Desktop Handling Using ASL

YUVRAJ G. RANE[1], GITESH V. SAGVEKAR[2], ROHIT R. PARAB[3], OMKAR D. DIKE[4]
[1, 2, 3] *Finolex Academy of Management and Technology, University of Mumbai.*
[4] *Assistant Professor, Finolex Academy of Management and Technology, University of Mumbai.*

Abstract- During the corona virus outbreak in 2020, the need of contactless ways to carry out day-to-day task to help reduce spreading of infection was highlighted. There are traditional ways of Human-computer interaction which use physical devices like keyboard or mouse. This paper focuses on the achievement of effective contactless human–computer interaction using only webcam and hand signs to navigate computer screen and perform various task. American Sign Language is a universal sign language used by the people with Hearing and Speech Disabilities for communication in their daily activities. It is completely vision-based communication language which provides standard set of hand signs. In this research paper, presented an optimal approach for human- computer interaction which uses ASL signs detected using Machine Learning algorithm for mouse navigation.

Indexed Terms- Dataset Generation, Feature extraction and Representation, CNN, Machine Learning, Keras, Human-Machine Interaction, GUI Application

I. INTRODUCTION

Desktop handling using American Sign Language is an important research problem for enabling contactless Human-machine interface. This project provides an efficient and correct way to enable contactless operation of computer using set of alphabets of American sign language. In this project, American Sign Language used by the user are captured and detected. The basic objective of this project is to develop a computer based intelligent system that will enable people to communicate with computer system using their natural hand gestures and perform general tasks such as navigating the mouse, switching tabs, clicking operation. The idea uses image processing, machine learning to capture and identify hand sign made by the user and uses it to take action assigned to it. Hence the objective of this project is to develop an intelligent system which can act as a communicator between the people and the computer system and can make the communication both effective and efficient. The objective can be further expanded to perform electronic devices functionalities.

II. LITERATURE SURVEY

Our project Desktop handling using ASL is based on hand gesture detection and classification using CNN Model to provide human machine interface. Various approaches used for recognition and classification of hand gestures are mentioned in this literature survey.

- Gesture recognition based on wearable devices:

In this approach, wearable devices ae used to provide information about hand sign to the computer. Wearable devices are robust and accurate hence the methods based on wearable devices are more accurate. The overall accuracy and speed in this approach is high, however this method is expensive as compared to other methods as it uses special hardware.

- Vision based gesture recognition approach:

The vision-based gesture recognition approach relies on camera and image features for identification and classification of hand gesture. The process usually involves segmentation, modelling, matching and recognition of image of user making hand gesture. camera and computer vision-based approach was used in [1] for hand gesture recognition. The approach uses image feature extraction to differentiate hand gesture captured. Convex hull and convexity defects algorithms from image processing were used to identify the hand sign. Space between adjacent fingers causes convexity depends which are used to count no. Of fingers in image. This approach is curently under development. [2] used Sobel's algorithm and hand gesture recognition (HGR) algorithm to identify hand gesture from given image. The method assumes stationary background to get better results. The approach used 12 MP iBall C12 camera for capturing colored images. The size of image was reduced to increase computation speed. Hand segmentation is done based YCbCr color space into YCbCr color plane. Also, Sobel's algorithm is used to extract region of Interest by finding the edge where gradient input binary image has maximum value. Hand gestures are used for performing various mouse activities such as left and right click. For each input image, corresponding centroid is calculated. Centroid moves based on movement of hand, which is used for controlling mouse navigation. Here, the efficiency of tracking the hand is improved by using red and blue colored caps on the fingers to make centroid looking more prominent.

The above approaches come under the application of image processing and computer vision. This type of approaches do not require any special type of hardware. However, accuracy of this approaches greatly depends upon the environment of the user making hand signs. The factors like lighting, Orientation of camera, presence of background affects the accuracy of model.

- Gesture recognition based on soft computing:

It is one of the recent approaches that are being used for hand gesture detection. It uses Machine learning based methods like neural network classifiers. [3] used approach which consists use of Hybrid dwt – Gabor filter to extract hand image from the input image. DWT provides highest frequency component of image. As Gabor filters give highest response at points where the texture of image changes, they are used for detecting edges. Different models used for classification based on SVM, Random Forest, KNN and CNN. The highest accuracy obtained was for CNN model which is 97%.

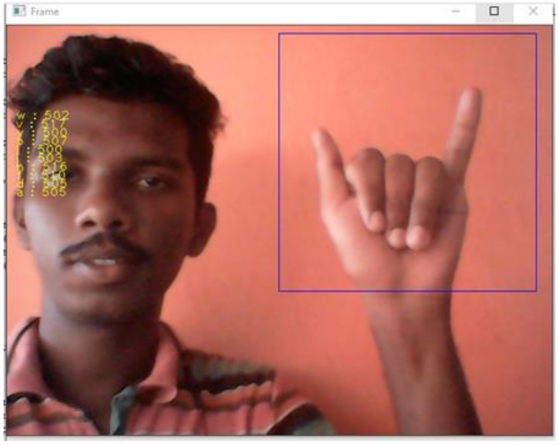
Morphological operations like Contour extraction and polygon approximations were used in approach of [4] for extraction of hand gesture from the input image. A CNN Model was trained with the help of back propagation to increase accuracy. The authors tried 4 different CNN model with different no. Of convolution and pooling layers. CNN model having 2 convolutional and 2 pooling layers achieved 94.7 % accuracy whereas 3 other models with 4, 7 and 9 convolutional layers resulted in accuracy over 96%. [5] used joint bilateral filtering from image processing to extract hand sign from input image as the dataset. Joint bilateral filtering is improved version of gaussian filtering. Also, depth image denoising method was used along with that to retain the edges in images. To enhance the images for training the model, near filling algorithm was used. Image classifier was developed using CNN. The Model was trained on 8 signs and Error back propagation and support vector machine were used to enhance accuracy of model. Recognition rate up to 98.52% was achieved. [6] used convolutional neural network (CNN) and stacked denoising autoencoder (SDAE). Hand gesture images with uniform dark background were used for training purpose. Hand signs made by user were extracted using hand segmentation algorithm. The hand segmentation algorithm tracks the boundry of white pixels in the image. 3 CNN models were trained with 2, 3 and 4 hidden layers respectively. 98.13 %, 96.32 %, 93.54 % accuracy was achieved respectively, calculated with the help of testing data set.

III. METHODOLOGY

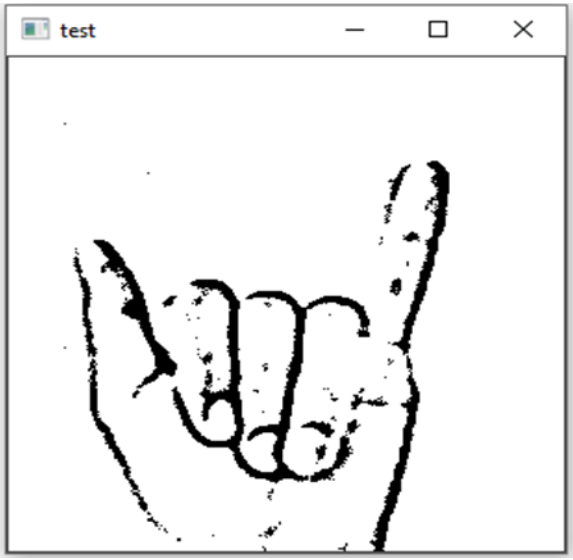
The system is a vision-based approach. All the signs are represented with hands gestures and so it reduces the burden of using extra electronic devices.

- Dataset Generation

The main obstacle in developing this project was getting the dataset as per our requirement. Due to inability of finding any dataset of American Sign Language as per our requirement, we generated our own dataset of 11 ASL signs consisting 'a', 'd', 'f', 'h', 'i', 'l', 'o', 'v', 'w', 'x', 'y' and one more 'blank' class. 2000 images for training and 1000 images for testing were captured for each ASL symbol. All the images were grayscale images of dimension 128 x 128 pixels. The low-resolution images were used as it reduces the overhead while training of CNN model and also provides better results than high resolution images.



Gaussian blur filter was used to extract features from the captured image. This was done with the help of OpenCV library in Python.



At the end we had the image dataset of 25000 images in training data set and 10000 images in testing dataset. Place figure captions below the figures; place table titles above the tables.

- Development of Machine learning model:

In this approach, we used Convolutional Neural Network (CNN) based classifier to classify the sign done by user to one of the 12 signs available which are 'a', 'd', 'f', 'h', 'i', 'l', 'o', 'v', 'w', 'x', 'y', 'blank'.

Details of Layers used in CNN model are as follows

1. 1st Convolution Layer:

The images used in Dataset have resolution of 128x128 pixels. It acts as input to the first layer. First layer is convolutional layer with 32 filter weights and stride of size 3x3 pixels. This will result in a 126X126 pixel image, one for each filter-weights.

2. 1st Pooling Layer:

Max pooling is used in this layer, with stride 2x2 size. It takes the maximum value present in that stride. This is done to reduce the dimensionality of image. The images obtained as output of this layer has dimensions 63x63 pixels.

3. 2nd Convolution Layer:

Output of 1st pooling layer is then passed through a convolutional layer with filter weight of 32 and stride of 3x3. This gives us image of size 60 x 60 pixels as output.

4. 2nd Pooling Layer:

The resulting images are down sampled again using max pool of 2x2 and 30 x 30 resolution of images are obtained as output.

5. 1st Densely Connected Layer:

These images are fed to a Fully connected layer consisting of 128 neurons. The input to this layer is an array of 30x30x32 = 28800 values. For this layer dropout value 0.5 was used to avoid overfitting. Output of this layer is given as input to second densely connected layer.

6. 2nd Densely Connected Layer:

Now the output from the 1st Densely Connected Layer is used as an input to a fully connected layer with 96 neurons. Dropout value of 0.5 is used in this layer as well.

7. Final layer:

Final layer has no. of neurons equal to no. of classes (ASL symbols used) present i.e., 12. SoftMax function was used as an activation function to normalize the output between 0 and 1 to use as probability.

- Activation Function

We have used Rectified Linear unit (ReLU) function as activation function. It returns maximum value between 0 and passed input. Thus, it returns 0 for all the negative values. This switches off the certain neurons in the network which provide negative values, making the model non-linear and better for learning complex features. It also speeds up the training process by removing the problem of vanishing gradients and thus reducing the computational time.

- Pooling Layer

We reduce the number of parameters by adding max pooling layer of pool size (2,2) with ReLU activation function. This results in lessening the computation cost and reduces overfitting.

- Dropout Layer

This layer randomly “drops off” certain number of neurons in layer by turning them to 0 during model training. This helps in reducing overfitting of the model on training dataset.

- Optimizer

Adam optimizer was used for optimizing the model in response to the values given by loss function, in this case, categorical cross entropy. Adam provides the advantages of adaptive gradient algorithm (ADA GRAD) and root mean square propagation (RMSProp), two extended versions of stochastic gradient descent algorithm.

- Training and Testing

Grayscale images of size 128 x 128 pixels of user making ASL sign are used for training and validation purpose. Gaussian blur filter is applied to those images to reduce the noise and focus on Region of Interest.

After developing the model as specified above, we feed this image dataset to model for training and validation.

The output of model is result set of 12 values. Each value represents how likely the images falls under that category. We normalize this values in the range of 0 to 1 using SoftMax as the activation function.

The first output of model will differ than the actual output expected because of randomly initialized weights and biases. To increase the accuracy of model we train it on the labeled dataset. We use categorical cross entropy as loss function, which gives output as zero when the predicted output is equal to the expected output, otherwise it gives non-zero positive value as an output. Thus, we increase performance of model by minimizing the cross entropy. We do this by optimizing weights by using gradient descent method. We use Adam optimizer which is best gradient descent optimizer for reducing cross entropy.

- Developing GUI Application

We developed a Graphical User Interface (GUI) application using Tkinter Library in Python. The model trained in the previous steps is loaded from the memory. The application uses openCV library to capture the user image making hand sign. The Camera feed is shown in the GUI as well. A 300 x 300 sized square is highlighted at top right corner of the camera feed. This is our Region of Interest. This RoI is captured, resized to size 128 x 128 and fed to the model. The CNN model, then, classifies this image into one of the 12 classes as 'a', 'd', 'f', 'h', 'i', 'l', 'o', 'v', 'w', 'x', 'y', 'blank'.

Based on the symbol that is predicted, the Mouse navigation action is assigned to that symbol is taken. Actions taken were 'a':'scrollUp' , 'd':'MouseUp' , 'f':'Click' , 'h':'MouseLeft' , 'i':'enterKey' , 'l':'MouseRight' , 'o':'DoubleClick' , 'y':'MouseDown' , 'v':'scrollDown' , 'w':'showTab' , 'x':'shiftTab' , 'blank':'Idle Mouse'. This mouse navigations and functions are achieved with the help of “pyautogui” library which provides various methods to interface with cursor and navigation.

IV. RESULTS

We achieved 93.76% accuracy in our trained model. The model was implemented in the GUI application and was used for prediction of ASL symbol made by the user and to interface with the computer using mouse navigations.

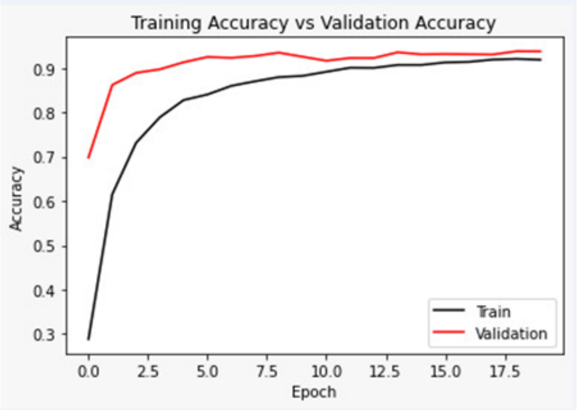
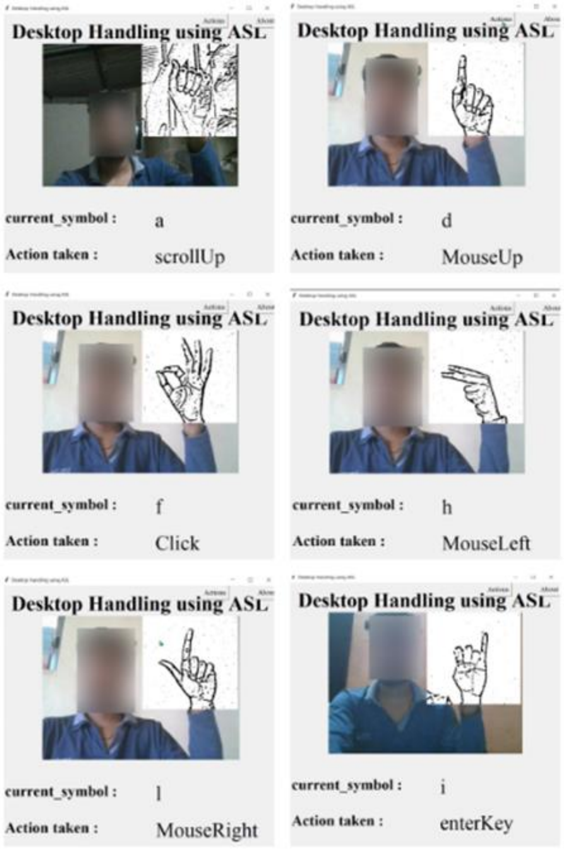


Fig a) Graph showing training vs validation accuracy

The model successfully predicts label of the ASL symbol made by the user. Based on the ASL symbols detected, actions corresponding to respective symbols were performed.

Following are the screenshots of Desktop handling using ASL application successfully capturing the user making ASL hand sign and detecting the label of that hand sign and performing the action assigned to it.



Desktop Handling using ASL
current_symbol : a
Action taken : scrollUp

Desktop Handling using ASL
current_symbol : d
Action taken : MouseUp

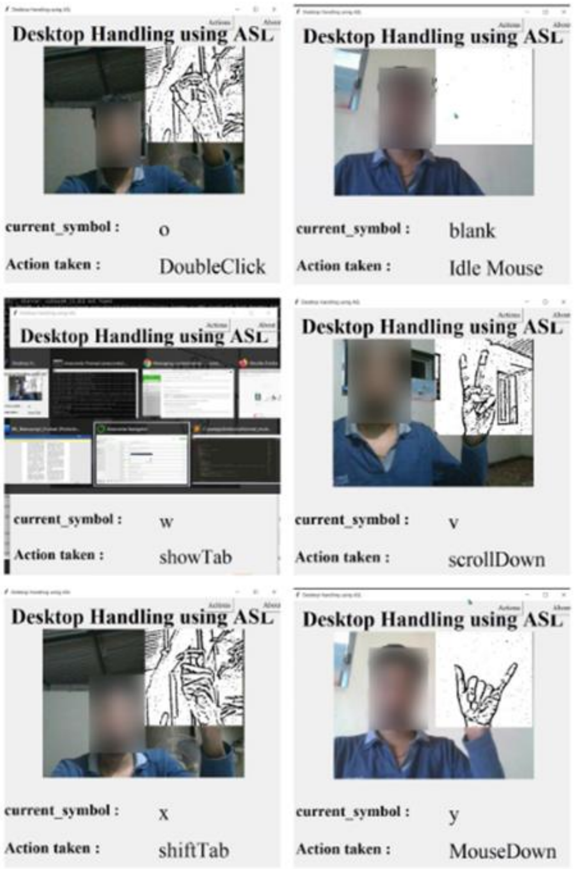
Desktop Handling using ASL
current_symbol : f
Action taken : Click

Desktop Handling using ASL
current_symbol : h
Action taken : MouseLeft

Desktop Handling using ASL
current_symbol : l
Action taken : MouseRight

Desktop Handling using ASL
current_symbol : i
Action taken : enterKey

Fig. b) Frontend for GUI application (1)



Desktop Handling using ASL
current_symbol : o
Action taken : DoubleClick

Desktop Handling using ASL
current_symbol : blank
Action taken : Idle Mouse

Desktop Handling using ASL
current_symbol : w
Action taken : showTab

Desktop Handling using ASL
current_symbol : v
Action taken : scrollDown

Desktop Handling using ASL
current_symbol : x
Action taken : shiftTab

Desktop Handling using ASL
current_symbol : y
Action taken : MouseDown

Fig. c) Frontend for GUI application (2)

V. MOTIVATION

Nowadays the most popular receivers are the computers and because of COVID-19 contactless interaction is getting more importance. Deaf people communicate among themselves using sign language but when it comes to desktop handling, it becomes difficult to them. Our project aims at taking the basic step to overcome the communication gap between deaf people and Computer System using American sign language. Effective extension of this project to words and common expressions may not only make the deaf people communicate faster and easier with computers, but also provide a boost in developing autonomous systems for understanding and aiding them.

CONCLUSION

There are various methods available for human-computer interactions, but most of these methods need user to physically handle the device. Amidst the time of pandemic an effective method for contactless human-computer interface was necessary. The main focus of this project was to enable user to operate desktop with the help of ASL symbols. An effective method for contactless human-computer interaction is provided using this project. Hopefully, this Desktop handling using ASL project will serve as guideline for new methods of Human-computer interaction.

REFERENCES

[1] A. Pradhan and B. B. V. L. Deepak, "Obtaining hand gesture parameters using image processing,"2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Chennai, 2015, pp. 168-170, 2015.

[2] V. Bhame, R. Sreemathy and H. Dhumal, "Vision based hand gesture recognition using eccentric approach for human computer interaction,"2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 949-953,2014.

[3] V. Ranga, N. Yadav, and P. Garg, “American sign language fingerspelling using hybrid discrete wavelet transform-gabor filter and convolutional neural network,” Journal of Engineering Science and Technology, vol. 13, no. 9, pp. 2655–2669, 2018.

[4] Raimundo F. Pinto, Carlos D. B. Borges, Antônio M. A. Almeida, Iális C. Paula, "Static Hand Gesture Recognition Based on Convolutional Neural Networks", Journal of Electrical and Computer Engineering, vol. 2019, Article ID 4167890, 12 pages, 2019. https://doi.org/10.1155/2019/4167890

[5] Li, G., Tang, H., Sun, Y. et al. Hand gesture recognition based on convolution neural network. Cluster Comput 22, 2719–2729 (2019). https://doi.org/10.1007/s10586-017-1435-x

[6] Oyedotun, Oyebade & Khashman, Adnan. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications. 28. 10.1007/s00521-016-2294-8.