ECE – GY 6143 Introduction to Machine Learning

PROJECT

A Proof Of Concept on Smart Home Automation Using Hand Gesture Recognition System

Vignesh Selvaraj Nadar
vsn2005
N13395682

ABSTRACT

Visual interpretation of hand gestures is a natural method of achieving Human-Computer Interaction (HCI). In this project, I present an approach to set up a smart home where the appliances can be controlled by the 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 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 desktop where the actual gesture recognition is implemented using Machine Learning techniques.

INTRODUCTION AND OVERVIEW

This project began as an idea to translate American Sign Language to Text in Real Time using Machine Learning, but the problem turned out to be very complicated while trying to implement it. While lots of data sets are available online to model a sign image classifier with efficiencies upwards of 97%, real-time translation of sign languages is vastly complex due to factors like there are a lot of shorthand notations, the signs are not usually perfectly made, etc. I do hope to tackle these problems in a project later on.

But what my algorithm ended up becoming was a highly accurate real time classifier of certain gestures which are given as an input to a webcam. The model is already trained on these gesture images before it is used to classify it. And once the chain of real time gestures was being identified, I progressively used the recognized chain of inputs as specific commands and was able to control a set of LED lights attached to an Arduino Uno, connected to the personal system.

This project is not a comparison of different models, but a model creation using transfer learning technique. A machine learning model is created by figuring out the weights and biases by repurposing a trained generic model.

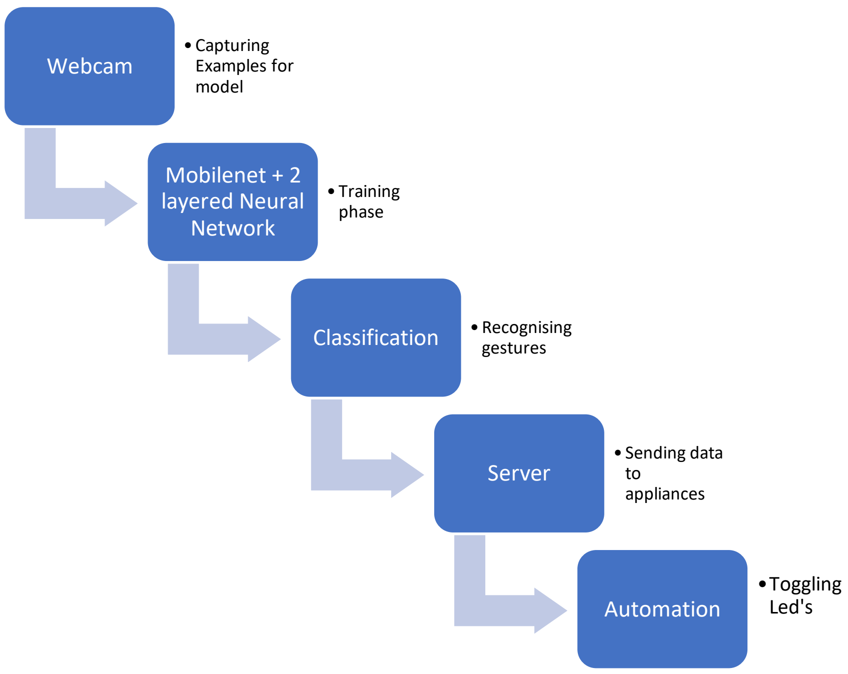
In this project, I am using MobileNet. This is because these are light weight deep neural networks that use depth separated convolution layers with the perfect tradeoff between Latency and Accuracy. The neural networks are primarily used for Mobile vision and Embedded Systems, which is perfect for the current project setup.

METHODOLOGY

Components:

So, there are two major components to the project. One of them is the implementation of the Machine Learning model which will classify the gestures provided to the system/webcam.

The other component is the hardware which controls the LED lights. Commands recognized by the gesture recognition system is sent to this component which in turn processes the data and controls the lights accordingly.


Webcam
• Capturing Examples for model
Mobilenet + 2 layered Neural Network
• Training phase
Classification
• Recognising gestures
Server
• Sending data to appliances
Automation
• Toggling Led's

HOW IT WORKS:

Classification of Gestures→

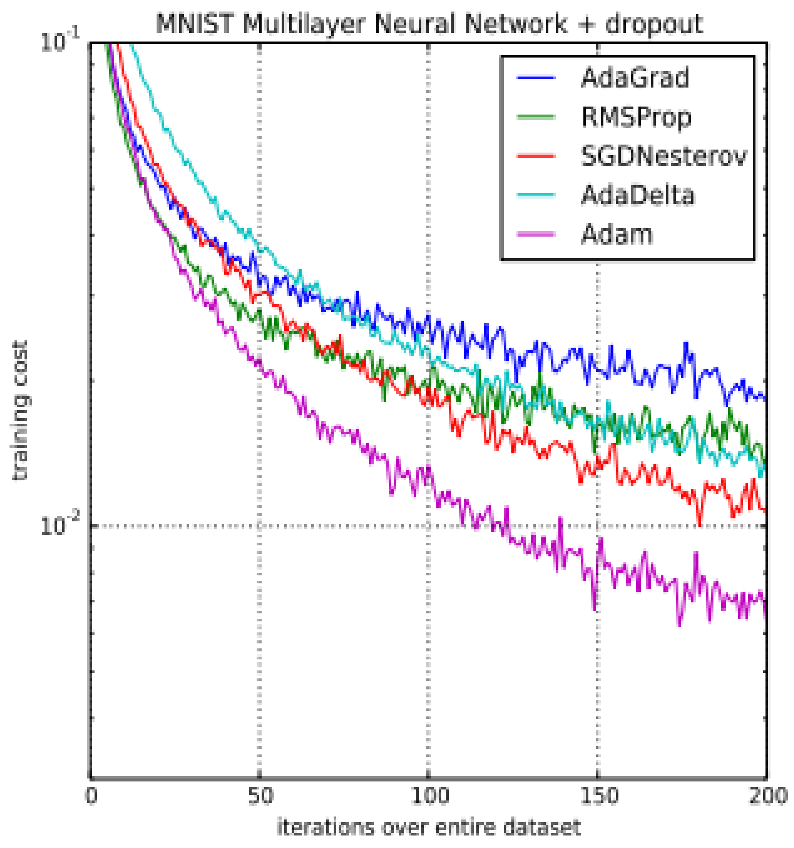
This is a machine learning algorithm, written in JavaScript, as I wanted to utilize the HTML CSS User Interface for interacting with the project, and also the result data had to be transferred to the external Arduino Device.

I am using TensorFlow as the development library as it is easier to use and has robust and dynamic functions for modelling deep neural networks.
For the model itself, I am using transfer learning using MobileNet.
In the implementation,

- The User Interface has the option to provide gestures for specified Appliances like TV, AC, etc. and certain commands like Start and Stop. Also, the user has certain options to provide custom hyperparameters like number of Epochs, Learning Rate, etc.
- Once the user provides pictures for each gesture and hits the train button, the algorithm is called where I create a 2-layer sequential model of the form:

```
model = tf.sequential({
  layers: [
    // Flattens the input to a vector so we can use it in a dense layer. While
    // technically a layer, this only performs a reshape (and has no training
    // parameters).
    tf.layers.flatten({inputShape: [7, 7, 256]}),
    // Layer 1
    tf.layers.dense({
      units: ui.getDenseUnits(),
      activation: 'relu',
      kernelInitializer: tf.initializers.varianceScaling(
          {scale: 1.0, mode: 'fanIn', distribution: 'normal'}),
      useBias: true
    }),
    // Layer 2. The number of units of the last layer should correspond
    // to the number of classes we want to predict.
    tf.layers.dense({
      units: NUM_CLASSES,
      kernelInitializer: tf.initializers.varianceScaling(
          {scale: 1.0, mode: 'fanIn', distribution: 'normal'}),
      useBias: false,
      activation: 'softmax'
    })
  ]
 });
```

- By creating a separate model, rather than adding layers to the MobileNet model, we "freeze" the weights of the MobileNet model, and only train weights from the new model.
- This model is fit with the test sample gestures provided by the User earlier, and an Adam Optimizer is used to increase the efficiency or reduce the Loss of the model.



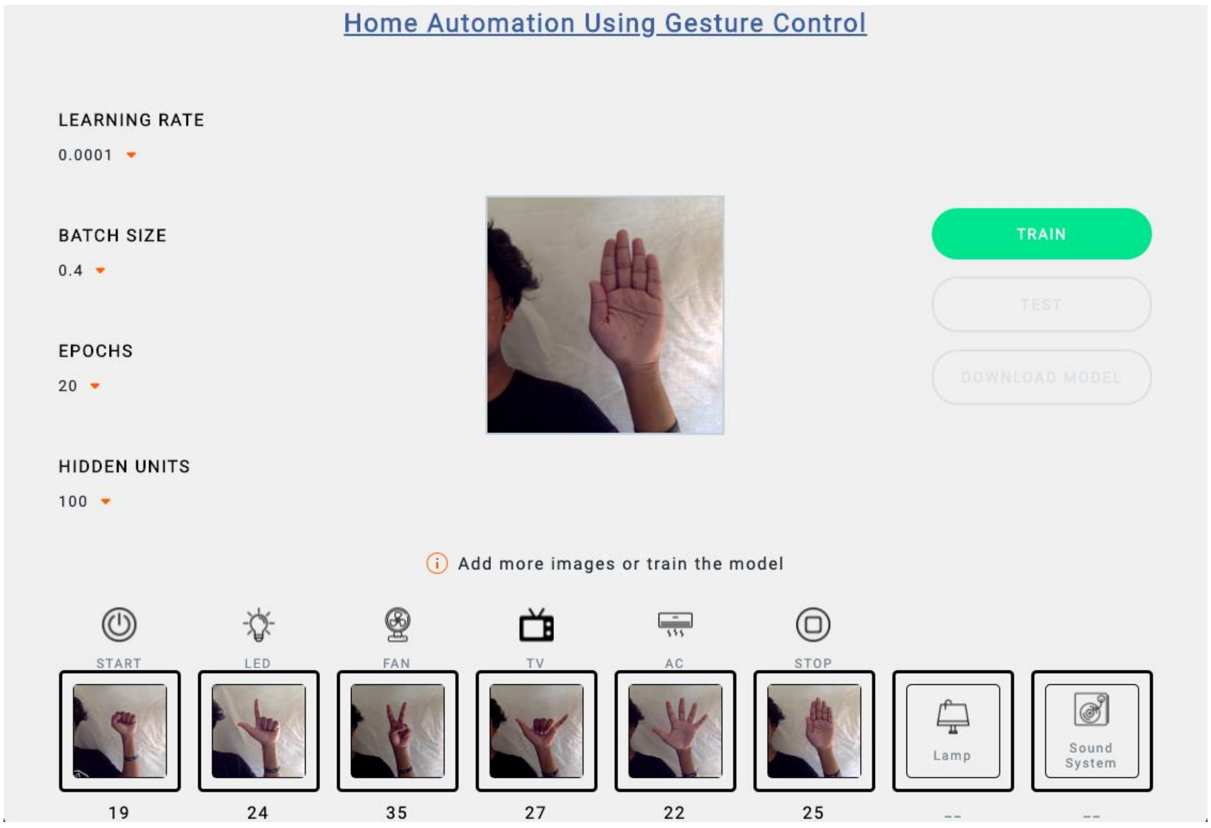
- Once the model is trained, it is ready to recognize real time gestures provided to it. Once the user starts providing gestures to the system, the model tries to identify the class to which the gesture belongs to and it sends this information to the other JS files, which in turn make sense of the data and transfer it to the Arduino accordingly.
- For example, the Gestures recognized by the model could be [‘Start’, ‘AC’, ‘TV’, ‘Stop’, TV’]. The other script files would then parse this information as a command – “Start the TV and AC, Stop the TV”, this would turn on 2 LEDS representing the AC and TV and turn off the one representing TV.

Parsing the recognized commands and executing it →

- After the prediction of the class that the gesture belongs to we creating an XMLHttpRequest().A XMLHttpRequest (XHR) objects is created to interact with the node server.
- With XMLHttpRequest object we are sending a list in the request body. The Server is using express framework for handling the request. A request is generated continuously for every new command or prediction and command is sent to the server in the form of a list.
- The server on receiving the command as list unwraps the list and send it to the Arduino Uno. I am using Johnny five framework to communicate with the Arduino. The Arduino is serially connected to the laptop using a serial port.
- The Arduino contains LED’s attached to it. The LED’s represent an appliance that the user will have in the working environment. On receiving the data from the Server, the Arduino switches on and off the particular LED representing the particular appliance.

RESULTS:

1. User Interface and Providing Input for Training ->



Home Automation Using Gesture Control

LEARNING RATE
0.0001

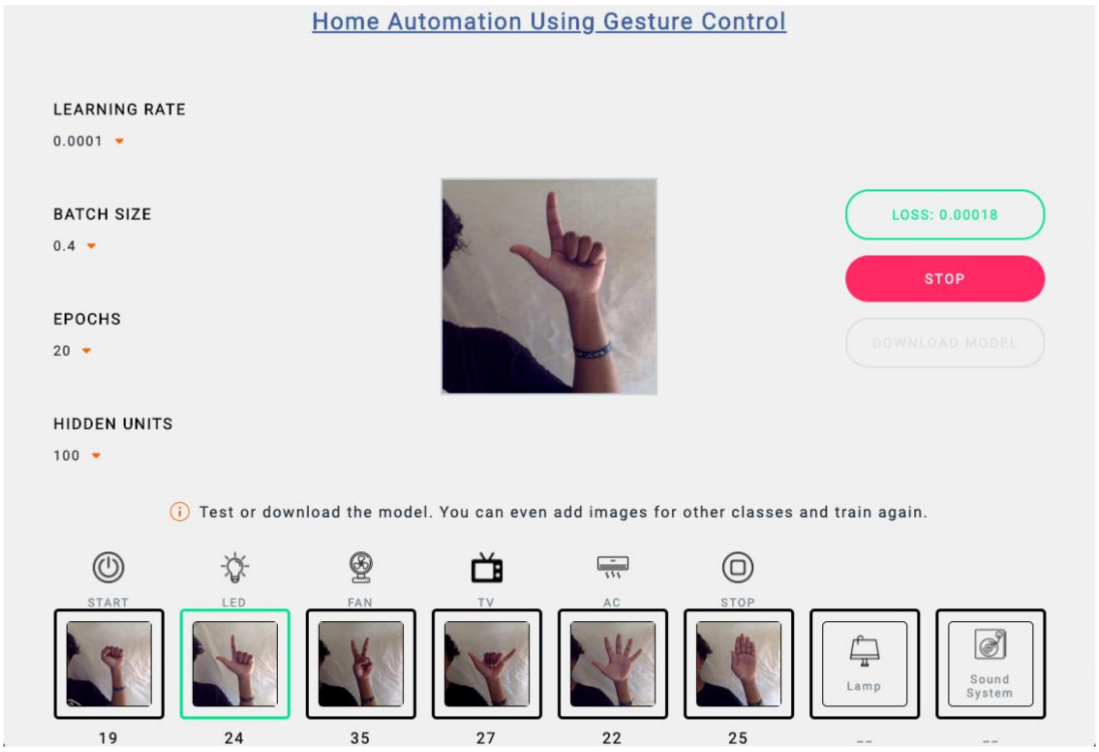
BATCH SIZE
0.4

EPOCHS
20

HIDDEN UNITS
100

Add more images or train the model

2. Using the Model:



Home Automation Using Gesture Control

LEARNING RATE
0.0001

BATCH SIZE
0.4

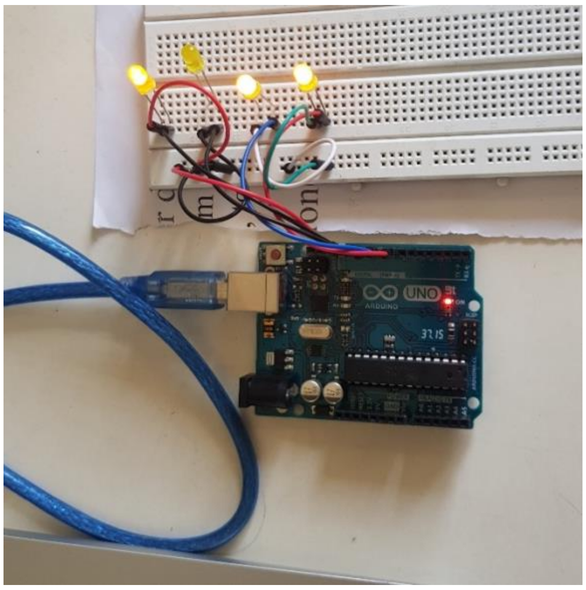
EPOCHS
20

HIDDEN UNITS
100

LOSS: 0.00018

Test or download the model. You can even add images for other classes and train again.

3. Controlling the LEDs :



CONCLUSION:

Light Weight Deep Learning Models like MobileNet can be a great tool to develop such systems like the one mentioned above. Even with providing casual and slightly varying images to train the model, it was still able to achieve >99% accuracy.

Hand gestures are a powerful way for human communication, with lots of potential applications in the area of human computer interaction. Vision based hand gesture recognition techniques have many proven advantages compared with traditional devices.

THANK YOU!