2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)
Galgotias University, Greater Noida, UP, India. Oct 30-31, 2020

Models for Hand Gesture Recognition using Deep Learning

Manasi Agrawal
Department of Computer Engineering
MIT College of Engineering
Pune, India
Email: manasi1211@gmail.com

Rutuja Ainapure
Department of Computer Engineering
MIT College of Engineering
Pune, India
Email: ainapurer7@gmail.com

Shrushti Agrawal
Department of Computer Engineering
MIT College of Engineering
Pune, India
Email: shrush10agrawal@gmail.com

Simran Bhosale
Department of Computer Engineering
MIT College of Engineering
Pune, India
Email: simranbhosale123@gmail.com

Dr. Sharmishta Desai
School of CET
MIT World Peace University
Pune, India
Email: sharmishta.desai@mitwpu.edu.in

***Abstract*—According to the World Health Organization, 5% of the world population, approximately 466 million people, is deaf and/or mute or has disabling hearing loss. There is often a wall of distinction between handicapped people and normal people. We communicate to share our thoughts but for a disabled person (mainly deaf and dumb), it becomes difficult to communicate. Inability to speak is considered to be a true form of disability. For such people, sign language or Braille is the only means of communication. Sign Language is a way of communication using hand gestures. However, it becomes difficult for them to communicate with others as most don't understand sign language. Hence, we aim at bridging this communication gap, between a deaf/mute person and others, by developing a system which acts as a mediator between both. We propose a hand gesture recognition system which works in 4 steps: 1. Generate a live stream of hand gestures using web-cam. 2. Form images from the video using video frames. 3. Preprocess these images. 4. Recognize sign language hand-gestures and convert into text/audio output. The system is implemented using the concepts of image processing and neural networks. We have tested the proposed models using Kaggle dataset, our dataset and a dataset formed after combining both. We propose to eliminate the ambiguity introduced in the results by inculcating variation in the background. Most of the models give similar accuracy of the test results for both plain and cluttered background.**

***Index Terms*—Gesture Recognition, Image Processing, Sign Language Interpretation, Neural Network.**

I. Introduction

Sign languages conveyed using hand gestures is used by deaf and dumb people as it provides a way of communication with the world around them. As a majority of the deaf-mute people are born to normal parents, they have to undertake grave efforts to learn sign language. All the more, their family members also have to take the efforts of learning sign language. Thus, sign language is irreplaceable. However, ordinary people will never inflict themselves with the pain of learning sign language. An ordinary person will hardly ever feel the necessity of interacting with a deaf and mute person or try to communicate with them considering the communication barrier.

In addition to this, the world is directly or indirectly very unfair to the differently-abled people. Every day they fight a new battle just to survive in this fast world. It takes great courage and continuous support and motivation to lead such a hard and difficult life. We noticed that the major problem faced by differently-abled people is their dependence on their family members or mentors. Thus, they face unfair rejections multiple times, one of the example being employment.

We came across an article which claimed that deaf and mute people who had cleared civil services exams and were thus eligible for the job in the government sector were denied their positions only due to their disability. We believe that changing one disabled person's life will change all the lives around him/her, thus changing thousands of lives at a time. With the advancement in medical science and technology, it is now possible to overcome some of the disabilities. For instance, prosthetic arms and legs can aid an amputee and increasing awareness about organ donations can save lives. However, one cannot overcome from being mute.

Hence, to help such people and ease their communication with others so that they can have some normalcy in their lives, we realized the need to bridge this communication gap. This suggests a need for a sign language recognition system which will enable them to overcome their difficulties in everyday life to some extent.

In this work, we aim to interpret sign language hand gestures from a live stream captured by the webcam using CNN (Convolutional neural network) which is a type of deep neural network and is most commonly used for visual imagery. We are using one of the well-known models for transfer learning - Inception V3, which is a CNN model and is 48 layers deep. Transfer learning helps to retain the knowledge that the model has learnt during its original training & allows users to define their own layers and retrain a couple of existing

978-1-7281-6324-6/20/$31.00 ©2020 IEEE

layers and apply it on their own smaller datasets. The other model that we used does not use a base model & parameters were also tuned differently in each training.

II. RELATED WORKS

Literature Review reveals that there are various systems developed to work as a sign language translator. Kshitij Bantupalli and Ying Xie [1] have developed a sign language translator using CNN model named Inception to extract spatial features from video stream. They used a Long Short-Term Memory(LSTM) [6], a Recurrent Neural Network(RNN) model to extract temporal features from the video series. The model dropped accuracy when tested with different skin tones, the inclusion of faces in the video and variation in the clothing of the subject. Another real-time system [2] developed by Murat Taskiran, Mehmet Killioglu, and Nihan Kahraman for ASL recognition used CNN for feature extraction and as a classifier; convex hull algorithm and skin color detection for hand position determination. However, only a uniform black background was used to conduct experiments and thus the real-life problem of background clutter was ignored. Soeb Hussain and Rupal Saxena [3], proposed a hand gesture recognition system to command a computer using static and dynamic hand movements. They used a CNN based classifier trained using transfer learning over VGG16. [4] HSV skin color algorithm is used to segment the hand area. Tracing of hand for dynamic gestures is a time-consuming process. However, they have improved accuracy in comparison with CNN architecture. Neel Kamal Bhagat, Vishnusai Y, Rathna G N [5] have developed a model using deep learning and Image processing to identify ISL hand gestures in real-time. The model gives good performance for hands of any size and different background conditions but leaves scope for further research for achieving real-time performance. Siming He [6] has developed a sign language translator model using deep learning. He described the design of the hand locating algorithm based on deep learning. He used 3D CNN along with the recognition algorithm grounded on recurrent neural network-LSTM for feature extraction. The data set is restricted in scope and does not include all the sign language words and yet this model gives high accuracy. Rangel Daroya, Daryl Peralta, and Prospero Naval, Jr [7] proposed a method that was based on DenseNet using deep learning. It was concluded that some letters faced trouble in recognition due to the resemblance in their appearances.

III. METHODOLOGY

A. Dataset Preparation

Sign Language MNIST dataset from Kaggle is modified using a set of our own images. The MNIST dataset has 3000 images for gestures of each letter. We randomly selected 750 images from the dataset for each letter. We then added 50 more images per letter to this dataset, captured in varied lighting conditions modelled by different subjects. This gave us a total of 800 images of each letter. Furthermore, Image Data Augmentation is performed to further improve the variety in our dataset. Our augmentation included techniques like a random rotation about 20 degrees, height & width shift, zoom, brightness and horizontal flip. We have chosen two different subsets of our dataset for experimentation purpose. In one subset, there are only 6 letters: A, B, C, I, L, P. In the second subset, there are a total of 13 letters. These include A, B, C, L, M, O, P, Q, S, U, W, Y, Z. Both these subsets were randomly divided into a training set and validation set in the ratio of 8:2.

B. Training the Models

Deep neural networks have a branch called convolutional neural networks(CNN) which are used to interpret visual imagery. These networks are named so because of its similarity in connectivity pattern with the animal neural systems. The striking feature of CNN compared to its ancestors is that it automatically detects the relevant features without requiring any human interference. For example, if given pictures of cats and dogs, it learns the important features by itself and classifies the image into a cat or dog accurately up to a certain percentage.

CNN has several hidden layers between its input and output layer. The hidden layers of a CNN typically consist of a sequence of convolutional layers that convolve with multiplication or other dot product. RELU, one of the most common activation layers, is applied followed by additional processing layers such as pooling layers, dropout layers, fully connected layers and normalization layers which are referred to as hidden layers; since the activation function as well as final convolution masks the inputs and outputs. We have trained two different types of models - one which uses the concept of transfer learning and the second one where a base model isn't required.

1) No Base Model: It takes an input frame of size 200x200. The first convolution layer has a filter of 3x3. It is followed by a max-pooling 2D layer of 2x2 filter. These two together form one layer and a few such layers are stacked together. Dropout layer and Batch Normalization layer are added alternately to reduce overfitting of the model. The Dropout layer eliminates some of the trained neurons, thereby reducing total trained neurons. For implementing stochastic gradient descent and effective noise handling, we used Adam optimizer for this model.

2) Using Transfer Learning: This concept allows us to retrain the terminal layer of an existing model, resulting in a notable drop in not only the training period but also the volume of the dataset required. We used one of the well-known and efficient models for transfer learning, Inception V3, which is 48 layers deep. We retrained the final layer, which allowed us to preserve the knowledge that the model had acquired during its primary training and implement it on our smaller dataset, resulting in much more precise classifications without the necessity for extensive training and investment of computational power.

The input shape of data was fixed to 150x150. We defined some layers on top of the Inception v3 to suit our requirements and the include_top argument was set to “false”, meaning that the fully connected layer at the top was excluded. The “mixed7” layer of the InceptionV3 was chosen as the last layer and we built our layers on top of it(Dropout and Dense layers).

Both our models are trained using both subsets of our entire dataset and for a few different epochs. Since the output is categorical, the softmax activation function is suitable to our requirements for the last dense layer. All training and validation accuracies were recorded along with the graphs of accuracy & loss during training and validation and then were compared to see which model gives better results.

IV. EXPERIMENTAL SETUP & OUTCOMES

A. Testing

For testing static images, we compiled a small dataset other than the modified Kaggle dataset of hand gestures. The images are converted to specific dimensions according to the model requirement. The trained models are loaded and applied to these images to obtain a label for every image. This label is then converted to a letter associated with it.

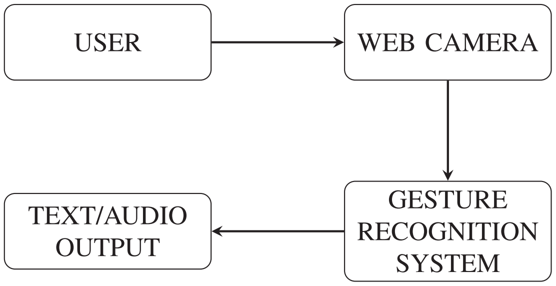

Fig. 1. Overview of System Architecture

Fig. 1. shows the outline of the system architecture.

For video testing, the video is captured from the webcam and broken down into relevant image frames. These image frames undergo preprocessing to match the dataset used for training.

TABLE I
USING INCEPTION V3 AS THE BASE MODEL - ACCURACY TABLE

Sr.No	No. of Letters	No. of Epochs	Training Accuracy	Validation Acuuracy	Image Testing Accuracy	Video Testing Accuracy on Plain Background	Video testing Accuracy on Cluttered Background
1	6	20	81.91%	79.36%	82.35%	37.5%	40%
2	6	30	82.59%	80.34%	35.29%	37.5%	40%
3	6	50	82.48%	88.67%	82.35%	75%	65%
4	13	15	66.53%	62.97%	48.14%	35%	30%
5	13	25	66.78%	61.44%	48.14%	47.5%	45%
6	13	35	68.6%	61.41%	48.14%	31.66%	28.33%

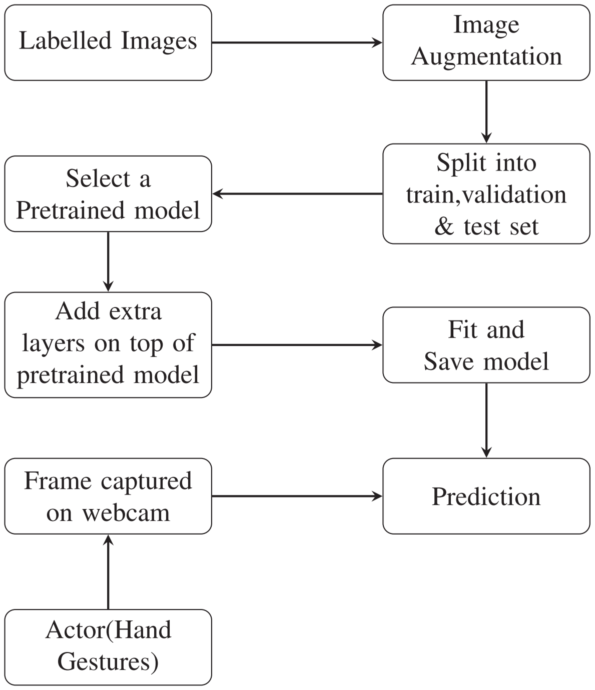
Labelled Images
Image Augmentation
Split into train,validation & test set
Select a Pretrained model
Add extra layers on top of pretrained model
Fit and Save model
Prediction
Frame captured on webcam
Actor(Hand Gestures)

Fig. 2. System Flow Diagram

Images are resized to meet the purpose since captured frames from the video may have different sizes. The frames tested on the trained model generate a label which is then converted to the letter associated with it.

Further, the converted letters are passed as a string and converted to audio using the google-text-to-speech library in the English language. The **testing accuracy** is obtained by dividing the number of correctly identified images by the model by the total number of images it is tested on. Fig. 2. shows the detailed working and data flow of our system.

B. Experimental Results

1) For InceptionV3 Model: The result table (Table I) implies that the models give similar accuracies even if we introduce variation in the background. As epochs increase, the number of accurately recognized letters also increase. For example –

For the Inception V3 Models using 13 letters dataset –

For 15 epochs – B C O M S were recognized. For 25 epochs – B C L M O P Q W Y were recognized

For the Inception V3 Models using 6 letters dataset –

For 20 epochs – A B C L were recognized but a lag was experienced.

For 50 epochs – A B C L I were recognized relatively quicker.

The difference between the video testing results of the above two examples clearly show the effect the number of epochs have on the accuracy and speed of the recognition of gestures.

TABLE II
USING NO BASE MODEL - ACCURACY TABLE

Sr.No	No. of Letters	No. of Epochs	Training Accuracy	Validation Acuuracy	Image Testing Accuracy	Video Testing Accuracy on Plain Background	Video testing Accuracy on Cluttered Background
1	6	20	65.18%	68.68%	29.41%	15%	15%
2	6	35	67.45%	69.01%	23.52%	25%	15%
3	6	50	68.52%	67.97%	23.52%	7.5%	7.5%
4	6	100	68.8%	69.6%	29.41%	8.33%	17.5%
5	13	20	49.14%	52.21%	14.81%	7.69%	3.8%
6	13	35	52.45%	52.21%	7.4%	2.5%	2.5%
7	13	50	61.07%	69.18%	3.7%	10%	10%

2) Using No Base Model: Table II shows the models which were trained for different number of epochs which gave similar accuracies even after we introduced variation in the background. The testing accuracies are quite low due to less number of layers. Even after training on 100 epochs, the testing accuracy did not change much.

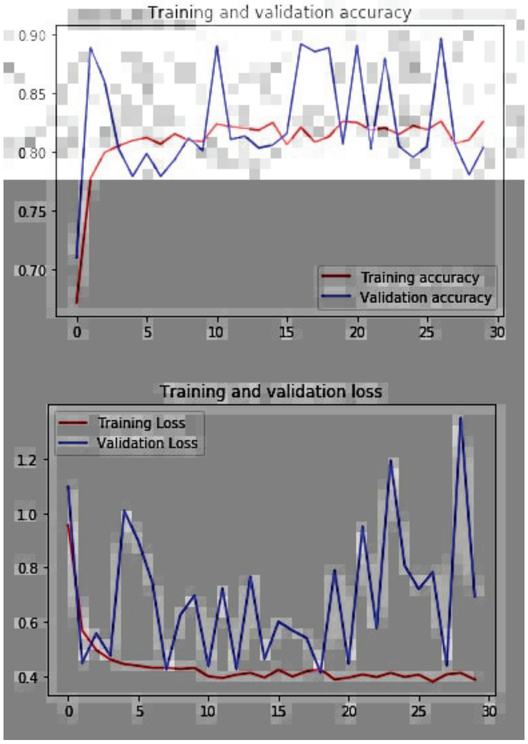

Fig. 3. Inception V3 - 30 epochs for 6 letters

C. Graphical Results

In Fig. 3., The first graph represents the training and validation accuracy of the model for each of the 30 epochs trained on 6 letters dataset. The second part of the graph shows the training & validation loss of the model for each of the 30 epochs trained on 6 letters. It can be observed from the graph that the validation accuracy and loss is not stable and fluctuates constantly over a wide range. Similarly, Fig. 4. shows the plots of training & validation accuracies and losses over the course of 50 epochs for 6 letters. Both figures represent the Inception V3 model.

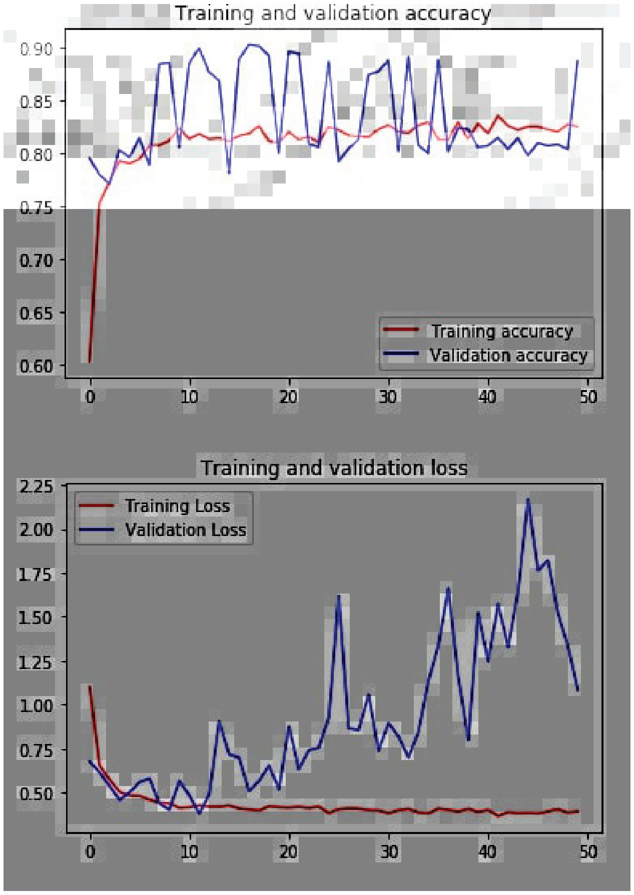

Fig. 4. Inception V3 - 50 epochs for 6 letters

Fig.5. shows the plots for training & validation accuracy and loss for the "No Base Model". This model was trained for 100 epochs on the 6 letters dataset. Fig. 6. shows the plots of the model trained for 50 epochs on 13 letters dataset.

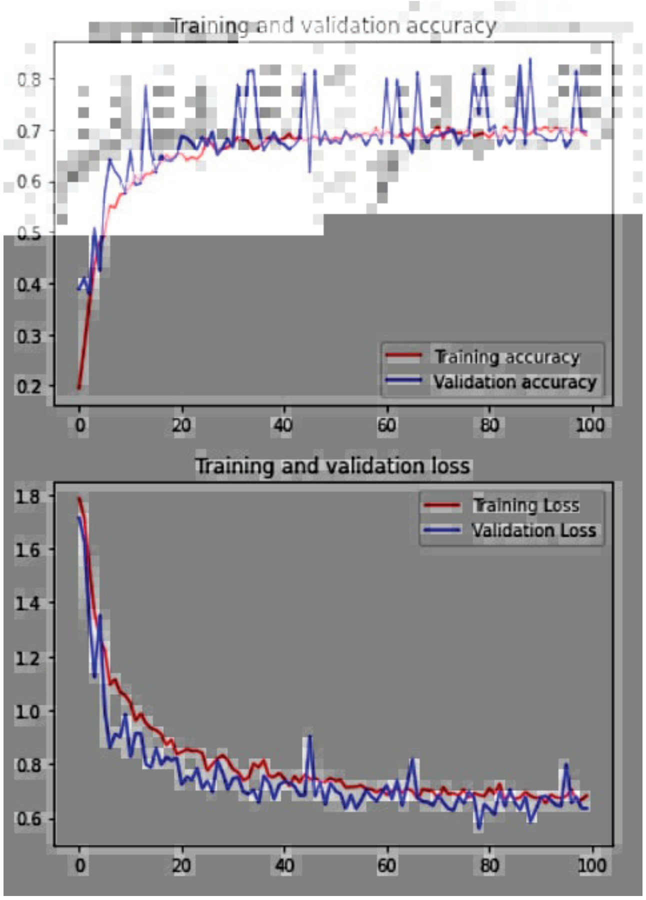

Fig. 5. No Base Model - 100 epochs for 6 letters

V. CONCLUSION

After comparing the results, we noticed that the models usually work well with lesser letters. When more gestures are added, it increases ambiguity and thus the model doesn't respond well. The accuracy increases up to a certain peak value and then overfitting is observed as we go on increasing the epochs. The model which works well for a few letters doesn't work the same for a larger dataset. So, as we add more letters to our dataset, we need to increase the layers in the model as well. The models trained with Inception V3 show higher accuracies since the model has several layers (48) as compared to the second model which has only 4-5 layers at the most. Even with parameter tuning and data augmentation, the accuracy doesn't change favorably.

VI. FUTURE WORK

- Modifying and extending the dataset to cater recognition of all the English alphabets.
- Implementing video processing to recognize gestures of letters J and Z; since it cannot be done using static image frames.

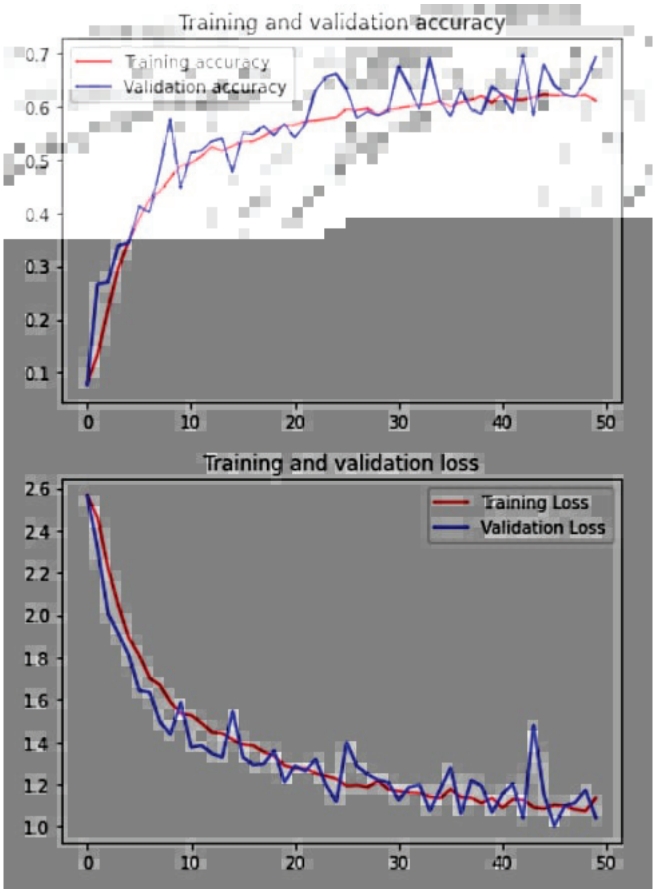

Fig. 6. No Base Model - 50 epochs for 13 letters

- Increasing the speed of gesture to text or audio conversion and modifying the dataset to aid gesture recognition of various types of sign languages differing regionally.
- Improving/maintaining accuracy.
- Addition of a module providing a reverse conversion - text to a pictorial representation of sign language alphabet gesture.
- Inclusion of gesture-controlled home automation module with remote mobile connectivity and security alarm.

ACKNOWLEDGMENT

We would like to thank our mentor, Dr. Sharmishta Desai for helping us throughout the process. We would also like to thank the faculty at MITCOE whom we consulted time and again and the staff at MITCOE who allowed us to use the college resources to the full potential. It would not have been possible to complete this research without the support we got in the way.

REFERENCES

[1] K. Bantupalli and Y. Xie, ”American Sign Language Recognition using Deep Learning and Computer Vision,” 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4896-4899.

[2] M. Taskiran, M. Killioglu and N. Kahraman, ”A Real-Time System for Recognition of American Sign Language by using Deep Learning,” 2018 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, 2018, pp. 1-5.

[3] S. Hussain, R. Saxena, X. Han, J. A. Khan and H. Shin, ”Hand gesture recognition using deep learning,” 2017 International SoC Design Conference (ISOCC), Seoul, 2017, pp. 48-49.

[4] Karen Simonyan, Andrew Zisserman, “Very deep convolutional network for large scale image recognition”, ICLR (Internaltional Conference on Learning Representations), 2015.

[5] N. K. Bhagat, Y. Vishnusai and G. N. Rathna, ”Indian Sign Language Gesture Recognition using Image Processing and Deep Learning,” 2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 2019, pp. 1-8.

[6] S. He, ”Research of a Sign Language Translation System Based on Deep Learning,” 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), Dublin, Ireland, 2019, pp. 392-396

[7] R. Daroya, D. Peralta and P. Naval, ”Alphabet Sign Language Image Classification Using Deep Learning,” TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), 2018, pp. 0646-0650.

[8] Ms. Swati S. Avhad, Ms. Sharmishta Desai, ”A Research Review on Indian Sign Language Recognition using Machine Learning,” JCSEST, 2020, Vol. 6, No 2

[9] Sharmishta Desai, ”Dynamic Induction Model for Students Behavior Analysis”, IJSSE, 2019, Vol. 7, Issue 1

[10] Sharmishta Desai and S.T. Patil, 2018. Boosting Decision Trees for Prediction of Market Trends. Journal of Engineering and Applied Sciences, 13: 552-556.

[11] Nidhi Kela, Sharmishta Desai, ”COMBINING GENETIC ALGORITHM WITH OTHER MACHINE LEARNING ALGORITHM FOR CHARACTER RECOGNITION”, IJCEA, 2018

[12] Sharmishta Desai, Kishanprasad Gunale, Utsav Pataskar, Saurabh Ghatnekar, Sanjiv Gupta, Swapnil Sawant, ”DRONE ASSISTED DEVELOPMENT OF AGRICULTURE (D.A.D.A.): A SURVEY”, IJCEA, 2018

[13] Desai, Sharmishta Patil, S.. (2017). Big Data Classification Using Distributed Optimized Hoeffding Trees. Journal of Machine Intelligence. 2. 14-20. 10.21174/jomi.v2i1.101.

[14] Alabhya Farkiya, Prashant Saini, Shubham Sinha , Sharmishta Desai, ”Natural Language Processing using NLTK and WordNet”, IJCSIT, Vol. 6 (6) , 2015, 5465-5469

[15] K. Wadewale, S. Desai, ”Survey on Method of Drift Detection and Classification for time varying data set”, IRJET, Volume 02, Issue 09, 2015

[16] , S. Desai and S. T. Patil, ”Efficient regression algorithms for classification of social media data,” 2015 International Conference on Pervasive Computing (ICPC), Pune, 2015, pp. 1-5, doi: 10.1109/PERVASIVE.2015.7087040.

[17] S. Desai, S. Roy, B. Patel, S. Purandare and M. Kucheria, ”Very Fast Decision Tree (VFDT) algorithm on Hadoop,” 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, 2016, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860037.