Journal of Mines, Metals and Fuels, 71(5): 607-613; 2023. DOI: 10.18311/jmmf/2023/34158

Print ISSN : 0022-2755
Journal of Mines, Metals and Fuels
Contents available at: www.informaticsjournals.com/index.php/jmmf

Sign Language Detection and Translation

Rimika Bhaumik[1*], Sudipta Patra[2], Debdipta Chakraborty[1], Subhadeep Basack[1], Pallabi Mazumder[1] and Paromita Das[1]

[1]Electronics and Communication Engineering, Amity University, Kolkata Kolkata, India.
E-mail: rimikabhaumik@gmail.com / debdipta778@gmail.com / subhadeep.basack@gmail.com / pallabimazumder2002@gmail.com / pdas1@kol.amity.edu
[2]Electronics and Communication Engineering, Amity University Kolkata, Dhanbad, India.
E-mail: sudiptapatra1102@gmail.com

Abstract

Communication between the general public and the deaf community is difficult. Most people find it difficult to communicate without an interpreter since sign language is not universally understood. This research suggests utilizing machine learning methods to build an effective Sign Language Detection and Translation model employing real-time dataset. This model could be employed in school and other places, facilitating communication between impaired and non-impaired people. The suggested approach can be used to recognize sign language with ease using Keras and Tensorflow.

Keywords: *Machine Learning, OpenCV, Keras, Convolutional Neural Network, American Sign Language.*

1.0 Introduction

Communication is essential for people to function as a species. It is a fundamental and effective technique for communicating thoughts, feelings, and points of view. Many people experience either hearing loss, speaking difficulty, or both. Mute is a handicap that prevents speech and renders those who have it mute. Casual gestures and official signals are the two main categories of sign language. It is the strongest and most successful method for bridging the social contact and communication gap between them and able-bodied persons.

By converting sign language into spoken words and the other way around, sign language interpreters assist close the communication gap with the hearing impaired. For virtual conferences like Zoom meetings and other similar events, real-time captioning can be created by implementing predictive model technology to automatically recognize Sign Language symbols. This would greatly increase access of such services to those with hearing impairments as it would go hand-in-hand with voice-based captioning, creating a two-way communication system online for people with hearing issues.

Between the hearing audience and the deaf, a communication channel can be established. Three actions need to be taken right away to remedy our problem:

1. The first step is to record the person signing on camera (input).
2. Identifying a sign for each frame of the video.
3. Using categorization scores, reconstructing, and showing the most likely Sign (output).

2.0 Literature Review

A. Classification of Forearm EMG and IMU Signals for Signing Exact English by a Sign Language Interpreter Sangit Sasidhar and Soo Pei Yi Jane

The deaf frequently encounter a severe language barrier that prevents them from communicating with hearing persons.

[*]Author for correspondence

This research involved designing a method to let the hearing and the deaf communicate more effectively. From the processed signal, a number of time- and frequency-domain parameters are retrieved. A 48-word Signing Particular English word list was used to test the artificial neural network classifier, which had an average classification rate of 97.12%.

B. Surface EMG Signal-Based American Sign Language Recognition System by Celal Savur and Ferat Sahin

Thanks to a cutting-edge system called the Sign Language Recognition (SLR) system, people with hearing impairments can interact with society. In this paper, a surface electromyography-based method for comprehending American Sign Language (ASL) was proposed.

The objective of this research is to aid ASL users in spelling words and phrases by assisting them in recognizing the alphabet's letters. 26 English alphabet letters, one for the home position, and 27 ASL motions were recorded using sEMG signals from the subject's right forearm. We gathered the parameters for the time domain, frequency domain, power spectral density (band power), and average power. Principal component analysis (PCA), which was used to obtain uncorrelated features, was performed after feature extraction. Principal component analysis (PCA), which was used to obtain uncorrelated features, was performed after feature extraction. Support Vector Machine and Ensemble Learning methods were applied as a classification, technique and the outcomes were contrasted with tabular data. The findings of the study.

C. Jian Wu, Zhongjun Tian, Lu Sun, Leonardo Estevez, and Roozbeh Jafari's Real-time American Sign Language Recognition Using Wrist-worn Motion and Surface EMG Sensors

Persons with hearing loss can communicate with people who can hear and speak by using a sign language recognition (SLR) device. The development of a substantial human computer interface that can read hand gestures and determine the user's intent is due in large part to the popularity of wearable computers. Four subjects and the top 40 words are each given a feature pick. The experimental findings demonstrate that our system achieves a 95.94% identification rate following feature selection and conditioning. The outcomes also demonstrate how two when compared to just the inertial sensor, modalities perform better. We found that only one sEMG channel (out of four) was situated on the wrist and area underneath the wristwatch are adequate.

D. Classifying Electromyography Data to Identify Finger Movement for use in Robot Control Vahid Azimirad, Mahdiyeh Hajibabazadeh, and Maryam Ali Mohammadi Soltanmoradi

An innovative method for classifying electromyography (EMG), that is used to control robots, is presented in this work. The EMG waves of a person's muscles are crucial for monitoring how the prosthesis moves. Instead of, two EMG probes are attached to the human forelimb and are utilized to gather EMG data. Wavelet coefficients and tunes for moment and rate of recurrence, such as Autoregressive and Number of Zero Crossings, are both recognized as features. On the other hand, the Support Vector Machine (SVM) is a classification technique. Results demonstrate that the proposed technique is about 80% accurate. Finally, to demonstrate the value and relevance of the findings, the results of the categorization system are applied to a stationary robot known as Tabriz- Puma.

3.0 Technology

Prerequisites:

1. Python (3.7.4)
2. IDE (Jupyter)
2. Numpy (version 1.16.5)
3. cv2 (openCV) (version 3.4.2)
4. Keras (version 2.3.1)
5. Tensor flow (as keras uses tensor flow in back end and for image preprocessing) (version 2.0.0)

4.0 Algorithm

A. OpenCV

Real-time performance, which is essential in contemporary systems, is a key function of OpenCV. OpenCV is an extensive open-source toolkit for computer vision, machine learning, and image processing. It can analyze images and videos to identify objects, such as faces and objects, as well as writing from human hands. The OpenCV array structure can be processed for analysis by combining Python with a variety of modules, such as NumPy. By applying mathematical processes to the vector space features of a visual pattern, we may identify it. OpenCV 1.0 was the initial release. To support real-time applications for efficient processing, OpenCV's key focus during development. It's all written in C/C++ that has been enhanced to take use of multi-core processing.

B. Keras

A kind of artificial intelligence called "deep learning" aims to solve extremely complicated problems by simulating how the human brain functions. To help divide the problem into smaller components that can each be tackled separately, deep learning uses neural networks with numerous operators inserted in nodes. But it can be quite challenging to implement neural networks. This problem is addressed with the Keras deep learning framework. Numerous backend neural network computations are provided as well. It is reasonably easy to understand and utilize Keras because it provides a high level of abstraction, a Python frontend, and a variety of compute back-ends. As a result, Keras is less intuitive for novices yet slower than other deep learning frameworks. With Keras, one can switch between different back ends. Keras supports the following frameworks:

C. Tensorflow

The TensorFlow Object Detection API, an open-source framework built on top of TensorFlow, makes it simple to develop, train, and apply object recognition models. Pre-trained models are present in their framework through a feature called Model Zoo. 1. The dataset for "Common Objects in Context." 2. The Open Images dataset KITTI Dataset

TensorFlow combines the deep learning with machine learning models and techniques. It offers a practical Python front-end and functions efficiently in optimized C++. Programmers can use TensorFlow to build a graph of computations to execute. Each node in the network stands for a mathematical operation, whereas each link represents data. As a result, the developer may focus on the overall logic of the program rather than having to figure out how to link the output of one function to its input in another. Currently, the most used software library is TensorFlow. TensorFlow is well-liked because deep learning has many real-world uses. Uses for TensorFlow, an open-source deep learning and the machine learning framework, include text-based applications, voice search, and image identification. Deep Face, Facebook's image recognition program, makes use of TensorFlow.

5.0 Implementation

A. Library

- NumPy: NumPy is the name of the essential NumPy library for Python. It is a Python library that provides a variety of derived objects, such as masked arrays and matrices, multidimensional array objects, and a collection of routines for quick and smooth operations on arrays, such as sorting, selecting, shape manipulation, logical and mathematical operations, I/O, fundamental linear algebra, discrete Fourier transforms, basic statistical operations, random simulation, and several others.
  The processing of numerical data is done with it.
- Pandas: This package for data analysis offers data frames. Data in rows and columns can be picked. NumPy is the name of the primary Python library for scientific computing.
- OpenCV: The sizable open-source library for computer vision, machine learning, and image processing is called OpenCV. Among numerous programming languages that OpenCV supports are Python, C++, and Java. It can look for individuals, objects, or even handwriting in pictures and movies. When it is used with additional libraries, such as the highly efficient NumPy library for numerical operations, the number of weapons in your arsenal rises. NumPy and OpenCV can be used to accomplish any task.
- Mediapipe: Unlike the MediaPipe Python framework, which offers direct access to the core MediaPipe C++ framework elements like Timestamp, Packet, and calculator graph, fully prepared Python solutions hide the technical details of the framework and only return the callers with readable model inference results. The MediaPipe framework serves as the foundation for the pybind11 library. A C++/Python language coupling allows Python to access the C++ core framework. The following information assumes that the reader is already familiar with the MediaPipe C++ framework. In any case, Framework Concepts offer valuable knowledge.
- Keras: This Python-based open-source software library supports artificial neural networks. A TensorFlow library interface is part of Keras. It is made easier to deal with text and image data, and Keras offers numerous implementations of well-known neural network building pieces like layers, objectives, activation functions, and optimization. These features simplify dealing with text and image data as well as writing the necessary coding for deep neural network code.

B. Problem Statement

The most widely used and meaningful method of communication for society's deaf and mute individuals is the sign language. Communication between the deaf and the general population might be significantly improved with the use of a real-time sign language detector.

In this study, we propose a system design for hand gesture recognition based on the detection of various relevant shape-based parameters, such as orientation, the center of mass (centroid), the status of the fingers and thumb in terms of raised or folded fingers of the hand, and their respective

placement in the image. In order to do this, it is crucial to consider how similar the four-finger and one-thumb structures of human hands are. Additionally, this might be set up as a method for instructing those learning sign language.

C. Background Dataset

The sign language that is most frequently used in, you guessed it, the United States and Canada, or American Sign Language. The data set contains a collection of illustrations of American Sign Language alphabets that have been divided into 24 folders to reflect the different classes. It is a collection of 2400 images, 100 images for each of the 24 classes.

Five subjects in total are recorded making these gestures. We evaluated and formulated models to classify hand gestures for the 24 letters of the alphabet, except Z and J.

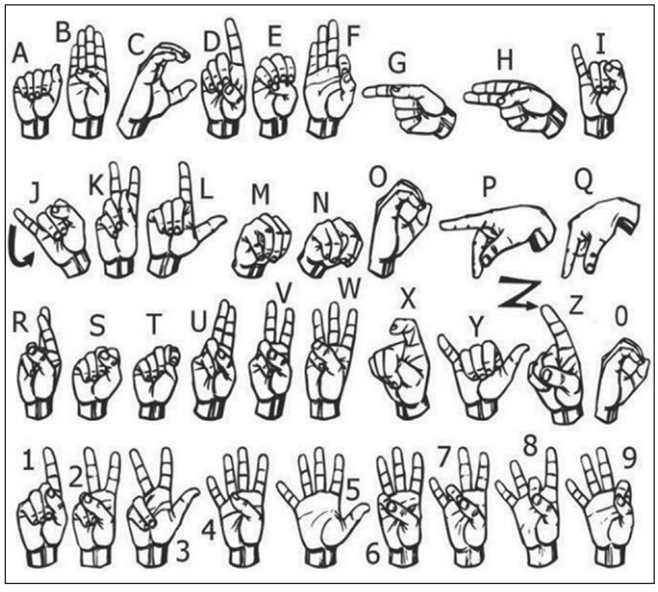


Figure 1: American sign language reference picture

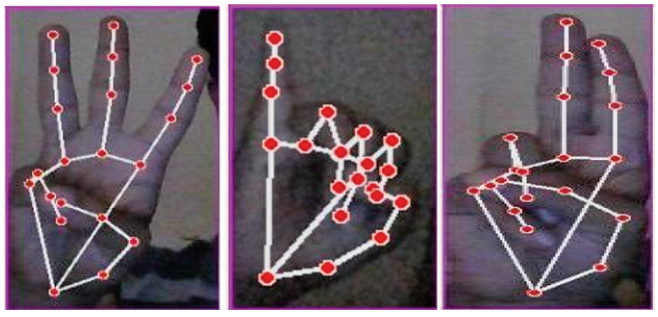


Figure 2: Real time dataset recorded by the five authors

D. Workflow

Data Collection

For collecting the training data, firstly all the necessary libraries are imported: Cvzone and mediapipe. Then using cv2.video capture the webcam module of the device being used is opened. Directories are created for the alphabets A-Z containing the real time train dataset, and almost 250-300 images are recorded per alphabet for training the model.

The letters J and Z in American Sign Language are dynamic so they can't be detected by the test classifier module. Now, from Cvzone. Hand Tracking Module the library: HandDetector is imported through which we are setting the maximum number of hands that can be detected at a time, by initializing the variable max Hands. Through the inbuilt function find Hands the image is passed, and the hand skeleton of the image is detected. The hand portion of the image is then recognized and separated from the background image using the border box holding the hand dimensions.

Finally, a white backdrop is provided via a matrix containing 8-bit integer values ranging from 0-255 to adjust the dimensions and the aspect ratio (height/width) of the cropped image.

Figure 3: The dataset

Model training and prediction in real time

Convolutional neural networks surpass other neural networks, when given inputs such as images, voice, or audio. There are three different primary layers of the convolutional Neural Network, namely the convolutional layer, the pooling layer and the fully connected layer.

Convolutional layer: The convolutional layer, which comprises most of the computation in CNN, is its central component. Among other things, it requires input data, a filter, and a feature map. The three dimensions of the input are

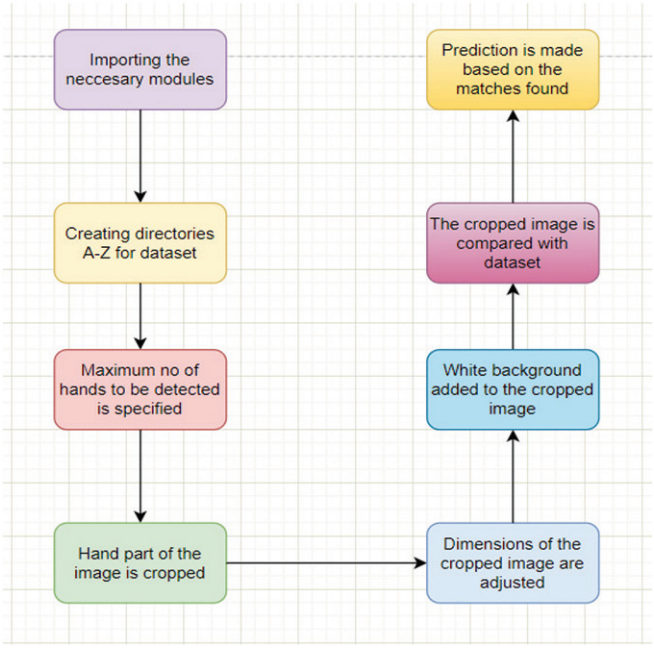


Figure 4: Flowchart depicting data collection

height, breadth, and depth. By moving through the image's receptive fields, a feature detector – often referred to as a kernel—determines if the feature is there. Typically, the kernel size is a 3×3 matrix. A dot product between the input pixels and the filter is then computed after the filter has been applied to a specific area of the image. The output array is then given this dot product. The filter moves forward one step and then continues the operation until the kernel has completely covered the image. A feature map or activation map is the final product of the series of dot products from the input and filter. A CNN adds nonlinearity to the model by applying a Rectified Linear Unit (ReLU) correction to the feature map after each convolution operation.

Pooling layer: Pooling layers, commonly referred to as down sampling, is a technique for reducing dimensions. Additionally, this lowers the amount of input parameters. Similar to how the convolutional layer does it, the pooling technique sweeps a filter across the entire input, but this filter lacks weights. Therefore, the kernel makes use of an aggregation function to add values from the receptive field to the output array. The two most common types of pooling are maximal and average. The filter progresses across the input while using max pooling, selecting the input pixel with the greatest value to deliver to the output array. The average value in the receptive field is calculated and then transferred to the output array when the filter passes over the input in average pooling. In comparison with average pooling, maximum pooling is employed more frequently.

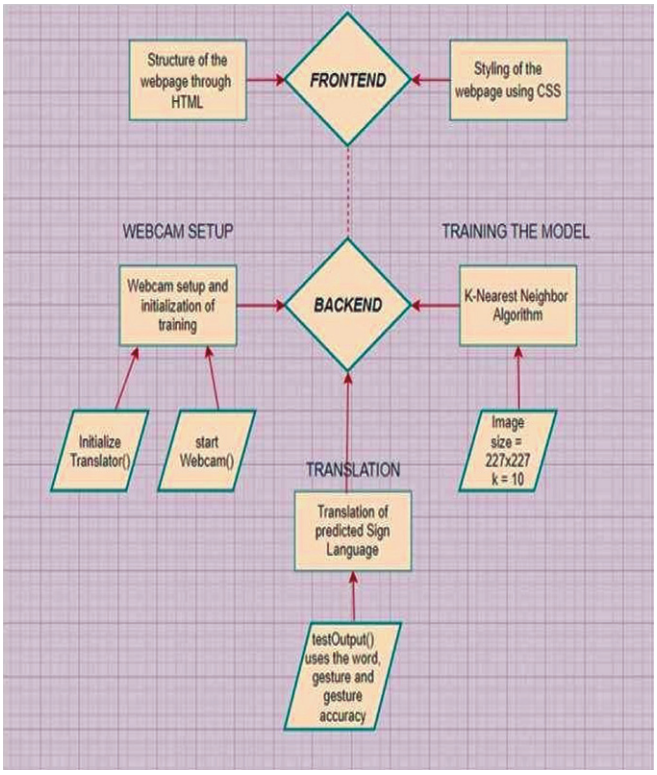
Fully connected layer: The pixel values of the input image are not directly connected to the output layer in partially linked layers. The completely connected layer, on the other hand, has a direct coupling between each node in the input layer and every node in the output layer. The features that were collected from the levels above and their corresponding filters are used in this layer to carry out the classification process. FC layers usually employ a softmax activation function, which produces a probability ranging from 0 to 1, to correctly identify inputs. It is common practice to use the ReLu activation function in convolutional and pooling layers.

Web application building

Language Used:

1. HTML
2. CSS
3. JavaScript

Working procedure:



Advantages:

1. Model is being trained thoroughly at the real time.
2. The entire model can be used to implement any sign languages like ISL, ASL, etc.
3. Minimum only 30 images are required for the model to be trained and performed at the real time.
4. Different personalized hand gestures can also be implemented effortlessly.
5. Interpretation of the sign also makes communication especially with the blind people at ease.

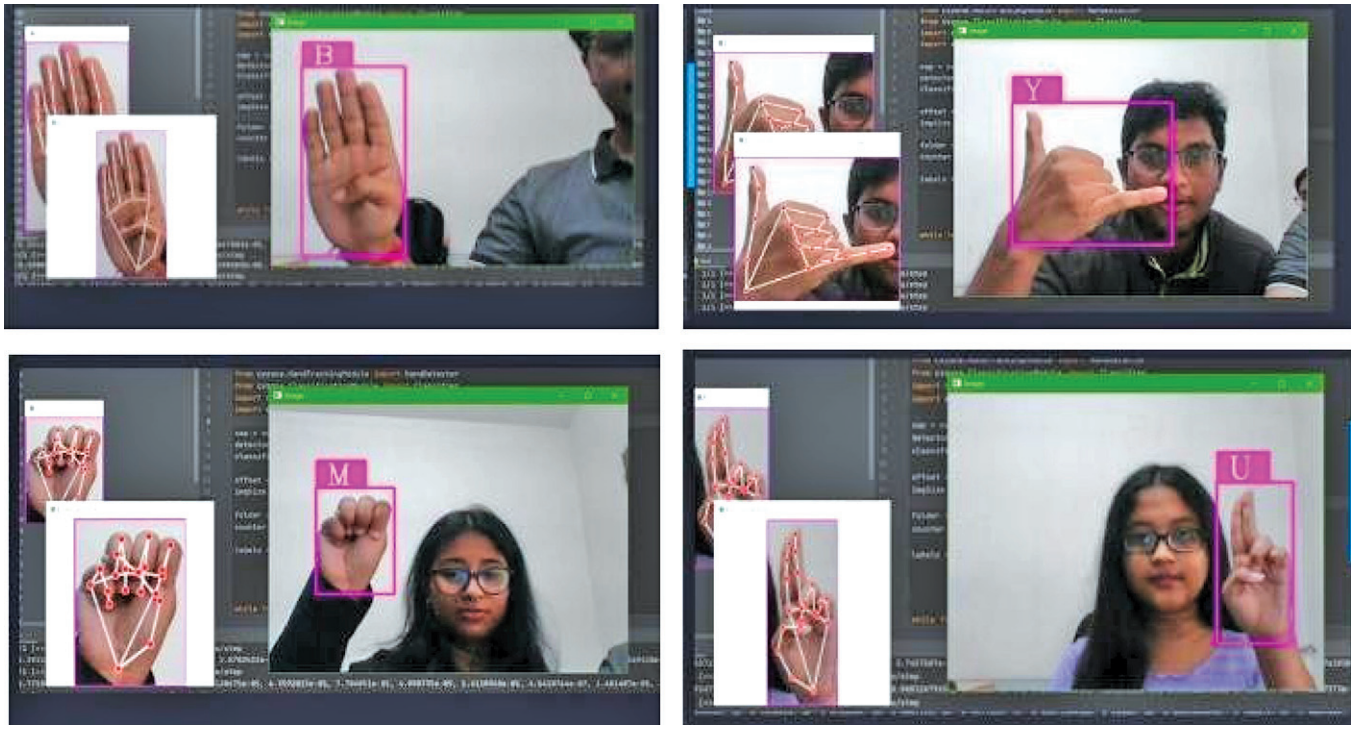


Figure 5: Real-time prediction

6. Furthermore, Video Calling options for physically challenged people can be employed with the existing model.

E. Prediction

When the subject gesticulates an action in front of the webcam, the camera captures the hand gesture, and based on the accuracy of the model, will predict what sign is being displayed.

Our four subjects demonstrated the signs M, Y, U, and B which was correctly predicted by the model as shown in the reference pictures.

6.0 Application

- The dataset is easily expandable and adaptable to.
- The worth of the disabled people's labor can be acknowledged at worldwide gatherings by using the sign detection model, which makes them easier to interpret.
- The paradigm is accessible to everyone and may be utilized by anyone with a basic understanding of technology.
- This strategy can be used at the elementary school level to introduce sign language towards children as early as possible.

7.0 Future Scope

- Extending our idea to other sign languages, such as American or Indian sign languages.
- Increasing the neural network's efficiency in symbol recognition and improvement in the model's ability to recognize expressions can also increase the overall accuracy of the model.
- This model can be very helpful for translation purpose. The deaf people can easily translate their thoughts to people who don't understand sign language using this detection and translation model, which aims at first detecting the hand sign, and then converting the text to voice.
- This will open employment opportunities for deaf people in education and other fields.
- Further an Android application can be built to access the features of our model easily.

8.0 Conclusion

First-vision understanding of sign language has some limitations because some gestures will end up looking the same. But by placing more cameras in various locations, this ambiguity can be eliminated. This enables what one camera cannot see to be perfectly visible to a different camera. Although the system wasn't very effective, it showed that a

first-person sign language translation system could be constructed using simply cameras. It was discovered that the model frequently mixed together different signs, including U and W. We can infer from the model's output that, given conditions of controlled light and intensity, the suggested system can produce reliable results. Additionally, adding new gestures is simple, and the model will be more accurate if there are more photographs captured at various angles and frames. As a result, expanding the dataset makes it simple to scale up the model.

9.0 Acknowledgments

We sincerely express our deep sense of gratitude to Prof. Paromita Das, our faculty guide for her constant support, and supervision throughout the research. Many thanks to Prof. Sayanti Banerjee, Prof. Dr. Pushan Kumar Dutta, Prof. Kalyan Chatterjee, Prof. Subhasish Roy, and Prof. Dr. Semanti Chakraborty for their valuable advice throughout the study in Sign Language Detection and Translation project. We would like to thank the ECE Dept. of Amity School of Technology Kolkata, Amity University Kolkata for technical assistance and supply of academic resources related to the project.

10.0 References

1. L. Kin, T. Tian, R. Anuar, Z. Yahya, and A. Yahya, (2013): "Sign Language Recognition System using SEMG and Hidden Markov Model," Conference on Recent Advances in Mathematical Methods, Intelligent Systems and Materials, pp. 50–53.
2. S. A. K. Mehdi Y. N., (2002): "Sign language recognition using sensor gloves," Proceedings of the 9th International Conference on Neural Information Processing, vol. 5, pp. 2204–2206.
3. Savur and F. Sahin, "American Sign Language Recognition System by Using Surface EMG Signal," in *International Machine Learning and Application*
4. Conference ICMLA, 2015. K. Elissa, "Title of paper if known," unpublished.
5. Ethem Alpaydin, Introduction to Machine Learning, 3rd ed. The MIT Press, 2014.
6. L. Breiman, "Random Forests," *Journal of Machine Learning,* vol.45, no. 1, pp. 5–32, 2001.
7. C. Savur, (2015): "American Sign Language Recognition System by Using Surface EMG Signal,".
8. National Office for Empowerment of Persons with Disability (NEP)," *Annual Report 2012*, pp.94, 2012.
9. The Foundation for The Deaf Under the Royal Patronage of Her Majesty the Queen (13-07-2014). *The school for the deaf persons*.
10. W. C. Stokoe, "Sign language structure: An outline of the visual communication systems of the American deaf," *Journal of deaf studies and deaf education*, vol. 10, no. 1, pp. 3–37, 2005.
11. Barberis, N. Garazzino, P. Prinetto, G. Tiotto, A. Savino, U. Shoaib,and N. Ahmad, (2011): "Language resources for computer assisted translation from Italian to Italian sign language of deaf people," in Proceedings of Accessibility Reaching Everywhere AEGIS Workshop and International Conference, Brussels, Belgium (November 2011).
12. B. Grieve-Smith, (2002): "Signsynth: A sign language synthesis application using web3d and perl," in Gesture and Sign Language in Human Computer Interaction, pp.134-145, Springer.
13. C. Manresa, J. Varona, R. Mas, and F. Perales, (2005): "Hand tracking and gesture recognition for human-computer interaction," *Electronic letters on computer vision and image analysis*, vol.5, no.3, pp.96-104.
14. K. Oka, Y. Sato, and H. Koike, (2002): "Real-time fingertip tracking and gesture recognition," *Computer Graphics and Applications*, IEEE, vol.22, no.6, pp.64-71.
15. K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, (2014): "Fusion of inertial and depth sensor data for robust hand gesture recognition".
16. T. Starner, J. Weaver, and A. Pentland, (1998): "Real-time American sign language recognition using desk and wearable computer-based video," Pattern Analysis and Machine Intelligence, *IEEE Transactions on*, vol.20, no.12, pp.1371-1375.
17. Vogler and D. Metaxas, (2001): "A framework for recognizing the simultaneous aspects of American sign language," *Computer Vision and Image Understanding*, vol.81, no.3, pp. 358–384.
18. T. E. Starner, (1995): "Visual recognition of American sign language using hidden markov models.," tech. rep., DTIC Document.
19. B. Ajiboye and R. F. Weir, (2005): "A heuristic fuzzy logic approach to emg pattern recognition for multifunctional prosthesis control," Neural Systems and Rehabilitation Engineering, *IEEE Transactions on*, vol.13, no.3, pp.280–291.
20. J.U. Chu, I. Moon, and M.-S. Mun, (2005): "A real-time emg pattern recognition based on linear-nonlinear feature projection for multifunction myoelectric hand," in Rehabilitation Robotics. ICORR. 9th International Conference on, pp. 295–298, IEEE, 2005.
21. Y. Li, X. Chen, X. Zhang, K. Wang, and J. Yang, (2011): "Interpreting sign components from accelerometer and semg data for automatic sign language recognition," in Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, pp. 3358–3361, IEEE.
22. Sherrill, P. Bonato, and C. De Luca, (2002): "A neural network approach to monitor motor activities," in Engineering in Medicine and Biology. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint, vol.1, pp. 52–53, IEEE, 2002.