ISSN: 3068-014X

Research Article

Archivos de Ciencia e Investigación

Sign Language Translation Using AI & ML

Ankita Gandhi, Jaykumar Rameshbhai Makwana*, Viraj Rajendra Bhalodiya, Sharad Kalpeshkumar Patel and Priyank Kiritbhai Upadhyay

Department of Computer Science & Engineering, Parul University, India

***Corresponding author**
Jaykumar Rameshbhai Makwana, Department of Computer Science & Engineering, Parul University, India.

Submitted: 2025 July 17; **Accepted:** 2025 Aug 18; **Published:** 2025 Sep 12

Citations: Gandhi, A., Makwana, J. R., Bhalodiya, V. R., Patel, S. K., Upadhyay, P. K. (2025). Sign Language Translation Using AI & ML. *Arch Cienc Investig, 1*(2), 01-10.

Abstract

The communication of the deaf depends greatly on sign languages. But low awareness and knowledge of these sign languages have limited the effectiveness of communication and interaction and hence integration. In this paper, we describe an AI-centric system for real-time translation of sign languages with support for English, Indian, Turkish, and Arabic signs and alphabets. The system has the ability to recognize and be responsive to the palm gesture of alphabets and numbers in these languages and is ready to enable full-fledged communication. The system leverages advanced machine learning algorithms to ensure portability, which makes it accessible for everyday use and immediately deployable. Key features include a user-friendly interface, as well as the ability to integrate with the majority of e-learning platforms to offer sign language lessons and to be continuously improved based on users' feedback through adaptive learning. Working with the deaf community, language experts, and teachers is part of the effort to fine-tune the system and improve its practical impact. The project will demonstrate the potential of ML and AI technologies in breaking the communication barrier for the deaf towards achieving full accessibility, inclusivity, and active participation in all kinds of social, educational, and professional environments.

Keywords: Artificial Intelligence (AI), Machine Learning (ML), Real-Time Gesture Recognition, CNN, YOLO v11

1. Introduction

Sign Language Translation Using AI and ML is an intriguing project, something that places within the portfolio advanced technologies offering human interpreters real-time translation between spoken language and sign language gestures. The system should be implemented using algorithms based on Artificial Intelligence (AI) and Machine Learning (ML) to accurately recognize applied sign language gestures, including alphabets and numbers, in such sign languages as American Sign Language (ASL), Indian Sign Language (ISL), Turkish Sign Language (TSL), and Arabic Sign Language (ArSL). The main goal of the project is to create an AI and ML model that will be capable of translating from spoken to sign language with portability and user friendliness while bridging educational practitioners or students' communication barriers with the teaching content or lessons. Additionally, it will be able to link to present communication networks and has continuous improvement mechanisms using users' comments and feedback to further enhance the system. The project shall enable the deaf to communicate, hence breaking the barrier between the deaf community and others in various social, educational, and professional settings. The application shall be developed taking ethical considerations in account, which includes issues related to data security and inclusivity. The deaf people will also be included in the development; therefore, a sign language expert and so cultural sensitivities in all rounds of the practical application will be considered to enhance its practical utility.

1.1 Research Background

A survey of research literature sheds light on the state-of-the-art technologies used for automatic recognition and translation of sign languages. In “A Survey on Sign Language Machine Translation” (2023), for instance, authors explained a sign language video-to-text translation system that they built, with its BLEU-1 score equating to 53.97; however, its current performance lags due to the strong reliance on publicly available datasets. In a peer-reviewed study “Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning” (2021), the model is trained to recognize hand movement in four sign languages: ASL, ISL, TSL, Arabic; and achieves an impressive 99% accuracy rate. The “An Argentinian Sign Language Dataset” (2023) used the LSA64 dataset for its translation into Argentinian Sign Language with a successful 95.95% accuracy rate, although it applies to no other

sign language. There is indeed a version described regarding “Sign Language Recognition System” (2023) wherein they have indicated the integration involved OpenCV and MediaPipe Holistic with lack of explicit results. The work by "Indian Sign Language Recognition Using Mediapipe Holistic" (2023) worked with Speeded Up Robust Features and Support Vector Machines and achieved an accuracy of 85% which opens perspectives up for AR and VR technologies. Each investigation is within the dynamics of the landscape of sign recognition emphasizing the need for advanced approaches, varied datasets, and further research that can be applied more generally [5].

2. Methodology

2.1 Data Collection and Dataset

For this research, a custom dataset was created to recognize and classify hand gestures in American Sign Language (ASL). The dataset includes a wide range of hand gestures corresponding to various words and phrases commonly used in ASL. Data collection was carried out using real-time videos captured from diverse sources, including both indoor and outdoor environments to increase variability and robustness. Each video was recorded with high-resolution cameras, ensuring that the fine details of hand gestures were captured accurately. The dataset was carefully labeled with each gesture corresponding to its ASL meaning, providing annotated images and videos that would serve as the basis for model training.

2.2 Preprocessing

Images and videos were collected and preprocessed before training to help the models learn well and generalize well. Images were reduced to a certain specific resolution to maintain uniformity and reduce computation. Techniques of background nullification were applied to remove background noise and superfluous artifacts; this helped keep the hand gestures by themselves. The dataset was balanced with augmentation techniques such as rotation, flipping, and scaling. Different variations of hand gestures were also simulated. Additionally, pixel values were normalized to standardize the data, ensuring that the input would be fair and unbiased, based on the difference in lighting conditions and camera settings.

2.3 Model Selection

The major models applied for gesture recognition were the different YOLO (You Only Look Once) versions such as v5, v7, v8, v10, and v11. It has been chosen due to the efficiency that it has already proven with real-time object detection tasks. Since the presentation requires that the hand gesture be quickly and accurately localized and classified, YOLO's architecture is well-suited for this task. Choosing multiple versions of YOLO was decided so that any changes in performance over the different iterations could be investigated. Another benchmark used was Google Teachable Machine which, having great ease of implementation and capable of simultaneously training deep learning models with ease and without high demands on computational resources, also provided a benchmark for the comparison.

The models were trained on the ASL dataset as provided above after they were split into training, validation, and test sets to make the models generalize to unseen data. It was run on GPUS due to speed requirements in computation. For the YOLO models, learning rate and epochs were tuned with the batch sizes. The starting learning rate was taken as 0.001, while the batch sizes usually depended on the model and resource constraints. I iterated over 100 epochs; the model was based on performance carried out at each epoch to avoid overfitting. It was trained on a scalable platform that offers cloud resources for large datasets that are computationally expensive to run and scale.

2.4 Evolution

It used another hold-out test of images and videos to gauge how well it generalized. The performance was scored with accuracy, precision, recall, or F1. All of them surveyed how accurate the model was on its affirmative predictions, though precision and recall theorize correctly detecting gestures with the least number of false positives. F1 balances between these two, ensuring all bases are covered in estimating predictive power and can be quite general in real-life applications. Other models were tested for real-time applicability.

2.5 Comparison and Analysis

The different versions of YOLO performance were compared against Google Teachable Machine to identify the most effective model for ASL gesture recognition. In general, their performance was compared on their ability to correctly classify ASL gestures, particularly taking into consideration the time taken to make detections and how robust these models were towards changes in the environment. Indeed, the results of this work demonstrated that the increased versions of YOLO, v8, v10, and v11, outperformed the earlier v5 and v7 in terms of achieving high positive predictive values. The Google Teachable Machine well demonstrated its competitive performance vis-a-vis set up that is simpler and faster but lags in terms of precision and choice of modeling complex gestures. This analysis throws up the very important trade-offs on model complexity versus accuracy versus speed in real-time applications critical for converting sign languages.

2.6 System Architecture

The core system is proposed to support a functionality which will process signs in American Sign Language and translate it into text on a real-time basis, prepared with the use of AI and ML from three primary modules working coherently to carry the required operations on sign language gestures, present the output text, and deliver the processed output in a user-intuitive manner. These major building blocks are labeled Data Collection & Preprocessing, Model Training & Evaluation, and Real-time Translation System.

2.7 Data Collection and Preprocessing

The allusion part of the system is based on a self-created ASL dataset with many gestures included in it. Each gesture is labeled so that in the future it would represent some specific word or phrase. Data collection is followed up by several image preprocessing steps to

get everything ready for model training. Image Normalization: This means resizing the images then converting them to grayscale and tuning them to meet the model's input format.

Model Training and Evaluation: The training process is done to apply the data augmentation technique. For now, the model needs to be used for detecting whether the right values are being predicted. Evaluate the model for finesse, in this case, further checking the model fitting metrics produced for bias and variance.

Multiple versions of YOLO (You Only Look Once) models can be used for training to achieve real-time ASL recognition. Each of them has its unique advantages with respect to speed and accuracy on object detection tasks.

For the purpose of comparison, YOLO v5 was considered for the baseline model. YOLO v7 and YOLO v8 indicate better performance with high detection accuracy and great results in translations.

2.8 Model Training and Evolution

Though YOLO v10 and YOLO v11 are newer, they have inconsistencies in some cases; therefore, the performance of these networks will be less stable for specific ASL signs.
Perform training of YOLO models based on the annotated dataset aimed at recognition and detection of ASL gestures within images and video frames. Evaluate their work according to standard metrics: accuracy, precision, recall, F1 score. Unseen data separated from training is used for unbiased evaluation to compare all versions of the model fairly and comprehensively.

2.9 Real-time Translation System

The real-time translation system records live video of ASL gestures and subsequently passes it through the following layers of the system:
Input Layer: The capture of video frames through a webcam or any external camera; further preprocessing of frames is done in terms of resizing and normalizing them as per the requirements of the input dimensions of the YOLO models trained by the authors.

Gesture Detection: Analyzing the preprocessed frames by the YOLO models to locate hand gestures corresponding to a specific ASL sign.

Translation Layer: Output of the YOLO models, which is classification output, is used as the means of translating detected gestures into text. Post-processing might be performed to improve accuracy, as well as the contextual output.

Output Display: The translated text is delivered in a GUI, where the end-user will be able to use it to perceive the real-time output. This particular feature can also be further integrated into more extended communication systems for enabling smooth conversations between people with or without hearing impairments.

In this section, we give results of the project on AI- and ML-powered sign language translation, presenting a comparison between different versions of YOLO (v5, v7, v8, v10, and v11) and the Google Teachable Machine for American Sign Language (ASL) recognition. The principal goal of this work is to evaluate and compare the performance of each model regarding their accuracy in recognizing sign language in real-time based on our own ASL dataset.

2.10 Flowchart

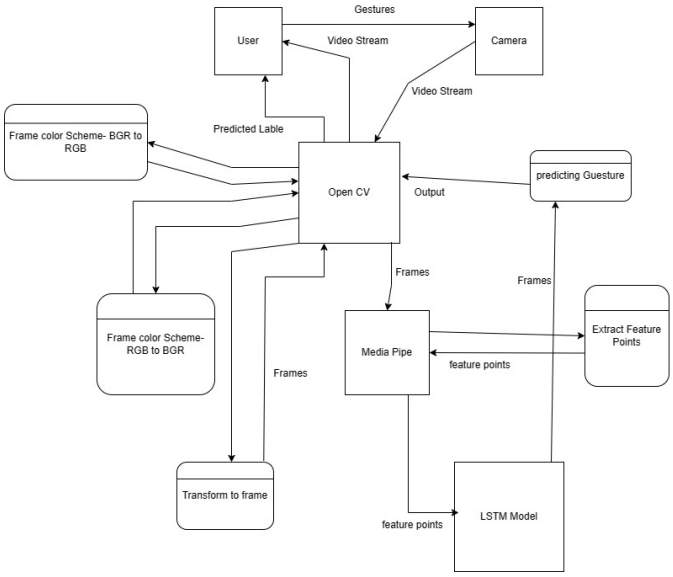

Figure 1: Data Flow Diagram

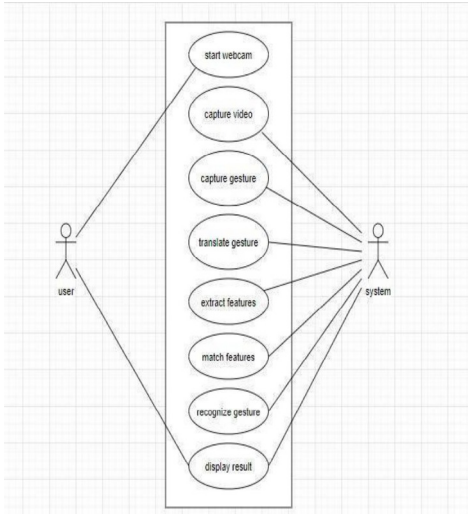

Figure 2: Use Case Diagram

3. Results

The paper throws light on the results of our project on translating sign languages through AI and ML, emphasizing a comparative assessment of the different YOLO versions (v5, v7, v8, v10, and v11) and Google Teachable Machine for the recognition of American Sign Language (ASL). The main goal was to evaluate as well as compare the accuracy of each model for real-time recognition of sign language based on a custom dataset prepared on ASL.

3.1 Dataset and Model Training

A custom dataset for training was created to include an array of gestures in American Sign Language accurately and appropriately labeled with annotations. The dataset includes a balanced distribution of gestures for different words or phrases to make sure that enough data is provided for the learning models.

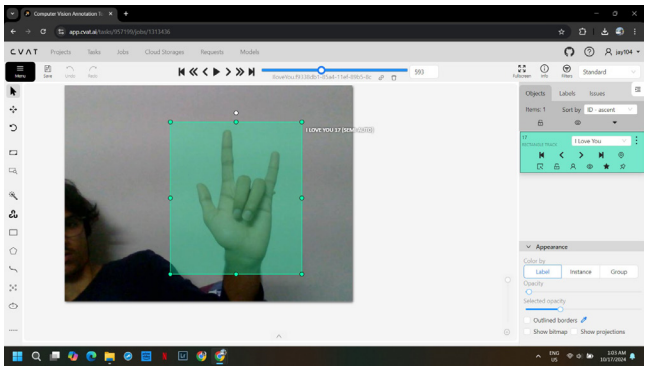

Figure 3: Dataset Creation

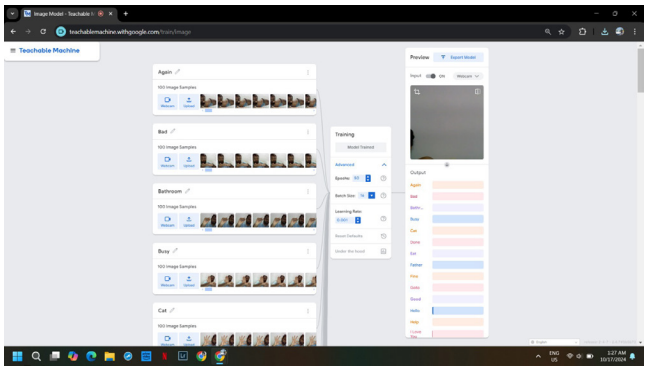

Figure 4: Model Training

3.2 Yolo Versions Comparison

ASL was the dataset that was used to train YOLO models (v5, v7, v8, v10, and v11). One main change was noticed in the area of accuracy amongst these versions. YOLO v7 and v8 performed better than the other models in terms of both detection speed and accuracy; they thus granted the best performance for ASL recognition. These two models showed best results in the recognition of signs in varying conditions, such as different lighting and hand positions, which are standard real-time sign language recognition pitfalls.

Feature	YOLOv5	YOLOv7	YOLOv8	YOLOv10/ YOLOv11	Google Teachable Machine
Developer	Ultralytics	Original YOLO authors	Ultralytics	Unofficial/ Speculative	Google
Release Date	2020	2022	2023	Future/ Experimental	2019
Architecture	Efficient PyTorch model	E-ELAN, extended blocks	Anchor-free detection	Speculated improvements	Simple architecture, no coding
Use Case	General-purpose detection	Real-time object detection	Unified detection & segmentation	Niche or experimental tasks	Educational & prototyping
Ease of Use	Moderate	Moderate to advanced	Easy (good docs)	Advanced (if custom)	Very easy (no code)
Model Customization	Highly customizable	Moderate to highly customizable	Very customizable	Custom tweaks likely needed	Minimal/no customization
Training Environment	Coding required	Coding required	Coding required	Coding required	No coding required
Accuracy	Good	Excellent	Very good to excellent	Speculative	Fair
Real-time Performance	Very fast	SOTA for real-time performance	Fast, real-time performance	Speculative	Not designed for real-time

Figure 5: Yolo Comparison Table

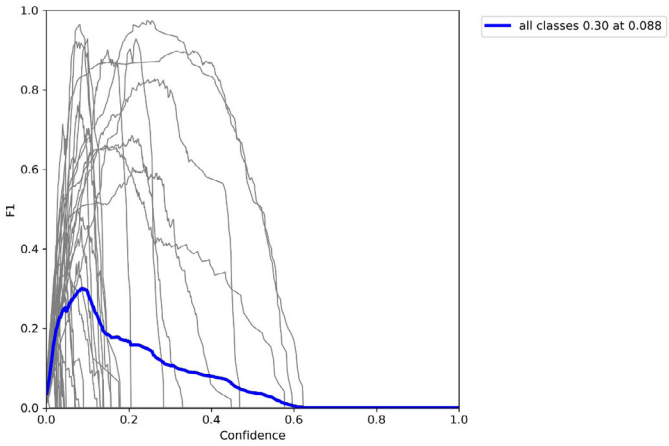

Figure 6: R Curve

The efficiency of v5 is retained, although with slightly lower accuracies as compared to YOLO v7 and v8 versions. In newer versions of the model, v10 and v11 show some inconsistencies and weaker results under certain conditions, hence slower recognition. Google Teachable Machine has also been tested and compared for performance, but the accuracy typically places relatively lower than that of the YOLO versions, more specifically real-time detection and translation.

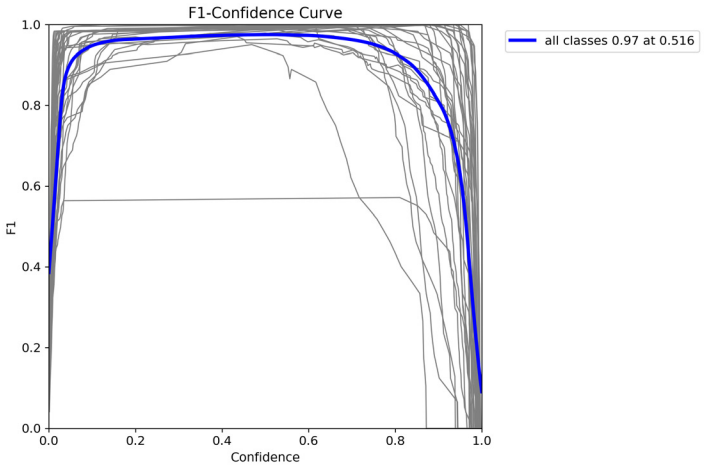

Figure 7: F1 Score

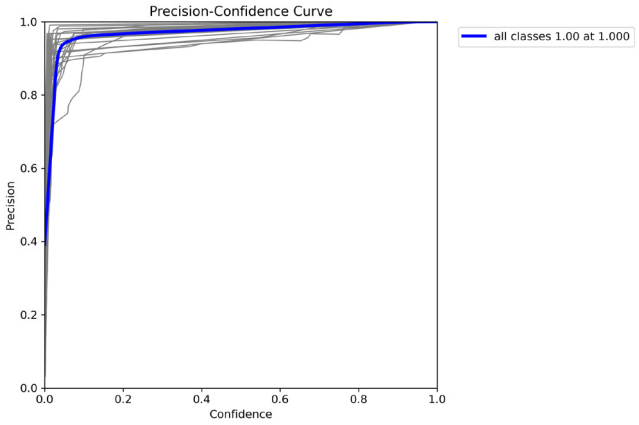

Figure 8: Precision Curve

3.3 Accuracy Metrics

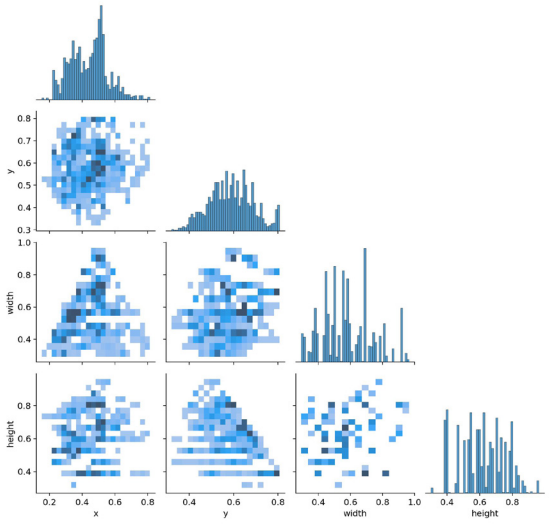

Figure 9: Confusion Metrics

To assess the models, metrics like accuracy, precision, recall, and F1 score were computed. YOLO v7 and v8 have the best accuracy, with YOLO v7 being marginally better in precision and recall. The Google Teachable Machine, simple and quick for initial testing, would be less consistent across different signs and particularly less so with the more complex or less frequently used ASL gestures.

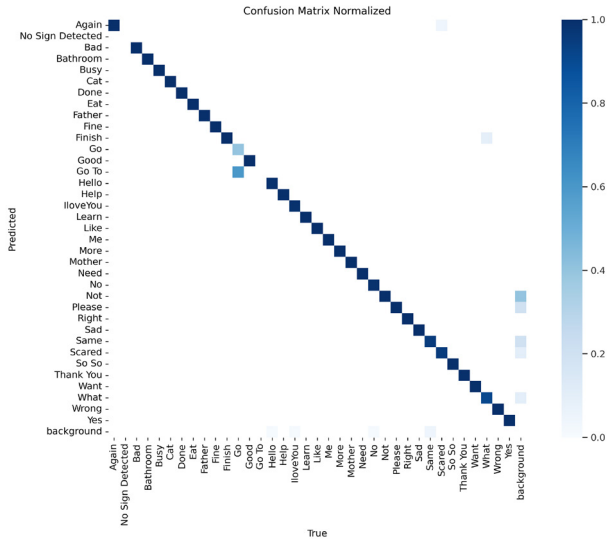

Figure 10: Confusion Matrix

3.4 Real-Time Performance

Figure 11: Detected Images

In real-time, sign detection lags behind not more than the previous second. It marks YOLO version 7 and version 8 as prospective models meant to replace the static systems with real-time translation applications. These models achieve high frame rates and, when tested on video streams, too low latency. It thus makes them applicable to such real-time applications.

Figure 12: Detection

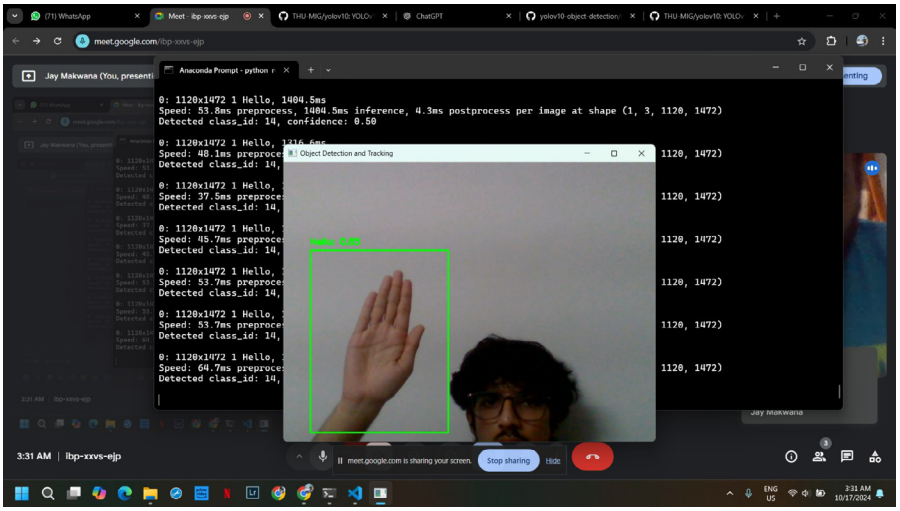

Figure 13: Real Time Detection

3.5 Limitations and Areas for Improvement

While the results proved promising, several limitations shadow them. For some signs, especially due to the complexity emerging from hand orientation and motion, just some of the YOLO versions, among which include v10 and v11, performed rather poorly. The dataset, though quite extensive, may still be broadened to include more variations of signs, hand shapes, and backgrounds for generalization towards better model performance. Furthermore, more tuning of models would be appropriate since it still is real-world application, and, normally, lighting and noise conditions vary very considerably.

3.6 Future Directions

Future work will focus on making the dataset far more diversified—with various samples on the table—and range further expanded on ASL signs to be part of training. More advanced machine learning models like transformers or hybrid CNN and LSTM-based models could really lend weight to the efficiency of such a system in handling complex sign language gestures. Additionally, user feedback will be collected, such that systems are intuitive and can be accessed by users in their real-world settings.

4. Conclusion

To sum up, this paper discusses the tremendous potential that arises from the application of methods founded upon Artificial Intelligence and Machine Learning in translation questions of sign languages. Comparison research Between versions YOLO v5, v7, v8, v10, and v11 and Google Teachable Machine in a bid to enhance the detection American Sign Language recognition was carried out. The models for sign language detection and real-time

translation into text were trained using the proposed ASL gestures dataset. It was observed that the different versions of YOLO and Teach By Example had completely heterogeneous levels of accuracy, with versions v7 and v8 offering statistically the best performances both in terms of speed in detection and detection accuracy hence beating other YOLO versions and the Teacher. This research identifies the critical impact of dataset quality, model choice, and evaluation affordances for the optimization of sign language recognition systems. Further research will lead to more improvements in the real-time translation capacity of the system with the continual improvement of the dataset and upgraded models. This project is proof of the way in which AI and ML can be harnessed towards the provision of more inclusive and accessible communication- bridging the gap between the deaf and hard-of-hearing community and the rest of society. As we continue to develop and improve these systems, we will be making increasingly better accommodations for facilitating more seamless communication, which in turn shall create an atmosphere of inclusivity for the people who everyone has always felt were left out, thus empowering individuals with diverse needs around communication and helping to build an interconnected and loving world [1-26].

Acknowledgements

This study benefited from the support of Prof. Ankita Gandhi through his insightful recommendations and academic direction. He has also been very critical in a kind way and has thus motivated us with his pragmatic attitude. He gave us the constant encouragement that gave us the drive we required to put in a lot of effort.

References

1. Núñez-Marcos, A., Perez-de-Viñaspre, O., & Labaka, G. (2023) A survey on Sign Language machine translation. *Expert Systems with Applications, 213*, 118993.
2. Halder, A., & Tayade, A. (2021). Real-time vernacular sign language recognition using mediapipe and machine learning. *Journal homepage: www. ijrpr. com ISSN, 2582*(7421), 2.
3. Avina, V. D., Amiruzzaman, M., Amiruzzaman, S., Ngo, L. B., & Dewan, M. A. A. (2023). An AI-Based Framework for Translating American Sign Language to English and Vice Versa. *Information, 14*(10), 569.
4. Hafeez, A., Singh, S., Singh, U., Agarwal, P., & Jayswal, A. K. (2023). Sign Language recognition system using Deep Learning for deaf and dumb. *International Research Journal of Modernization in Engineering Technology and Science.*
5. Das, S., Imtiaz, M. S., Neom, N. H., Siddique, N., & Wang, H. (2023). A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. *Expert Systems with Applications, 213*, 118914.
6. Qahtan, S., Alsattar, H. A., Zaidan, A. A., Deveci, M., Pamucar, D., & Martinez, L. (2023). A comparative study of evaluating and benchmarking sign language recognition system-based wearable sensory devices using a single fuzzy set. *Knowledge-Based Systems, 269*, 110519.
7. Alsulaiman, M., Faisal, M., Mekhtiche, M., Bencherif, M., Alrayes, T., Muhammad, G., ... & Alfakih, T. (2023). Facilitating the communication with deaf people: Building a largest Saudi sign language dataset. *Journal of King Saud University-Computer and Information Sciences, 35*(8), 101642.
8. Shin, J., Hasan, M. A. M., Miah, A. S. M., Suzuki, K., & Hirooka, K. (2024). Japanese sign language recognition by combining joint skeleton-based handcrafted and pixel-based deep learning features with machine learning classification. Comput. *Model. Eng. Sci, 139*(3), 2605-2625.
9. Shin, J., Musa Miah, A. S., Hasan, M. A. M., Hirooka, K., Suzuki, K., Lee, H. S., & Jang, S. W. (2023). Korean sign language recognition using transformer-based deep neural network. *Applied Sciences, 13*(5), 3029.
10. Jintanachaiwat, W., Jongsathitphaibul, K., Pimsan, N., Sojiphan, M., Tayakee, A., Junthep, T., & Siriborvornratanakul, T. (2024). Using LSTM to translate Thai sign language to text in real time. *Discover Artificial Intelligence, 4*(1), 17.
11. Amin, M., Hefny, H., & Ammar, M. (2021). Sign language gloss translation using deep learning models. *International Journal of Advanced Computer Science and Applications, 12*(11).
12. Goyal, K. (2023). Indian sign language recognition using mediapipe holistic. *arXiv preprint arXiv:2304.10256.*
13. Shen, X., Yuan, S., Sheng, H., Du, H., & Yu, X. (2023). Auslan-daily: Australian sign language translation for daily communication and news. *Advances in Neural Information Processing Systems, 36*, 80455-80469.
14. Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., & Gogoi, D. (2023). Real-time assamese sign language recognition using mediapipe and deep learning. *Procedia Computer Science, 218,* 1384-1393.
15. Harini, R., Janani, R., Keerthana, S., Madhubala, S., Venkatasubramanian, S. (2023). Sign Language Translation. *6th International Conference on Advanced Computing & Communication Systems (ICACCS).*
16. Minu, R. I. (2023, February). A extensive survey on sign language recognition methods. In *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 613-619). IEEE.
17. Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7784-7793).
18. Fernandes, L., Dalvi, P., Junnarkar, A., & Bansode, M. (2020, August). Convolutional neural network based bidirectional sign language translation system. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 769-775). IEEE.
19. Park, H., Lee, J. S., & Ko, J. (2020, January). Achieving real-time sign language translation using a smartphone's true depth images. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)* (pp. 622-625). IEEE.

20. Shin, J., Miah, A. S. M., Suzuki, K., Hirooka, K., & Hasan, M. A. M. (2023). Dynamic Korean sign language recognition using pose estimation based and attention-based neural network. *IEEE Access, 11,* 143501-143513.
21. Ronchetti, F., Quiroga, F. M., Estrebou, C., Lanzarini, L., & Rosete, A. (2023). LSA64: An Argentinian sign language dataset. *arXiv preprint arXiv:2310.17429.*
22. Muchada, M. J. (2023). Sign Language Recognition System. *Harare Institute of Technology.*
23. Muchada, M. J. (2023). Advances in Sign Language Dataset and Sign Language Recognition System. *Harare Institute of Technology.*
24. Strobel, G., Schoormann, T., Banh, L., & Möller, F. (2023). Artificial intelligence for sign language translation: A design science research study. *Communications of the Association for Information Systems, 53,* 42-64.
25. Roy, P., Han, J. E., Chouhan, S., & Thumu, B. (2024). American Sign Language Video to Text Translation. *arXiv preprint arXiv:2402.07255.*
26. Rani, R. S., Rumana, R., & Prema, R. (2021). A review paper on sign language recognition for the deaf and dumb. *International Journal of Engineering Research & Technology (IJERT), 10*(10).