Journal of Information Systems Engineering and Management
2025, 10(26s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Machine Learning Approaches for Barakhadi Recognition of Devnagari Sign Language

Ms. Deepali R. Naglot[1], Dr. Deepa S. Deshpande[2]
[1]*Assistant Professor, Jawaharlal Nehru Engineering College, MGM University, Chh.Sambhajinagar, Maharashtra, India.*
[2]*Professor, Jawaharlal Nehru Engineering College, MGM University, Chh. Sambhajinagar, Maharashtra, India.*

ARTICLE INFO

Received: 24 Dec 2024

Revised: 18 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Hearing impairments affect millions of individuals worldwide, including approximately 63 million people in India who depend on sign language for communication. However, a lack of awareness and understanding among the general population generates significant communication challenges. Recognizing Devnagari Sign Language (DSL), widely used in India, remains an underexplored area despite advancements in Sign Language Recognition (SLR). Key challenges in SLR include variations in lighting, diverse hand gestures, and complex backgrounds, which hinder effective real-time translation. This study introduces a system for recognizing the complete Barakhadi of DSL, comprising 408 characters. A robust dataset of static hand signs, captured in varied environmental conditions, is utilized for training. Multiple classifiers such as SVM, FFNN and Random Forest are leveraged and compared to identify the DSL Barakhadi characters. Customized RNN-LSTM model evaluated by employing diverse optimization strategies. The system's performance is measured using precision, recall, and F1-score metrics. The proposed RNN-LSTM model, optimized using the Adam algorithm, achieves an outstanding accuracy of 99.62%, demonstrating its effectiveness in DSL recognition.

Keywords: Barakhadi, Sign language recognition SLR, LSTM, FFNN, Devnagari Sign Language DSL

INTRODUCTION

As stated by the World Health Organization, Hearing impairment impacts nearly 5% of the global population. Approximately 63 million people in India are affected by hearing impairments. Individuals with auditory disabilities primarily depend on sign language for their means of interaction; however, this can pose challenges for those who are unfamiliar with it. [1]. Creating effective solution that enables engagement among the broader society and individuals with hearing impairments is crucial to bridge this communication gap.

Due to different lighting conditions, complicated backgrounds, range of hand positions, camera angles, occlusion, and variances in sign production recognition of real time Sign Language is a highly intricate field. Although research in this field has been going on since the early 1990s, it hasn't progressed enough, primarily due to limitations in sensor and computer technology. Various national sign languages have been examined by numerous researchers. However, there hasn't been much focus on Devnagari Sign Language, which is widely used in India for communication. DSL's foundation is the Devnagari script, which is extensively used in South Asia for languages like Hindi, Marathi, Nepali, and Sanskrit. Developing a system for DSL interpretation can greatly enhance communication in India for deaf and dumb.

A vowel (Swar) in Devnagari is referred to as 'Svr', which denotes the act of articulating or vocalizing. From 'क' to 'ळ', there are 34 consonants in the Devnagari script as shown in Figure 1 and 2. The script is the basis for writing Marathi/Hindi in Devnagari. The merging of vowels and consonants in Devnagari is recognized as "Barakhadi", which includes not only the fundamental characters but also particular symbols associated with religious affiliations based on particular vowels [2].

In accordance with the outlined requirements, the paper's objective is to use a range of techniques to identify and classify all 408 Barakhadi characters in Devnagari Sign Language. In order to provide a comprehensive analysis and reliable findings, a structured dataset comprising people of various ages and genders has been developed. The 408 distinct Barakhadi character classes created by combining 12 vowel and 34 consonants signs. Keypoint coordinates of hand landmarks are also retrieved from concatenated images of both hand signs in order to analyze hand gestures in depth. A different classification method used for identifying Barakhadi characters in Devnagari Sign Language to improve the accuracy and efficiency of the recognition system. The proposed system's performance is analyzed using metrics such as precision, recall, and F-score and different optimizers.

Figure 1.Swar of Devnagari Sign Language Alphabets

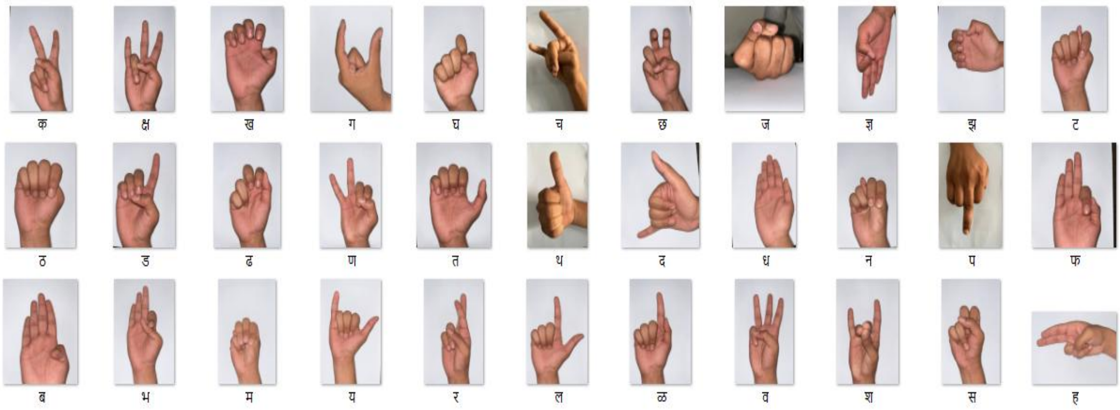

Figure 2.Consonants of Devnagari Sign Language Alphabets

RELATED WORK

(Gaikwad & Admuthe, 2024) [1] study proposed LSTM network for dynamic recognition system of ASL. The OpenCV and MediaPipe Holistic generated dataset, was trained over 1000 epochs, achieved a maximum accuracy of 97.53%. Recognized signs were processed using a natural language processing (NLP) module such as “keytotext” and “T5 model” to generate meaningful sentences, enhancing communication between individuals with hearing impairments and no knowledge of sign language. Future work proposes include more signs by enlarging the dataset and extending the system for other sign languages.

(Sarkar et al., 2024) [3] proposed a Graph Convolutional Networks (GCNs) based ASL recognition system with residual connections. For input into the GCN model, 21 Hand gesture landmarks extracted using MediaPipe, are represented as graphs. The system achieves 99.14% validation accuracy on the ASL Alphabet dataset, showcasing high generalization performance via 5-fold cross-validation. This approach outperforms in capturing complex spatial relationships for ASL rather than CNN. Future research will emphasize on advanced models and larger datasets, enhancing assistive technologies for the deaf and speech-impaired communities.

(Srivastava et al., 2024) [4] presents a continuous SLR system utilizing MediaPipe Holistic and a LSTM model. The system was trained on a dataset of 45 signs covering alphabets and frequently used words, collected using OpenCV and Python. Facial, hand, and body movements tracked by MediaPipe Holistic to gather landmarks. The six-layer NN model with three LSTM layers achieved 88.23% accuracy in both real-time and testing environments. The approach showcased effective and efficient sign recognition mode, further expansion to cover additional gestures and various sign languages.

(Sanaullah et al., 2022) [5] presented a video-driven automated gesture identification system leveraging a dataset of 204 videos containing 1,882 key frames with recognition rate of 94.66%. They used CNN to classify gestures and NLP for Sentence Formation. The methodology focused on recognizing both singular and plural gestures to translate them into corresponding words or sentences in real time. Key challenges addressed include differentiating between single and multiple gestures to improve communication accuracy. Future directions emphasize optimizing the system for mobile platforms to enhance accessibility and handle complex machine learning tasks efficiently.

(Sreemathy et al., 2023) [6] introduces a Python-based SLR system capable of classifying eighty signs with a self-created dataset. Two models were proposed: YOLOv4 and SVM integrated with MediaPipe, achieving accuracies of 98.8% and 98.62%, respectively. YOLOv4 demonstrates robustness against transition gestures but requires high computational power, while SVM with MediaPipe trains faster but is sensitive to noise. An expert system combining both models improves real-time gesture prediction accuracy.

(Camgöz et al., 2020) [7] presents system which combining Continuous SLR and translation using CTC loss, eliminating the demand for explicit gloss annotations. Tested on the PHOENIX14T dataset with doubling previous BLEU-4 scores (9.58 to 21.80). The model leverages sign-specific pre-trained features, improving recognition and translation tasks. Future work aims to incorporate multiple articulators, such as faces, hands, and body, for enhanced linguistic understanding.

METHODS

Figure. 3 illustrates the proposed system, which includes data collection, image resizing for concatenation, image augmentation followed by hand landmark collection, classification using SVM, FFNN, random forest, and customized LSTM-RNN, and model testing.

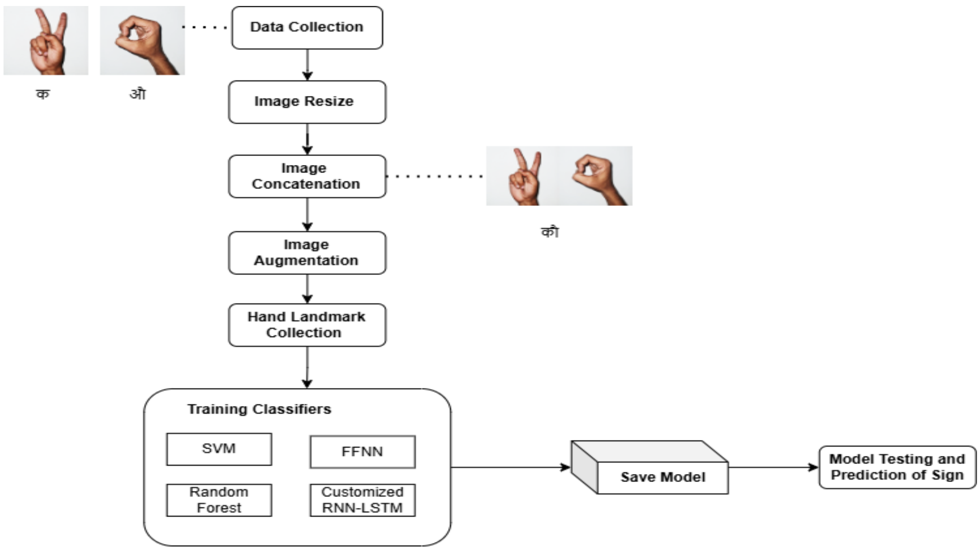

Figure 3. Proposed System

1. Data Collection, Image Concatenation and Augmentation

The dataset comprises images representing the 47 alphabets, encompassing both vowels and consonants of Devnagari Sign Language, ranging from 'अ' (Aa) to 'ज्ञ' (Dnya), as illustrated in Figureb1 and 2. The dataset was collected using a mobile camera under varying lighting conditions and diverse backgrounds. It includes contributions from 15 individuals, both male and female, aged between 15 and 30 years [8, 9].

The combination of vowel signs with consonants forms the foundation of characters in the Devnagari script. For instance, Figure 4 illustrates the alphabet 'क' combined with 12 vowels. Similarly, all 34 consonants are paired with these 12 vowels, resulting in the complete Barakhadi of the Devnagari script, comprising a total of 408 characters, which serve as class labels. We have employed a diverse image augmentation methods aimed at enriching the diversity and resilience of our dataset. We first applied the Horizontal Flip approach, which introduces a variety of orientations by horizontally mirroring the image with a 50% chance to facilitate generalization. To simulate varying degrees of image focus and clarity different blurring techniques such as Motion Blur with a 20% probability, Median Blur with a 10% probability, Gaussian blur with blur limit of 3. The Shift-Scale-Rotate transformation also applied with a 20% probability which included shifts, scales, and rotations with limits of 0.0625, 0.2, and 45 degrees, respectively. Model invariance to translation, scale changes, and rotation is promoted by this composite transformation. To improve contrast and replicate various lighting scenarios techniques for color augmentation were used Applied with a 30% chance, they include Sharpening, CLAHE (Contrast Limited Adaptive Histogram Equalization), Random Contrast adjustment, Embossing, and Random Brightness change. Additionally, Hue-Saturation-Value Adjustment was included to account for changes in lighting and color circumstances, which modifies hue, saturation, and brightness with a 30% chance. After Image augmentation, the total dataset comprised 306000 images distributed across 408 classes as 750 images of each Barakhadi alphabet.

2. Hand Landmark Collection:

After image augmentation techniques, the next step is to extract the keypoint coordinates of hand landmarks from the augmented images. We used MediaPipe Hands, a machine learning-based hand tracking technology to extract hand landmarks from both the images. The coordinates for several keypoints, such as the wrist, thumb, index finger, middle finger, ring finger, and pinky finger joints, were retrieved for every image. The information is arranged in a CSV file with columns for each hand landmark's X and Y coordinates and class label [10]. A detail of hand landmarks is given in Table 1.

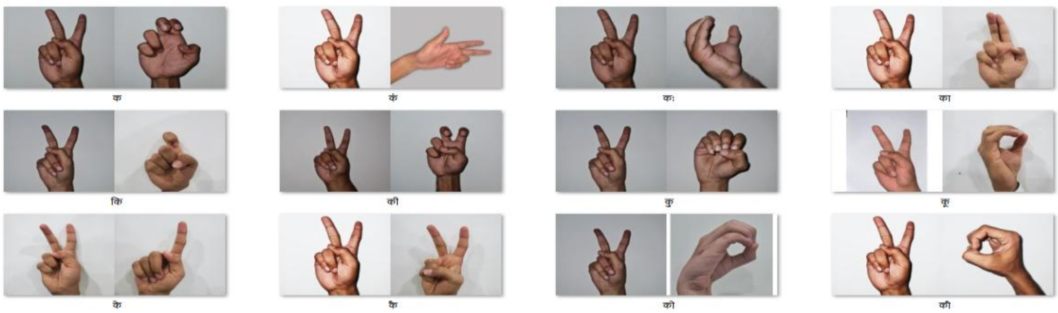

Figure 4. Sample of alphabet 'क' Concatenation with 12 vowels

Table 1: Hand Landmarks

Fingers	Hand Landmarks	Landmarks of 1st Hand Image (X and Y coordinates)	Landmarks of 2nd Hand Image (X and Y coordinates)
WRIST	-	2	2
THUMB	CMC,MCP,IP,TIP	8	8
INDEX	MCP,PIP,TIP,DIP	8	8
MIDDLE	MCP,PIP,TIP,DIP	8	8
RING	MCP,PIP,TIP,DIP	8	8
PINKY	MCP,PIP,TIP,DIP	8	8
Total Landmarks		84	

3. Normalization

To standardize the range of attribute values in the dataset, decimal scaling normalization was employed. This method adjusts each data point by dividing it by a power of 10, chosen according to the highest absolute magnitude in the dataset. Mathematically, the normalized value x_i' for each data point x_i is computed as given in Eq. 1:

$$x_i' = \frac{x_i}{10^j} \tag{1}$$

Where j is the smallest integer such that $\max(|x_i|) < 10^j$. This makes sure that all feature values are scaled proportionally; maintaining their relative differences while bringing them into a comparable range of 0.0 to 1.0[11].

4. Classification

4.1. Feed Forward Neural Network:

A fully connected neural network (FFNN) model is created in order to efficiently classify the dataset, as illustrated in Figure 5a. Features that match the dimensionality of the training data are accepted by the input layer. To add non-linearity, the Rectified Linear Unit (ReLU) activation function is applied across multiple dense layers of the framework. The next hidden layers are set up with increasingly fewer neurons (512, 256, and 128 accordingly) to identify hierarchical structures in the input data. To mitigate overfitting, dropout layers are introduced after each dense layer, randomly disabling a fraction of neurons throughout training. The output layer, containing neurons equivalent to the total distinct classes, employs a softmax activation function to compute the probability distribution across the target categories [12].

4.2. Customized RNN Using LSTM:

To recognize and categorize Barakhadi of Devnagari sign language, the proposed system makes use of a customized RNN architecture with Long Short-Term Memory (LSTM) layers. LSTMs are especially useful for sequential data as they avoid the vanishing gradient issue that standard RNNs frequently encounter. This allows the model to capture both long-term and short-term dependencies in the input data. In order to handle sequence learning tasks, LSTM networks use a number of gating mechanisms such as cell state, input, forget, output gate and mathematical model of gates given in Eq.2-7[13, 14].

Forget Gate

It regulates the elimination of unnecessary information from the prior cell state. The decision is based on the current input, previous hidden state, and passed through a sigmoid activation:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f \tag{2}$$

Where f_t : Forget gate vector, W_f: forget gate's Weight matrix , h_{t-1} : The previous time step's hidden state, b_f : forget gate's Bias vector, σ: Sigmoid activation function

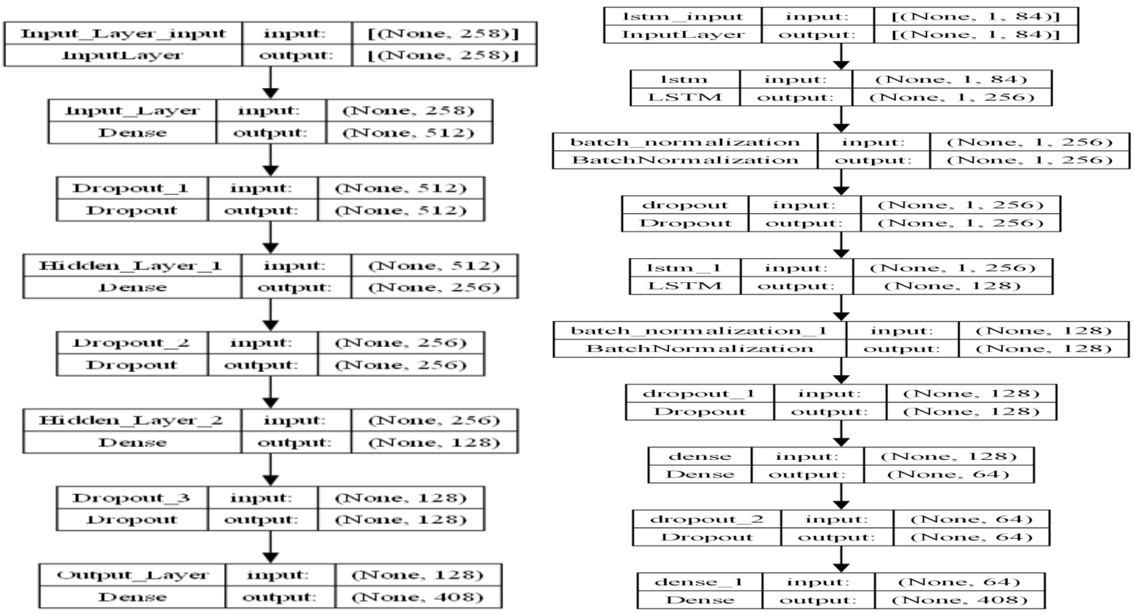

Figure 5a.FFNN Structure, Figure 5b.LSTM Structure

Input Gate

It identifies and stores essential new information. It consists of two steps:

A. Compute the input gate activation:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i \quad (3)$$

B. Compute the candidate cell state:

$$\breve{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C \quad (4)$$

Cell State Update

The old cell state is updated to by forgetting irrelevant information and incorporating new updates:

$$C_t = f_t * C_{t-1} + i_t * \breve{C}_t \quad (5)$$

Output Gate

Determines the output based on the updated cell state, filtered by a sigmoid gate. The filtered cell state passes through tanh to regulate its range:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

The proposed model consists of multiple layers, starting with an LSTM layer that returns the full sequence of hidden states, with 256 units, allowing it to preserve temporal information. Next, a batch normalization layer is applied to stabilize and speed up training by standardizing the intermediate activations, followed by a dropout layer with a 0.5 rate to mitigate overfitting. A second LSTM layer with 128 units processes the sequence representation from the first layer and outputs a fixed-size vector, condensing the temporal information. The output from the second LSTM layer is subsequently fed into two fully connected dense layers. The first layer consists of 64 units with ReLU activation, while the second layer contains 408 units, corresponding to the total number of gesture categories, and utilizes a activation function softmax to compute class probabilities. Regularization techniques, like dropout layers are incorporated into the model, after the dense layers to further prevent overfitting, ensuring that the model generalizes well to unseen data [15, 16]. Figure 5b shows customized LSTM structure.

RESULTS

The experimental setup is conducted using a laptop with a 64-bit architecture, running the Windows 10. The system is equipped with RAM of 8 GB and an Intel Core i3-6100U processor, which operates at a clock speed of 2.30 GHz, built on an x-64 architecture. The Devnagari barakhadi recognition model is developed within a Jupyter Notebook environment, utilizing TensorFlow and scikit-learn libraries for machine learning. The data utilized for the training phase and testing consisted of 30600 images captured via a mobile camera, representing 408 distinct classes, with 750 images per class. The 80% dataset assigned for training and the remaining 20% reserved for testing. The Devnagari Sign Language Recognition model's performance is assessed using precision, recall, and F1-score metric as given in Eq. 8-11. Here, TP and TN corresponds True Positives and Negatives, FP and FN represents False Positives and Negatives [17].

The complete training dataset was run through the network 50 epochs, during the training of the model FFNN and customized RNN-LSTM. The model processed 32 samples at once before adjusting its parameters by using a batch size of 32. Sparse categorical cross-entropy was used as the loss function, making it ideal for classification tasks with multiple distinct classes where target labels are represented as integer values. It determines the discrepancy between the model's predicted probabilities and corresponding ground truth labels. As illustrated in Fig. 6, the majority of optimizers, including Adam, RMSprop, SGD, and Nadam, stabilizing early in the training period, showed quick convergence to almost perfect accuracy [18].

The training and validation phase results, including accuracy and loss trends over 50 epochs for customized LSTM architecture. As depicted in Fig 8, the model achieves near-perfect training accuracy 100% and robust validation accuracy 95%, indicating high learning capacity and good generalization to unseen data.

$$\text{Accuracy}=\frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (8)$$

$$\text{Precision}=\frac{TP}{(TP + FP)} \quad (9)$$

$$\text{Recall}= \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{10}$$

$$\text{F1-Score} = 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})} \tag{11}$$

The effectiveness of various classifiers for Devnagari sign language recognition as depicted in Fig 7. Among the models evaluated, the Customized RNN LSTM architecture attains the highest accuracy of 99.62%, demonstrating its exceptional ability to capture and model sequential dependencies in the data. The Feed Forward Neural Network (FFNN) follows closely with an accuracy of 97.14%, showcasing its proficiency to learn complex patterns, though it lacks the temporal modelling strength of the RNN LSTM. The SVM achieves an accuracy of 97.51%, highlighting its robustness for classification tasks, despite being a non-neural approach. In contrast, the Random Forest model records an accuracy of 93.66%, reflecting its effectiveness for simpler, non-sequential data but its limitations in handling the temporal nature of SLR. These results emphasize the efficiency of sequence-aware architectures, such as the RNN LSTM, for achieving superior capability in Devnagari sign language recognition.

As there is currently no existing system specifically developed for recognizing Barakhadi of Devnagari Sign Language, so the proposed system's performance is compared with studies on other sign languages such as ASL and ISL of 26 English alphabets. Notably, previous works achieved recognition rates of 96% and 93.8% by applying advanced techniques like RNN-LSTM for ASL and ISL, respectively. The proposed system utilizes LSTM for Devnagari Sign Language Barakhadi recognition, outperforms with an impressive recognition rate of 99.61% as shown in Table 2.

Table 2: Comparison with other related studies

Author	Sign Language	Technique used	Recognition Rate (%)
Sreemathy R. [6]	Indian Sign Language word	YOLOv4	98.62%
Avola, D [17]	American Sign Language	RNN-LSTM	96%
Ezhil Tharsan S [16]	Indian Sign Language	RNN-LSTM	93.8%
Proposed System	Devnagari Sign Language	LSTM	99.61%

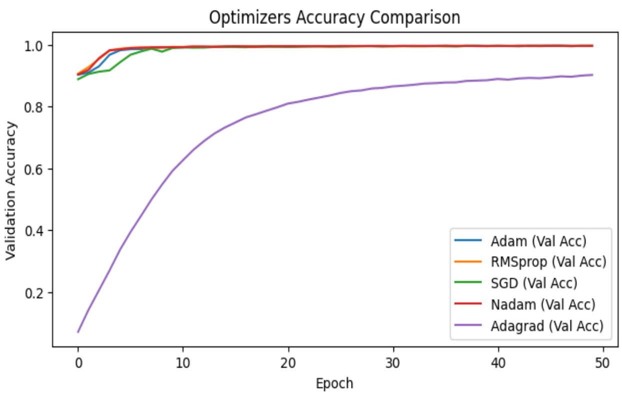

Figure 6.Different Optimizers Accuracy

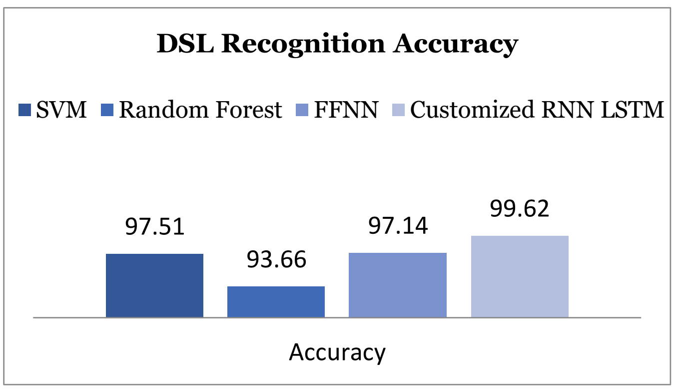

Figure 7. Performance Comparison of Different Classifiers

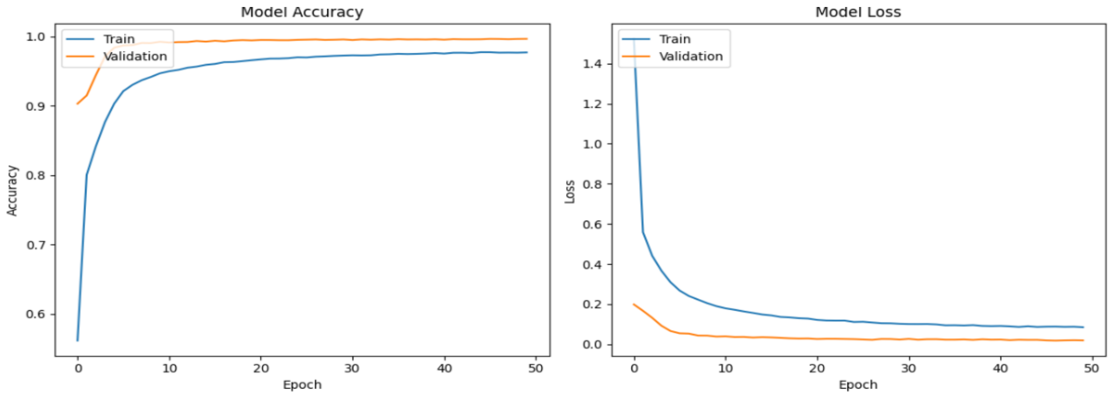

Figure 8.Customized RNN LSTM Model Accuracy and Loss

CONCLUSIONS AND FUTURE WORK

This research demonstrates a machine learning approach to identifying the entire Barakhadi of Devnagari Sign Language (DSL), which consists of 408 characters formed of a combination of vowels and consonants. A dataset extended to 306,000 images using advanced image augmentation techniques, utilized to train and evaluate the models. The augmentation approaches comprising horizontal flips, blurring, shift-scale-rotate transformations, and color adjustments, significantly broadened the dataset's range and variety and helps to improve the model's generalization capability. The hand landmarks feature extracted using MediaPipe to enhancing the system's performance. Among the classifiers examined in the study, the customized RNN-LSTM architecture achieved the highest accuracy of 99.61%. Future work could focus on expanding the dataset to include dynamic gestures and real-time recognition of all 408 brakhadi alphabets. To make suitable for real-time applications, incorporate temporal and spatial features into the model would enable it to process continuous sequences. To bridging communication gaps integrate this system into portable devices or mobile applications could make DSL recognition accessible to a wider audience.

REFRENCES

[1] Gaikwad. R.,; Admuthe, L. (2024).; Sign Language Recognition of Words and Sentence Prediction Using Lstm and Nlp. Journal of Theoretical and Applied Information Technology, 102(4), 1541–1548.

[2] Shazid wahid khandakhani,; Sachikanta Dash,; Sasmita Padhy,; Rabinarayan Panda.; (2024). Implementation of Deep learning Methods to Marathi Hand Written Characters and its Pattern Recognition by Using Generative AI. Journal of Computational Analysis and Applications (JoCAAA), 33(07), 979–994.

[3] Sarkar, U., Chakraborti, A., Samanta, T., Pal, S., & Das, A. (2024). Enhancing ASL Recognition with GCNs and Successive Residual Connections. http://arxiv.org/abs/2408.09567

[4] Srivastava, S., Singh, S., Pooja, & Prakash, S. (2024). Continuous Sign Language Recognition System Using Deep Learning with MediaPipe Holistic. Wireless Personal Communications, 137(3), 1455–1468. https://doi.org/10.1007/s11277-024-11356-0

[5] Sanaullah, M., Kashif, M., Ahmad, B., Safdar, T., Hassan, M., Hasan, M. H., & Haider, A. (2022). Sign Language to Sentence Formation: A Real Time Solution for Deaf People. Computers, Materials and Continua, 72(2), 2501–2519. https://doi.org/10.32604/cmc.2022.021990

[6] Sreemathy, R., Turuk, M. P., Chaudhary, S., Lavate, K., Ushire, A., & Khurana, S. (2023). Continuous word level sign language recognition using an expert system based on machine learning. International Journal of Cognitive Computing in Engineering, 4(November 2022), 170–178. https://doi.org/10.1016/j.ijcce.2023.04.002

[7] Camgöz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 10020–10030. https://doi.org/10.1109/CVPR42600.2020.01004

[8] Pansare, J., & Ingle, M. (2018). A Real-Time Devnagari Sign Language Recognizer (α-DSLR) for Devnagari Script. In Lecture Notes in Networks and Systems (Vol. 18, pp. 75–84). Springer. https://doi.org/10.1007/978-981-10-6916-1_8

[9] Deshpande, A.M., Kalbhor, S.R. (2020). Video-Based Marathi Sign Language Recognition and Text Conversion Using Convolutional Neural Network, Lecture Notes in Electrical Engineering, vol 569. Springer, Singapore.

[10] Lugaresi,C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). MediaPipe: A Framework for Building Perception Pipelines. http://arxiv.org/abs/1906.08172

[11] Naglot, D., & Kulkarni, M. (2016). Real time sign language recognition using the Leap Motion Controller. Proceedings of the International Conference on Inventive Computation Technologies, ICICT 2016, 2016, 1–5. https://doi.org/10.1109/INVENTIVE.2016.7830097

[12] P. Suresh Kumar, H.S. Behera, Anisha Kumari K, Janmenjoy Nayak, Bighnaraj Naik, Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades, Computer Science Review, Volume 38, 2020,100288, ISSN 1574-0137, https://doi.org/10.1016/j.cosrev.2020.100288.

[13] Samaan, G.H.;Wadie, A.R.,Attia, A.K.; Asaad, A.M.; Kamel, A.E.,Slim, S.O.; Abdallah, M.S.; Cho, Y.-I.MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition. Electronics 2022.

[14] Ezhil Tharsan S., Dharshan S., Dinesh G., Saraswathi S. . Real Time Indian Sign Language Detection using LSTM and Keypoint Extraction. International Journal of Computer Applications. 184, 21 (Jul 2022), 1-7. DOI=10.5120/ijca2022922235

[15] Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., & Massaroni, C. (2019). Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures. IEEE Transactions on Multimedia, 21(1), 234–245. https://doi.org/10.1109/TMM.2018.2856094

[16] Lee, C. K. M., Ng, K. K. H., Chen, C. H., Lau, H. C. W., Chung, S. Y., & Tsoi, T. (2021). American sign language recognition and training method with recurrent neural network. Expert Systems with Applications, 167. https://doi.org/10.1016/j.eswa.2020.114403

[17] Luque, A.; Carrasco, A.; Martín, A.; Lama, J.R. Exploring Symmetry of Binary Classification Performance Metrics. Symmetry 2019, 11, 47. https://doi.org/10.3390/sym11010047

[18] Gousia Habib, Shaima Qureshi, Optimization and acceleration of convolutional neural networks: A survey, Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 7, 2022, Pages 4244-4268, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2020.10.004.