DEVELOPMENT OF MOBILE APPLICATION FOR THE AID OF HEARING AND SPEECH-IMPAIRED PEOPLE

Ms. DEEPA M
Assistant Professor (Sr. Gr)DEPARTMENT OF ELECTRONICS AND COMMUNICATION
PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH
Email: deepa@psgitech.ac.in

PRAKALYA E
DEPARTMENT OF ELECTRONICS AND COMMUNICATION
PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH
Email: 19ec135@psgitech.ac.in

ASWATHI R
DEPARTMENT OF ELECTRONICS AND COMMUNICATION
PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH
Email: 19ec106@psgitech.ac.in

VEDHA R
DEPARTMENT OF ELECTRONICS AND COMMUNICATION
PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH
Email: 19ec157@psgitech.ac.in

SOWMYA S
715519106049
DEPARTMENT OF ELECTRONICS AND COMMUNICATION
PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH
COIMBATORE, INDIA
Email: 19ec149@psgitech.ac.in

Abstract—Background: Communication forms the basis of information exchange between individuals. The process of conveying one's thought processes depends on the different types of communication, orality being the direct and highly effective option. 5-10 percent of the population is not equipped with this necessity. This brings in the need for alternatives, viz, sign language.

Sign Language employs the visual manner of expression. There are over 300 sign languages across the world, and furthermore, variations stemmed from the regionalization of accents. This confers to the issue of difficulty in understanding speech and hearing-impaired people across the world. Barriers within accents of the same language are characteristic. This paper works at trying to bridge this gap in communication, by using Machine Learning to create a sign-language translation system that converts gestures to the spoken word.

Keywords—Sign-Language Translation, Machine learning, Software Application

I. INTRODUCTION

Sign language uses gestures and other visual features to convey messages. Expressions form the core of communication for hearing and speech-impaired people. Learning sign language would involve an understanding of the intricacies of grammar and regional dialects and accents. This poses a difficulty for indigenous variety in this form of communication. American Sign Language (ASL) is very popularly used around the world. Indian sign language (ISL) has a wide usage of around 1.8 million users but there is a national shortage of certified interpreters limiting it to a small 300 number. India, Pakistan, Bangladesh, and possibly Nepal are prominent users of ISL.

A. Need for this Project

Despite having 300 sign languages handy, there is still going to be the standing issue of having to learn everything. Going through the process for people who do not necessarily require sign language is cumbersome. This leads to miscommunication or a lack of expression between the speech and hearing-impaired people to the rest of the community which is illiterate in their form of communication.

TABLE I. THE OVERLOOK OF TOOLS USED

TOOLS USED		
Purpose	***Hardware***	***Software***
Model Training and Application	1. PC with operating systems: linux-ubuntu 16.04 or 17.0 or Windows 10 or 11 2.Android phone	Flutter 3.3 Android Studio Dolphin 2021.3.1 Python 3.9.0 Teachable machine Open CV Tensor Flow Kaggle or Jupyter Notebook

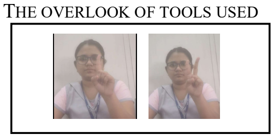


Fig. 1. Sign-language inputs. The one on the left is a sign for number 1 and on the right is for number 2.

The ease of carrying a digital interpreter around gives the entire process simple. Since the aim is for an application for translation of sign language, something as simple as the internet and installation process gives you access to the herculean communication technique of sign language and the benefits of cutting down barriers between the rest of the community and the specially-abled.

Working with ISL-based translation applications with a bonus of basic teaching techniques is thus an important need. It would benefit the users and non-users of sign language in India, Pakistan, Bangladesh, Nepal, and generally anybody willing to learn it or communicate it.

a) Planned approach:

The proposed flow of translation would include-

1. *Development of model with different classes of datasets.*
2. *Tracking and detection of inputs.*
3. *Pre-processing of inputs followed by recognition and classification.*
4. *Getting the desired outputs.*
5. *Verification of accuracy and methods to improve it.*
6. *Linking the model to the user interface application.*

b) Software Translation system:

The proposed system is expected to translate all gesture patterns of any person present in any background. With the advanced aid of Machine Learning and Computer Vision, this process has become quite easy. The complexity of hardware requirements is eliminated. The set of software tools used would be Flutter, Android studio, Python 3.9.0, Tensor flow lite, Open CV, Kaggle, or Jupyter Notebook.

B. Anticipated Output

The desired output would be an accurate translation of the gestures to text. The inputs are processed for location, detection, and differentiation. Post this, the model is expected to use the previously trained results to classify the gestures into the respective alphabet, number, or phrase. The display of the output is in text for the user to read. This, incorporated with an application platform would be viable for online and digital media zones.

II. PREVIOUS WORK IN THE FIELD

The speech-impaired and hard-of-hearing community uses sign language, a visual language, to communicate with one other and the hearing community. In recent years, there has been a lot of research being done on the development of machine learning-based sign-language recognition systems. A significant source of research articles in this area has been the IEEE platform. This review of the literature provides an overview of the machine learning-based sign-language recognition systems in chronological order of articles published on the IEEE platform. The survey provides insights into the most recent breakthroughs, their limitations, and potential paths while tracing the history of sign-language recognition systems from their earliest days to the present.

The first paper we came across was ‘Machine Learning Techniques for Indian Sign Language Recognition’[1] authored by Kusumika Krori Dutta, and Sunny Arokia Swamy Bellary in 2017 on the IEEE platform. It dealt with single and double-handed gesture translation using Machine Learning tools and focusing on the usage of Matlab for an aim of 92% accuracy, furthering a goal of 100% recognition power.

In the same year, the IEEE journal ‘Sign language learning system with Image Sampling and Convolutional Neural Network’[2] was worked by Yangho Ji, Sunmok Kim, and Ki-Baek Lee. A video with recognizable motion was concatenated into screenshots of 2-D using a cheap camera for a realistic touch. The data set chosen to work on had very finite differences increasing the difficulty level of the experiment. This was fed to a convolutional neural network (CNN) and it gave test results with 86% accuracy, which could further be increased with more situational data sets.

In 2018, ‘American Sign Language Recognition using Deep Learning and Computer Vision’[3] was written in IEEE by Kshitij Bantupalli and Ying Xie which had a CNN model used. It was an Inception i.e. a model developed by Google for image recognition. It used two different approaches for classification:

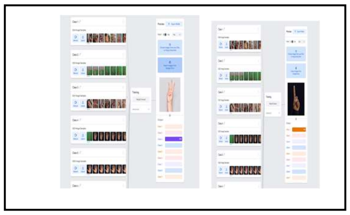
a. Using the predictions from the Softmax layer and
b. Using the output of the global pool layer.

2018 had two more literary works ‘Time Series Neural Networks For Real-Time Sign Language Translation’[4] and ‘Real-time Indian Sign Language (ISL) Recognition’ [5]worked upon by Sujay S Kumar, Tenzin Wangyal, Varun Saboo, Ramamoorthy Srinath, and Kartik Shenoy, Tejas Dastane, Varun Rao, Devendra Vyavaharkar respectively.

The former paper deduced how Sign-Language to text can be achieved solely based on visual cues. The system was made to recognize a subset of Sign- Language vocabulary by using a single camera module. This involved identification of SL glosses as the first stage with the translation of said SL glosses to English sentences being the second stage. The latter however worked using the signs following the techniques of Object Stabilisation, Face elimination, Skin color Extraction, and Hand extraction. Implementation of these techniques before the actual extraction reduced the probability of error in the detection.

This was done using single hand gestures but can be easily extended to double hand gestures. The paper worked as a good base for the proposed approach in this project and helped us understand the areas of difficulty we would face. It gave a clearer picture of what we would be doing throughout and areas where there is scope for improvement.

Effective translation of sign language from videos on smartphones that do not require any special equipment like a camera or gloves. This is achieved through ml and neural networking. It can be further developed for one-handed or double-handed sign language. Being a project orienting towards software-based solutions, this paper titled ‘A virtual sign language translator on smartphones’

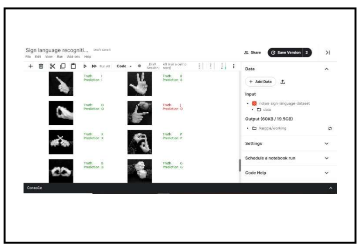


[6]by Yun-Jung Ku, Min-Jen Chen, and Chung-Ta King in 2019 IEEE was indeed helpful.

In the same year, Paranjoy Paul and G N Rathna gave us a look into Indian Sign Language Recognition work on the Indian Academy of Sciences platform[7]. Their model used segmented RGB hand gestures to improve accuracy. It also used depth-based segmentation to remove the overheads of dynamic backgrounds.

It could detect the usage of both the hands where edges of one hand can get overlapped or be nullified due to the other hand giving an accuracy of 89.30%.

Sandrine Tornay, Marzieh Razavi, and Mathew Magimai Doss, in the year 2020, wrote a paper 'Towards Multilingual Sign Language Recognition'[8] in IEEE. Sign language data collection has a big resource scarcity when it comes to information about the differences in the signs of various languages, its variety. It was dealt with by collecting data as subunits of hand shape and hand movements, and a mix of both. The combination of both worked as a good form of assessment.



Field of sign language recognition will continue into the foreseeable future and the number of potential beneficiaries of such solutions is simply too great to ignore. The most common design deploys a CNN network to derive discriminative features from raw data since this type of network offers the best properties for this task. This is achieved by post-estimation models and depth imaging. With this inference, 'Deep Learning for Sign Language Recognition'[9] by Muhammad Al-Qurishi, Thariq Khalid, and Riad Souissi in 2021 IEEE gave an insightful approach.

Dissecting the paper 'Neural Network based Real Time Sign Language Interpreter for Virtual Meet'[10], D.A Janeera, K. Mukilan Raja, U K R Pravin, and M Krishor Kumar in IEEE, gave us the idea of how a translator built within a video conference tool facilitates the conversion of sign gestures made by deaf-mute to captions so that all clients in the video call can understand them. It used a skin segmentation algorithm and has an accuracy of 97%. Being oriented towards commercializing our model on similar online and video conferencing platforms, this was very judicious.
Having overall read around ten papers, our survey tightened towards the technical opportunities available at hand and gave us a look into what we should look forward to as an ideal machine learning model that translates sign language to textual content with high accuracy levels. It taps into the software streams useful to us and user interface ideas to take in for possible commercialization.

III. Translation Model

The flow that the model follows is as such:

1. Image acquisition from the camera
2. Hand-region segmentation
3. Hand-detection and tracking
4. Posture Recognition
5. Classify Gesture
6. Display as Text

Fig 2. Teachable machine training done.

Firstly, for alphabet and number detection, total number of classes are taken, and names, symbols, and number of images in each class are identified.

1. Getting live feed from camera

These are converted to grayscale, and after feature extractions are done, the outputs are classified. The breakdown of the process would include- Getting a live feed with less to no distortions from the web camera for the purpose of data collection.

2. Installation of Media Pipe and Open CV

This is done using the installation of Open CV and Media Pipe. They aid with the required hand tracking functions. OpenCV is a programming library for real time computer vision. The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining. Using production-level tools to automate and track model training over the lifetime of a product, service, or business process is critical to success. Media Pipe is a framework that offers customisable solutions in ML for a live media feed.

3. Hand detection and deployment

This will enhance the detection capability of the model. After the creation of a custom hand tracking module, it is deployed for the location and detection of hands (both left and right).

4. Pre-processing the dataset

The next stage is the pre-processing of feed data collected. Cropping only the hand part of the entire spanning of the webcam, setting backgrounds of similar ratios size-wise, further checking the aspect ratios and making it centralized for the purpose of clarity. Making a white image, and transferring the cropped image on it. Making the image appear at the centre for clarity. This makes it easy for the two types of inputs used: Images and a live detection.

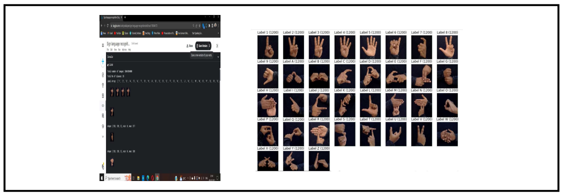


Fig 4. Pre-processing data and collection of inputs from a live feed

5. Saving Images

This step is to take multiple images using the web camera and saving it for creating multiple datasets. We can have different backgrounds for increasing accuracy.

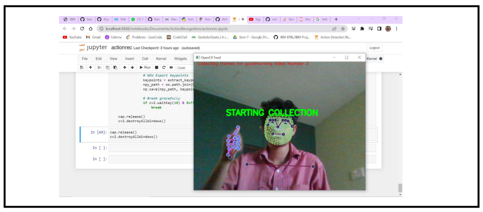
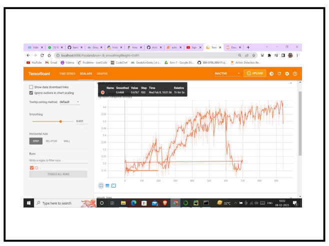


Fig 5. Picture inputs of hand gestures on model.



6. Training the model

After collecting enough datasets to get correct the outputs, we can train the model to prepare the data to read and train accordingly.[11] All pixel data from the datasets provided are converted and processed to be read by the algorithm used. Usually binary conversion helps speed up, and ease the training process. This basis is further extrapolated to a randomized input situation. During random data implementation, we need to keep in mind to normalize the inputs, as in, add noise and changes to make it realistic. Using Mediapipe and Open CV, we take the mentioned randomized inputs and run them on the developed model to get desired outputs in textual way.

The data is basically detected, analyzed by following the pre-processing methods, and comparatively referred to with the previously provided data for as long as inputs are being given. Elaborating, this is done by processing varying probabilities of the data against the ones during training and give results based on the closeness.

The setup we deal with involves two major funtions: being able to translate pictures uploaded by the user, and translating the live action inputs given. This thus makes way to the two modelling styles used, CNN for translation of pictures uploaded, and long short-term memory networks (LSTM) to handle action inputs like phrases from the user.

After training the model, we can test the accuracy of the model's classification capability. The Epoch training accuracy curve is simulated using TensorBoard. The final step is to make sure all the desired outputs are produced.

IV. Outputs

A. *Model training outputs*

Model training outputs are seen to be reaching expectations so far, giving us results in very admissible amounts of time (varying from 30ms to 55ms max). This adds to the advantage of improved conversational support, as users can expect very little delays in the process of translation.

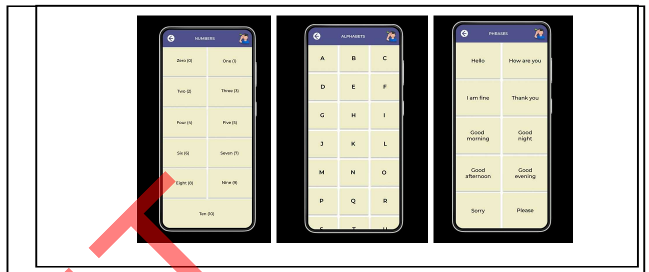


Fig 6. The outputs of numbers and actions collected from model.

B. *Accuracy calculations*

Fig 7. The accuracy epoch curve

The accuracy calculations are taken while simultaneously running the model on tensor flow. The epoch training flow graphs were read to see an average of 60-70% accuracy as shown in Fig 6. This is something we are continuing to work on, making the training process more repetitive and constant, we are looking forward to improved accuracies, to be more beneficial, and to meet user expectations.

V. Mobile Application

A suitable user-interface is suggested to be implemented by the linking of the model trained to translate sign-language to text. This being the final step, helps transfer the technology to users, and also be a viable commercial option.
Our work has both uploading and live feed collection input options, with an added bonus of videos recorded of ourselves teaching the sign-language.
Flutter and Android Studio Dolphin were used as the basis software for the application. Flutter worked as the framework. Android Studio Dolphin is fully built-in the app as an emulator for running the application.

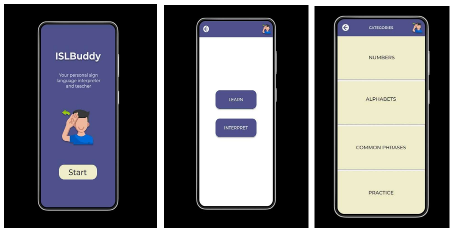


Fig 8. The proposed look of the UI

VI. CONCLUSION

The obvious act of exchanging information between one another, about what we think or feel, depends on how we communicate it.
Sign- Language is the efficient alternative used by the hearing and speech-impaired people. Thus, being able to understand sign language becomes important.
There is an opportunity to use machine learning models to better interpret sign languages, and improve their efficiency.
Awareness of sign language is not extensive, so providing facilities to learn it is necessary.

REFERENCES

[1] Dutta, K.K. and Bellary, S.A.S., 2017, September. Machine learning techniques for Indian sign language recognition. In *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)* (pp. 333-336). IEEE.

[2] Ji, Y., Kim, S. and Lee, K.B., 2017, April. Sign language learning system with image sampling and convolutional neural network. In *2017 First IEEE International Conference on Robotic Computing (IRC)* (pp. 371-375). IEEE.

[3] Bantupalli, K. and Xie, Y., 2018, December. American sign language recognition using deep learning and computer vision. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 4896-4899). IEEE.

[4] Kumar, S.S., Wangyal, T., Saboo, V. and Srinath, R., 2018, December. Time series neural networks for real time sign language translation. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 243-248). IEEE.

[5] Shenoy, K., Dastane, T., Rao, V. and Vyavaharkar, D., 2018, July. Real-time Indian sign language (ISL) recognition. In *2018 9th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1-9). IEEE.

[6] Ku, Y.J., Chen, M.J. and King, C.T., 2019, November. A Virtual Sign Language Translator on Smartphones. In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)* (pp. 445-449). IEEE.

[7] Mariappan, H.M. and Gomathi, V., 2019, February. Real-time recognition of Indian sign language. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)* (pp. 1-6). IEEE.

[8] Tornay, S., Razavi, M. and Doss, M.M., 2020, May. Towards multilingual sign language recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6309-6313). IEEE.

[9] Al-Qurishi, M., Khalid, T. and Souissi, R., 2021. Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. *IEEE Access*.

[10] Janeera, D.A., Raja, K.M., Pravin, U.K.R. and Kumar, M.K., 2021, April. Neural Network based Real Time Sign Language Interpreter for Virtual Meet. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1593-1597). IEEE.

[11] Sarkar, A. (2019, May 28). Garimella, M. (2022) Sign Language Recognition with Advanced Computer Vision. Towards Data Science. https://towardsdatascience.com/sign-language-recognition-with-advanced-computer-vision-7b74f20f3442.

[12]