electronics

Article

Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism

Diksha Kumari * and Radhey Shyam Anand

Department of Electrical Engineering, IIT (Indian Institute of Technology) Roorkee, Roorkee 247667, India; r.anand@ee.iitr.ac.in
* Correspondence: dkumari@ee.iitr.ac.in

Abstract: Sign language is a complex language that uses hand gestures, body movements, and facial expressions and is majorly used by the deaf community. Sign language recognition (SLR) is a popular research domain as it provides an efficient and reliable solution to bridge the communication gap between people who are hard of hearing and those with good hearing. Recognizing isolated sign language words from video is a challenging research area in computer vision. This paper proposes a hybrid SLR framework that combines a convolutional neural network (CNN) and an attention-based long-short-term memory (LSTM) neural network. We used MobileNetV2 as a backbone model due to its lightweight structure, which reduces the complexity of the model architecture for deriving meaningful features from the video frame sequence. The spatial features are fed to LSTM optimized with an attention mechanism to select the significant gesture cues from the video frames and focus on salient features from the sequential data. The proposed method is evaluated on a benchmark WLASL dataset with 100 classes based on precision, recall, F1-score, and 5-fold cross-validation metrics. Our methodology acquired an average accuracy of 84.65%. The experiment results illustrate that our model performed effectively and computationally efficiently compared to other state-of-the-art methods.

Keywords: sign language; gestures; attention mechanism; CNN; LSTM

check for updates

Citation: Kumari, D.; Anand, R.S. Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism. *Electronics* **2024**, *13*, 1229. https://doi.org/10.3390/electronics13071229

Academic Editors: Jongnam Kim, Heung-Il Suk and Youngbong Kim

Received: 26 February 2024
Revised: 19 March 2024
Accepted: 22 March 2024
Published: 26 March 2024

1. Introduction

More than 70 million people are prone to hearing and speech impairments, according to the World Federation of the Deaf. It signifies that a considerable portion of the world population has a communication gap with others in society, which hinders their daily interactions and promotes inequality. Sign language is a non-verbal interaction mode that uses gestures and facial expressions to convey thoughts and emotions. A sign is composed of mainly two components, namely manual and non-manual. The former consists of palm orientations, shape, and movement of hands. The facial expressions and body postures constitute the non-manual elements. The main motive of sign language recognition (SLR) is to develop a mode of communication between the deaf community and people without any hearing impediments, and it plays an essential role in implementing human–computer interaction (HCI).

SLR understands the language of gestures and transforms them into text or voice forms. There are two categories of SLR: one is isolated based on word sign gestures, and another is continuous, which considers sentence-level sign gestures. We focused on isolated SLR in this paper. Sign language possesses various regional and dialectal variations; many of the same sign gestures can be expressed using different hand shapes and motions. Due to this variability, it is challenging for SLR to predict and distinguish similar gestures. Earlier SLR techniques acquired data using data gloves or specific sensors for capturing the position, velocity, and orientation of hands. Although SLR delivers more accurate results through these devices, this approach is quite expensive and unnatural for the signers.

With the further advancement of deep learning, researchers focused on vision-based SLR techniques that use cameras for capturing data and making the process more natural. The vision-based approach faces challenges like cluttered backgrounds, varying illumination and image resolutions, and occlusions. All these factors make the architecture of SLR more complex. Numerous scholars have proposed machine-learning and deep-learning methods to recognize sign language. The conventional tradition-based SLR approach uses various feature descriptors to extract relevant features and separate classifiers for recognition. A lot of preprocessing is required to enhance the model's efficiency. These approaches could be more efficient in dynamic sign language gestures [1–5].

The traditional SLR methods involve extracting handcrafted features followed by classification using the hidden Markov model (HMM) or dynamic time warping (DTW) for learning the temporal features from the video sequence. In [6], the authors designed an HMM-based framework using the camera for real-time hand tracking to recognize ASL words in 40 classes [7]. In 1997, a scholar from Germany recognized isolated words using HMMs with an accuracy of 89.8%. Then, some researchers used an artificial neural network (ANN) for SLR. In [8], Huang and Huang recognized 15 hand gesture classes with a 3D Hopfield neural network with an accuracy of 91%. The authors in [9] proposed a framework based on similarity assessment for the recognition of Chinese sign language using trajectory features by DTW and visual and contour features. Hikawa and Kaida [10] proposed a hybrid ASL framework using a self-organizing map (SOM) for computing features and a Hebbian network to classify 24 classes. Selecting an appropriate feature descriptor for developing any model is a major problem that requires a long process of hits and trials.

With the advancement of technology, deep learning neural networks like CNN have been widely used in SLR systems. It has revolutionized the field of computer vision with its enhanced performance. It eliminated the extra steps involved in traditional approaches by combining feature extraction and classification stages. Generally, models like two-dimensional (2D)-CNN, three-dimensional (3D)-CNN, and networks with multimodal inputs are used by the researchers for SLR [3]. The deep learning neural network proposed in 2012 on the ImageNet dataset encouraged researchers to develop more CNN frameworks like VGG-16, VGG-19, Inception V3, MobileNetV2, etc. Pigou et al. [11] used 2D-CNN as a backbone model for feature extraction from the video frame sequence and classification model using fusion techniques. Unfortunately, the CNNs failed to capture temporal information and efficiently extracted spatial features from the video sequence. Then, 3D-CNNs were introduced, which have proven effective in simultaneously removing spatial and temporal features. The author in [12] used 3D-CNN for dynamic hand gesture classification with depth and intensity input data and achieved relatively high accuracy. Still, the network is computationally expensive due to many parameters. The 2D-CNN can only extract spatial features but fails to learn temporal features, which is crucial in dynamic SLR. The recognition of dynamic sign language gestures is quite challenging compared to static SLR. It involves extracting spatial and temporal features from long sequential data. Hence, some researchers used 3DCNNs to learn spatiotemporal features from the video sequence, which gave good results but suffered from high computational time. Many researchers have employed recurrent neural networks (RNNs), which are effective at processing sequential data but insufficient for learning spatial characteristics [13]. However, these networks need help with the problem of gradient explosion for learning long sequential data. The introduction of long short-term memory (LSTM) solved this problem. Many researchers extracted features from images using CNN and fed those to RNN. The author in [14] developed an ASL recognition system using the Inception network for spatial and RNN for temporal learning. In Ref. [15], the authors proposed an architecture combining pre-trained CNN (InceptionV3) and LSTM for ISL recognition tasks for spatial feature extraction and sequential learning, respectively. In Ref. [16], the authors implemented a robust cascaded SLR network using a single shot detector (SSD), CNN, and LSTM from video frame sequences to learn spatiotemporal information. More neurons are needed to store more information, which further makes the model complex and results in an overload of information. This issue can be resolved using an attention mechanism by selecting and processing the most relevant information from the observed data and discarding the redundant information with limited resources. This mechanism works on the principle of the human visual system to pay attention to significant regions helpful in detecting objects. Natural language processing tasks used attention-based networks; now, they are used extensively in the computer vision domain [17]. In our work, we have proposed a hybrid network using the benefits of CNN and LSTM with an attention mechanism for recognizing isolated sign language words. We have used the MobileNetV2 network, which has a lightweight structure with depth-wise convolutions for filtering valuable features from the frame sequence to minimize the complexity of the overall architecture [18]. Then, we cascaded the attention module to LSTM to optimize the weight distribution and learn the spatiotemporal relationships for acquiring the desired information from the temporal sequence. The attention module reduces the loss of information and improves the stability of the model.

The major contributions of this paper are listed as follows:

- We propose a CNN and LSTM-based method with an attention mechanism that is substituted over the output layer of the LSTM to detect the spatiotemporal features.
- The attention layer assigns different weights by employing probability distribution to focus on relevant cues in the sequence for the recognition of sign language gestures.
- The designed architecture is lightweight, with an optimum parameter count, and computationally efficient compared to the related existing methods.
- Various performance metrics and the K-fold approach were used to assess the model's efficiency and contributed to assuring the model's resilience.

The remaining portion of this paper is arranged as follows: Section 2 explains the model architecture. Then, the network's performance is analyzed and compared to the existing methods using various metrics in Section 3. Finally, in Section 4, we present our conclusions.

2. Materials and Methods

In this section, we present the dataset description, data pre-processing, and working mechanism of the proposed architecture.

2.1. Dataset

For the evaluation of our methodology, the Word Level American Sign Language (WLASL) dataset [19] mentioned in the paper "Word-level Deep Sign Language Recognition from Video: A New Largescale Dataset and Methods Comparison", published in 2020, was used. It is the largest signer-independent ASL dataset collected by native American signers with different backgrounds, signing styles, and dialects from 20 websites. It comprises over 2000 words from over 100 signers, with 21,083 samples. Hence, it is a challenging dataset with various subsets, with 100, 300, 1000, and 2000 glosses [20]. This paper experimented on a subset of 100 glosses, with 2038 videos created by 97 signers. Figure 1 shows the sample video frames from the WLASL dataset.

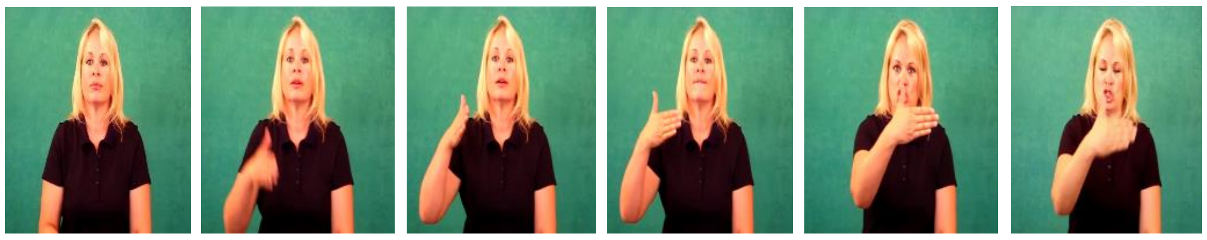


Figure 1. Sample frames of sign "Fish" captured from the WLASL dataset.

2.2. Dataset Pre-Processing

Since the length of video frames differs for each video, we used padding to equalize it according to the maximum frame sequence length. Then, we resized the frame to a size of 224 × 224. We normalized the resized frames by dividing each pixel value by 255. This operation sets the pixel values in the range [0, 1], making the input data fit for training machine learning models, especially neural networks.

2.3. Model Architecture

This section describes the architectural design of the model. The proposed model is an end-to-end framework for the classification of sign gestures. The model architecture, as shown in Figure 2, follows various stages explained in the subsections. First, the sign language gesture videos are given as input to the designed framework. The videos are transformed into an image/frame sequence and fed to CNN for feature extraction. CNN plays a crucial role in computer vision, especially in image processing, object recognition, and classification tasks. The human visual system inspires the structure of CNN to learn hierarchical local and global features from images using various layers. It mainly consists of the convolutional layer that extracts meaningful features from the image data. Finally, the pooling layers are employed to reduce dimensionality. The fully connected layers convert the feature vector obtained into a resulting output that predicts the classes using a probability distribution. There is no need for training from scratch as it uses the weights of an already trained network on the ImageNet database to finish the problem on the new dataset. Transfer learning strategy enhances the model's efficiency [21].

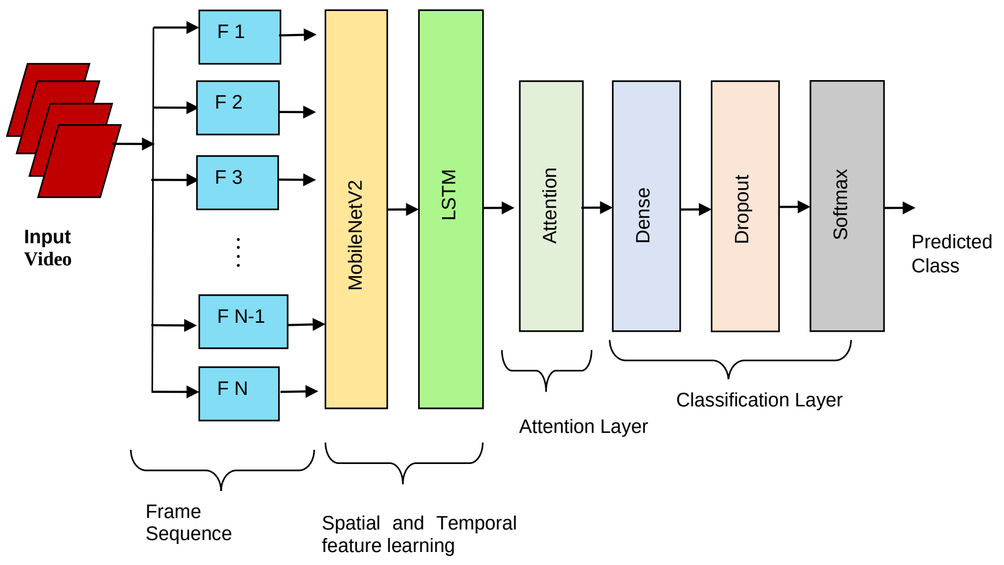


Figure 2. The block diagram of the proposed approach, illustrating various steps involved in sign language classification.

We used MobileNetV2 [22], a pre-trained network proposed by Google, for spatial feature extraction from the frames. It empowers deep learning neural network applications on mobile devices due to its compact, low-latency, and energy-efficient CNN framework. It uses depthwise separable convolutions that divide the normal convolution process into two separate phases: depthwise and pointwise convolutions. This step optimized the computational efficiency and size of the model and minimized the parameters. The enhanced version of MobileNet, known as MobileNetV2, comprises various components, such as linear bottlenecks with skip connections, inverted residual blocks, and squeeze and excitation blocks. These blocks play an essential role in improving the performance while maintaining accuracy.

It fulfills the diverse computational needs as the magnitude and dimensions of the network can be modified [23]. The spatial features are captured from the last max-pooling layer with a size of 1 × 1 × 1280. Thus, the proposed model generates feature vector sequences that have the shape of m × n, where m represents the size of the vector, and n gives the video frame length. After extracting the feature vector from the MobileNetV2 network, it is fed to the LSTM to learn the temporal relationship with the sequence. The LSTM network proposed by Hochreiter and Schmidhub [24] is a variant of RNN. It consists of memory cells and control units that save the sequential information for extended intervals. The design structure of LSTM consists of three types of gates: input, forget, and output gates with a sigmoid activation function. The cell state is an essential part of LSTM that shrinks the values between 0 and 1 using the sigmoid activation function and helps pass the information throughout the module. Traditional RNNs must deal with the gradient vanishing and gradient explosion problems while processing long sequential data. The gates in the LSTM module decide whether to keep or remove the information in the cell state, providing a solution to the problems. The LSTM module comprises three main gates: forget, input, and output gates. Each neuron of the memory cell of the LSTM has a present state and a hidden layer.

The motive of the forget gate is to decide whether to discard or keep the data from the cell state using a sigmoid function. If the sigmoid value is 1, the cell state saves the information, whereas if the output is 0, it discards it entirely.

In Equation (1), F_t is the forget gate, x_t is the information stored in the memory cell, and h_t represents the output in every cell. The weight vector $\left(W_f\right)$ and bias $\left(b_f\right)$ of the forget gate are adjusted by the sigmoid activation function (σ).

$$F_t = \sigma\left(W_f\,[h_{t-1},\, x_t] + b_f\right) \tag{1}$$

Then, in the next step, LSTM decides which information to store in the cell state after processing the sequence of inputs using Equation (2). The input gate I_t, with the activation function, regulates the value between −1 and 1 to generate the new candidate values as $\tilde{C}_t$, represented in Equation (3). After this, based on the two values I_t and $\tilde{C}_t$, the cell state is updated and multiplied by f_t to predict whether to keep the information of the previous state or not [25,26].

$$I_t = \sigma(W_i\,[h_{t-1},\, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_c\,[h_{t-1},\, x_t] + b_c) \tag{3}$$

$$C_t = F_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{4}$$

Finally, the output gate uses the sigmoid activation function to decide which memory information to pass on to the next layer using sigmoid activation and determines the final hidden state, as shown in Equations (5) and (6).

$$O_t = \sigma(W_o\,[h_{t-1},\, x_t] + b_o) \tag{5}$$

$$H_t = O_t \times tanh(C_{\,t}) \tag{6}$$

The sequence of spatial features acquired from the video frames at each moment 't' is represented as X of length n.

$$X = [x_0,\, x_1,\, \ldots\ldots,\, x_n] \tag{7}$$

The input feature sequence is passed on to the LSTM network 1280 hidden units that consist of several memory cells and output a series of hidden states that summarize the temporal information as follows:

$$h = [h_0, h_2, \ldots\ldots, h_n] \tag{8}$$

After this, the Bahdanau attention mechanism [27] is integrated into the LSTM model output, informing the model to focus on a particular portion of the feature sequence. The attention layer learns the importance of each hidden state. Attention-based architecture attends to every hidden state at every time step, and then, the most informative one is decided for computing predictions, as shown in Figure 3.

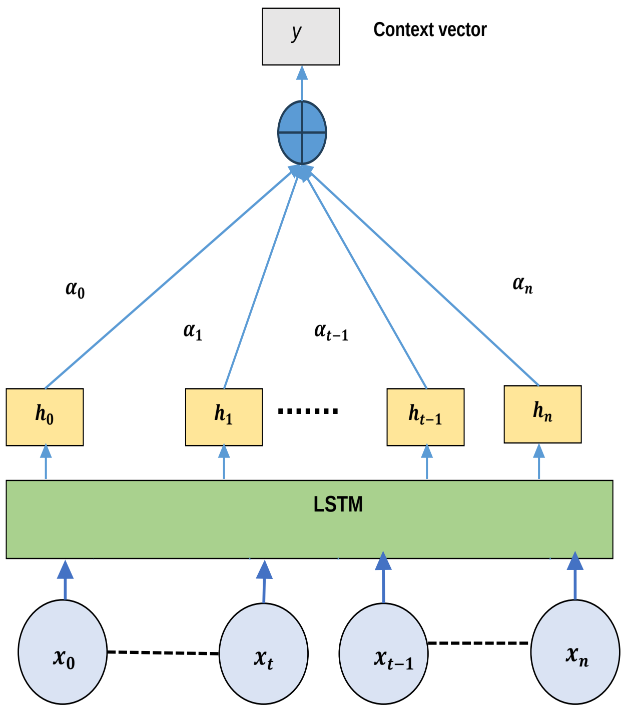


Figure 3. Structure of LSTM fused with an attention mechanism.

First, the attention weights α_{ti} are calculated based on the matching of x_t and h_i. The attention mechanism normalizes the attention weights using a feed-forward framework with a hyperbolic tangent as an activation function, as shown in Equation (9), where v^t and W_a are the weight vector and weight matrix, respectively.

$$\alpha_t = v^t \tanh(w_a[x_t; h_i]) \tag{9}$$

The obtained attention score is further normalized using the softmax function as follows:

$$\alpha_{ti} = \frac{\exp(score(x_t,\ h_i))}{\sum_{i=1}^{t} \exp(score(x_t,\ h_i))} \tag{10}$$

The context vectors at time t, which is the sum of hidden states of the input sequences, are weighted by the attention score using Equation (11). In this equation, α_{ti} represents the present sequence's output probability and demonstrates the sign gesture's final state in the frame sequence [4,28].

$$c_t = \sum_{i=1}^{t} \alpha_{ti} h_i \tag{11}$$

The dynamic gesture spatiotemporal feature vector $\boldsymbol{c_t}$, extracted from the LSTM output, is fed to the dense layer with 128 units. Then, we used a dropout layer with a dropout value of 0.7 to avoid overfitting while training the model. Finally, the fully connected layer with neuron units is equivalent to the total classes in the dataset, and the softmax activation function gives the prediction of sign classes. Algorithm 1 lists the important steps in the proposed architecture.

Algorithm 1. Hybrid CNN-LSTM with attention to sign language recognition.

```
Input: Input sign gesture video frame sequence
Output: Prediction of class labels
Step 1: Spatial feature extraction
# Loop over each frame in the frame sequence to compute features
for frame in frame sequence do
  for frame in frame sequence do
            $Features \leftarrow MobileNetV2$ feature extractor ()
end for
Step 2: Attention-based LSTM
 for $t$ in range (feature sequence length $\boldsymbol{X}$):
      # Compute hidden state $\boldsymbol{h_t}$ using Equations (1)–(6)
           Set LSTM output $\boldsymbol{h}$ to the sequence of hidden states $\boldsymbol{h} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_t)$;
           for each hidden state sequence $\boldsymbol{h}$ do
                  Compute attention weights $\boldsymbol{\alpha_{ti}}$, context vector $\boldsymbol{c_t}$ using Equations (9)–(11).
                         context vector = Attention ($\boldsymbol{x_t}$, $\boldsymbol{h_t}$)
           end for
# Apply a Dense layer with ReLU activation to the context vector
dense1 = Dense (128, activation="relu") (context vector)
# Apply Dropout layer with a dropout rate of 0.7
dropout = Dropout (0.7) (dense1)
# Apply a Dense layer with Softmax activation for classification of sign gesture class
output = Dense (100, activation="softmax") (dropout)
end for
```

3. Results and Discussion

The efficiency and superiority of the implemented methodology are analyzed in this section, with a large-scale WLASL dataset. First, we described the implementation details of the experimental setup and then demonstrated the performance of the proposed framework with various statistical measures. Then, a comparative analysis is illustrated with the other state-of-the-art networks based on accuracy and computational efficiency.

3.1. Implementation Details

The proposed network extracts frames from the sign language videos, and pre-trained MobileNetV2 was adopted as a feature extractor. For the training of the model, an adaptive moment estimation (Adam) was used for optimization with beta 1, beta 2, and epsilon as 0.99, 0.999, and 1×10^{-7}, respectively. The exponential decay learning rate scheduler, with an initial learning rate of 0.001, decay steps of 10,000, and decay rate of 0.96, was used with the optimizer to decrease the learning rate with training that will allow the model to properly converge at the optimal point where the loss is minimum. The "categorical cross-entropy" loss function was used in the training. The epochs and batch size were set to 200 and 32, respectively. We experimented using NVIDIA GTX 1060 GPU with the Python framework on the Windows 10 operating system, with configurations of Intel(R) Core (TM) i7 processor, 2.40 GHz CPU, and 16 GB RAM. The model was trained for approximately one hour.

3.2. Performance Measures

The network performance was measured through statistical metrics like accuracy, recall, precision, F1-score, and K-fold cross-validation. These performance metrics are critical in determining the efficiency of deep learning models across diverse positive and negative samples. Accuracy is a widely used metric that demonstrates how many instances are classified adequately from all classified samples. The precision value is the ratio of accurately categorized positive classes to the total sum of optimistic classifications. Recall indicates how many samples are predicted as accurate out of all the predictions. It measures how good the network is at predicting all the positives. The F1-score metric combines precision and recall to create balance. It gives the harmonic mean between precision and recall [21,29].

The evaluation metrics are calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{12}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

where TP, FP, TN, and FN signify the value of true positives, false positives, true negatives, and false negatives. Furthermore, K-fold cross-validation (K = 5) tests the validation of the classifier. This technique partitions the data into 'K' folds, and the model is trained for each fold, ensuring the efficiency of the network is not affected by the train-test split. Figure 4a,b demonstrates the accuracy and loss graphs of the model with an attention mechanism.

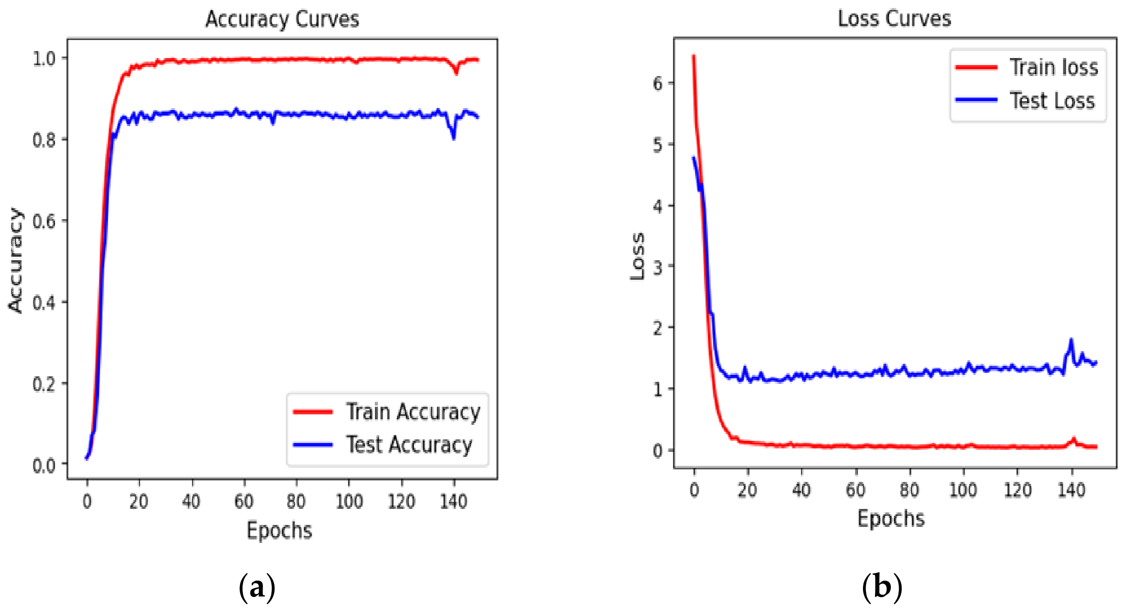


Figure 4. Graphs for (**a**) accuracy and (**b**) loss.

In Figure 5, the bar chart demonstrates each fold's precision, recall F1-score, and accuracy. The highest accuracy of 85.47% is achieved for the 4th fold with precision, recall, and F1-score values of 86, 87, and 84, respectively. The average values of accuracy, precision, recall, and F1-score are 84.65%, 86.8, 87.4, and 84.4, respectively. These metrics give comprehensive details of the model's efficiency and indicate that the model has performed relatively well, suggesting that it has made correct predictions while reducing false positives and negatives. The confusion matrix in Figure 6 illustrates the per-class accuracy of the dataset for K = 4-fold using the designed architecture.

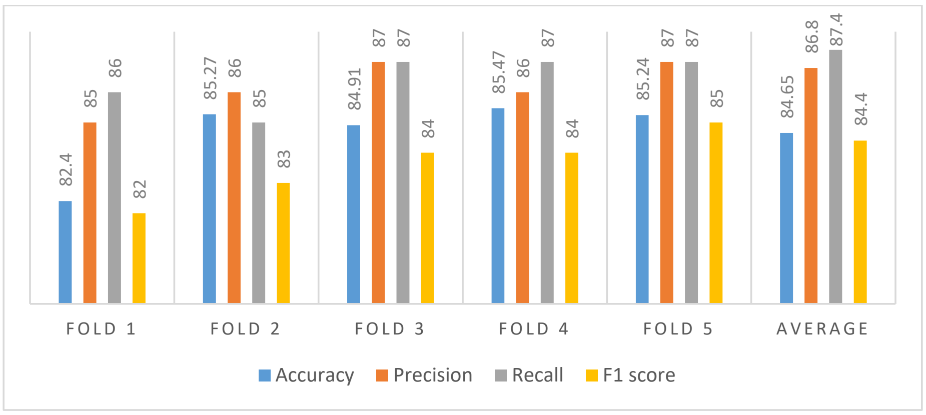


Figure 5. Performance metric analysis for each fold.

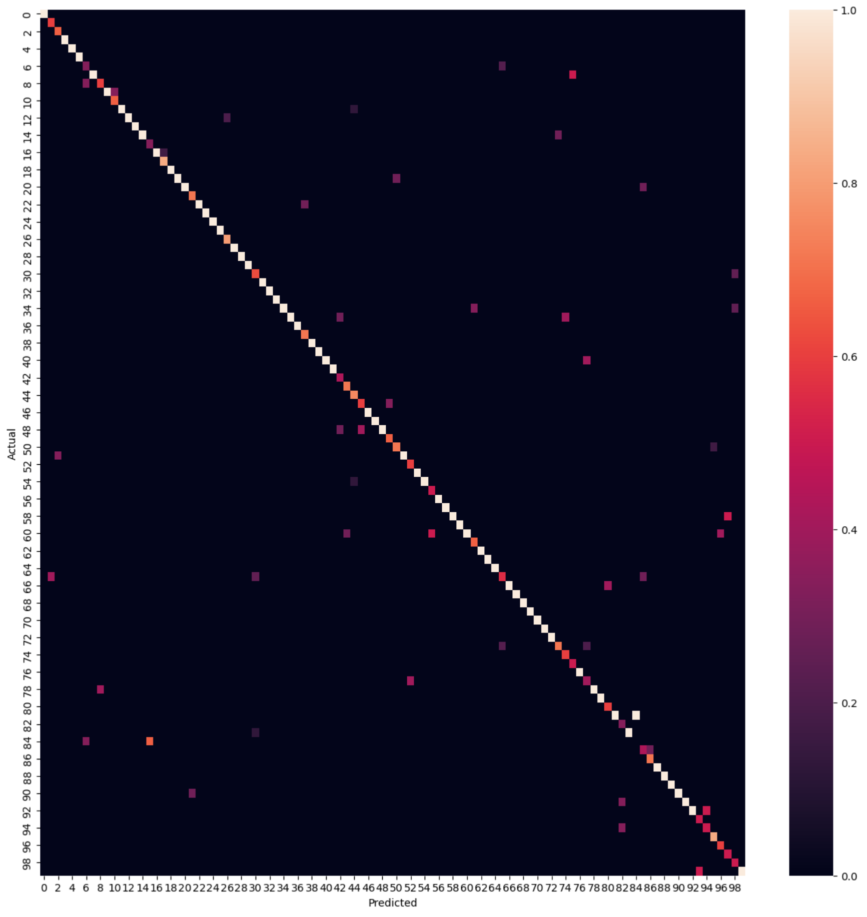


Figure 6. Confusion matrix.

Our model showed good performance across almost the whole dataset; however, it was not able to predict sign classes 'full', 'soon', 'thin', 'Thursday', 'tell', 'pizza', and 'why'. The sign class 'bar' was misclassified with class 'theory' and 'brother'. The irregularity of spatial and temporal features caused by similarity in the movement of gestures can be the reason for this. After analyzing the dataset, we found that different signers have performed the same sign class differently, which led our model to result in some wrong predictions.

3.3. Comparison with the Existing Techniques

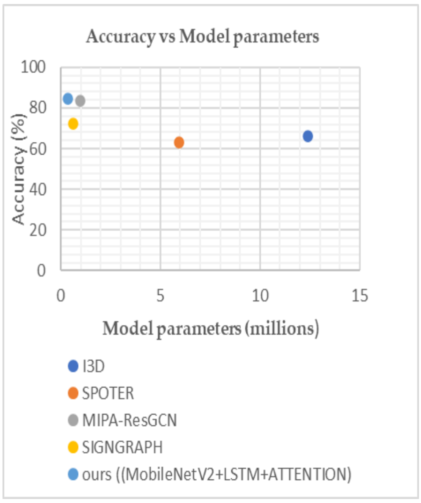
The performance of the proposed methodology was compared to other RGB-based and pose-based frameworks tested on the WLASL dataset, as shown in Table 1. The authors in [19] used VGG16 as a backbone model to extract spatial features with stacked Gated Recurrent Unit (GRU) and Inception 3D convolutional networks for sequential learning. The authors in [30] coarsely aligned the isolated signs and employed a temporal attention strategy to recognize the sign language. A full transformer network, with a Swin transformer as a spatial encoder and a mask future transformer, has been used for recognizing word signs [31]. Pose-based methods: Sign Pose-based Transformer (SPOTER) and Graph Convolutional Network (GCN) with BERT were used to detect sign videos. It was observed from the comparison that by using only pose data as input, the accuracy is considerably less than using full RGB data. Pose data lack in providing knowledge about fingers and facial expressions, which are very useful for recognizing sign language gestures. Our proposed network with hybrid CNN and LSTM with an attention mechanism performed better than the existing methods by achieving 84.65% prediction accuracy.

Table 1. Performance comparison of the model with other state-of-the-art methods.

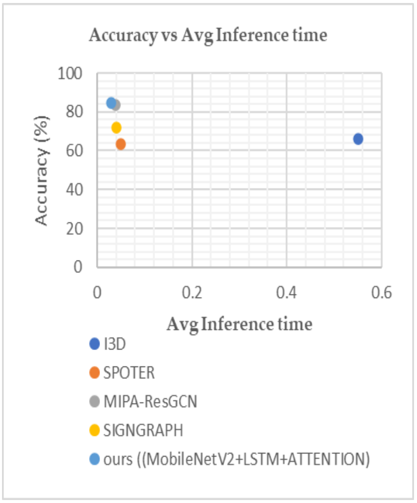
Method Used	Input	Parameters	Training Epochs	Avg Infer Time	Top-1 Accuracy (%)
I3D [19] (2020)	RGB	12.4 M	200	0.55 s	65.89
TK-3DConvNet [30] (2020)	RGB	-	-	-	77.55
Full Transformer Network [31] (2022)	RGB	-	-	-	80.72
GCN-BERT [32] (2021)	Pose	-	100	-	60.15
SPOTER [20] (2022)	Pose	5.92 M	-	0.05 s	63.18
MIPA-ResGCN [2] (2023)	Pose	0.99 M	350	0.0391	83.33
SIGNGRAPH [1] (2023)	Pose	0.62 M	350	0.0404	72.09
Ours (MobileNetV2 + LSTM + ATTENTION)	**RGB**	**0.374 M**	**150**	**0.0302**	**84.65**

3.4. Computational Performance Analysis

In this study, we conducted a comparison of our proposed architecture with I3D (an appearance-based method), SPOTER, (MIPA-ResGCN), and SIGNGRAPH (pose-based methods) to evaluate the computational efficiency based on the number of model parameters and average inference time taken to process each video. Our model has 0.374 million parameters, which are less than those of MIPA-ResGCN, SIGNGRAPH, I3D, and SPOTER, with 0.99 million, 0.62 million, 12.4 million, and 5.92 million model parameters, respectively. Figure 7 shows that our model is computationally efficient, with fewer parameters, less average inference time and reasonable accuracy than the other existing methods.



(**a**)



(**b**)

Figure 7. Comparison of the accuracy of the proposed model in terms of (**a**) model parameters (in millions) and (**b**) average inference time (in sec).

We also analyzed the time complexity of the models using Big-Oh notation, which computes the total time for each operation in the algorithm. This is the most commonly used notation for handling worst-case scenarios, and it also describes running times and space bounds based on a parameter 'n' that varies according to the task [33]. In case of hybrid CNN with attention-based LSTM, we have used MobileNetV2, which consumes O $\left(\sum_{i=1}^{L} C_{L-1} \times M_L^2 \times C_L \times S_L^2\right)$, where M, S, and C represent the dimensions of the feature map, size of the kernel, and channel count for every convolutional layer L. Then, we used LSTM with an attention mechanism for classification, which has a computational complexity of [2·O ($k \times h$)], where k signifies the total outputs and h is the number of neuron units in the hidden layer. The attention mechanism has time complexity of O ($n \times m$), where n represents the dimensionality of hidden states and m gives the length of the sequence.

The I3D Convolutional network is 3DCNN, which involves a 3D filter for convolution operations. The time complexity is O $\left(\sum_{i=1}^{L} C_{L-1} \times M_L^2 \times C_L \times S_L^3\right)$. The S_L^3 term signifies the cubic kernel that makes the network computationally expensive. In [32], the authors used a transformer-based network with a multi-head attention mechanism that consumes O (6 × n^2 × d) + O (6 × n × d^2), where n is the sequence length and d represent the dimensionality of the input embeddings. There are six layers for each encoder and the decoder with nine heads. Also, the parameter count of this model is 5.92 M, which makes it time-consuming during training and inference. The architecture proposed in [1,2] employed a graph convolutional network (GCN) that has a time complexity of O ($k \times e \times d + k \times n \times d^2$), where the variables n, e, K, and d represent the total number of nodes, edges, layers, and dimensions of the node hidden features utilizing pose data as input.

Our model achieved high recognition accuracy with fewer model parameters, fewer training epochs, less average inference time, and less temporal complexity than other models. It also maintained an appropriate balance between accuracy and efficiency.

4. Conclusions

Both spatial and temporal features play a vital role in implementing dynamic SLR. This paper presents a novel hybrid CNN-attention-based LSTM framework for recognizing isolated word sign language gestures from the video frame sequence. We employed MobileNetV2 CNN to extract salient features from the video frames in this work. The attention mechanism is based on weight allocation and detects the most significant information by distributing higher weights. Hence, it positively optimized the traditional LSTM model by allowing it to dynamically weigh the significance of different input elements. It improves task performance with long sequences, alleviating the need to compress all information into a fixed-size representation. It provides some interpretability as we can visualize where the model is "looking" when generating an output. The features from the CNN are passed in sequence to LSTM to learn the long-term dependencies. An attention layer is incorporated into the hidden layers of LSTM to enhance its efficiency and determine more detailed spatiotemporal information from the data. We have experimented on the WLASL dataset with 100 classes. Our system achieved a classification accuracy of 84.65%, reporting nearly 2% to 3% improvements over the relevant existing methods.

Author Contributions: Study conception and design: D.K.; data collection: D.K.; analysis and interpretation of results: D.K. and R.S.A.; draft manuscript preparation: D.K. and R.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors have no conflicts of interest to declare in this paper.

References

1. Naz, N.; Sajid, H.; Ali, S.; Hasan, O.; Ehsan, M.K. Signgraph: An Efficient and Accurate Pose-Based Graph Convolution Approach Toward Sign Language Recognition. *IEEE Access* **2023**, *11*, 19135–19147. [CrossRef]
2. Naz, N.; Sajid, H.; Ali, S.; Hasan, O.; Ehsan, M.K. MIPA-ResGCN: A multi-input part attention enhanced residual graph convolutional framework for sign language recognition. *Comput. Electr. Eng.* **2023**, *112*, 109009. [CrossRef]
3. Wang, F.; Zhang, L.; Yan, H.; Han, S. TIM-SLR: A lightweight network for video isolated sign language recognition. *Neural Comput. Appl.* **2023**, *35*, 22265–22280. [CrossRef]
4. Huang, J.; Zhou, W.; Li, H.; Li, W. Attention-based 3D-CNNs for large-vocabulary sign language recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2822–2832. [CrossRef]
5. Das, S.; Biswas, S.K.; Purkayastha, B. A deep sign language recognition system for Indian sign language. *Neural Comput. Appl.* **2023**, *35*, 1469–1481. [CrossRef]
6. Starner, T.; Weaver, J.; Pentland, A. Real-time american sign language recognition using desk and wearable computer-based video. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1371–1375. [CrossRef]
7. Grobel, K.; Assan, M. Isolated sign language recognition using hidden Markov models. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 1, pp. 162–167.
8. Huang, C.L.; Huang, W.Y. Sign language recognition using model-based tracking and a 3D Hopfield neural network. *Mach. Vis. Appl.* **1998**, *10*, 292–307. [CrossRef]
9. Wang, L.C.; Wang, R.; Kong, D.H.; Yin, B.C. Similarity assessment model for Chinese sign language videos. *IEEE Trans. Multimed.* **2014**, *16*, 751–761. [CrossRef]
10. Hikawa, H.; Kaida, K. Novel FPGA implementation of hand sign recognition system with SOM–Hebb classifier. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 153–166. [CrossRef]
11. Pigou, L.; Dieleman, S.; Kindermans, P.J.; Schrauwen, B. Sign language recognition using convolutional neural networks. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6–7 and 12, 2014, Proceedings, Part I 13*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 572–578.
12. Molchanov, P.; Gupta, S.; Kim, K.; Kautz, J. Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–15 June 2015; pp. 1–7.
13. Huang, Y.; Huang, J.; Wu, X.; Jia, Y. Dynamic Sign Language Recognition Based on CBAM with Autoencoder Time Series Neural Network. *Mob. Inf. Syst.* **2022**, *2022*, 3247781. [CrossRef]
14. Bantupalli, K.; Xie, Y. American sign language recognition using deep learning and computer vision. In Proceedings of the 2018 IEEE International Conference on Big Data, Seattle, WA, USA, 10–13 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4896–4899.
15. Aparna, C.; Geetha, M. CNN and stacked LSTM model for Indian sign language recognition. In *Machine Learning and Metaheuristics Algorithms, and Applications: First Symposium, SoMMA 2019, Trivandrum, India, December 18–21, 2019, Revised Selected Papers 1*; Springer: Singapore, 2020; pp. 126–134.
16. Rastgoo, R.; Kiani, K.; Escalera, S. Video-based isolated hand sign language recognition using a deep cascaded model. *Multimed. Tools Appl.* **2020**, *79*, 22965–22987. [CrossRef]
17. Ming, Y.; Qian, H.; Guangyuan, L. CNN-LSTM Facial Expression Recognition Method Fused with Two-Layer Attention Mechanism. *Comput. Intell. Neurosci.* **2022**, *2022*, 7450637. [CrossRef] [PubMed]
18. Bousbai, K.; Merah, M. A comparative study of hand gestures recognition based on MobileNetV2 and ConvNet models. In Proceedings of the 2019 6th International Conference on Image and Signal Processing and their Applications (ISPA), Mostaganem, Algeria, 24–25 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
19. Li, D.; Rodriguez, C.; Yu, X.; Li, H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1459–1469.
20. Boháček, M.; Hrúz, M. Sign pose-based transformer for word-level sign language recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 182–191.
21. Das, S.; Biswas, S.K.; Purkayastha, B. Automated Indian sign language recognition system by fusing deep and handcrafted feature. *Multimed. Tools Appl.* **2023**, *82*, 16905–16927. [CrossRef]
22. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
23. Hassan, N.; Miah, A.S.M.; Shin, J. A Deep Bidirectional LSTM Model Enhanced by Transfer-Learning-Based Feature Extraction for Dynamic Human Activity Recognition. *Appl. Sci.* **2024**, *14*, 603. [CrossRef]
24. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
25. Venugopalan, A.; Reghunadhan, R. Applying Hybrid Deep Neural Network for the Recognition of Sign Language Words Used by the Deaf COVID-19 Patients. *Arab. J. Sci. Eng.* **2023**, *48*, 1349–1362. [CrossRef] [PubMed]

26. Tay, N.C.; Tee, C.; Ong, T.S.; Teh, P.S. Abnormal behavior recognition using CNN-LSTM with attention mechanism. In Proceedings of the 2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), Kuala Lumpur, Malaysia, 25 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
27. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
28. Natarajan, B.; Rajalakshmi, E.; Elakkiya, R.; Kotecha, K.; Abraham, A.; Gabralla, L.A.; Subramaniyaswamy, V. Development of an end-to-end deep learning framework for sign language recognition, translation, and video generation. *IEEE Access* **2022**, *10*, 104358–104374. [CrossRef]
29. Lanjewar, M.G.; Panchbhai, K.G.; Patle, L.B. Fusion of transfer learning models with LSTM for detection of breast cancer using ultrasound images. *Comput. Biol. Med.* **2024**, *169*, 107914. [CrossRef] [PubMed]
30. Li, D.; Yu, X.; Xu, C.; Petersson, L.; Li, H. Transferring cross-domain knowledge for video sign language recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6205–6214.
31. Du, Y.; Xie, P.; Wang, M.; Hu, X.; Zhao, Z.; Liu, J. Full transformer network with masking future for word-level sign language recognition. *Neurocomputing* **2022**, *500*, 115–123. [CrossRef]
32. Tunga, A.; Nuthalapati, S.V.; Wachs, J. Pose-based sign language recognition using GCN and BERT. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 31–40.
33. Umar, S.S.I.; Iro, Z.S.; Zandam, A.Y.; Shitu, S.S. Accelerated Histogram of Oriented Gradients for Human Detection. Ph.D. Thesis, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 2016.