Hand Gesture Recognition and Implementation for Disables using CNN'S

Kollipara Sai Varun, I. Puneeth and T. Prem Jacob

***Abstract*—Hand Gestures are playing the major role in today's industry. The development of the gesture recognition or detection will help people in different ways. Development of many automated technologies like machine learning, deep learning, neural networks and computer vision will help in using futuristic methods like gesture controlling and recognition. These gestures can be used in such a way it can help with people who have difficulty in controlling or operation systems or devices. Here the development of the model is in such a way it can help us to recognition and implementation. For such developments these models will be working with Convolution Neural Networks and Back Propagation methodologies which can help for the easy working of these models.**

***Index Terms*—Computer Vision, Convolution Neural Networks, Deep Learning, Image Processing.**

I. Introduction

THE communication or transfer of data in between human and human is really easy and understandable. But when it comes to human and machine it's really difficult because even machine knows all the languages humans can speak and understand, they cannot communicate with that knowledge or data. So for improving that communication features with machines we can develop some interactive techniques like gesture recognition [1]. This is becoming a trending topic in the field of computer science and its applications related to deep learning technology [2].

Hand Gesture acknowledgment is essential for structuring no touch or control interfaces in vehicles. Such technologies enable drivers to drive while at that time connecting with different controls, e.g., sound and cooling, and also there are line enhance drivers' security and solace. In the recent decades, numerous vision-based powerful hand signal acknowledgment [3-5] calculations were presented.

To perceive motions, distinctive highlights, for example, handmade spatiotemporal descriptors and enunciated models were utilized. As signal classifiers, concealed Markov models, contingent irregular fields and bolster vector machines (SVM) [3] have been broadly utilized. Notwithstanding, vigorous order of signals under broadly fluctuating lighting conditions, and from various subjects is as yet a testing issue.

Computer-human communication [6] refers to the way how the human communicate to the computer/machine, and since the machine is not useful until a human trains the machine for a particular task. There are mainly 2 characteristics that will be checked when developing a man-machine communication model as mentioned in: machine's performance and usage. The Model performance refers to how well the machines are performing to communicate with the human and usage refers to weather all the provided functionalities are performing according to the development.

Gestures can be in any form like hand image or pixel image or any human given pose that require less computational difficulty or power for making the devices required for the recognitions to make work. Different techniques are being proposed by the companies for gaining necessary information/data for recognition handmade gestures recognition models [7] [8]. Some models work with special devices such as data glove devices and color caps to develop a complex information about gesture provided by the user/human.

Recently, visual communications like augmented reality and virtual reality has attracted attention since many companies have introduced futuristic devices for the company's growth. For example, smart glasses for realistic features and many applications that have been developed in many companies. Virtual glass and augmented devices displays have also been introduced and are being used in futuristic applications. Hand gesture recognition is widely used as an interface to give commands to these reality controlled devices.

The system that acquires a perfect balance between the concepts considered as well performed and super powerful systems. Gestures can acts as communication barrier between the human and machines. Also can develop between people who cannot communicate properly.

The rest of the paper is organized as below. Section II and III describes the method and system implementation of the work respectively. Section IV explains the CNN working. Section V discuss about the results. Section VI and VII concludes the paper with conclusion and further development respectively.

Kollipara Sai Varun, I. Puneeth and T. Prem Jacob are with the Computer Science Department, UG Student, Sathyabama University, Chennai, India. saivarunprabas@gmail.com puneeth2011@gmail.com premjac@yahoo.com

978-1-5386-7595-3/19/$31.00 ©2019 IEEE

II. METHOD

The model uses a convolutional neural networks [9] for hand gesture recognition. Most of the researchers/developers classified that gesture recognition system into mainly getting the input image/data from the camera or from any input providing devices. This section introduces the way for operating the data provided from the user from cameras or other input devices. The layers will be created and trained with many sample data and will try with a real time data for the checking it with many random input's.

A. Data Set

The dataset used for implementation of this model is a CNN [10-13] usable dataset of images. It consists of more to 20,000 images with more than 1.2 G.B. This data will be implemented in our model for creating a trained model which is used for recognizing the gestures and for the implementation of those gestures for some operations in systems. This data consists of sample images of Palm, Fist, Fist Moved, Thumb, Index, OK, Palmmoved, Down, C symbol, I symbol in Fig. 1 & Fig. 2.

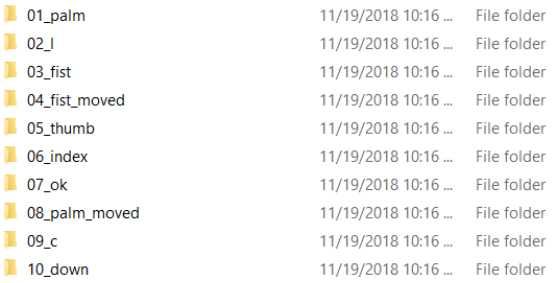

Fig. 1. This is the data downloaded from the kaggle repository that will help us identifying total 10 different types of hand representations with individual labels and there are 10 samples of the data.

The system includes four parts: acquiring a gesture sample, gesture sample processing, run-time gesture recognition and a control system. For the first step in acquiring a sample from user, use OpenCV [10] to activate the system camera to get the original image Fig. 6, Fig. 7, Fig. 8. The plotting and how the system can see the image and the model is made to perform operation accordingly. And the downloaded data set will be having different data sets. The system is made to work on each and every one of the sample data that is provide by the kaggle dataset repository. The path flow of the processing is provided in Fig. 3.

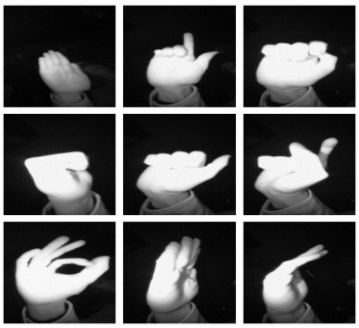

Fig. 2. Dataset Samples – So this are some sample images that are available in that folders provided in Fig. 1. So we will train out Network based on these images.

B. Procedure

The data which user will be giving as an input will be processed by a neural network and will be training that data in the CNN model. Once the data is successfully trained it can take sample data from the user and make it recognize and make the model remember and work on the model to perform an operation when the same gesture is invoked.

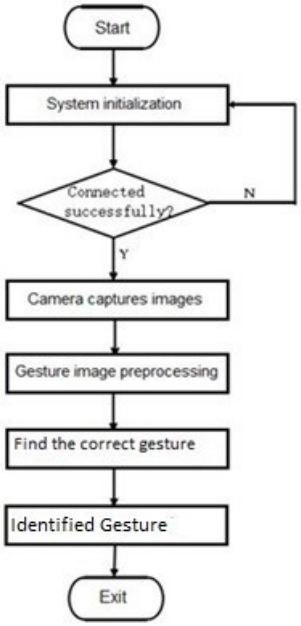

Fig. 3. The flow represents the steps that are present in the model working and how the output will be.

```
In [7]: lookup = dict()
        reverselookup = dict()
        count = 0
        for j in os.listdir('../leapgestrecog/leapGestRecog/00/'):
            if not j.startswith('.'): # If running this code locally, this is to
                                      # ensure you aren't reading in hidden folders
                lookup[j] = count
                reverselookup[count] = j
                count = count + 1
        lookup

Out[7]: {'01_palm': 0,
         '02_l': 1,
         '03_fist': 2,
         '04_fist_moved': 3,
         '05_thumb': 4,
         '06_index': 5,
         '07_ok': 6,
         '08_palm_moved': 7,
         '09_c': 8,
         '10_down': 9}
```

Fig. 4.The is the initial step that is performed – Execution of the dataset. Here we are labeling the data based on the downloaded folders and labeling based on folder names.

C. Gesture Classification

The developing and analysis of the input hand image, the gesture classification method [11] is used to recognize the input gesture from the recognized data can be seen in Fig. 3. The execution of dataset is shown in Fig. 4. Recognition process affects a certain selection of attributes, parameters and an algorithm. The system is made to work accordingly and that can be seen in Fig. 5.

We are designing our model network based on some features so when it comes to features in a excel data we can see many parameters like variance, mean and standard deviation and also we can plot some graphs based on the vales

of the attributes. For images we will be converting the color to a greyscale or black and white image which are known as masks and using those the neural networks can save the images based on features. The plan is detailed by properly training CNN with all kinds of hand gesture representations[14,15].

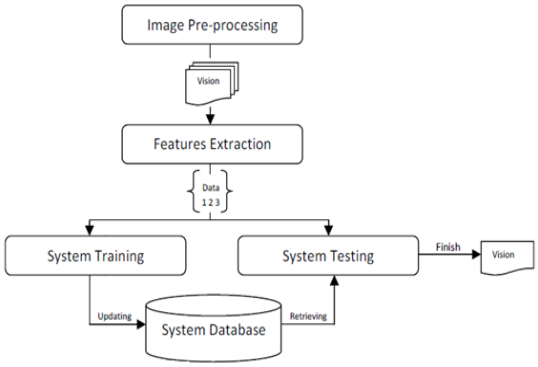

Fig. 5.The is the main step that is performed – Feature Extraction and Image Understanding. We can separate our data into testing and training, then processing with proper data.

III. System Implementation

The system working conditions and environment is based on Anaconda Environment interface design, with OpenCV, Tensor Flow, Keras, Matplot Lib, Numpy libraries and some of the sub packages of these libraries. Camera resolution is 1920*1080 and with fps of 40 (Default System Camera).

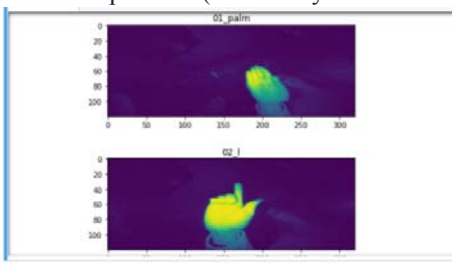

Fig. 6. These are some sample images of trained data that will represent the way we have converted the image into system understanding way.

While the device is ON and when the model is run. The model will open a tab of the camera which takes input from the user. The model is designed for the recognition and further working is done by the commands given to the system and how the user wants the gestures to make recognize. At the same time, the mouse pointer movement will be captured and made operate without any human interaction. The model working and implementation is shown in the figures.

A. Model Understanding

This is a sample image of how the system will convert the input into computer understandable images [11].

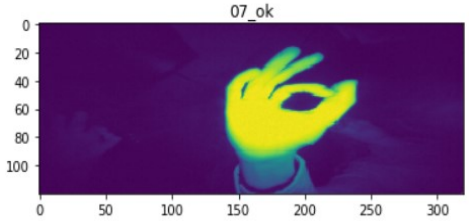

Fig. 7. OK Representation

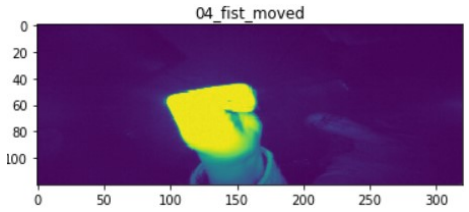

Fig. 8. Fist Moved Representation

So the images which are tained in the CNN using testing and training data is here.

IV. CNN Working

Convolution Neural Networks [16] are part of deep learning technologies [17] which is high level of machine understanding technique. So the neural networks are layered processing algorithm where it contains input layer, output layer and several middle layers. So this middle layer comprises of processing layers like convolution, pooling, recurrent, dropout, noise, normalization and many more.

So in the model which we are going to implement consist of some of these layers that would be processing images faster and loading the features based on the number of times the training is going on / number of samples that it's been training with. Keras [18] is the neural network library that will be imported to make CNN work on our system.

V. Results

The final performance of the hand gesture recognition system using deep learning models on the hand gesture recognition kaggle dataset. User uses data from the 10 folders provided in the dataset and will be checking for the accuracy of the model how it is being performed. The model created for the training is given in the Fig. 9.

```
In [15]: model=models.Sequential()
         model.add(layers.Conv2D(32, (5, 5), strides=(2, 2), activation='relu', input_shape=(120, 320,1)))
         model.add(layers.MaxPooling2D((2, 2)))
         model.add(layers.Conv2D(64, (3, 3), activation='relu'))
         model.add(layers.MaxPooling2D((2, 2)))
         model.add(layers.Conv2D(64, (3, 3), activation='relu'))
         model.add(layers.MaxPooling2D((2, 2)))
         model.add(layers.Flatten())
         model.add(layers.Dense(128, activation='relu'))
         model.add(layers.Dense(10, activation='softmax'))
```

Fig. 9. CNN created for the gesture processing and recognition

Fig. 10 shows how the CNN model is working during training. The users can apply all forms of inputs for the accuracy testing to the defined neural network (CNN) in order to prevent overfitting. Data pre-processing and cleaning will be playing as the key components to recognize and classify them accordingly.

```
In [17]: [loss, acc] = model.evaluate(x_test,y_test,verbose=1)
         print("Accuracy:" + str(acc))

         2000/2000 [==============================] - 21s 10ms/step
         Accuracy:1.0
```

Fig. 10. CNN model accuracy. This is available from the trained model from the images of training and testing set.

In the process of creating the model, user will be training our model with large data for better accuracy and for better prediction. For this, user will be using some CNN's so that the training will be more effective [13] [15].

Creating such models that will be creating a network which will be including dense layer, sequential layers and flatten layers. The model will be providing max-pooling for every convolution layers for conversion of image into computer understandable form.

After evaluation of the model and compilation of the model, user can see the accuracy of the model by using new data from the input provided. So when a user gives new input to the model it will be showing the output based on how it has been trained and the better the model is trained the performance will be. So if the batch size is increased along with a number of epochs for better performance so the user need to be careful with the device which they are selecting because large data needs more power which needs to be provided from powerful GPU's [12][14]. So the device selection is also a major part in the process of creating a neural network.

VI. Conclusion

From the models that are developed we can conclude that it is able to handle some hand gestures provided by any person and help us to identify what the gesture is. So the main point which we can look into is that the machine is able to understand on the images and is able to identify what the images are that is really helpful in many ways.

VII. Further Development

The development of this model can be created and that can be made work with more models and more examples and that models are given the possibilities to the model to convert the inputs and samples into operations that can be performed on the system. And the better the operations can be the more easily it will be to make the system accessible to everyone.

We can also develop a model with this as a base model where we can train the system to operate any game without using keyboard and mouse that would help in different situations.

References

[1] Donsong Ya, Yongjun Xie, et.al, (2018) Design on Computer System based on hand gesture recognition. 2018 IEEE15th international conference on networking, Sensing and control, pp 1-4.

[2] Prof. Quan Min Zhu, (2018) International Journal of Computer Applications in Technology, Vol 57, No 4. pp 1-16.

[3] Hong Dai, (2018). Research on SVM improves algorithm for large data classification. IEEE 3rd international conference on big data analytics, pp 181-185.

[4] M.Mernik, et.al, (2018). Journal of visual languages and computing. ISSN: 1045-926X.

[5] N.Paragios, (2018) Computer vision and understanding. Vol 116. pp 102-114.

[6] Guojen Wen, Zhiwei Tong, et.al, (2009), Man machine interaction in machining center. International workshop on intelligent systems and applications. pp 1-4.

[7] S.D. Bharkad, et.al. (2017). international conference on computing methodologies and communication, pp 1151-1155.

[8] Emile A. Hendriks, (2008), Sign Language Recognition by Combining Statistical DTW and Independent Classification. IEEE Transactions on pattern analysis and machine intelligence. Vol: 30, Issue: 11. pp 2040-2046.

[9] Ravi Prasad K. Jagannath, et.al, (2018), Optimized Laplacian Generalized Classifier Neural Network. 9th International Conference on Computing, Communication and Networking Technologies, pp 1-5.

[10] Gaurav Raj, et.al, (2016), Blur image detection using Laplacian operator and Open-CV. International Conference System Modeling & Advancement in Research Trends, pp 63-67.

[11] G.M.Foddy, et.al, (2008), Multiclass and Binary SVM Classification: Implications for Training and Classification Users. IEEE Geoscience and Remote Sensing Letters. Vol: 5, Issue: 2. pp 241-245.

[12] Minhua Yang, et.al, (2010), an improved random decision trees algorithm with application to land cover classification. 18th International Conference on Geoinformatics. pp 1-4.

[13] Dr. T. Prem Jacob, Mr. Karthick Kumar, (2017), Motion Activated Security Camera using Raspberry Pi. International Conference on Communication and Signal Processing, IEEE, pp.no. 1598 – 1601.

[14] Masataka Mito, Yuichiro Mori, et.al, (2018), A Single Filter CNN Performance for Basic Shape Classification. 9th International Conference on Awareness Science and Technology, pp 139-143.

[15] Jaejin Lee, Hyoungwook Nam, et.al, (2017), Performance analysis of CNN frameworks for GPUs. IEEE International Symposium on Performance Analysis of Systems and Software, pp 55-64.

[16] Convolution Neural Networks [Online]. Available: http://www.wikipedia.org/wiki/Convolution_neural_networks [Accessed: 12-Jan-2019].

[17] Deep Learning [Online]. Available: http://www.wikipedia.org/wiki/Deep_Learning [Accessed: 12-Jan-2019].

[18] Keras [Online]. Available: http://keras.io [Accessed: 12-Jan-2019].