Recognizing Hand Gesture of American Sign Language using Machine Learning

Nikhataara Jakati[1], Sahana Desai[1], Pavana Gatnur[1], Kshama Desai[1], Sharada S[2]

[1]Student, Computer Science Department, Jain College of Engineering and Technology, Hubballi, Karnataka, India - 580032
[2]Professor, Computer Science Department, Jain College of Engineering and Technology, Hubballi, Karnataka, India - 580032

---***---

Abstract - *Millions of people around the world suffer from hearing disability. This large number demonstrates the importance of developing a sign language recognition system converting sign language to text for sign language to become clearer to understand without a translator. Neural Network Algorithm is proposed based on American Sign Language. The neural network of this system used extracted image features as input and it was trained using softmax and ReLU algorithm to recognize which letter was the given letter with the average accuracy of 96%.*

***Key Words*:** Hand gesture recognition; artificial neural network; multilayer perceptron; softmax; American sign language(ASL); finger spelling

1. INTRODUCTION

Over 5% of the world population, which means 360 million people, including 32 million children and 328 million adults, has hearing disability according to World Health Organization (WHO) statistics [1]. Hearing impaired people generally use sign languages for communicating with other people. But mostly hearing people do not know sign language. When considering large number of people who suffer from hearing disability, it is revealed how important providing them opportunity to communicate with hearing people who do not have knowledge of sign language

A need to develop such a sign language recognition system arises day by day. The important key points of such a sign language system are reducing cost and obtaining more accurate rate efficiently. Developing a sign language system based on machine learning for automatically recognition sign language and converting sign language to text helps hearing people to communicate and understand hearing impaired people. The proposed system uses the images in the local system or the frame captured from webcam camera as input. Processed input image is given to the classifiers which use Convolution Neural Network Algorithm. It classifies the image and converts into model. Finally the predicted result is produced as text.
In this below figure, converting sign language to text by an automated sign language recognition system based on machine learning is proposed to satisfy this need. Convolution Neural Network Algorithm is used for the system.

2. LITERATURE SURVEY

Chuan-Kai Yang, Quoc-Viet Tran and Vi N.T. Truong have proposed a system recognizing static hand signs of alphabets in American Sign Language from live videos and translating into text and speech. AdaBoost and Haarlike classifiers have been used for the classification during training process. After the training process, the classifier can recognize different hand postures. Process of testing the system consists of three stages: preprocessing stage, classification stage and text to speech stage. In “Preprocessing Stage”, frames from the video stream are extracted and methods of image processing are used to obtain the features from the image. In “Classification Stage”, the processed images in the preprocessing stage are used as input and classification is done by using Haar Cascade Algorithm. In “Text To Speech Stage”, text recognized by the classifier is converted to speech by using SAPI 5.3. Performance measures of the system are: 98.7% precision, 98.7% recall, 98.7% sensitivity, 99.9% specificity and 98.7% F-score. [2]

Yi Li has proposed Hand Gesture Recognition System using on Microsoft Kinect for Xbox. The system is built on Candescent NUI project and uses Open NI framework for data extraction from the Kinect sensor. There are three main processes in the proposed system: Hand Detection, Finger Identification, and Gesture Recognition. In “Hand Detection” process, firstly hands are separated from background by using depth information. Then two clusters of hand pixels are obtained by using K-means Clustering Algorithm to be able to detect hand by merging two clusters. Afterwards, convex hulls of hands are determined by using Graham Scan Algorithm and detection of hand contours is done by using contour tracing algorithm. When a single hand gesture is used, system gives accuracy which varies from 84% to 99% . When same gesture is performed with both hands, accuracy varies from 90% to 100%.[3]

Anis Diyana Rosli, Adi Izhar Che Ani, Mohd Hussaini Abbas, Rohaiza Baharudin, and Mohd Firdaus Abdullah have proposed a spelling glove work recognizing the letters of American Sign Language alphabet. The system has been designed targeting deaf-mute people to communicate with normal people. After Microcontroller finds the combination position of each finger in library, LCD displays correct alphabet. Recognition rate of the system is 70%.[4]

Md. Mohiminul Islam, Sarah Siddiqua and Jawata Afnan have proposed another Hand Gesture Recognition study based on American Sign Language. The system works in four steps for gesture recognition including image acquisition, preprocessing, feature extraction and feature recognition. In "Image acquisition" step, a database of 1850 images of 37 signs is created by collecting image samples of each sign of the sign language from different people. "Preprocessing" step prepares the image received from camera for feature extraction step by removing noise and cropping image to obtain portion from wrist to fingers of a hand for sign detection. "Feature extraction" step applies different algorithms for feature extraction of hand gesture recognition system including K convex hull for fingertip detection, eccentricity, elongatedness, pixel segmentation and rotation. Artificial Neural Network, Softmax and relu Algorithm is used for training. Gesture recognition rate of the system is 96%. [5]

Jun-Wei Hsieh, Teng-Hui Tseng, Wan-Yi Yeh and ChunMing Tsai have proposed a sign language recognition system in order to detect English letters and numbers. The color data, skeleton data and depth data which are obtained from the input Kinect are used for detecting palm area of hands. Then Otsu thresholding method is used for extracting palm and morphology closing operation is used for closing the holes in the palms. Then SURF descriptors and features are extracted. Finally, Brute-force and SVM are used for recognition the letters and numbers in the sign language. The accuracy rate obtained by classification of the numbers and letters with SVM is 100%. The alphabet is also trained by SVM with a recognition rate of 70.59%. [6]

M. Deriche, S. I. Quadri and M. Mohandes have proposed an image based recognition system for Arabic Sign Language. Region growing technique and Gaussian skin model are used respectively for face detection and hand tracking. Hidden Markov Models is used for classification of the signs. Proposed system has accuracy rate of 93%. [7]

Raja S. Kushalnagar, Lalit K. Phadtare and Nathan D. Cahill have proposed a system for synthesis of American Sign Language The proposed system uses MS Kinect and Open NI library. Skeletal and depth data is read in from the Kinect, then detection of the palm orientation and classification of hand shape is done. A three dimensional extension of the shape context classification algorithm is used for the classification of hand shape. The classification method classifies correctly 10 shapes of 40 hand shapes of the Hamnosys set. [8]

A novel training method for sign language recognition has been proposed by Shuqiong Wu and Hiroshi Nagahashi. The system proposes a new training method for Haar-like features based on AdaBoost classifier, including a hand detector which combines a skin-color model, Haar-like features and frame difference based on AdaBoost classifier for detecting moving right or left hand and a new tracking method which uses the hand patch extracted in the previous frame in order to create a new hand patch in the current frame. The detecting rate of the system is 99.9% and the rate of tracked hands which are extracted in proper size is more than 97.1%. [9]

Chana Chansri and Jakkree Srinonchat have presented a study of recognizing Thai sign language.The proposed system receives the color and depth information from the Kinect sensor for hand detection. Then Histograms of Oriented Gradients technique is used for feature extraction of images. Finally the extracted features are trained bu using Neural Network. The accuracy rates are obtained from different distances from Kinect sensor such as 08.m, 1.0m and 1.2 m are in order of 83.33%, 81.25%, 72.92%.[10]

3. METHODOLOGY

The proposed system works as following steps: Model Building Phase and Prediction Phase in Fig -1.

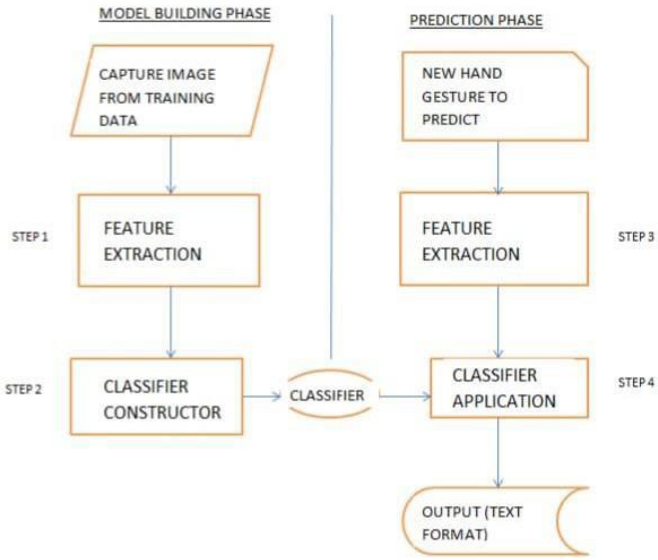

Fig -1: Block Diagram

3.1 Model Building Phase

3.1.1 Image Processing

The images in the local system or the frame captured from webcam camera are used as input to the system. After processing input image, then classifiers classify the image which class it belongs to. Convolution Neural Network learning technique is used to classify images which class they belong to. Feature extraction from the images is obtained in this way: input images in to the system are converted to a numerical format which means converting each image to a series of RGB pixels. Then images are normalized to be in the same shape. To do that, each observation is resized to 76x66 px. Then resized images are flattened to get 2D array using as features for the classifier. On the other hand histogram features are obtained by getting flattened histogram of input images to the system. Then classifier are trained with Convolution Neural Network Algorithm. Finally the predicted results which are obtained from classifiers are produced as text.

3.1.2 Classification

Convolution Neural Network(CNN) is a type of neural network model which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification.

The system for recognizing sign language using multilayer perceptron neural network was implemented by Python programming language.

This technology helps to bridge the gap between the speech and hearing impaired people and normal people. Work made a system, which helps dumb people where they can significantly communicate with all other people using their normal gesture we will define our Convolutional Neural Network (CNN) Model

After defining our model, we will check the model by its summary in the below Fig -2.

After successful training, we will visualize the training performance of the CNN model and calculate the accuracy with Confusion Matrix.

3.2. Prediction Phase

According to the result we have in the model we will calculate the confusion matrix with accuracy and loss is further discussed in experimental results.

4. TECHNOLOGY USED

4.1 Python (3.7.4)

Python is an interpreter, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
dropout (Dropout)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
conv2d_2 (Conv2D)	(None, 3, 3, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 1, 1, 128)	0
dropout_2 (Dropout)	(None, 1, 1, 128)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 512)	66048
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 25)	12825

Total params: 171,545
Trainable params: 171,545
Non-trainable params: 0

Fig -2: Model Sequential Table

4.2 IDE (Jupyter)

Jupyter Notebook provides you with an easy-to-use, interactive data science environment across many programming languages that doesn't only work as an IDE, but also as a presentation or education tool.

4.3 Numpy (version 1.16.5)

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

4.4 OpenCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV. This OpenCV tutorial will help you learn the Image processing from Basics to Advance, like operations on Images, Videos using a huge set of OpenCV programs and projects.

4.5 Keras (version 2.3.1)

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3 Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, R, Theano, and PlaidML.

4.6 TensorFlow

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming

4.7 Softmax

Softmax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector. The most common use of the softmax function in applied machine learning is in its use as an activation function in a neural network model.

Specifically, the network is configured to output N values, one for each class in the classification task, and the softmax function is used to normalize the outputs, converting them from weighted sum values into probabilities that sum to one. Each value in the output of the softmax function is interpreted as the probability of membership for each class.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

σ = softmax

$\vec{z}$ = input vector

e^{z_i} = standard exponential function for input vector

K = number of classes in the multi-class classifier

e^{z_j} = standard exponential function for output vector

e^{z_j} = standard exponential function for output vector

Fig -3: Softmax Function

4.8 ReLU

In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input. The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

5. EXPERIMENTAL RESULTS

The Accuracy and Loss is evaluated in terms of graphical representation with respect to model. Here, in the below Fig -4 and Fig -5 shows the graphical representation of accuracy and loss evolution respectively.

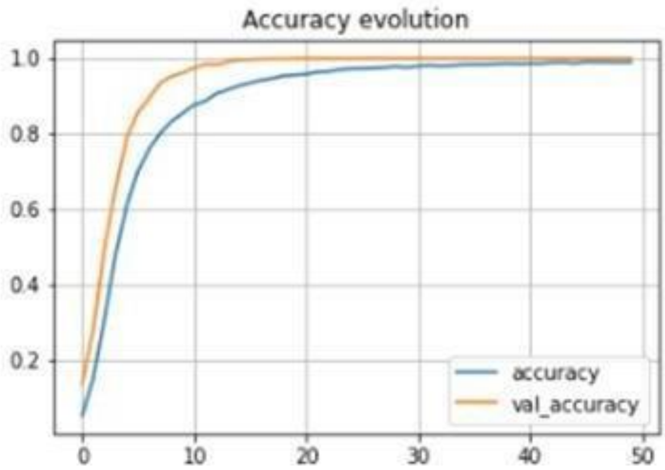

Fig -4: Accuracy Evolution

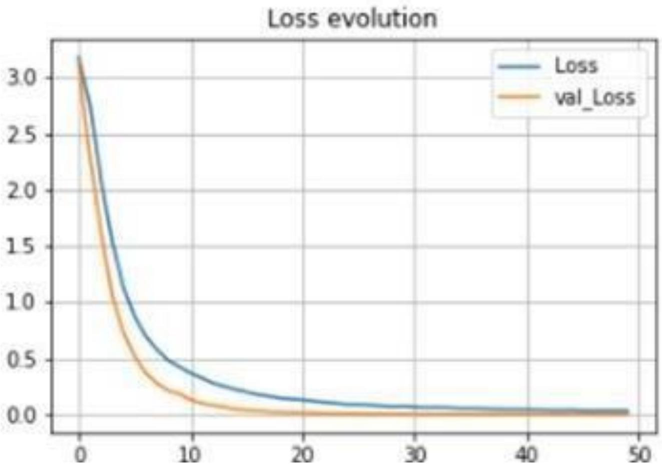

Fig -5: Loss Evolution

Confusion Matrix: A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

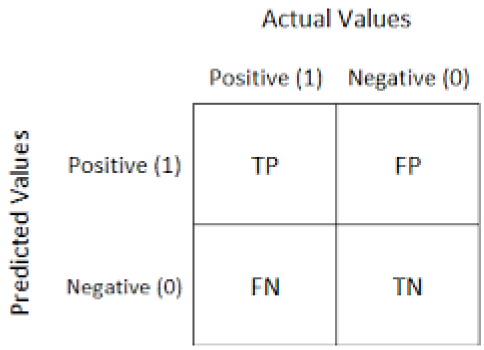

Fig -6: Formula of Confusion matrix

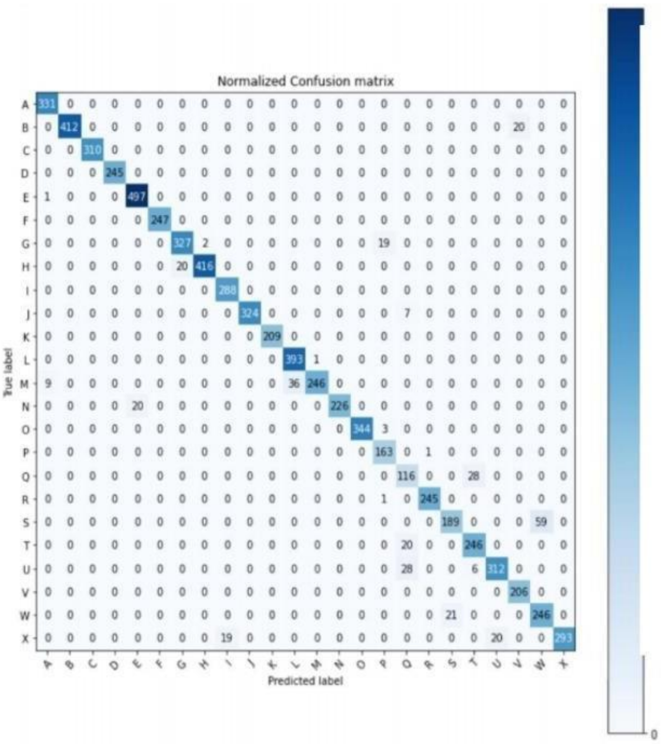

Fig -7: Normalized Confusion matrix

```
#Classification accuracy
from sklearn.metrics import accuracy_score
acc_score = accuracy_score(y_test, predicted_classes)
print('Accuracy Score = ',acc_score)

Accuracy Score =  0.9742987149357468
```

Fig -8: Accuracy Score

The CNN model has given 100% accuracy. Now, we will obtain the average classification accuracy score. We can conclude that the Convolutional Neural Network has given an outstanding performance in the classification of sign language symbol images. The average accuracy score of the model is more than 96% and it can further be improved by tuning the hyper parameters. We have trained our model in 50 epochs and the accuracy may improve if we have more epochs training. However, more accuracy is also an achievement than 96% is also achievement. As shown in Fig - 8 accuracy score.

6. CONCLUSION AND FUTURE WORKS

In method, Recognizing Hand Gesture using American Sign Language realized the time constraints and difficulties of creating a dataset from scratch. Looking back, it would have been nice to have had a dataset already to work off of. Some letters were harder to classify in our live demo such as "a" vs. "i" since they only differ by a very small edge (the "i" has the pinky pointing up). Although our classification system works quite well as has been demonstrated through tables and images, there is still a lot of scope for possible future work.

The CNN model has given 100% accuracy. Now, we will obtain the average classification accuracy score. We can conclude that the Convolutional Neural Network has given an outstanding performance in the classification of sign language symbol images. The average accuracy score of the model is more than 96% and it can further be improved by tuning the hyper parameters. We have trained our model in 50 epochs and the accuracy may improve if we have more epochs training. However, more accuracy is also an achievement than 96% is also achievement. As shown in Fig - 8 accuracy score.

Possible extensions to this work would be extending the gesture recognition system to all alphabets of the ASL and other non-alphabet gestures as well. In future work, we can take the images of real-time to test our application from web camera, and convert the result to text form. We feel that we can also improve upon the speed of our real-time system by coding in C. The framework of this work can also be extended to several other applications like controlling robot navigation using hand gestures.

REFERENCES

[1] V. N. T. Truong, C. K. Yang and Q. V. Tran, "A translator for American sign language to text and speech," 2016 IEEE 5th Global Conference on Consumer Electronics, Kyoto, 2016, pp. 1-2. Doi: 10.1109/GCCE.2016.7800427

[2] Yi Li, "Hand gesture recognition using Kinect," 2012 IEEE International Conference on Computer Science and Automation Engineering, Beijing, 2012, pp. 196-199.doi: 10.1109/ICSESS.2012.6269439

[3] A. I. C. Ani, A. D. Rosli, R. Baharudin, M. H. Abbas and M. F. Abdullah, "Preliminary study of recognizing alphabet letter via hand gesture," 2014 International Conference on Computational Science and Technology (ICCST), Kota Kinabalu, 2014, pp. 1-5.doi: 10.1109/ICCST.2014.7045002

[4] M. M. Islam, S. Siddiqua and J. Afnan, "Real time Hand Gesture Recognition using different algorithms based on American Sign Language," 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Dhaka, 2017, pp. 1-6.doi: 10.1109/ICIVPR.2017.7890854

[5] W. Y. Yeh, T. H. Tseng, J. W. Hsieh and C. M. Tsai, "Sign language recognition system via Kinect: Number and english alphabet," 2016 International Conference on Machine Learning and Cybernetics (ICMLC), Jeju, 2016, pp. 660-665. doi: 10.1109/ICMLC.2016.7872966

[6] "Deafness and hearing loss," World Health Organization.[Online].Available: http://www.who.int/mediacentre/factsheets/fs300/en/. [Accessed: 19-Aug-2017].

[7] M. Mohandes, S. I. Quadri and M. Deriche, "Arabic Sign Language Recognition an Image-Based Approach," Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on, Niagara Falls, Ont., 2007, pp. 272-276.doi:10.1109/AINAW.2007.98

[8] L. K. Phadtare, R. S. Kushalnagar and N. D. Cahill, "Detecting handpalm orientation and hand shapes for sign language gesture recognition using 3D images," 2012 Western New York Image Processing Workshop, New York, NY, 2012, pp. 29-32. doi: 10.1109/WNYIPW.2012.6466652

[9] S. Wu and H. Nagahashi, "Real-time 2D hands detection and tracking for sign language recognition," 2013 8th International Conference on System of Systems Engineering, Maui, HI, 2013, pp. 40-45. doi: 10.1109/SYSoSE.2013.6575240R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[10] C. Chansri and J. Srinonchat, "Reliability and accuracy of Thai sign language recognition with Kinect sensor," 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Mai, 2016, pp. 1-4. doi: 10.1109/ECTICon.2016.7561403

BIOGRAPHIES

Nikhataara Jakati received the B.E. degree in computer science engineering from JCET, Hubballi, Karnataka in 2021. Currently, she is a Software Engineer. Her current research interests include in Machine Learning, Image Processing.

Pavana Gatnur received the B.E. degree in computer science engineering from JCET, Hubballi, Karnataka in 2021. Her current research interests include in Machine Learning, OpenCV.

Kshama Desai received the B.E. degree in computer science engineering from JCET, Hubballi, Karnataka in 2021. Her current research interests include in Artificial Intelligence and Cloud Computing.

Sahana Desai received the B.E. degree in computer science engineering from JCET, Hubballi, Karnataka in 2021. Currently, she is a Software Engineer. Her current research interests include in Machine Learning.