SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Enhancing Tamil Sign Language Recognition In Video Data Using Deep Learning

S. Thanga Pandeeswari[1], K. Balasubramanian[2],

[1]Research Scholar, Department of MCA, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India – 626126.Email id: thangapandeeswari67@gmail.com
[2]Professor, Department of Computer Applications, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India – 626126, Email id: ksbala75@gmail.com
Corresponding author mail id: thangapandeeswari67@gmail.com

Abstract
Tamil Sign Language (TSL) gratitude using video datasets is a critical step in connecting letter fences for the unhearing community by identifying sign language gestures using automated computer vision systems. Recognizing and classifying gestures in a real-time video context is a challenging task due to the variability of signs, variability of signers, background variability, and altering illumination. The inherent spatial and temporal dependencies and combinations are difficult for traditional methods to capitalize on. To address these challenges, the Deep Convolve Spatial Layer–Gated Recurrent Neuronet (DCSL-GRN) model is proposed, effectively apprehending both spatial and temporal dependencies in video sequences. The DCSL-GRN model integrates convolutional layers for spatial feature extraction and Gated Recurrent Units (GRUs) for temporal modeling, significantly enhancing the accuracy and robustness of TSL gesture recognition. The Tamil Sign Language video dataset from Kaggle is used in the research. Sobel edge detection enhances image contrast by emphasizing key edges, while GrabCut segmentation isolates gesture regions accurately. Fourier Transform extracts frequency-based features, and Gaussian augmentation increases data diversity for better generalization. With a classified accuracy of 98%, precision of 97%, recall of 98%, and F1-score of 97%, experimental findings validate the model's resilience to failure that show its exceptional performance. A Graphical User Interface (GUI) was created to display Tamil sign letter outputs as combined text. Moreover, the use of Retrieval-Augmented Generation (RAG) allows for contextual awareness and increased accuracy and interpretability of the outputs. An implemented method using a Python tool, the suggested approach offers an accurate, accessible answer for real-time credit and translation of Tamil Sign Language.

Keywords: sign language recognition, Sobel edge detection, GrabCut Segmentation, Fourier Transform, Deep Convolve Spatial Layer Gated Recurrent Neuronet (DCSL-GRN), Graphical User Interface (GUI), Retrieval-Augmented Generation (RAG), AI Innovation.

Introduction

Signed philological is an important mode of exchange for deaf people because it allows them to communicate complex emotions in multiple dimensions through hand movements, face expressions, and body movements. TSL is a sign word that is mostlyused by Tamil speaking deaf individuals and it is estimated that this language has more than 60,000 users in India alone [1]. TSL has significant importance culturally and linguistically, but the recognition and analysis of TSL have received comparatively less attention than other global sign languages, for example, ASL or BSL. The lack of reliable and automated recognition solutions results in an enormous communication barrier between signers who do sign and all other non-signers. Therefore, intelligent solutions must be devised to mimic human performance while recognizing TSL gestures in real time [2].SLRis variably aninterestingmission due to a number of inherent technical obstacles. These include intra-sign variability, inter-sign variability, and variations in the environment such as lighting, occlusion, and image-dynamic backgrounds [3, 4].

Traditional computer vision ML techniques often struggle to demonstrate robust performance as they analyze each frame as an independent entity. Spatial features, which represent the properties of the hand shape, location and orientation, represent static information, whereas temporal features describe continuity of motion and evolution of the gesture across images. Analysis of only one of these dimensions results in performance accuracy and generalization failure while operating in unconstrained real-world conditions [5, 6].Recent developments in DL, CNNs and RNNs have reshaped gesture and activity recognition. Deep models can capture multi-level spatial representations and account for temporal dependencies across video sequences for more accurate interpretation of dynamic gestures [7, 8]. Additionally, attention mechanisms and hybrid architectures have been able to recognize continuous sign language sequences to effectively encode contextual dependencies and long-range motion [9, 10].Improving TSL recognition is not only a technology challenge but also a socially responsible motivation. A TSL recognition system that identify TSL accurately in real time could enable communication inclusivity, access to education, and digital empowerment of deaf people. In this regard, a capable, video-based human language DL framework for TSL is a significant step toward linguistic inclusivity and the dissolution of barriers to communication that have lasted in the Tamil community to this day [11, 12].

Objective of This Research: A strong and effective system for identifying TSLgestures portrayed in a dynamic video setting is developed and tested using a DL-based approach. The research uses a DLbased method that uses the DCSL-GRN model;this describes the use of gates for recurrent units to learn a timeline and layers of convolution to extract spatial features.After discussing the setting of a dynamic gesture recognition system with systematic pre-processed learning of features and implementation of a graphical interface in real-time.

Key contributions

- Collected TSL video dataset from Kaggle, converting videos into structured frames for effective gesture analysis and training.
- Applied Sobel edge detection and GrabCut segmentation to enhance gesture boundaries and accurately isolate hand regions from complex backgrounds.
- Used the Fourier Transform to convert gesture images into frequency-domain features, capturing spatial patterns and enhancing discriminative gesture representation.
- Integrated DCSL-GRN to learn spatial-temporal dependencies, achieving robust and accurate TSLrecognition.

Organization of research: Introduction is demonstrated in Section 1, related works are summarized in Section 2, method details are covered in Section 3, results are outlined in Section 4, the findings and previous research are discussed in Section 5, and the research is concluded in Section 6.

2. Literature Review

A DL-based architecture was industrialized to transform sign gestures into both text and synthesized speech, utilizing CNN and NLP models. The approach achieved 97.6% accuracy with strong F1-scores, as reported in [13], though its adaptability remains limited when applied to diverse or unseen sign variants.A sophisticated multi-modal recognition framework combined ResNet, a Transformer encoder, and CTC-based decoding for Sinhala Sign Language that enhanced translation at the sentence level. The model specified in [14] reached a WER of 12.70; however, the limited dataset and the set environment inhibited generalization.The combination of a DCNN and RNN-LSTM model has shown effective continuous gesture recognition and sentence formation. The complementary fashion of the two networks enhanced the understanding of the sequences, proving to be accurate at nearly 90% [15]. Itis resource-heavy to execute and is sensitive to responsiveness under complex motion modes.In research [16], 2D and 3D CNNs were used for both standing and active datasets to offer a real-time, efficient model. Although batch normalizing and geographic augmenting increased overall accuracy, the model's robustness decreased in cluttered or uncertain backdrops.Additionally, a TSL translation system was created that uses SIFT and edge detection from Canny to handle localized movements. The system successfully handled 31 alphabets and numerous finger combinations, overcoming basic communication barriers [17], but its recognition ability was limited to isolated symbols rather than full phrase-level interpretation.

To categorize South Indian Sign Language movements across age ranges, ResNet-50 and Inception-V3transfer learning-based models,likeVGG-16, were used. With 95.20% specificity and 92.45% verification accuracy, the Inception-V3 model successfully classified simultaneous gestures into ten groups [18]. However, information limitations continue to prevent modeling applicability.The communication problems of deaf children, and a system based on DL that detects alphabets and numerals using a CNN were presented. CNN-based recognition was emphasized as providing better accuracy than contemporary algorithms [19]. However, any productivity improvement could potentially be enhanced even further if sophisticated extraction for characteristic methods was included as well.A tagged dataset called TLFS23 was developed for Tamil finger-spelling recognition and is made up of 2, 55, and 155 images organized over 248 classes. The dataset aimed to facilitate research into vision-based gesture-to-gesture translation and the design of DL-based as CNN interpreters for users with hearing impairments [20]. However, the dataset consisted only of static motions, leaving out the ability to make use of dynamic sequence learning.To recognize actions in ISL pictures, a novel model combining LSTM and GRU systems was proposed. The hybrid solution handled spatiotemporal sequences well, obtaining around 97% accuracy across 11 indications [21]. Despite demonstrating excellent accuracy, the model faced computational challenges during long-sequence training.

The further improvement of recognition and translation was observed with the help of Hybrid CNN + Bi-LSTM architecture and MediaPipe implementation in real-time posture extraction and NMT + GAN implementation in video production. The framework obtained 95% accuracy in classification and BLEU score of 38.06 [22]. Despite this, it was a successful technique, but it was not easily implementable in a large-scale real-time application because of its complexity.A MediaPipe-based algorithm was developed to detect movements in Assamese Sign Language using a dataset containing 2,094 samples [23]. The feed-forward NN was able to attain an accuracy of nine various gesture types. Nevertheless, the vocabulary of the dataset was not that large to allow a continuous sentence-level debate.Five NN models were evaluated for ASL alphabet recognition: AlexNet, ConvNeXt, Efficient Net, ResNet-50, and Vision Transformer. The ResNet-50 model had the highest accuracy, exceeding the others [24]. The models mostly concentrated on individual alphabets without surrounding word-level knowledge, despite their outstanding outcomes.

3. Methodology

The proposed method combines the steps of collecting the dataset, refining and segmenting the input, deriving significant features, enhancing data diversity, and performing classification in TSL. The collected video datasets are segmented using GrabCut, processed through Fourier-based feature extraction, and augmented with Gaussian before classification with the DCSL-GRN. Figure 1 shows the entire methodological flow for the proposed research framework.

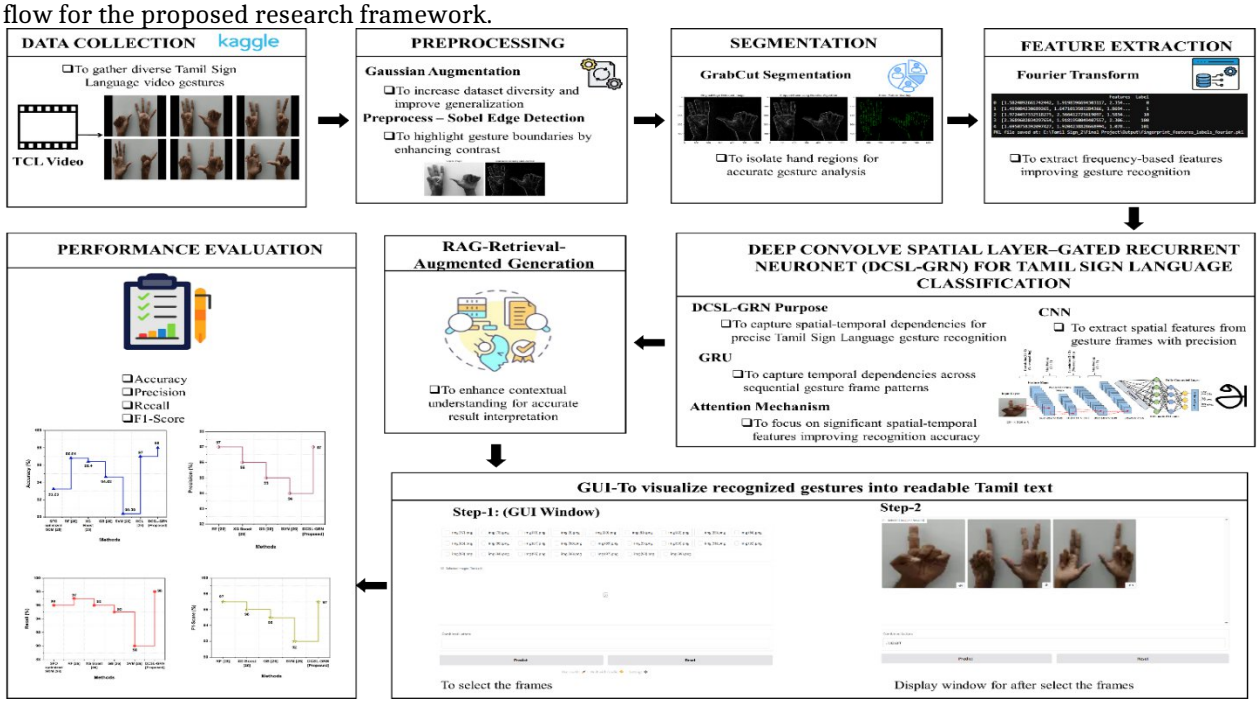

Figure 1: Methodological flow of the proposed DCSL-GRN-based TSL recognition system.

Data collection

The data collection process requires collecting TSL video datasets from publicly available repositories, including Kaggle [25]. In these videos, dynamic hand gesture sequences represent the Tamil sign alphabet. Figure 2 represents the sample data of the TSL.To maintain balanced evaluation and lessen overfitting, the dataset is divided into 80% for training and 20% for testing during model development.Each video is segregated into separate frames as part of Video-to-Frame Conversion to conduct frame-wise analysis. This step ensures efficient preprocessing and feature extraction, converting a continuous video stream into structured image frames, following which segmentation and recognition stages take place.

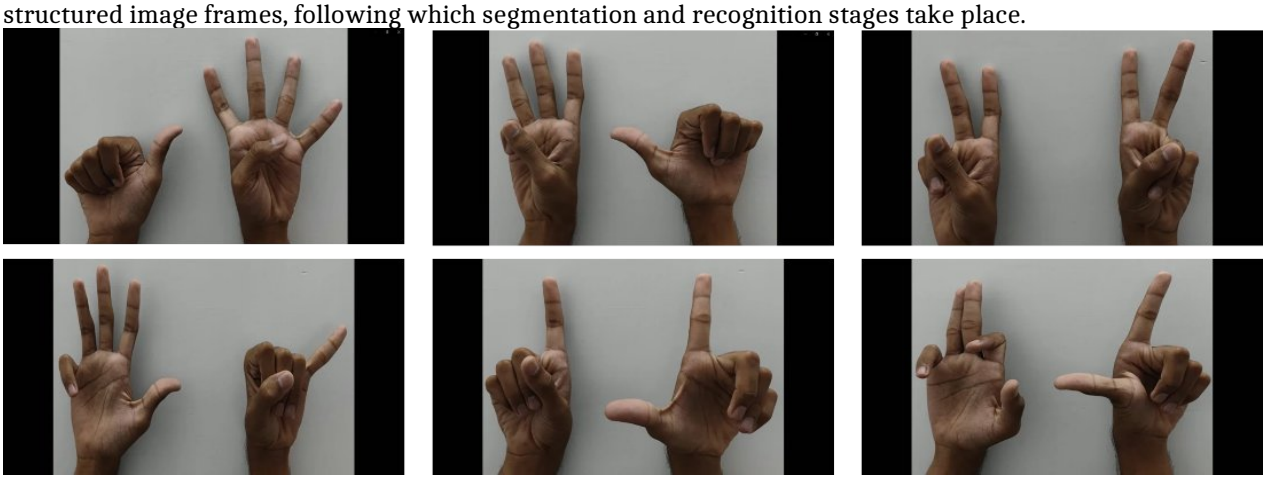

Figure 2: Sample Tamil Sign Language video frames representing gesture data used for recognition

Data Augmentation (Gaussian) for Dataset Enhancement

Gaussian noise augmentation adds more and more variation to the TSL data by injecting controlled amounts of variation in the pixel intensity, simulating natural conditions of variations in lighting and background as a natural world manifestation, which in turn can result in increased generalization by the model. Moreover, Gaussian augmentation does not only avoid overfitting but also improves the variability of the dataset, which guarantees improved learning and enhanced performance in gesture recognition.

Preprocessing Using Sobel Edge Detection for Tamil Sign Language Gesture Enhancement

The TSLvideo frame enhancement is done by the utilization of adetector to enhance the visibility of the hand sign in the frames. Sobel edge detection identifies both horizontal and vertical intensity gradients, which, when detected, accentuate boundaries in the various gestures, separate the hands and the backgrounds of the video frames, and enhance the accuracy of segmentation and feature extraction of the recognition task on the hands.

$$N = \sqrt{t_w^2 + t_z^2} \quad (1)$$

In Equation (1),t_w is the horizontal gradient that identifies vertical edges, and $t_z$is the vertical gradient that identifies the horizontal edges and N is the total edge strength in sign gesture frames. The turn-taking weight and the attractive edge detection precision in the gesture margins is expressed in Equations (2-4). The eight surrounding pixel values of a target pixel are indicated by b_0–b_7.

$$t_w = \left(b_2 + db_3 + b_4\right) - (b_0 + db_7 + b_6) \quad (2)$$

$$t_z = \left(b_0 + db_1 + b_2\right) - (b_6 + db_5 + b_4) \quad (3)$$

$$T_w = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \quad 2T_z = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \quad (4)$$

In Figure 3, the preprocessing phase utilizes the Sobel operator to highlight the gesture boundaries in Tamil Sign Language frames, increasing edge visibility and ensuring spatial feature extraction accuracy during the task of recognition with deep learning-based gesture recognition systems.

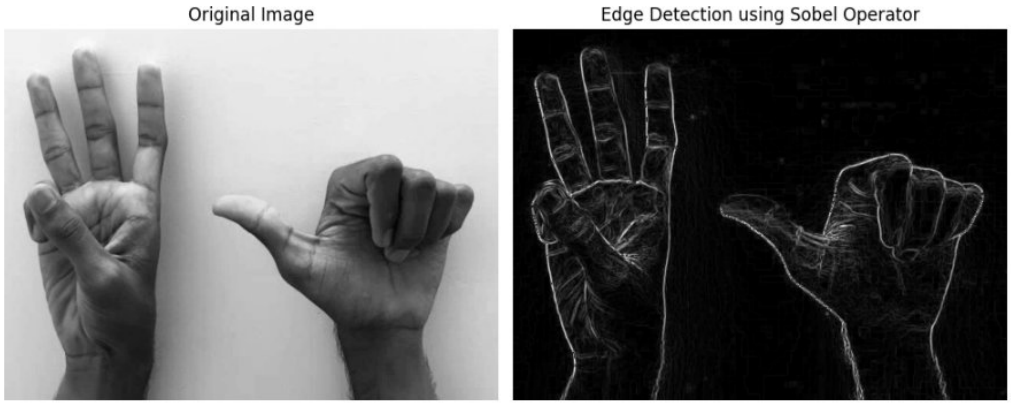

Figure3: Sobel edge detection enhancing gesture contours in TSL preprocessing

Segmentation Using the GrabCut Algorithm for Hand Gesture Isolation

The GrabCut segmentation approach is implemented on Sobel edge-detected images to segment the hand and gesture regions presented in TSLfreehand frames through initializing the mask and bounding box to distinguish the hand (foreground) from the background and subsequently iteratively refining the mask to ensure the gesture regions are isolated correctly. Once the gesture regions were appropriately isolated, the fingers were prescribed a green-striped overlay to emphasize the contours of the fingers to allow for clearer gesture regions for extraction and recognition of features.

$$F\left(\underline{\alpha}, \underline{\theta}, y\right) = V\left(\underline{\alpha}, \underline{\theta}, y\right) + U(\underline{\alpha}, y) \quad (5)$$

$$V\left(\underline{\alpha}, \underline{\theta}, y\right) = \sum_{n} - \log h(y_m; \alpha_m) \quad (6)$$

$$U\left(\underline{\alpha}, y\right) = \gamma \sum_{(n,m)\in D} \mathrm{dis}\,(n,m)^{-1} \left[\alpha_m \neq \alpha_n\right] \exp - \left(y_n - y_m\right)^2 \quad (7)$$

In Equations (5-7), $F(\grave{\alpha}, \grave{\theta}, y)$represents the total energy combining data $V(\grave{\alpha}, \grave{\theta}, y)$and smoothness $U(\grave{\alpha}, y)$. Here, $\grave{\alpha}$denotes pixel labels, $\grave{\theta}$defines GMM parameters, yis the pixel intensity, $\gamma$is a regulation factor, Dis an adjacent pair, $\mathrm{dis}\,(n,m)$designates pixel distance, and $y_n, y_m$are neighboring concentrations.

Figure 4 shows how the GrabCut algorithm segments the Tamil Sign Language gesture frames, isolating the hand regions from their more complicated backgrounds. This reduced noise and increased spatial fidelity to accurately model and recognize gestures from deep learning.

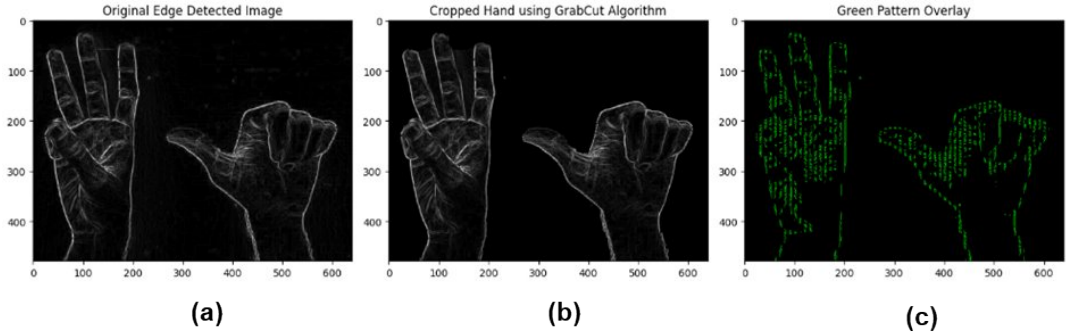

Figure 4: GrabCut segmentation isolating hand regions for precise gesture recognition (a) original image, (b) preprocessed image, and (c) segmented image

Feature Extraction Using the Fourier Transform for Gesture Pattern Analysis

The Fourier Transform retrieves features from TSLgesture images of the segmented fingerspelling by moving from the spatial to the frequency domain. Before applying the 2D FFT to an image to obtain the frequency components, it is resized and normalized. After the FFT was conducted, the magnitude spectrum was shifted to center the frequency peaks, log-transformed, and compressed into one-dimensional feature vectors. Since the features extracted are based on frequencies, they will represent any texture and patterns that change. Data based on frequencies in the magnitude spectrum comprise a labeled dataset, which can be applied for

identifying gesture patterns for classification and actions for recognition of gesture patterns in future sequences.

$$(e*h)(s) = \int_{-\infty}^{\infty} e(s)\,h(s-\tau)\,d\tau = \int_{-\infty}^{\infty} e(s-\tau)\,h(s)\,d\tau \quad (8)$$

$$FT(e*h) = E \times H \text{and} FT(e \times h) = E * H \quad (9)$$

In Equations (8 and 9),$e(s)$and $h(s)$represent input and impulse response functions, while $(e*h)(s)$denotes their convolution over time shiftτ. FT designates the Fourier Transform, E and $H$are the altered frequency-domain symbols of $e(s)$and$h(s)$, respectively, and "×" and "*" denote multiplication and convolution operations.

The Fourier Transform displayed in Figure 5 converts frames of the Tamil Sign Language gestures into features in the frequency domain, characterizing periodic motion trends and greater discriminative information, therefore aiding deep learning-based gesture classification and labeling.

```
                                        Features  Label
0  [1.5824092661742442, 1.9198396694303117, 2.354...      0
1  [1.419084230689265, 1.6471013501284366, 1.8694...      1
2  [1.9724457332518275, 2.366412725619097, 1.5854...     10
3  [2.3689602034297654, 1.9181950049407557, 2.306...    100
4  [1.6950758392097427, 1.9204238828668994, 1.078...    101
PKL file saved at: E:\Tamil Sign_2\Final Project\Output\fingerprint_features_labels_fourier.pkl
```

Figure 5: Fourier Transform extracting frequency-based features for gesture recognition

Deep Convolve Spatial Layer Gated Recurrent Neuronet (DCSL-GRN) for Tamil Sign Language Classification

The DCSL-GRN unifies three-dimensional and chronological learning to facilitate sign language appreciation. It's a hybrid of the use of DeepCNN and GRU for sequence learning in sequential order models for a correct identification and understanding of hand gestures and movements, thus providing increased recognition accuracy and robustness.

Deep Convolutional Neural Network (DCNN) for Spatial Feature Extraction in Tamil Sign Language Recognition

The DCNN used in this research is developed to abstract and learn spatial geographies from single TSL video frames. It encodes complex hand motion, shapes, orientation, and movement using multiple convolving and pooling layers. The DCNN acquires low-level landscapes such as edges and textures and learns high-level abstract features that are task-relevant to the gestures. When combined with the GRU layer, it adapts temporally to the variations of sign sequences. This hybrid architecture integrates spatial learning and temporal education to advancesignclassification accuracy and give the capability to accurately identify sign patterns exhibited under different lighting conditions, background, and signer styles in video datasets.The output feature map at spatial position (j, i), $Y(j+l,i+m)$, is the input feature value, $X(l,m)$denotes the convolution filter weights, $c$is the bias term, $L$and $M$are filter magnitudes, and $t_l$indicates the stride or conversion index used in difficultyprocesses using Equations (10 and 11).

$$Z(j,i) = \sum_{l=1}^{L} \sum_{m=1}^{M} Y(j+l,i+m).X(l,m) + c \quad (10)$$

$$Z(j,i) = \max_{l,m} Y(t_l + j, t_l + i) \quad (11)$$

In Equations (12 and 13), $\sigma(a)_j$ symbolizes the softmax output for class j, $f^{(a_i)}$represents the beginning function for input a_i, L is the total number of classes, $E(\theta;w)$is the model output, $\theta_m$are model criticisms, and $w$represents network weights.

$$\sigma(a)_j = \frac{f^{a_i}}{\sum_{l=1}^{l} f^{a_l}} for i = 1,.....,L \quad (13)$$

$$E(\theta;w) = softmax\left(e_m\left(\theta_m, e_{m-1}\left(\theta_{m-1}, \ldots e_2\left(\theta_2, e_2\left(\theta_1, w\right)\right)\right)\right)\right) \quad (14)$$

In Equations (15-17), δsignifies the optimal agitationminimizing the norm $\|y\|_0$, y is the agitation vector, $P(w)$ and $P(w+y)$denotescalculations before and after perturbation, $w$is the input data, and $y^*$is the adversarially modified output.

$$\delta = \underset{y}{argmin} \|y\|_o \quad (15)$$

$$t.s.\ P(w) \neq P(w+y) \quad (16)$$

$$y^* = y + \delta \quad (17)$$

The model for TSLgesture classification using CNN approach is shown in Figure 6. The input gesture image takes several convolutions, pooling, and flattening layers, along with a fully associated layer and a softmax layer, which aids in the classification and identification of the state to extract the features associated with the gesture.

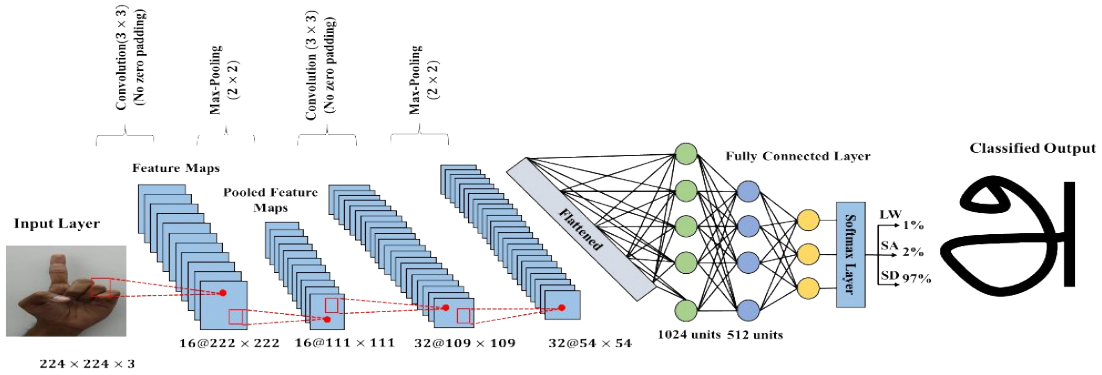

Figure 6: CNN architecture for sign language gesture recognition.

Gated Recurrent Unit (GRU) for Temporal Sequence Modeling in Frame-Based Gesture Recognition

The GRU models the temporal relationships between sequences of image frames that were derived from TSLvideos. GRU indicates how spatial features were learned over time after a DCNN was utilized to excerpt spatial geographies for each frame, thus indicating dynamic gesture information over time. Utilizing update and reset gates, the GRU maintains a balanced memory capacity while retaining critical temporal information and discarding task-irrelevant information. Generally, compared to RNNs or LSTMs, GRUs learn faster without high computational costs. Specifically, using GRU in the current study is useful for learning contextual frames for continuous gestures and producing long-term dependencies across frames, thus increasing sign accuracy and robustness for interpretation.In this GRU model, $q(s)$, $y(s)$, and $p(s)$characterize candidate activation, respectively. $w(s)$ denotes the current input, $g(s-1)$is the precedingsecreted state, x_{wg} and x_{gg} are massmediums, a_q, a_y, and $a_p$are bias relations, $\sigma$and $tanh$are beginningpurposes evaluated in Equations (18-20).

$$q(s) = \sigma(x_{wg}^{q} \times w(s) + x_{gg}^{q} \times g(s-1) + a_q) \quad (18)$$

$$y(s) = \sigma(x_{wg}^{y} \times w(s) + x_{gg}^{y} \times g(s-1) + a_y) \quad (19)$$

$$p(s) = tanh\left(x_{wg}^{p} \times w(s) + x_{gg}^{p} \times q(s) \times g(s-1) + a_p\right) \quad (20)$$

In these GRU Equations (21 and 22), $g(s)$is the efficientsecreted state, $y(s)$is the reset gate, $p(s)$is the applicant activation, $g(s-1)$is the previous state, $x_{gz}$is the mass, $a_g$is the bias term, and $z(s)$is the output.

$$g(s) = (1 - y(s)) \times g(s-1) + y(s) \times p(s) \quad (21)$$

$$z(s) = x_{gz} \times g(s) + a_g \quad (22)$$

Spatial Attention Mechanism for Focused Feature Learning in Tamil Sign Language Recognition

The Spatial Attention Mechanism aids the model in identifying and concentrating on the most informative fields of every extracted frame (e.g., pointing to the hands and fingers), while also decreasing interference generated by the background. In particular, the attention mechanism allows the model to produce an attention map with a superior weight allocated to the relevant spatial area of the frame and a lesser weight to irrelevant areas. This focused spatial attention improves the gesture recognition performance and accuracy, especially for gesture interpretation relative to complex or cluttered backgrounds. The attention mechanism, in conjunction with the DeepCNN and GRU, ensures the network is focusing on spatial features across extracted frames that best characterize the TSLgestures. Thus, the attention mechanism will provide

improved accuracy and contextualization of the gestures and corresponding outputs.The DCSL-GRN algorithm is designed to recognize TSLgestures promptly using Deep Convolutional and Recurrent neural networks. The algorithm uses DeepCNN to extract spatial features, GRU is used to model temporal dependencies, and the attention layer creates spatial attention to focus on the key gesture regions. The proposed integrated framework provides richer features, minimizes the background noise, and tracks the dynamism of motion to provide accurate and robust gesture recognition in real time across various lighting, backgrounds, and signer conditions.

Algorithm 1: DCSL-GRN for Tamil Sign Language Recognition

Input:
Tamil Sign Language video dataset
frame size (128×128);
regularization factor (γ); Sobel parameters ((t_w,t_z)); GrabCut parameters ((α¯,θ¯,y));
FFT window size; CNN filter size ((L,M)); GRU weights ((x_{wg},x_{gg},x_{gz})); and attention coefficients.
Output:
Optimized DCSL-GRN model (μ^(s|θ^{μ^})) and best recognition accuracy (f_{best}).
Initialization:
Load dataset (V); initialize CNN–GRU weights and attention parameters.
Normalize all frames and resize to (128×128).
Preprocessing (Edge Enhancement):
For each frame (f_i∈V):
a. Apply the Sobel operator using **Eq. (1)** to compute gradient magnitude (N).
b. Obtain enhanced frame (f'_i) emphasizing gesture contours.
Segmentation (Hand Isolation):
For each (f'_i):
a. Initialize GrabCut mask and bounding box.
b. Compute energy (F(α¯,θ¯,y)) using **Eq. (5)**–**(7)**.
c. Extract hand region → segmented frame (f''_i).
Feature Extraction (Fourier Domain):
For each (f''_i):
a. Apply FFT using **Eq. (8)**–**(9)** to obtain frequency features.
b. Flatten and normalize into a vector (X_i).
Spatial Feature Learning (Deep CNN):
a. Convolve (X_i) with filters using **Eq. (11) and (12)**.
b. Apply activation and softmax normalization using **Eq. (13) and (14)**.
c. Generate spatial feature map (Z(j,i)).
Temporal Modeling (GRU):
a. Feed sequential CNN outputs to GRU.
b. Update hidden state using **Eq. (21)**.
c. Compute final output using **Eq. (22)** for temporal encoding.
Spatial Attention Mechanism:
Compute attention weights over (Z(j,i));
enhance regions with higher gesture relevance;
generate attended feature representation (Z').
Classification and Optimization:
a. Concatenate spatial–temporal features.
b. Apply a fully connected layer and Softmax classifier.
c. Compute loss using cross-entropy and update via backpropagation.
d. Repeat until convergence across (E) epochs.
Output:
Return final trained model (μ^(s|θ^{μ^})) and best recognition score (f_{best}).

The most important hyperparameters chosen for the architecture of the DCSL-GRN are presented in Table 1. The input to the network consists of gesture frames in greyscale with dimensions 128 x 128 that go through a sequence of convolutional blocks, spatial attention layers, GRU layers, and fully connected layers. These parameters, combined with the regarding structure, the optimized Adam algorithm and the categorized cross-entropy loss model offer sufficient feature training in terms of spatial learning and precision for classification for TSL.

Table 1: Hyperparameter configuration of the DCSL-GRN in TSL

Module / Layer	Parameter	Specification
Input Layer	Image Dimension	128 × 128 pixels
	Color Channel	1 (Grayscale)
Convolution Block 1	Filters	32
	Kernel Dimension	3 × 3
	Activation Function	ReLU
	Padding Type	Valid
	Pooling Window	2 × 2
Convolution Block 2/ Block 3	Filters	64
	Kernel Dimension	3 × 3
	Activation Function	ReLU
	Padding Type	Valid
	Pooling Window	2 × 2
Spatial Attention Layer	Attention Mechanism	Spatial Attention
	Kernel Dimension	7 × 7
	Activation Function	Sigmoid
Flatten Layer	Output Dimension	25,088
Reshape Layer	Reshaped Output	(196, 128)
GRU Layer	Hidden Units	128
	Activation Function	tanh
	Recurrent Activation	Sigmoid
	Return Sequences	False
Dense Layer	Neurons	128
	Activation Function	ReLU
Output Layer	Classes	247
	Activation Function	Softmax
Training Configuration	Loss Function	Categorical Cross-Entropy
	Optimizer	Adam
	Learning Rate	0.001 (default)
	Epochs	10
	Batch Size	32
Regularization	Dropout	Not Applied
	Batch Normalization	Not Applied

Integration of Retrieval-Augmented Generation (RAG) for Context-Aware Tamil Sign Language Interpretation

The RAG framework improves TSL identification with the integration of contextual knowledge when managing information with gesture interpretation through deep learning. The retrieval module searches a knowledge base built from previously labeled gesture-meaning pairs, and then the generation module DCSL-GRN generates reliable linguistic outputs. The hybrid approach integrates the visual gesture recognition and the semantics. This allows for more accuracy in translation, flexibility, and efficiency in real-time interpretation. The integration of retrieval-based context (i.e., previous gesture meaning and attention) with generative modeling allows the system to become cognizant of gesture intent, distinguish between gesture ambiguity, and eventually generate Tamil text or speech translation of contextual meaningfulness for equitable communication.

4. Result

The DCSL-GRN framework of the research was assessed on its ability to identifyTSL. The outcomes of the evaluation suggest that the future model is effective in accurately recognizing gestures from video frames, exhibiting enhanced ability to learn spatial–temporal features more effectively, and showing improved reliability through preprocessing and augmentation, as well as improved accuracy of gesture recognition compared to traditional deep learning approaches.

Experimental Setup

Table 2 setup provides the computational environment required for training, testing, and real-time implementation of the Tamil Sign Language recognition model using deep learning frameworks.

Table 2: Experimental Hardware and Software Configuration for TSL Recognition

Category	Specification
Operating System	Windows 10 / 11 (64-bit), Ubuntu 20.04 LTS or later
CPU	Intel Core i7 / AMD Ryzen 7 or equivalent
GPU	NVIDIA GeForce GTX 1080 Ti / RTX series with CUDA support
GPU Toolkit	NVIDIA CUDA Toolkit 11.x, cuDNN 8.x
RAM	16 GB DDR4 or higher
Storage	512 GB SSD
Programming Language	Python 3.8 – 3.10
Deep Learning Frameworks	TensorFlow 2.x, PyTorch 1.10 or newer
Supporting Libraries	NumPy, SciPy, Pandas, Scikit-learn
Computer Vision Libraries	OpenCV 4.x, MediaPipe / OpenPose
Visualization Tools	Matplotlib, Seaborn
Development Environment	Jupyter Notebook / JupyterLab, Visual Studio Code (VS Code)
Environment Management	Anaconda Navigator / Conda virtual environments
Additional Tools	Tinker

Performance Evaluation

Table 3 introduced the metrics used to appraise the DCSL-GRN model's presentation in TSL recognition. These metrics facilitate, consistent, and reliable assessment of the replica's spatial-temporal gesture classification capabilities, and confirm its various abilities to deal with sign gestures that could be highly diverse and dynamic.

Table 3. Definition and significance of evaluation metrics used in the proposed DCSL-GRN model for Tamil Sign Language recognition

Metric	Definition (Related to Research)	Equation
Accuracy	DCSL-GRN model by evaluating how many gestures are correctly classified among all predictions, which represents the overall robustness of the recognition system for Tamil sign gestures. While *TP* and *TN* are true positive and negative,*FP* and *FN* are inaccurate ones.	$A = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	The proportion of all expected actions that are properly detected. This ensures that the model minimizes false gesture recognition from TSLvideos. FP only refers to incorrect positive predictions.	$P = \frac{TP}{TP + FP}$

Recall	Shows how well the model can recognize every importantTSL from the dataset, ensuring comprehensive gesture detection and reduced omission errors.	$R = \frac{TP}{TP + FN}$
F1-Score	Gives an equilibrium level of recall as well as precision, balancing the accuracy of movement detectionand reliability. It is crucial for evaluating recognition performance when data imbalance exists.	$F1 = \frac{2 * (precision * recall)}{precision + recall}$

The learning effectiveness of the model is offered in the form of accuracy and a loss performance output is illustrated in Figure 7. The training and validation accuracy in Figure 7 (a) continue to increase with epochs, demonstrating a successful convergence and generalization. The decrease in training and validation loss, as seen in Figure 7 (b), means that the model is optimized successfully with little evidence of overfitting. In general, the learning outputs validate that the model is stable, well-optimized, and efficient in classifying double-handed TSLgestures.

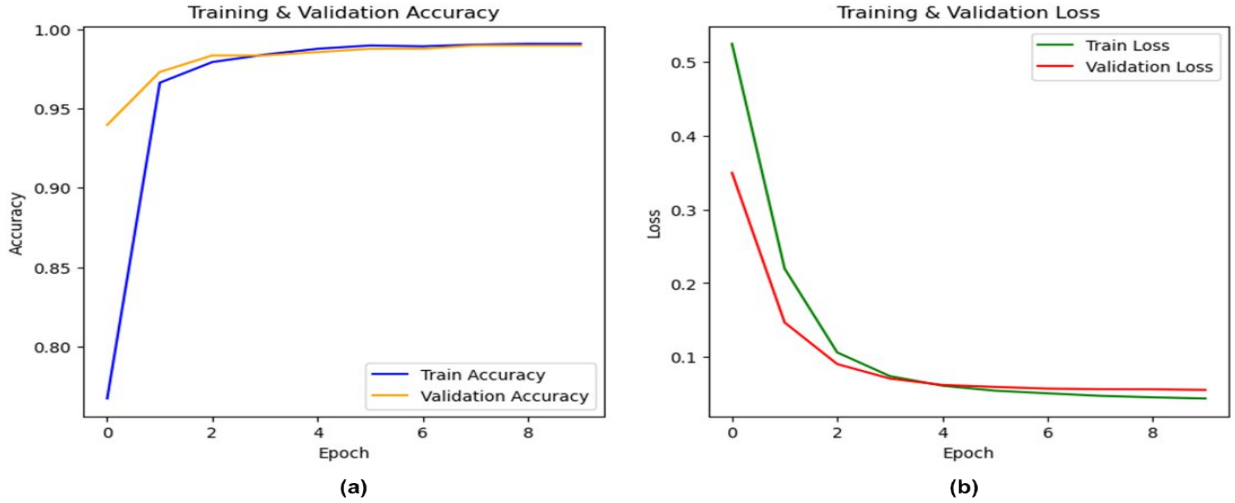

Figure 7: Training performance showing (a) accuracy and (b) loss reduction trends

Figure 8 shows the model's classification presentation with ROC and Precision–Recall curves. Figure8(a) shows the ROC curve with an AUC of 0.990, indicating excellent discriminative power on gesture classification. Figure8 (b) shows the average precision of 0.982 on the Precision–Recall curve and implies a solid balance between accuracy and recollection. The AUC and typical accuracy evaluations demonstrate the model's stability as well as its capacity to correctly recognize double-handed TSL movements.

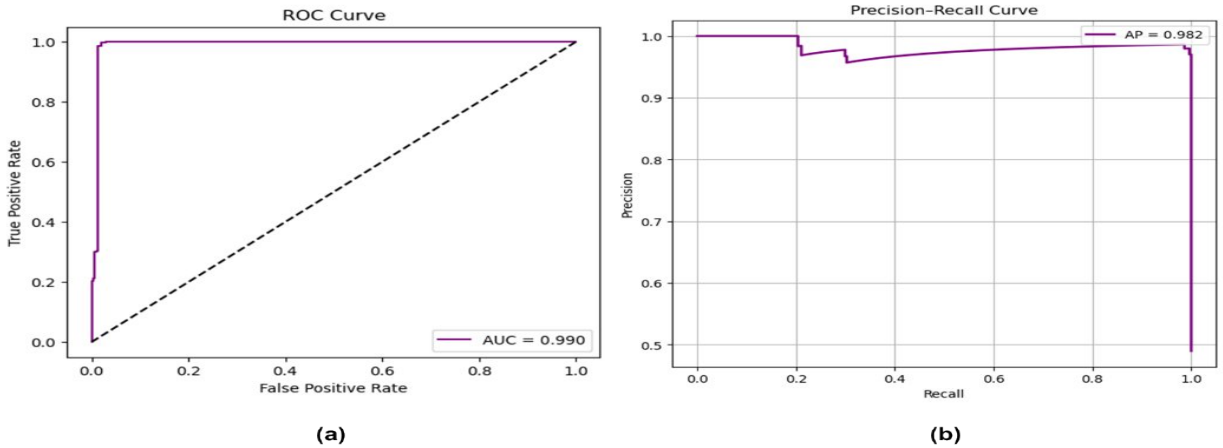

Figure 8: Performance evaluation showing (a) ROC curve and (b) Precision–Recall curve

The confusion matrix presented in Figure 9 evaluates the exactness of the DCSL-GRN model for the classification of TSLgestures. The slanting of the confusion matrix shows a high level of TP and

TNclassification and negligible misclassification. The presence of a clear diagonal indicates that the model was able to learn spatial-temporal features for classification well. Thus, it provides a safeguard of the accuracy and reliability of the organization's performance of recognizing sign gestures in the videos.

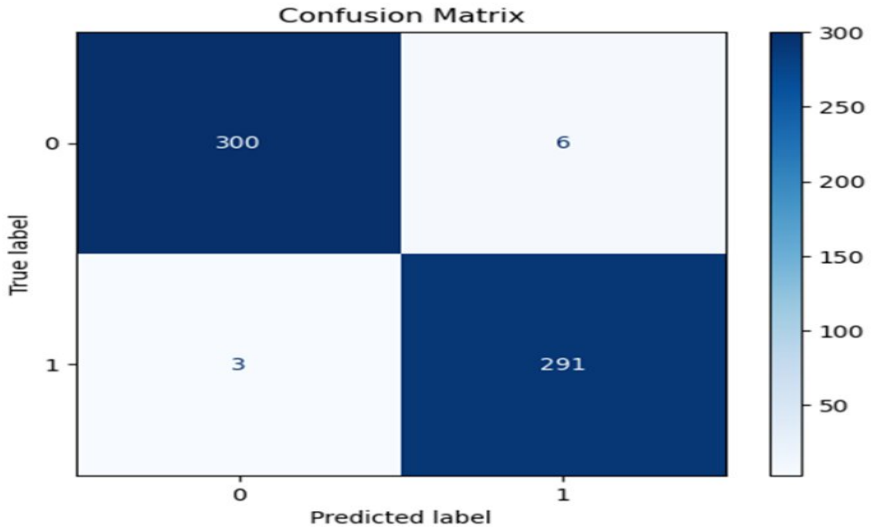

Figure 9: Confusion matrix illustrating classification performance of the proposed DCSL-GRN model.

Every neural layer, pooled layer, attentiveness layer, attention GRU layer, and thick layer has a different output size shown in Figure 10, reporting on the proposed model's aptitude to compress spatial features and ultimately output a temporal feature representation. Thus, it shows that TSLgesture representation can be accomplished efficiently through dimensionality reduction and feature extraction in the proposed DCSL-GRN model.

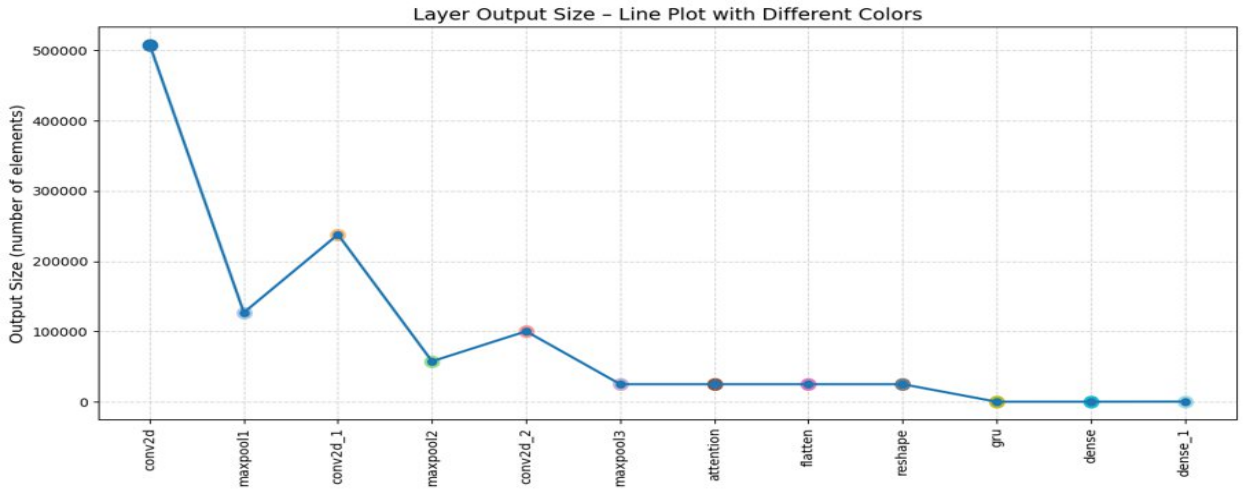

Figure 10: Layer-wise output size variation in the proposed DCSL-GRN model

The data in Table 4 summarizes the evaluation of existing techniques compared to the proposed model, DCSL-GRN. Although the traditional classifiers, SVM, RF, and XGBoost, show strong performance, DCSL-GRN produced balanced metrics. The performance illustrates that DCSL-GRN attained an improved spatial–temporal feature learning ability for gesture recognition of TSL. Figure 11 presents existing techniques vs. the proposed model, DCSL-GRN, and shows that the model achieves thereby validating improved recognition efficiency in gesture recognition.

Table 4: Performance comparison of existing methods with DCSL-GRN model

Methods	Accuracy	Precision	Recall	F-1score

RF [26]	96.81	97	97	97
XG Boost [26]	96.4	96	96	96
GB [26]	94.63	95	95	95
SVM [26]	90.39	94	90	92
HCL [27]	71	-	-	-
SFO optimized SOM [28]	93.23	-	96	-
DCSL-GRN [Proposed]	**98**	**97**	**98**	**97**

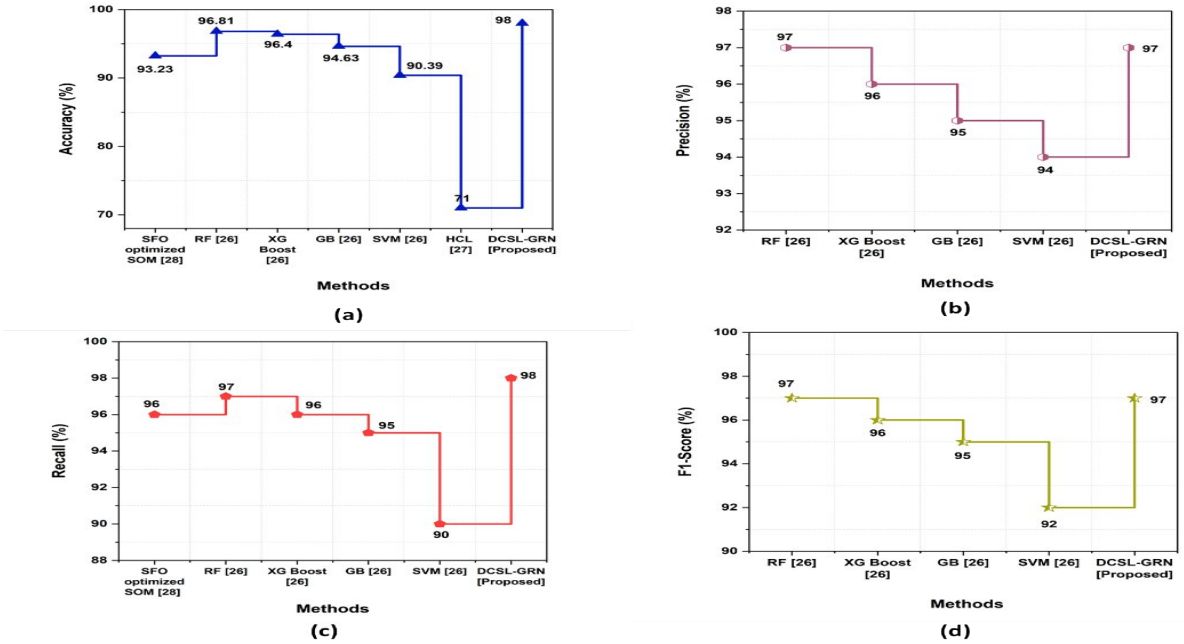

Figure 11: Comparative performance analysis showing (a) Accuracy, (b) Precision, (c) Recall, and (d) F1-Score for existing and proposed DCSL-GRN models

Graphical User Interface (GUI) Implementation

The GUI created allows for real-time engagement with the TSLrecognition system, which has a trained RAG model with a Spatial Attention layer. Figure 12 illustrates the two-step functionalities of the developed GUI for TSLrecognition. Specifically, it enables users to select gesture frames from a specific folder (Stage 1 represented in Figure 12 (a)) and then shows the predicted Tamil letters (Stage 2 represented in Figure 12 (b)).Created in Gradio, the interface visually displays recognized gestures alongside their outputs in text and as Tamil speech with Google Text-to-Speech (gTTS). This tangible interface illustrates an interface to the model's capabilities with gesture recognition and speech production while building bridges between sign inputs and meaningful spoken output in real-world applications.

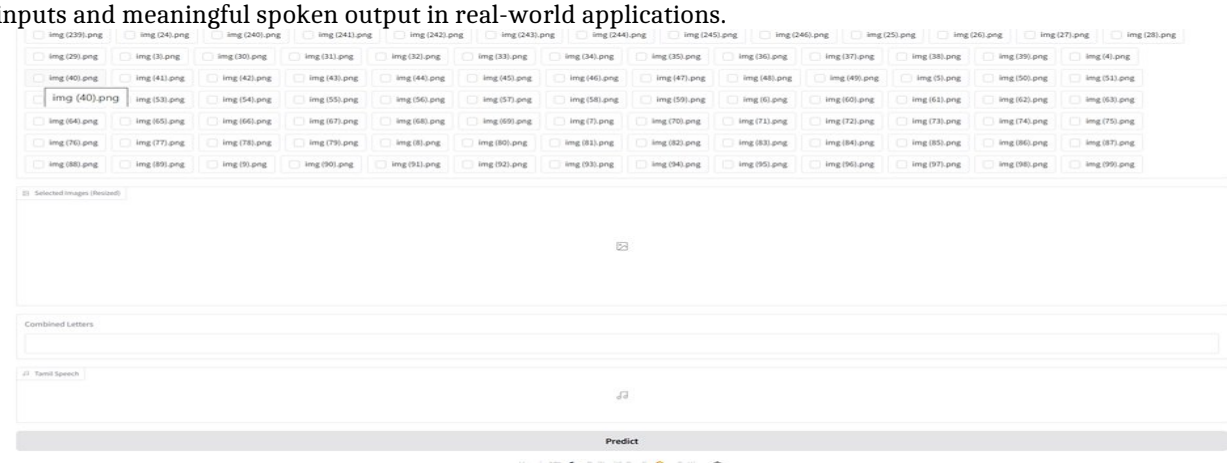

Figure 12: GUI interface Stage 1 (a) Gesture image selection from the dataset, allowing users to choose specific Tamil Sign Language gesture frames for recognition and prediction.

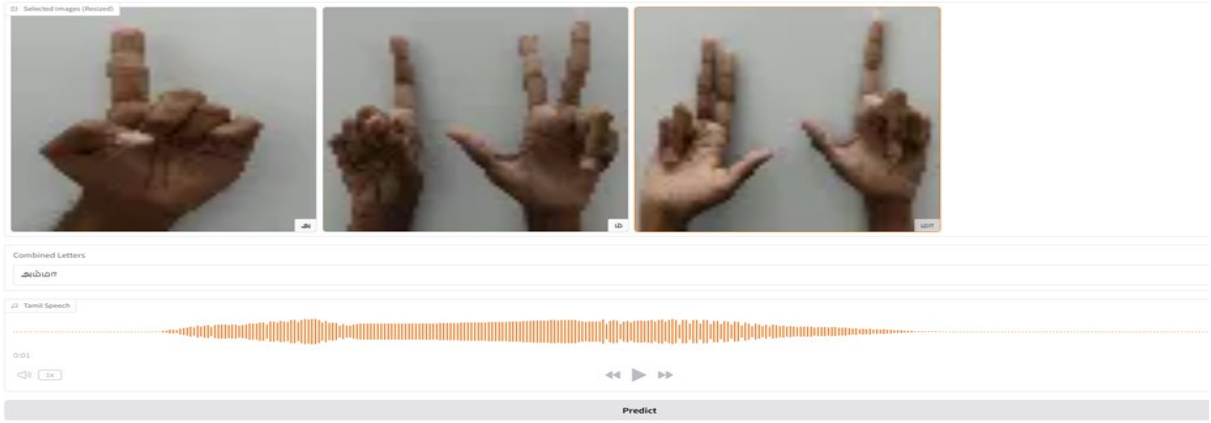

Figure 12: GUI interface Stage 2 (b) Recognized gesture outputs displayed as corresponding Tamil letters and combined text, demonstrating real-time conversion of sign gestures into meaningful linguistic representation.

4. Discussion

The substantial capability of the DCSL-GRN model is attributed to its hybrid architecture that synergistically integrates DCNN and GRU for sequential categorization modelling. This dual-layered architecture facilitates the simultaneous capture of gesture detail within a given frame and in the sequential temporal frames leading up to that frame. Incorporating spatial attention into the model improves interpretability by computing attentive areas that are more gesture-relevant.Gaussian augmentation added variability to the available dataset, which helped mitigate overfitting, helping to improve generalization in terms of lighting conditions, signer styles, and backgrounds. The model's convergence pattern displayed effectiveness of optimization and stability, with steadily increasing accuracy and decreasing loss. The high AUC of 0.990 and precision-recall balance of 0.982 further demonstrate reliability when detecting gestures and are critical for practical use in continuous and real-time recognition.

Comparison with related work

The DCSL-GRN framework demonstrates a more accuracy and interpretability improvements over current approaches. Conventional frameworks such as Random Forest (96.81%), and XGBoost (96.4%), worked well however they were not very robust in terms of time. Gradient Boosting (94.63%) and SVM (90.39) were not successful with sequential feature representation and SFO-optimized SOM (93.23) exhibited little adaptive capability. The HCL model scored 71 which is poor generalization. DCSL-GRN has an advantage over these, as it works well in terms of incorporating spatial-temporal dependencies with its gated recurrent and attention mechanisms. Earlier approaches, such as CNN–LSTM hybrids [14,15], obtained accuracies of approximately 90–97%, but at the expense of high computational resources and sensitivity to motion noise. Models based on ResNet and Transformer [17,18] also made progress in identifying static signals, but because they are devoid of temporal ability, they struggled to identify continuously overlapping movements. Incorporating spatial attention is an important difference from previous work focused on TSL [18], which relied on classical image processing techniques (SIFT and edge detection), rather than deep temporal learning. In addition, the use of the RAG framework adds a new layer of semantic reasoning that is not present in the vast majority of existing sign language systems [29]. While previous work has typically focused on classification models, this study [30] combines recognition with contextual inference and natural language translation, demonstrating a distinct advancement towards a more intelligent, human-like interpretation system.

Practical Implications

The OSAS bears great promise to improve accessibility to communication, education, and assistive technology. The real-time GUI created using Gradio facilitates communication between signers and non-signers, providing translations from TSL gestures to both text and speech. This technology can help create inclusive classrooms, allowing deaf students to take part in classes taught in written or spoken English via

automatic interpretation. In healthcare, public service, and customer support environments, the model can assist in interaction with hearing-impaired individuals without human interpreters. In addition, the DCSL-GRN and RAG combination allows for adaptation to multilingual systems of sign recognition, with easy retraining for other regional sign languages simply by adjusting the dataset and language modules.

Limitations and Future Work

Even though the suggested DCSL-GRN model is highly accurate and robust, its restrictions on vocabulary were limited to a small set of gestures and concentrated on the alphabet. Variations in signer style, lighting, and background can still have an influence on real-time recognition accuracy, and the fine finger articulations analysis is limited by the absence of 3D motion data. Future work will expand the dataset to continuous phrases, depth 3D or skeleton tracking Kinetic, and embrace Transformer to enhance the sequence ofKnowledge Portfolio optimization in the model to cellular and edge implementation, which will facilitate realistic, real-time.

5. Conclusion

The proposed DCSL-GRN model has mastered exemplary performance inTSL recognition through integration. Deep convolutional layers to extract spatial features, GRU for temporal sequence learning, and a Spatial Attention Mechanism for focused gesture analysis. It was using a system that was essentially using a Sobel edge detector, GraceCut segmentation, and a Fourier-based approach to improve the clarity and precision of gestures and feature extraction. The experimental results demonstrated a recognition accuracy of 98%, a precision of 97%, a recall of 98%, and an F1-score of 97%, confirming the model's effective performance.The value of 0.982 confirms that it has a good generalization and discriminatory ability. Additionally, the context-awareness was made possible by the integration of the RAG module. The DCSL-GRN framework, in general, provides trustworthy, smart, and universal real-time Tamil Sign Language recognition.

References

1. Podder, K. K., Ezeddin, M., Chowdhury, M. E., Sumon, M. S. I., Tahir, A. M., Ayari, M. A., & Kadir, M. A. (2023). Signer-independent Arabic sign language recognition system using a deep learning model. Sensors, 23(16), 7156. https://doi.org/10.3390/s23167156
2. Kyaw, N. N., Mitra, P., & Sinha, G. R. (2024). Automated recognition of Myanmar sign language using a deep learning module. International Journal of Information Technology, 16(2), 633–640. https://doi.org/10.1007/s41870-023-01680-2
3. Hdioud, B., & Tirari, M. E. H. (2023). A deep learning-based approach for the recognition of Arabic sign language letters. International Journal of Advanced Computer Science and Applications, 14(4). https://doi.org/10.14569/IJACSA.2023.0140447
4. Duraisamy, P., Abinayasrijanani, A., Candida, M. A., & Babu, P. D. (2023). Transforming sign language into text and speech through deep learning technologies. Indian Journal of Science and Technology, 16(45), 4177–4185. https://doi.org/10.17485/IJST/v16i45.2583
5. Kothadiya, D. R., Bhatt, C. M., Saba, T., Rehman, A., & Bahaj, S. A. (2023). SIGNFORMER: Deep vision transformer for sign language recognition. IEEE Access, 11, 4730–4739. https://doi.org/10.1109/ACCESS.2022.3231130
6. Yuniarno, E. M. (2024). Sign language recognition based on geometric features using deep learning. Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI, 13(2), 338–348. https://doi.org/10.23887/janapati.v13i2.82103
7. Subramanian, B., Olimov, B., Naik, S. M., Kim, S., Park, K. H., & Kim, J. (2022). An integrated mediapipe-optimized GRU model for Indian sign language recognition. Scientific Reports, 12(1), 11964. https://doi.org/10.1038/s41598-022-15998-7
8. Kasapbaşi, A., Elbushra, A. E. A., Omar, A. H., & Yilmaz, A. (2022). DeepASLR: A CNN-based human–computer interface for American Sign Language recognition for hearing-impaired individuals. Computer Methods and Programs in Biomedicine Update, 2, 100048. https://doi.org/10.1016/j.cmpbup.2021.100048
9. Huang, J., &Chouvatut, V. (2024). Video-based sign language recognition via ResNet and LSTM network. Journal of Imaging, 10(6), 149. https://doi.org/10.3390/jimaging10060149

10. Buttar, A. M., Ahmad, U., Gumaei, A. H., Assiri, A., Akbar, M. A., &Alkhamees, B. F. (2023). Deep learning in sign language recognition: A hybrid approach for the recognition of static and dynamic signs. Mathematics, 11(17), 3729. https://doi.org/10.3390/math11173729
11. Lu, C., Kozakai, M., & Jing, L. (2023). Sign language recognition with multimodal sensors and deep learning methods. Electronics, 12(23), 4827. https://doi.org/10.3390/electronics12234827
12. Noor, T. H., Noor, A., Alharbi, A. F., Faisal, A., Alrashidi, R., Alsaedi, A. S., &Alsaeedi, A. (2024). Real-time Arabic sign language recognition using a hybrid deep learning model. Sensors, 24(11), 3683. https://doi.org/10.3390/s24113683
13. Abdul Ameer, R. S., Ahmed, M. A., Al-Qaysi, Z. T., Salih, M. M., &Shuwandy, M. L. (2024). Empowering communication: A deep learning framework for Arabic sign language recognition with an attention mechanism. Computers, 13(6), 153. https://doi.org/10.3390/computers13060153
14. Haputhanthri, H. H. S. N., Tennakoon, H. M. N., Wijesekara, M. A. S. M., Pushpananda, B. H. R., & Thilini, H. N. D. (2023). Multi-modal deep learning approach to improve sentence-level Sinhala sign language recognition. International Journal on Advances in ICT for Emerging Regions (ICTer), 16(2), 21–30. https://doi.org/10.4038/icter.v16i2.7264
15. Jayanthi, P., Bhama, P. R. S., &Madhubalasri, B. (2023). Sign language recognition using deep CNN with normalized keyframe extraction and prediction using LSTM: Continuous sign language gesture recognition and prediction. Journal of Scientific & Industrial Research (JSIR), 82(7), 745–755. https://doi.org/10.56042/jsir.v82i07.2375
16. Jayanthi, P., Bhama, P. R., Swetha, K., & Subash, S. A. (2022). Real-time static and dynamic sign language recognition using deep learning. Journal of Scientific & Industrial Research, 81(11), 1186–1194. https://doi.org/10.56042/jsir.v81i11.52657
17. Priya, C. B., Ibrahim, S. S., Thangam, D. Y., Jency, X. F., & Parthasarathi, P. (2023). Tamil sign language translation and recognition system for deaf–mute people using image processing techniques. Applied Computational Engineering, 8(1), 380–386. https://doi.org/10.54254/2755-2721/8/20230193
18. Badiger, R. M., Yakkundimath, R., Konnurmath, G., &Dhulavvagol, P. M. (2024). Deep learning approaches for age-based gesture classification in South Indian sign language. Engineering, Technology & Applied Science Research, 14(2), 13255–13260. https://doi.org/10.48084/etasr.6864
19. Prakash, S. S., Devi, S. B. M., &Arulprakash, P. (2022). Educating and communicating with deaf learners using a CNN-based sign language prediction system. International Journal of Early Childhood Special Education, 14(2), 1308–5581.https://doi.org/10.9756/INT-JECSE/V14I2.245
20. Chirranjeavi, M., Aaruran, S., Gokulraj, V. A., Binoy, B. N., & Harikumar, M. E. (2024). TLFS23 Tamil language fingerspelling dataset. Data in Brief, 52, 109961. https://doi.org/10.1016/j.dib.2023.109961
21. Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A. B., & Corchado, J. M. (2022). DeepSign: Sign language detection and recognition using deep learning. Electronics, 11(11), 1780. https://doi.org/10.3390/electronics11111780
22. Natarajan, B., Rajalakshmi, E., Elakkiya, R., Kotecha, K., Abraham, A., Gabralla, L. A., &Subramaniyaswamy, V. (2022). Development of an end-to-end deep learning framework for sign language recognition, translation, and video generation. IEEE Access, 10, 104358–104374. https://doi.org/10.1109/ACCESS.2022.3210543
23. 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. https://doi.org/10.1016/j.procs.2023.01.117
24. Alsharif, B., Altaher, A. S., Altaher, A., Ilyas, M., &Alalwany, E. (2023). Deep learning technology to recognize the American Sign Language alphabet. Sensors, 23(18), 7970. https://doi.org/10.3390/s23187970
25. https://www.kaggle.com/datasets/s3programmerlead/tamil-sign-language-video-dataset/data
26. Perera, V., Ganesalingam, V., Ravi, V., & Thirunavukkarasu, J. (2024, December). Tamil Alphabet Sign Language Recognition Using Mediapipe and Machine Learning. In 2024, the 9th International Conference on Information Technology Research (ICITR) (pp. 1-5). IEEE.https://doi.org/10.1109/ICITR64794.2024.10857766
27. Anusuya, C., Roomi, S. M. M., &Senthilarasi, M. A. Signature-Based Dravidian Sign Language Recognition By Sparse Representation. International Journal on Natural Language Computing (IJNLC) Vol. 5.
28. Krishnaveni, M., Subashini, P., & Dhivyaprabha, T. T. (2018). An assertive framework for an automatic Tamil sign language recognition system using computational intelligence. In Nature-Inspired

Optimization Techniques for Image Processing Applications (pp. 55-87). Cham: Springer International Publishing.https://doi.org/10.1007/978-3-319-96002-9_3
29. Anithadevi, N., Palanisamy, S., Saranya Rubini, S., & Shrestha, S. (2025). MediaPipe-LSTM-Enhanced Framework for Real-Time Dynamic Sign Language Recognition in Inclusive Communication Systems. Engineering Reports, 7(7), e70142.https://doi.org/10.1002/eng2.70142
30. Kumar, M., Visagan, S. S., Mahajan, T., Natarajan, A., & Sreeja, P. S. (2025). Enhanced Sign Language Translation between American Sign Language and Indian Sign Language Using LLMs. IEEE Access.https://doi.org/10.1109/ACCESS.2025.3595943

List of Abbreviations

Abbreviation	Full Form (as in document / common expansion)
TSL	Tamil Sign Language
ISL	Indian Sign Language
ASL	American Sign Language
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
Bi-LSTM	Bidirectional Long Short-Term Memory
GRU	Gated Recurrent Unit
DCSL-GRN	Deep Convolve Spatial Layer–Gated Recurrent Neuronet
RAG	Retrieval-Augmented Generation
FFT	Fast Fourier Transform
GUI	Graphical User Interface
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
F1-Score	F1 Measure
ReLU	Rectified Linear Unit
GRN	Gated Recurrent Neuronet
Sobel	Sobel Edge Detection
GrabCut	Graph-Based Cut Segmentation
Adam	Adaptive Moment Estimation
FFN	Feed Forward Network
GPU	Graphics Processing Unit
CTC	Connectionist Temporal Classification
WER	Word Error Rate
BLEU	Bilingual Evaluation Understudy
GAN	Generative Adversarial Network
NMT	Neural Machine Translation
ResNet / ResNet-50	Residual Network (ResNet-50)
VGG / VGG-16	Visual Geometry Group Network (VGG-16)
Inception-V3	Inception Version 3 Network
EfficientNet	Efficient Convolutional Neural Network Model
ConvNeXt	Convolutional Neural Network Extension Architecture
AlexNet	AlexNet (CNN architecture by Krizhevsky et al.)
Transformer	Attention-Based Deep Learning Model
MediaPipe	Real-Time Framework for Pose and Gesture Detection
SIFT	Scale-Invariant Feature Transform
Canny	Canny Edge Detection Algorithm

PR Curve	Precision–Recall Curve
BatchNorm	Batch Normalization
Dropout	Dropout Regularization
MaxPool	Max Pooling
Softmax	Softmax Function
Epochs (E)	Training Iterations
Learning Rate (η)	Step Size during Optimization
Precision (P)	Positive Predictive Value
Recall (R)	Sensitivity
Accuracy (A)	Overall Correctness
Loss Function	Categorical Cross-Entropy
Augmentation	Data Augmentation
TLFS23	Tamil Language Fingerspelling Dataset (TLFS23)
SOM	Self-Organizing Map
SFO	SFO (document label: “SFO optimized SOM” — full form not specified in doc)
RF	Random Forest
XGBoost	eXtreme Gradient Boosting
GB	Gradient Boosting
SVM	Support Vector Machine
HCL	HCL (document label; full form not specified in doc)
gTTS	Google Text-to-Speech