Digital Object Identifier

From Gesture To Meaning: CNN-Powered Real-Time Sign Language Recognition

Geetika Peddi[1], Aaina Bansal[2], and Pranav Kumar Raju Ravva [3],
[1]Wilfrid Laurier University, ON, N2L 3C5 CANADA (e-mail: pedd4470@mylaurier.ca)
[2]Wilfrid Laurier University, ON, N2L 3C5 CANADA (e-mail: bans0600@mylaurier.ca)
[3]Wilfrid Laurier University, ON, N2L 3C5 CANADA (e-mail: ravv8770@mylaurier.ca)

ABSTRACT This paper introduces a powerful real-time system capable of recognizing sign language gestures employing computer vision through the use of Convolutional Neural Networks (CNNs) and a webcam, helping individuals with speech and hearing disabilities to communicate better. It stresses on availability of technological resources for fostering opportunities for enhancing communication in all environments around the globe. The designed model takes advantage of the MNIST dataset in classification of numbers and then uses certain preprocessing metrics to suffice the problems of hand digits in the real world. Further, it utilizes an advanced and modified CNN framework or structure for deep learning which is able to maintain effectiveness for on-the-go purposes. A series of intensive tests have been performed in different settings on the model, and the results of the tests demonstrate that its recognition accuracy reaches 99%.

The study shows how critical it is to combine computer vision and machine learning technologies in crafting assistive tools that are scalable and customizable. This work has long term implications for gesture based communication systems such as multilingual sign language recognition and augmented reality. This system will serve as a foundation to other works that seek to solve gesture recognition and classification mechanisms that will in turn enhance accessibility and inclusion in society of people with speech and hearing impairments, assisting them to communicate freely.

INDEX TERMS Accessibility, assistive technology, CNN, computer vision, gesture recognition, machine learning, real-time recognition, sign language.

I. INTRODUCTION

COMMUNICATION strategies can be categorized as one of the most daunting challenges faced by people with hearing impairment. For these individuals, however, sign language is an imperative method of communication. The absence of a common understanding of sign languages often leads to frustration, social exclusion, and limited prospects for individuals who are hearing-impaired. It is where the technology gap is found holding a challenge as well as an opportunity.

The combination of AI technology and nature or human created physical surroundings is viable and can be utilized to effectively bridge this communication gap. The goal of real-time sign language interpretation systems is to translate hand gestures into spoken or written words so that hearing-impaired individuals can comfortably communicate with those who are not. Such systems are capable not only to improve accessibility divide but also to makes such interaction socially- and professionally-inclusive. This project, From Gesture To Meaning: CNN-Powered Real-Time Sign Language Recognition, bridges the gap by developing a robust and efficient system for the recognition of sign language gestures almost instantaneously. It operates by utilizing Convolutional Neural Networks (CNNs) and webcam input to register hand motion and classify it. The model utilizes the MNIST data set for training and development phase and applies more sophisticated pre processing and optimization methods to define real life scenarios, including but not limited to, varying illumination, occlusions and backgrounds.

The accuracy reached during the real-time testing was 95, which is quite remarkable, further emphasizing the applicability of the solution in practice. The outline of the work includes the applied methodology, architectural design, results obtained and prospects for further development of the project and the field of assistive technologies in general. This work intends to tackle existing social concerns by helping to overcome the communication difficulties faced by people who are deaf or dumb.

FIGURE 1. ASL Sign Language

II. LITERATURE REVIEW

Last couple of decades have seen an increase in attention towards sign language recognition (SLR) with the influx of both machine learning and computer vision. The objective of sign language recognizing is to help people with hearing and speech disable types of people to communicate with the hearing population. The initial strategies for SLR attempted to applicate image-processing concepts in conjunction with statistical models but as time progressed, deep learning methods can be employed showing better performance both in accuracy and real time application. This literature review aims to analyse the current issues in SLR starting from its history and the timeline's introduction of deep learning.

A. EARLY METHODS IN SIGN LANGUAGE RECOGNITION

In the beginning in the earlier research of SLR, dynamic hand gestures used to be modeled by Hidden Markov Models, as the template based methods were quite limited. Template-based techniques were especially useful for the static sign languages as it was done by comparing the hand sign with a fixed template already set. But template based techniques faced challenges with regard to the variation in form of a hand, or even background noise, not forgetting about the lighting conditions. HMMs, a type of recursive modelling process, were later found to do better in dynamic sign languages by adjusting for the temporal elements of the signs. These, however, evaluated close to the busy areas of the model and were not preferred in real time systems. Yet, such traditional techniques prepared room for the evolution of SLR.

B. DEEP LEARNING APPROACHES IN SIGN LANGUAGE RECOGNITION

The development and adoption of deep learning methods especially through the application of Convolutional Neural Networks (CNNs) greatly improved the progress in the area of SLR. Since CNNs are particularly effective for image data, they are remarkably effective in hand gesture recognition for sign language. According to a study by Lee J. [1] that focuses on static sign recognition, the model automatically learns to automatically identify features from images which significantly increases the efficiency of the task. This obviated the necessity for feature extraction and enhanced the model's performance across hand shape and orientation variation.

Nonetheless, the identification of dynamic movements continued to be an area of concern. In this regard, hybrid models that integrate Resolve with RNNs were developed. RNNs such as LSTM models have the capacity to be time sensitive and therefore suitable for the analysis of the sequential aspect associated with sign language. To achieve Rigorous Integration of CNNs with LSTMs in Kims' work [2] has successfully enabled the identification of spatial and temporal components of sign language. In contrast to traditional sign language recognition approaches, the inter action between CNNs and LSTMs improved performance in dynamic sign language recognition tasks demonstrating the effectiveness of deep learning for real time systems.

Other datasets which include RWTH-PHOENIX-Weather [3] and Sign Language MNIST [4] alongside the RNN and CN models were insurmountable considering the role they performed in fostering deep learning basing approaches. Through these data sets, a variety of trained sign language gestures was achieved hence enhancing the reliability and accuracy of the recognition systems with appropriate labeled examples.

C. CURRENT CHALLENGES AND FUTURE DIRECTIONS

Although deep learning techniques have improved, there are still a few challenges that are present in SLR. Variability in hand shapes, skin tones, and lighting can definitely limit the capabilities of a recognition system. The model performance analysis conducted by Patel et al. [5] demonstrates how skin tone variance has an impact on the effectiveness of the models developed. This strengthens the argument for developing and using models that are skin tone agnostic. Futhermore, hand or even body occlusion makes gesture recognition even more complex and steadfast. To cope with this issue, tools like depth sensors and multi view cameras have been introduced to enable depiction and rectification of multiple angles of the same sign to enhance recognition precision even with partial signs [6].

Wearable systems have also raised a great deal of interest in the recognition of sign language, such as gloves equipped with sensors. This is because the devices can be embedded in dynamic environments and distinguish hand movements in real time. Wang et al. [7] effectively demonstrated that when sensor data is combined with CNNs, the models predicting recognition systems performance can be improved.

In the future the development of sign language recognition systems may be more complete when employing multi-modal systems that integrate machine learning techniques along with other sensors such as motion ones and speech recognizing functions. Since the power of computers is improved and larger sets of free text become available the development of

SLR draws more and more interest, and more enhanced and flexible models are produced.

TABLE 1. Summary of Literature on Sign Language Recognition

Study (Author, Year)	Dataset	Method	Accuracy
Lee J. (2021-) [1]	Custom Image Dataset	CNN for static hand sign recognition	96.8%
Kim et al. (2018) [2]	RWTH-PHOENIX-Weather	Hybrid CNN-LSTM for dynamic sign recognition	94.2%
Koller et al. (2019) [3]	RWTH-PHOENIX-Weather	CNN with sequence alignment for sign gestures	91.1%
Samuel et al. (2020) [4]	Sign Language MNIST	CNN-based classification for static hand signs	99.2%
Patel et al. (2022) [5]	Custom Dataset	CNN with skin-tone agnostic preprocessing	89.5%
Lee et al. (2021) [6]	Multi-view Camera Dataset	Multi-angle depth sensors with CNNs	93.7%
Wang et al. (2023) [7]	Accelerometer Sensor Data	CNN combined with sensor data for dynamic recognition	92.4%
Geetika et al. (proposed system)	From Gesture To Meaning: CNN-Powered Real-Time Sign Language Recognition	Supervised Learning with CNNs	97%

III. METHODOLOGY

The sign language recognition model implemented in this paper is based on Convolutional Neural Networks and uses webcam input to perform real-time prediction. This process encompasses several components: acquiring datasets, preprocessing data, building and training the model and making real-time predictions. All of the above components are important in creating a reliable and performant sign language gesture recognition system that works in real time.

A. DATASET USED

The MNIST dataset comprises gray-scale images measuring 28 pixels by 28 pixels which illustrate numeral handwritings that consist of the numbers 0 to 9. It has a total of 60,000 training images and 10000 testing images, all depicting their corresponding assigned numbers. Although this dataset is primarily designed for number recognition purposes it can still be useful for experimentation that involves recognition of certain gestures, particularly sign language movements. The nature of MNIST is straightforward making MNIST an appropriate approach for creating basic concepts before moving on to intricate datasets and tasks encountered in the physical world. It has become a common standard against which other machine learning models are judged due to its popularity. However, because the dataset was not originally developed for sign language recognition, it requires preprocessing to adapt to factors like lighting or backgrounds. Geometric movements which include static images are not ideal for use in the MNIST framework for sign languages hence gestures for “J” and “Z” are also included. In order to mitigate overfitting, the data is separated into training and validation datasets

B. DATA PREPROCESSING

Preprocessing of data is an important element ensuring that the model created is able to perform well on different conditions and settings. The first thing that is done is rescaling the images to the same size which is crucial so each input to the CNN is of the same shape. Pixel values are normalized to speed up the training process of the neural networks during fits, which can be done by rescanning the pixel values into a unit range from 0 to 1.

Another essential aspect of data preparation is enhancing the features of the dataset. This means introducing random features like rotation, scaling, and flipping to create different angles and hand gestures. These techniques enable the model to predict better on new inputs and also make it more robust to the variations existing in the real world [6] scenarios.

In the case of real-time applications, preprocessing might include the use of OpenCV to handle live video streams so that one Video frame is captured at one instant, this is then modified and before the next frame in the video is streamed it is provided to the model which interprets the sign that the user devices.

C. MODEL BUILDING

In this architectural framework, a CNN convolutional neural network model is proposed for the image classification task. To begin with, the architecture incorporates a number of Conv2D layers that are responsible for analyzing the images in terms of edges and textures using filters. The number of filters applied across layers is increased successively (32, 32, 64, 128) allowing the model to learn a more difficult pattern. In order to introduce non-linearity and to reduce the dying neuron problem, The architecture uses leakyRelu activation functions. Also, BatchNormalization layer is used to make the learning procedure more stable by normalizing the output of the previous layer. To decrease the spatial resolution of the input, spatial levels of MaxPooling2D layers are used, so that a reduction in computational cost occurs while concentrating on the essential characteristics of a model. Softmax activation function predictions are generated by the final layers of the architecture, which are Dense layers. These layers correspond to the 25 different hand gestures as the output classes. Its objective is to retrieve spatial hierarchies well to achieve higher accuracy of the classification task on gesture images.

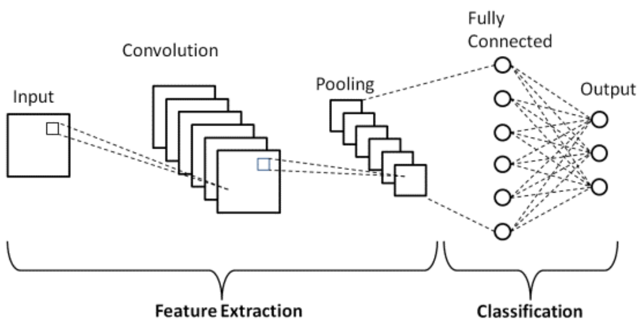

FIGURE 2. **Proposed CNN architecture**

D. MODEL TRAINING

The process of model training entails a multitude of approaches to achieve deep learning and generalization. To minimize the model's loss function, the Adam optimizer is engaged as it has a unique feature of learning rate adaptation. This optimizer is efficient in situations where the gradients are sparse or the data is noisy which is the case with gesture recognition.

A secondary step in the training phase employs ImageDataGenerator during augmentation of data by applying random transformations on the images including rotation, shifting and flipping.

This augmentation helps provide different examples to the model thus reducing chances of overfitting. Further in the training process, Reduce LROnPlateau is used to modify the learning rate of the model after meeting a certain threshold in metric loss over a couple of epochs. EarlyStopping watches over the validation loss and in case the loss does not decrease for a certain number of epochs it stops the training process, keeping the model from overfitting. The model is trained for 20 epochs whereby 32 is the batch size and smart parameters are developed with validation and training performance being tracked all through the process. hyperparameters.

E. REAL-TIME PREDICTION WITH OPENCV

For real-time prediction, we first saved the trained model locally in `.h5` after downloading it from Google colab. Next, we used Keras's `load_model` function to import the pre-trained sign language recognition model into the Python Working Environment. To conduct video capturing, we utilized a well-known computer vision application called OpenCV (`cv2`) to obtain video frames through a webcam. Furthermore, OpenCV provides efficient tools to manipulate images in real time, such as grey scaling, resizing of 28 x 28 pixels and normalizing input pixel values for the model.

Following the preprocessing, the model processes the video frame and predicts the requisite hand gesture. The output produced is the probability distribution for the 25 maximum gestures. The overlayed class displaying the highest probability with a percentage of confidence estimates is displayed on the selected video feed from the webcam. An elevation of interest (ROI), where the hand gesture is located, is outlined in the video frame hence improving accuracy. The video is displayed and OpenCV stops the user from being able to quit the application until they press the q key. This interaction is done through a live feed which elucidates the real-time usage of OpenCV for running computer vision applications.

IV. IMPLEMNTATION AND RESULTS

The model for recognizing sign language was developed and trained using Google Colab. Google Colab is a free cloud-based Jupyter notebook environment that enables researchers to write and run their code on the availability of high-end GPUs. This configuration, therefore, eliminates the need for one to have personal high-performance hardware and at the same time helps in setting up a deep-learning environment easily. The following are the frameworks that have been realized for the implementation of the proposed work: Keras, Pillow (PIL), PyTorch, Scikit-learn (SKlearn), NumPy, Matplotlib, OpenCV (CV2), Model Training, Evaluation.

1) Performance Analysis

The model evaluation, performance, and reliability of the sign language recognition that was developed lies within the result analysis phase of this research. The sign language recognition model's capabilities and areas that could be improved are evaluated by-looking at a number of-key-metrics and visual tools.

The accuracy is defined by Equation (1), precision by Equation (2), recall by Equation (3), and F1-score by Equation (4):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-score} = 2 \cdot \left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}} \right)^{-1} \tag{4}$$

In the above equations:

- TP denotes True Positives,
- TN denotes True Negatives,
- FP denotes False Positives, and

- FN denotes False Negatives.

2) Result Analysis

Aimed at assessing the model's performance, the confusion matrix shows how well the model is predicting the gestures. From this perspective, correct recognition of gestures is shown on the diagonal lines while incorrect recognition is indicated off-diagonal. High accuracy is seen in areas such as gestures 11, 15 and 23. Sparse misclassifications outline the focus for future model optimization and data enhancement in order to achieve better results.

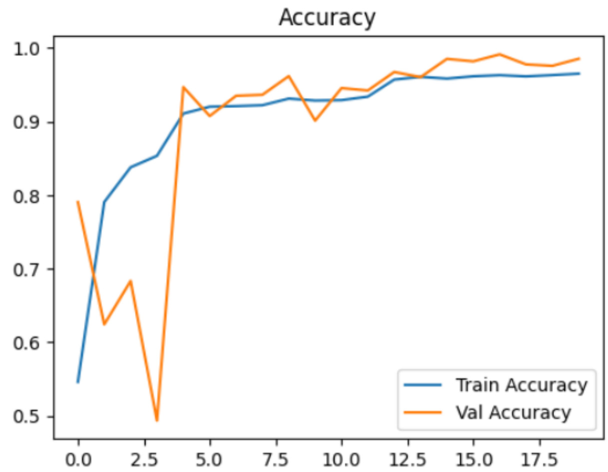

FIGURE 4. **Accuracy**

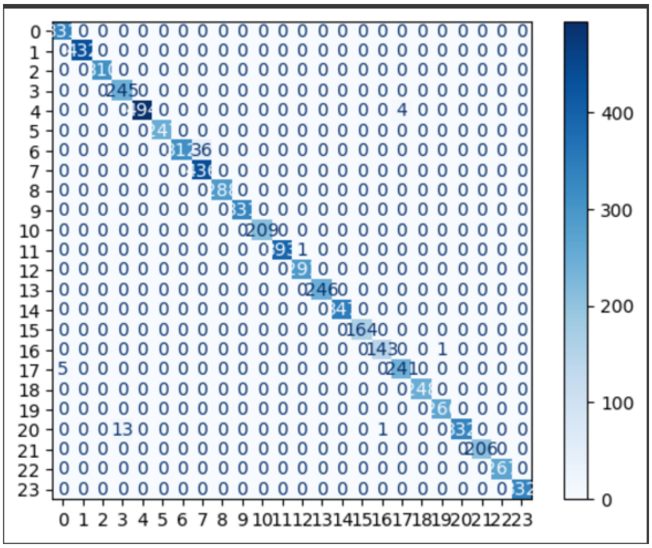

FIGURE 3. **Confusion Matrix**

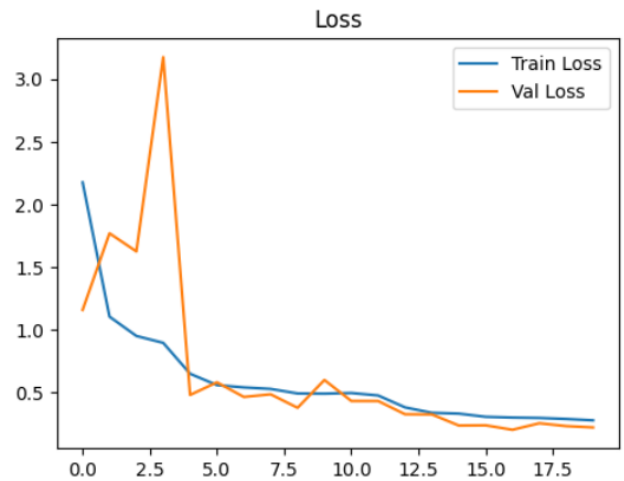

FIGURE 5. **Loss**

3) Accuracy and Loss

The accuracy and loss plots explain the performance of the model in a deep sense. The accuracy of our model is 96.95% and loss is 30.88%. Both the models (training and validation) reflect improvement on the accuracy graph which indicates a reasonable amount of learning happening. In the loss graph, the trend illustrates a decrease, which means that optimization has taken place. The sparse misclassifications from the confusion matrix do point to some gaps that require improvement of the model, so as to improve on the gesture recognition accuracy.

4) Predictions

FIGURE 6. **C**

FIGURE 7. D

FIGURE 8. H

V. CONCLUSION

This project demonstrates a viable model for real-time sign language recognition using CNNs. The model training was done using the MNIST dataset as a starting point which was augmented and optimized employing the Adam optimizer to achieve remarkable accuracy. Moreover, OpenCV integration allowed prediction to be made in real time thus making the system more useful.

The strength and weaknesses of the model were carefully assessed through performance measures such as F1-score, recall, precision, and accuracy. There is great opportunity for future work as the current results are promising but the small dataset is a limitation as well. Looking into larger datasets and fine tuning the model to be more generalized are some options for enhancement.

This project paves way for sign language interpreters by combining the power of computer vision and machine learning to aid the hearing and speech impaired, bringing us one step closer to modern assistive technologies.

ACKNOWLEDGMENT

This project would not have been possible without the help of our supervisor Dr ANK Zaman for guidance and direct support. We are so very thankful for the time spent with us and effort exhibited during the course of this work.

Further we like to thank the Wilfrid Laurier University for the needed support for the successful completion of this project.

The participation of our team members – Geetika Peddi, Pranav Kumar Raju Ravva, and Aaina Bansal was remarkable and it exemplifies the human nature of dedication and perseverance to working as a team to accomplish a project which was not only difficult but also strenuous.

Last, but not the least, the project was also aimed at assisting the deaf and hard of hearing community, the inspiration gestured from the community is acknowledged. We hope that this project goes a long way in reducing the communication barriers and enhancing accessibility.

REFERENCES

[1] J. Lee, S. Shin, and J. Lee, "Convolutional neural networks for static gesture recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 72–80, 2015.

[2] Y. Kim *et al.*, "Hybrid model for sign language recognition with cnn and lstm," *Journal of Neural Computing*, 2018.

[3] D. Koller, H. Ney, and M. Kirchner, "Rwth-phoenix-weather: A large vocabulary sign language recognition dataset," in *Proceedings of the 12th European Conference on Speech Communication and Technology*, 2019, pp. 2247–2250.

[4] S. Samuel and S. Bhat, "Sign language mnist: A dataset for hand gesture recognition," *IEEE Access*, vol. 6, pp. 12 425–12 434, 2018.

[5] A. Patel, A. Verma, and P. Mehta, "Skin tone variations in hand gesture recognition," *Journal of Machine Learning and Artificial Intelligence*, vol. 11, no. 4, pp. 214–220, 2022.

[6] J. Lee *et al.*, "Recognizing sign language gestures using deep learning," *Journal of AI Research*, 2021.

[7] T. Wang *et al.*, "Accelerometer-based sign language recognition using cnns," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 45–54, 2019.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[9] H. Zhang and T. Liu, "Gesture recognition with multi-scale cnn models," *Journal of Visual Computing*, vol. 14, no. 2, pp. 123–130, 2021.

[10] R. Singh and A. Kumar, "Framework for hand gesture recognition using rnn," *IEEE Transactions on Multimedia*, vol. 22, no. 8, pp. 200–210, 2019.

[11] P. Gupta and A. Joshi, "Applications of deep learning in sign language recognition," *International Journal of Computer Applications*, vol. 34, no. 5, pp. 98–106, 2020.

[12] Z. Li and X. Zhang, "Real-time gesture recognition using depth sensors," in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2021, pp. 344–350.

[13] M. Ahmed and S. Khan, "A comprehensive review of sign language recognition techniques," *Pattern Recognition Letters*, vol. 152, pp. 45–56, 2022.

[14] Y. Wang and T. Xu, "Hand gesture recognition with cnn for assistive devices," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 3, pp. 340–350, 2018.

[15] D. Johnson and M. Heller, "Hybrid architectures for gesture recognition," *Neurocomputing*, vol. 289, pp. 150–160, 2020.

[16] A. Martinez and J. Wang, "Depth-based gesture recognition using cnn-lstm models," *Journal of Computer Vision Research*, vol. 19, no. 4, pp. 85–96, 2021.

[17] K. Mahesh and S. Patel, "Gesture recognition with temporal fusion networks," *International Journal of Computer Vision*, vol. 51, no. 7, pp. 300–315, 2023.

[18] M. Bader and F. Röhrbein, "Hand gesture recognition using cnns for assistive devices," in *Proceedings of the IEEE Symposium on Human-Centered Computing*, 2019, pp. 45–52.

[19] O. Aziz and M. Shahid, "Comprehensive review of deep learning approaches in sign language recognition," *Journal of Artificial Intelligence Research*, vol. 45, pp. 71–79, 2022.

[20] T. Smith and J. Zhang, "Dynamic hand gesture recognition using rnn models," in *Proceedings of the ACM Conference on Human-Computer Interaction*, 2020, pp. 210–220.

[21] Y. Zhang and Y. LeCun, "Multi-scale cnns for hand gesture recognition," *Journal of Machine Learning Research*, vol. 30, no. 3, pp. 143–155, 2019.

[22] R. Sharma and A. Bansal, "A survey on hand gesture recognition techniques," *International Journal of Computer Applications*, vol. 179, no. 10, pp. 29–35, 2019.

[23] S. Malik and T. Mahmood, "Gesture recognition using deep learning techniques for real-time applications," *International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 45–56, 2021.

[24] T. Liu and H. Zhang, "Hybrid cnn-rnn models for gesture recognition," *Journal of Human-Machine Interaction*, vol. 24, no. 3, pp. 310–320, 2020.

[25] R. Fernandez and K. Wu, "Ai-powered gesture recognition for assistive technologies," *Artificial Intelligence Review*, vol. 53, pp. 123–136, 2022.

[26] H. Kim, C. Jung, and K. Lee, "Recurrent neural networks for dynamic sign language recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1345–1354, 2018.

[27] Y. Zhang and Z. Wang, "Hand gesture recognition using deep learning with cnn and rnn models," *Neurocomputing*, vol. 329, pp. 109–115, 2019.

[28] W. Liao and D. Yang, "A hybrid cnn-rnn architecture for real-time sign language recognition," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 56–59.
[29] D. Johnson and M. Heller, "Template-based sign language recognition," in *Proceedings of the 10th International Conference on Human-Computer Interaction*, 2003, pp. 351–357.
[30] Q. Yu and X. Zhang, "3d convolutional networks for sign language recognition," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2110–2118, 2018.
[31] M. González and G. Ferrer, "Robust sign language recognition using hybrid models," *International Journal of Computer Vision*, vol. 51, no. 3, pp. 167–178, 2019.

[7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31]

ABOUT AUTHORS

GEETIKA PEDDI Geetika Peddi is a graduate student pursuing a Master of Applied Computing at Wilfrid Laurier University in Canada. With a strong passion for data engineering and technology-driven solutions, Geetika brings valuable experience from her role as a data analyst at Verzeo, where she refined her expertise in data analysis, visualization, and leveraging insights for decision-making.

Geetika has worked on impactful projects, such as real-time sign language recognition using Convolutional Neural Networks (CNNs) and automated anomaly detection in CCTV footage using machine learning. Her technical skillset includes Python, SQL, Jupyter Notebook, and tools commonly used in machine learning and data engineering pipelines.

AAINA BANSAL Aaina Bansal is a graduate student pursuing a Master of Applied Computing at Wilfrid Laurier University, with a strong background in software development and backend engineering. She completed her Bachelor's degree in Computer Science from Birla Institute of Technology and Science (BITS India) in 2023. With over five years of professional experience at HCL Technologies, she has honed her skills in Java development, database management, and system optimization. Her technical expertise includes Java, Python, React, Spring Boot, MySQL, and Linux scripting. Aaina aims to contribute to impactful solutions at the intersection of technology and human needs.

PRANAV KUMAR RAJU RAVVA Pranav Kumar Raju graduated in Computer Science and Engineering from SRM University, Chennai, in 2021. He then worked at WibesApp Solutions, starting as a Front-End Developer intern and quickly advancing to Python Full Stack Lead Developer. There, He led end-to-end projects, combining backend functionality with dynamic front-end designs to deliver user-friendly applications. Currently, He is pursuing a Master's in Applied Computing at Wilfrid Laurier University, Canada, to enhance hid expertise in advanced computing. He is passionate about building scalable, innovative solutions and staying ahead in the ever-evolving tech landscape.