life situation, the input has to be captured using high resolution and independent camera setup.

While the proposed system gives us the flexibility to download the fully trained model, it is very complex to use the model again in the web-browser, where our project gets executed. Hence, every time the concept has to be demonstrated, we have to provide new training sample from the user as input before actual recognition takes place. Though this can be rectified in future scope, when the entire system is migrated out from a personal computer to an independent electronic device.

While the future of this system is meant to be as a closed loop, always running kind of system, the current proof of concept fails to implement that feature due to limitations provided by web browsers in the number of requests. This was actually rectified by setting up of the server and manually clearing the request list buffer.

Finally, even with this accuracy, which was achieved with the help perfectly aligned training inputs, it is very hard to replicate the same gestures during training under different circumstances. Since the machine learning part takes place over the browser, there is a chance of very high latency and errors creeping into the system, which ultimately results in misidentification of gestures during runtime. This can be rectified in the future by using high resolution cameras and powerful processors once the gesture recognition system has been migrated to an independent system of its own.

V. CONCLUSION

Hand gestures are indeed a powerful, natural and a novel way for human computer interaction, with lots of potential application in fields like Virtual Reality, Gesture Controlled Navigation in mobile devices, etc. Vision based hand gesture recognition techniques have many proven advantages compared with traditional devices. However, hand gesture recognition is a difficult problem and the current work we have done is only a small contribution towards achieving the results needed in the field of home automation using computer vision.

This concept presented a computer vision based system able to interpret static hand gestures. The system is using the personal computers web camera as way of providing training examples and, for the predictions, while being hosted locally on the server. The current system is not very efficient, although it is in controlled environments. This can be due to getting the training example from a low resolution camera or the server being not able to process the request as efficiently as needed.

On a concluding note, we feel that the concept provided in this paper is a very sound and technically viable for further research on this in the future. Not only are we able to control the LEDs using gestures, but it is a demonstration of the fact that such kind of a system can actually be implemented, that too right on the browser itself. Imagine the possibilities if this system is implemented on a standalone device.

REFERENCES

1. Donald A. Norman, “Design Principles for Human Computer Interfaces”, 1983.
2. Bergman, Johnson, “Towards accessible Human-Computer Interaction”, 1997.
3. Rajesh Singh, Anita Gehlot, Bhupendra Singh, “Introduction to Arduino and Arduino IDE and toolbox_arduino_v3”, 2019.
4. P. Voštinár, N. Klimová, J. Škrinárová, “Before We Start Arduino”, 2019.
5. Eugenia Cabrera, Maria & Manuel Bogado, Juan & Fermín, Leonardo & Acuña, Raul & Ralev, Dimitar, “Glove-Based Gesture Recognition System”, 2012.
6. Christopher Lee and Yangsheng Xu, “Online, interactive learning of gestures for human robot interfaces” Carnegie Mellon University, The Robotics Institute, Pittsburgh, Pennsylvania, USA, 1996
7. Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai, Tsukasa Ogasawara. “Hand Pose Estimation for Vision-based Human Interface”, 2003
8. Howard et al, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, 2017.
9. Tao Sheng, Chen Feng, “A Quantization-Friendly Separable Convolution for MobileNets”, 2018
10. S. L. Bangare, S. Gupta, M. Dalal, A. Inamdar “Using Node.Js to Build High Speed and Scalable Backend Database Server”, 2016
11. Manoj Kumar, Kailasa Akhi, Sai Kumar Gunti, Sai Prathap Reddy, “Implementing Smart Home Using Firebase”, 2016.

AUTHORS PROFILE

Vignesh Nadar is a Computer Science Engineer from Amity University Mumbai. He is an avid reader, and a technology enthusiast. He has worked on various projects that aim to alleviate the suffering of the common people, which include works like Flood Risk Management, Women Safety App, Sign Language Translator, etc.

Vaishnavi Sinha is a Computer Science Engineer from Amity University Mumbai. She actively started working for social causes right from the beginning of her academic career. Creator of Women Safety App, she has also researched on Skin Cancer detection using AI, etc.

Sushila Ratre is a Professor at Amity University Mumbai. She is also currently pursuing her PhD. She has an extensive record of teaching for the past 8 years. She has also worked on multiple projects that implement technology to solve social issues, while mentoring many young engineers to achieve the same.