Home Automation through Hand Gestures Using ResNet50 and 3D-CNN

Ankitha Raksha[1], Raghul Krishna Rajasekaran[1], Praveen Francis[1], Suhas Yogeshwara[1], Alexander I. Iliev[1, 2]

[1] SRH University Berlin, Charlottenburg, Germany,
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
{3104832, 3105481, 3105577, 3105758}@stud.srh-campus-berlin.de,
ailiev@berkeley.edu

Abstract. This paper talks about using hand movements for the operations of electrical equipment at home. With the use of the much-advanced algorithms - 3D-CNN and ResNet50 to increase the accuracy in detecting the hand gesture to correctly predict the right motion for the functioning of the electrical device. Eventually, the project focuses on the comparative study between different architectures so that we can determine the best-suited model for these kinds of image detection. We aim to bring about a good accurate model for detecting the hand signals.

Keywords: Hand Gestures, Home Automation, ResNet50, 3D-CNN.

1 Introduction

Home automation has been one of the farfetched ideas since the modern era for designing the house began. Since smart homes began as a project all of us started to wonder if we would have a fully functional operating home. After so many years of challenging work and advancement in technology, we have come to a point where we can automate the electrical device at home just by using simple hand gestures. This feature will not only ease out your movement across the room just to switch off the lights but also has a light for older people who have trouble walking or in case they forget to switch off a light, this can come very handy. For example, sliding hand increases or decreases the temperature of the heater or thumbs up, turn on all the lights or thumbs down turns off the lights in the room or pre-set settings for user depending on users' preferences of temperature or other controls. We use the technology of machine learning and artificial intelligence to identify the pattern and modify its corresponding control.

Briefly, we can say, it is possible to control domestic appliances (connected over the internet) using just your smartphone. For deploying this project, we use ResNet50, which was first introduced in 2012 for an image classification contest along with Alexie. As a method of comparison, we also make use of the 3D-CNN model which makes use of spatial and temporal dimension that makes it good for action detection.

Digital Presentation and Preservation of Cultural and Scientific Heritage. Conference Proceedings. Vol. 11, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2021. ISSN: 1314-4006, eISSN: 2535-0366

ResNet50 contains 50 layers of deep neural network and hence can increase the accuracy of image recognition.

Along with the available free sources of dataset with an enormous number of video clips of gestures to recognize the pattern and to train the model. Our objective is to control the appliances by having a human-computer interface with hand gestures. By having control over the appliances without having to manually operate them just by 'waving hand' or by 'pushing hands in or out. The aim is to differentiate the models (ResNet50 and 3D-CNN) and to achieve an accurate result. The automation process used in the model for the human-computer interface allows us to have control over various devices and systems and to avoid the risk of accidents or damages in the system or devices. Training the model in coherence with the dataset used for them helps in improving the performance of the network.

2 Literature Review

The interactions between humans and devices are much more advanced than before. To identify the hand movements and analyse which gesture it is, there are many ways to do so.

Hand gestures can be captured using electrical signals from the glove which has sensors attached in it, which is then put on hand. This can detect which hand gesture is it from the signals and map it to the appropriate result (Pomboza-Junez & Holgado-Terriza, 2015). But since the world is advancing towards more touchless services, we therefore also have vision-based detection (Shuai, Premaratne, & Vial, 17-19 Nov. 2013) where the Hidden Markov Model is used. In system (Nikhil & Shakshi, 2018) the authors have detected the hand gestures using MATLAB and corner point detection algorithm for classification of the hand gesture. The paper (Hatwar, Wahile, & Padiya, 2017) explains to integrate gesture recognition with electrical devices using MATLAB and a microcontroller. The paper (Naveenkumar, Padmaja, & Nagadeepa, 2015) presents releasable fpga based hand gesture recognition system is proposed by using a method called artificial neural network is used this method used to add learning capabilities of gesture recognition system that can be used for impaired people. The paper (Naveenkumar, Padmaja, & Nagadeepa, 2015) presents a framework for hand gesture recognition based on the information fusion of a three-axis accelerometer and multi-channel electromyography sensors, A decision tree and multi-stream hidden Markov models are used to get the results. Finally, paper (Xu, et al., 2011) hand motions can be captured with accelerometers placed around the forearm. Since accelerometers can also be integrated into mobile systems easily. For this, a neural network architecture consisting of two distinct kinds of recurrent neural network (RNN) cells is used (Koch, et al., 23-27.July.2019). But what really inspired us to produce using deep learning in detecting the hand gesture much more accurately was when the authors for paper (Sushmita, Ninad, Aboli, Shreyash, & Ashwini, 2020) (Mujahid, et al., 2021) have made use of CNN and YOLO algorithms for hand gesture classification. The papers (He, Zhang, Ren, & Sun, 2015) (Xu, Yang, & Yu, 2010) have extensively explained

the working and implementation of 3D- CNN and ResNet50 and prove to be one of the best algorithms for image detection.

3 Problem Statement

Given that, we have learned the knowledge from all the research done, we also thrive to implement more advanced methods. Previously the authors made use of MATLAB or any wearable hardware for detection of human hand gesture, it will be more convenient as we move to more used open software's and touchless devices. And instead of taking 3 gestures we found a dataset titled Jester dataset (Materzynska, Berger, Bax, & Memisevic, 27-28.Oct.2019), which has 27 different hand gestures. Moving towards more advanced and easier to implement algorithms for the ease of detecting the hand gestures has given rise to making this attempt. So, we have produced implementing 'Home Automation through hand gestures using ResNet50 and 3D-CNN'.

4 Dataset

The dataset used for our project is the Jester dataset which is considered as the large-scale gesture recognition dataset. Total of 148092 short clips from 1376 different people. The duration is 3 seconds for a single video clip with 27 different actions. With a total number of 5 million frames with 12 frames per second covered in this dataset. Gestures made are zooming in or out of fingers, pushing hands in or out. Jester dataset recommends 3D-CNN model to build the human-computer interface and with the vast amount of data being provided impacting the performance of the network with the 90% accurate rate being achieved in them. This dataset offers a real-world scenario. Unlike the others which provide a lack of variability in their background and with the limited gestures. The system is built so that the pattern is recognized and detects the correct gestures. There is a significant variation in the background of the actors which helps in improving by recognizing the complex gestures with regards to the contrast of the background. The model only detects the given gestures and by not responding to any other gestures unknown or known.

5 Methodology

We used the Jester dataset to train two different architectures of ResNet50 and 3d-CNN for predicting human hand gestures and selected the most accurate model. To do so we trained both the models for 26 hand movements. In both ResNet50 and 3D-CNN we make use of the 'ReLU' activation function and 64 filters. We make use of ImageDataGenerator for data augmentation. The images are sent to each model in 12 batch sizes. To monitor the training, we used Telegram Bot and Hyper Dash (Schreiber, n.d.). In both the cases Stochastic Gradient Descent (SGD) as optimizer for the loss function. The input dataset was sent in batches to each model and the output is derived in terms of index. The output from the model is mapped to certain functions of home automation

like turning on or off lights or increasing or decreasing the temperature which can be customized by the end-user as per their needs for more flexibility.

We have come across similar projects where the hand gesture is detected based on electrical sensors in glove or just be limited to the number of hand actions. But in our project, we take the idea one step further to add more hand gestures which can be detected using a camera, and the laptop camera also works to recognize your hand movements because of the features in OpenCV. The index of the hand movement is mapped onto a function like 'lights on' etc.

6 Model Architecture

6.1 ResNet50

Convolution neural networks show promising results in the field of computer vision. In convolutional nets, the important advantage is that they progressively learn more complex features, and the deeper the architecture the more promising the results, but there is a bottleneck for these deep architectures like [VGG] after a certain point model suffers from vanishing gradients. The problem of vanishing gradients has been overcome by deep residual networks in ResNet. We can see that it starts with one convolution layer and a pooling layer followed by 4 layers of similar behaviour. Each layer follows the same pattern with different feature maps [64,128,256,512], as shown in Figure 1, and the width and the height remain constant for the entire layer. The skip connection between the layers adds the output from previous layers to the next layers reducing the problem of vanishing gradients for deep convnets. ResNet comes from many sizes irrespective of size; it follows the same pattern, and in this project, we used the ResNet50 model for predicting human hand gestures.

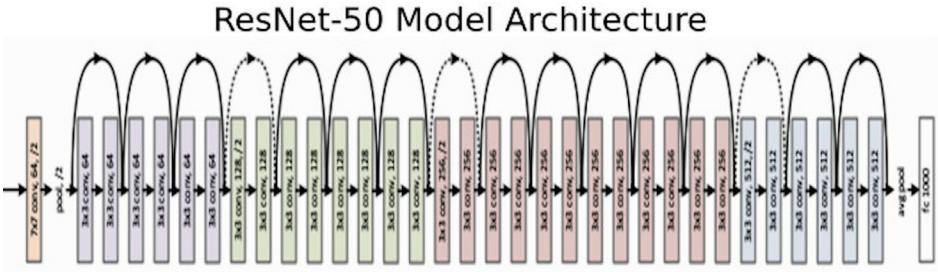

Fig. 1. ResNet50 Architecture

6.2 3D-CNN

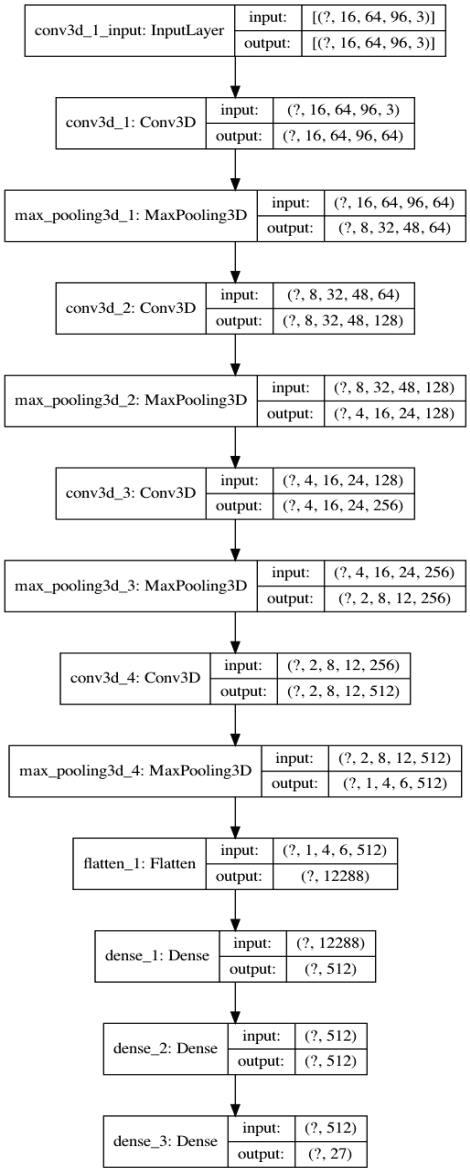

Fig. 2. 3D-CNN Model Architecture

Convolutional neural networks are a type of deep models which performs directly on the raw inputs. In a typical 2d convolution architecture the convolutions are applied on 2d feature maps to extract features from images, in this project we use videos as an

input, so we need to capture both the temporal and spatial features from the dataset. 3d convolution extracts features using 3D kernels forming a cube by stacking the video frames, so features are connected between one frame to another. There are many 3D-CNN architectures that are out there, but in this project we used the architecture shown in Figure 2, where it consists of 1 input 3D convolutional layers and 4 3D convolutional layers followed by 3D max pooling for 3D-convolutional layers below this is a flatten layer that flattens the output from the above layers and sends it to two dense layers and an output dense layer with softmax activation for predicting categorical events.

7 Training the Model

7.1 Hardware Requirements

The models used in the project are trained in amazon web services. The type of instance used for training is g4dn.2xlarge. The cost of the instance is 0.752 USD per Hour on-demand Linux price with Nvidia T4 vGPU with 16384 RAM and 8 vCPUs. The model's performance was monitored by Hyper Dash, an android app for monitoring machine learning models anywhere and a telegram bot to monitor and change the learning rate.

7.2 Model Training

7.2.1 Residual Block

Neural networks are universal function approximators and the accuracy increases as the number of layers increases. But there is a limit to the number of layers added to improve accuracy. So, if neural networks were approximators of universal functions, they should be able to learn any simple or complex function. But it turns out that, thanks to some problems like fading gradients and dimensional curses, if we have a deep enough mesh, it might not be possible to learn simple functions like an identity function. Now, this is clearly undesirable. Also, if we keep increasing the number of layers, we will find that the accuracy will start to saturate at times and eventually decrease. And this is usually not due to over-dressing. So, shallower networks learn better than their deeper networks, which is counterintuitive. But this is what we see in practice and is often referred to as the degradation problem. The solution to all the problems above is given by the residual block by skipping layers.

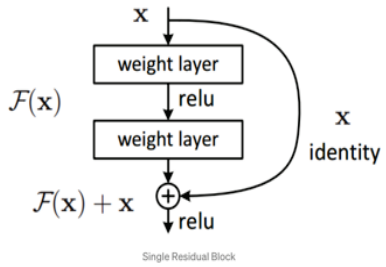

Fig. 3. Single residual block

Consider a Neural network block, where input x and B(x) the true distribution. The residual R(x) is given by:

$$A(x) = B(x) - x$$

$$H(x) = R(x) + x$$

The residual block tries to learn the actual output B (x). From the formula above, you can see that the layers are trying to learn the residuals because there is an identity connection due to x. A (x) In summary, the neural net learns the actual output (B (x)), while the residual network hierarchy learns the residuals (A (x)). Therefore, its name is the remaining blocks. Residual block F(x) representation mathematically represented as:

$$y = F(x, \{Wi\}) + x$$

where y is the output function, x is the identity or the input to the residual block F (x, $\{W_i\}$) is the residual block. W_i represents the weight layers in residual block.

From the paper (Yin, Li, Zhang, & Wang, 2019), mathematical properties of ResNet have been explored. The paper showed that the residual block can be formulated by a partial differential equation, where ResNet is equivalent to the path integral formula. The partial differential equation of ResNet is given as:

$$\sum_{K} \sigma_1^2 \frac{\partial^2}{\partial x} + \sigma_2^2 \frac{\partial^2}{\partial y} + p_1 p_2 \frac{\partial^2}{\partial x \partial y} + p_1 q_2 \frac{\partial}{\partial x} + p_2 q_1 \frac{\partial}{\partial y} + q_1 q_2$$

where σ1, σ2, p1, p2, q1, q2 are trainable parameters. K is the number of convolution kernels.

7.2.2 3D-CNN Description

- 3D-CONVOLUTION LAYER: In the 3d-CNN architecture, there were 5 3d-Conv layer was used with different with many sizes of filter [64,128,256,512]

for each layer with constant kernel size of (3,3,3) and strides of (1,1,1) with ReLU activation. a 3d-convolution layer can be used to find patterns and features across three dimensions (depth, height, and width). In this layer, a cube of kernel size (3,3,3) with varied sizes of filter depending on the position of the layer moves through the volumetric cube of input frames stacked from video to identify features from 3 dimensions with stride of (1,1,1) i.e., moves one pixel at a time through the three dimensions. The Feature map of a 3D-CNN is represented as:

$$v_{xy}^{abc} = tanh\left(h_{xy} + \sum_m \sum_{p=0}^{P_x-1} \sum_{q=0}^{Q_x-1} \sum_{r=0}^{R_x-1} w_{xym}^{pqr} v_{(x-1)m}^{(a+p)(b+q)(c+r)}\right)$$

where v_{xy}^{abc} is the feature map in the x[th] layer, tanh is the hyperbolic tangent function, h_{xy} is the bias for the feature map, Pi, Qi, Ri are the dimensions of the 3d kernel.

- MAX POOLING-3d: A total of 4 max-pooling 3d layers was used with constant configuration of pool size (2,2,2) and strides of (2,2,2). A max-pooling layer is typically used in the CNN architecture to reduce the dimensionality of the input and feed it to the following layer without losing information which saves many computational resources. In this architecture max-pooling 3d down-sampled the input by taking a maximum value from 2 X 2 X 2 cubes moved through a stride of (2,2,2) from input from the convolution layer thereby reducing the dimensionality
- FLATTEN: Flatten layer in this architecture converts the 3d output from max-pooling 3d layer to 1-D layer and feeds it to the following dense layer.
- DENSE LAYER: There are two fully connected layers of number of units 512 with rely upon activation and an output layer of size equal to the number of classes i.e., 27 with activation of SoftMax was used.

- ACTIVATION FUNCTIONS:

  RELU: The rectified linear activation function will output if the input value is positive else the values will be changed to zero hence ReLU plays a vital role in the vanishing gradients problem.

  SOFTMAX: Softmax activation is used in the output layer to convert vectors of numbers to vectors of probabilities where the probabilities can be mapped to different categorical classes.

7.3 Training Cost

The data was trained on ResNet50 using AWS services since the data is huge and takes a lot of time and resources from the local engine. We trained the model for 26 epochs which took 12 hours and 3D-CNN for 20 epochs it took 8 hours 30 minutes to train. Initially we took 40 epochs, but the 3D-CNN model was saturated at the 20[th] epoch and ResNet50 at 26[th] epoch, and over 35[th] epoch we see a declining in accuracy level. The training was monitored using Hyper Dash. This training in total cost around 15.7$, the summary of which we can see in the table below:

Table 1. Cost Overview

Model name:	No. Of epochs:	Total time for training:	Cost/hr.:	Total cost:
ResNet50	26	12:00:00	0.752$/hr.	9$
3D-CNN	20	8:22:21	0.752$/hr.	6.7$

8 Results

The model built was able to detect the hand gestures and the correct movements for the operation of the house appliances. When we run the code, it opens the camera, analyses the hand gesture, and shows the appropriate results, it shows the confidence of that label along with the output function, which can be seen in the Figure below. Given 148092 videos to train for both 3D-CNN and ResNet50 we recognize that ResNet50 was giving better results when compared to 3D-CNN, we owe this efficiency to the deep layers of ResNet50 which picked up the action of hand much quicker.

Fig. 4. Sample Output

8.1 Training and Validation Accuracy for 3D-CNN

Initially the accuracy is gradually rising for both validation and training dataset, but later validation accuracy becomes stagnant at a lower range when compared to training accuracy. We have managed to achieve around 80% accuracy for validation dataset and 98% accuracy for training dataset, this is clear sign that the model is overfitting:

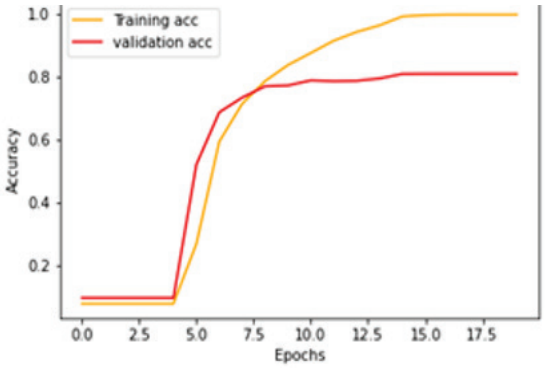

Fig. 5. Training and Validation accuracy for 3D-CNN

8.2 Training and Validation Accuracy for ResNet50

Both training and validation increase steeply and settle down at around 80% but there is no overfitting in this model.

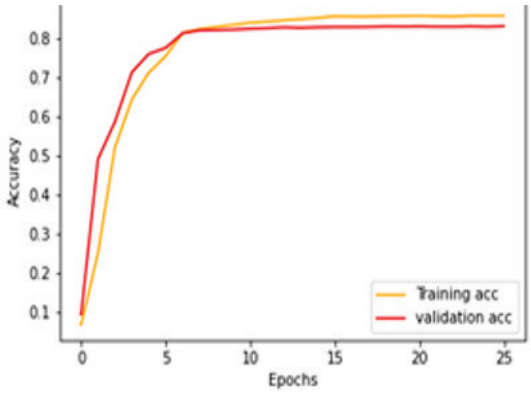

Fig. 6. Training and Validation accuracy for ResNet50

9 Conclusion

The main objective of the project was to differentiate between the models ResNet50 and 3d-CNN where the models were trained on Amazon web service. The trained models were able to give an accurate result and among the two ResNet50 was able to provide a satisfactory result in detecting the hand motions where the home appliances operations can be controlled. Though the model ResNet50 had a longer training time than 3D-CNN but with 26 epochs as shown in the cost overview table 1, the model had higher accuracy than 3D-CNN in detecting the image recognition.

10 Future Scope

The system can be extended in following ways:

1. Compare the existing models with other models like LSTMs
2. Human hand gesture recognition is not limited to home automation exploring the possibilities of gesture recognition in various applications.

References

Hatwar, P. D., Wahile, N. A., & Padiya, I. M. (2017, March). Home Automation System Based on Gesture Recognition System. *International Journal of Emerging Technologies in Engineering Research (IJETER), 5*(3).

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep Residual Learning for Image Recognition*. Retrieved from Cornell University: https://arxiv.org/abs/1512.03385

Koch, P., Dreier, M., Maass, M., Bohme, M., Phan, H., & Mertins, A. (23-27.July.2019). A Recurrent Neural Network for Hand Gesture Recognition based on Accelerometer Data. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE.

Materzynska, J., Berger, G., Bax, I., & Memisevic, R. (27-28.Oct.2019). The Jester Dataset: A Large-Scale Video Dataset of Human Gestures. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Seoul, Korea (South): IEEE. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/document/9022297/authors

Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H. (2021, April 28). Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. *Applied Sciences, 11(9)*(4164).

Naveenkumar, N., Padmaja, V., & Nagadeepa, C. (2015, February). Implementation of Gesture Recognition System for Home Automation using FPGA and ARM Controller. *International Journal of Science and Research (IJSR), 4*(2), 2099-2105.

Nikhil, A., & Shakshi, M. (2018, March 22). Home Automation Using Hand Gestures. *Iconic Research And Engineering Journals, 1*(9), 49-53.

Pomboza-Junez, G., & Holgado-Terriza, J. A. (2015). Control of home devices based on hand gestures. *IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*. Berlin, Germany: IEEE.

Schreiber, A. (n.d.). *Hyper Dash*. Retrieved from https://github.com/hyperdashio/hyperdash-sdk-py

Shuai, Y., Premaratne, P., & Vial, P. (17-19 Nov. 2013). *2013 5th IEEE International Conference on Broadband Network & Multimedia Technology* (pp. 63-69). Guilin, China: IEEE.

Sushmita, N., Ninad, K., Aboli, P., Shreyash, K., & Ashwini, J. (2020). Gesture Controlled Home Automation Using CNN. *International Research Journal of Engineering and Technology (IRJET), 7*(3), 5391-5395.

Xu, W., Yang, M., & Yu, K. (2010). 3D Convolutional Neural Networks for Human Action Recognition. *Proceedings of the 27th International Conference on Machine Learning (ICML-10). 35(1)*, pp. 495-502. Haifa, Israel: IEEE.

Xu, Z., Xiang, C., Yun, L., Vuokko, L., Kongqiao, W., & Jihai, Y. (2011, MARCH 22). A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 41*(6), 1064 - 1076.

Yin, M., Li, X., Zhang, Y., & Wang, S. (2019, April 16). *On the Mathematical Understanding of ResNet with Feynman Path Integral*. Retrieved from Cornell University: https://arxiv.org/abs/1904.07568

Received: June 05, 2021
Reviewed: July 15, 2021
Finally Accepted: July 22, 2021