Contents lists available at *Dergipark*

Journal of Scientific Reports-A

journal homepage: *https://dergipark.org.tr/tr/pub/jsr-a*

E-ISSN: 2687-6167 ***Number 55, December 2023***

RESEARCH ARTICLE

Receive Date: 27.09.2023 *Accepted Date: 17.10.2023*

Deep learning-based isolated sign language recognition: a novel approach to tackling communication barriers for individuals with hearing impairments

Naciye Nur Arslan[1*], Emrullah Şahin[2], Muammer Akçay[3]

[1]*Kütahya Dumlupınar University Faculty of Engineering, Departement of Software Engineering, 43000 Kütahya,,*
ORCID: 0000-0002-3208-7986
[2] *Kütahya Dumlupınar University Faculty of Engineering, Departement of Software Engineering, 43000 Kütahya,,*
ORCID: 0000-0002-3390-6285
[3] *Kütahya Dumlupınar University Faculty of Engineering, Departement of Software Engineering, 43000 Kütahya,,*
ORCID: 0000-0003-0244-1275

Abstract

Sign language is a primary and widely used means of communication for individuals with hearing impairments. Current sign language recognition techniques need to be improved and need further development. In this research, we present a novel deep learning architecture for achieving significant advancements in sign language recognition by recognizing isolated signs. The study utilizes the Isolated Sign Language Recognition (ISLR) dataset from 21 hard-of-hearing participants. This dataset comprises 250 isolated signs and the x, y, and z coordinates of 543 hand gestures obtained using MediaPipe Holistic Solution. With approximately 100,000 videos, this dataset presents an essential opportunity for applying deep learning methods in sign language recognition. We present the comparative results of our experiments, where we explored different batch sizes, kernel sizes, frame sizes, and different convolutional layers. We achieve an accuracy rate of 83.32% on the test set.

Keywords: Sign language recognition; deep learning; landmarks; ISLR

1. Introduction

Sign Language Recognition (SLR) is an essential field of research that aims to develop technology to recognize sign language and translate it into written or spoken language [1]. Sign language (SL) is an essential tool for

* Corresponding author.
E-mail address: naciye.arslan@dpu.edu.tr.

communicating with individuals in the deaf community. In sign language, communication is achieved using visual language such as hand gestures, facial expressions, and body language. Sign language varies in structure and syntax between different countries and regions. SLR is necessary to increase communication between deaf individuals and the hearing world and to facilitate the inclusion of deaf individuals into society [2].

Recognizing sign language has various difficulties. In order to distinguish meanings in sign language, it is necessary to make a detailed analysis of body parts, especially hands, arms, and facial movements [3]. Sign language meanings may vary due to signs made by different people when creating a data set. The same sign may appear differently due to position, speed, and acceleration variability during sign movements. Environmental factors such as lighting and background can also make it difficult to recognize signs.

SLR research is divided into two main branches: isolated and continuous [4]. The system, known as isolated SLR, is designed to recognize a single sign performed without any other signs or movements. Isolated SLR is used in applications where single signs need to be recognized, for example, in sign language dictionaries or educational environments. On the other hand, continuous SLR means that the system recognizes sign language sentences or multiple signs performed sequentially. Continuous SLR is often used in applications such as sign language translation systems or real-time communications [5].

In recent years, deep learning has emerged as a powerful tool to overcome the challenges of sign language recognition. Deep learning models, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) and their variations, have demonstrated the state-of-the-art in isolated and continuous sign language recognition tasks [6].

Sign language recognition can be made using landmarks, defined as points corresponding to important joint points on the human body. Landmarks are extracted from video data using computer vision techniques and used to train machine learning models for sign language classification. Landmark-based SLR has gained popularity recently due to its simplicity and effectiveness. Using landmarks to represent sign language movements makes it possible to capture significant spatiotemporal information without requiring large amounts of training data or complex deep learning models.

This work presents a new CNN-based high-performance deep learning architecture for isolated sign language recognition. Our unique architecture, focused on American Sign Language, uses Convolutional Neural Network (CNN) to identify input signals from video-derived point data, achieving the best performance in the current benchmark dataset for ISLR. This work demonstrates the effectiveness of our new CNN-based deep learning architecture combined with landmark-based features for isolated sign language recognition. It provides a basis for further research in this exciting and rapidly evolving field.

2. Related Works

Sign language recognition has become a popular research topic in recent years. Sign language recognition studies for various regions have been carried out in the literature using CNN, LSTM models, and transformer structures.

Aly and colleagues [2] introduced a framework employing DeepLabv3+ and BiLSTM for recognizing Arabic sign language, yielding promising outcomes. Sincan et al. [3] established a foundation for assessing the performance of CNNs and LSTM models in Turkish Sign Language by comparing various architectural approaches. De Coster et al. devised a system that combines OpenPose, a real-time pose estimation framework with transformers, and multi-headed attention to recognize sign language in a Flemish Sign Language dataset, achieving state-of-the-art results [7]. Rastgoo et al. proposed a cascading model for recognizing Persian sign language, focusing on static and dynamic movements. They employed the Single Shot Multi-Box Detector (SSD) and LSTM [8]. Rastgoo et al. introduced a multimodal deep learning model utilizing RGB and depth videos for sign language recognition across three publicly available datasets, demonstrating improved accuracy [9]. Sharma et al. suggested a deep transfer learning approach for continuous sign language sentence recognition. They leveraged pre-trained models to enhance performance [8]. Aloysius et al. conducted a comprehensive review of the literature on vision-based continuous sign

language recognition, emphasizing recent advancements in deep learning techniques and evaluating their effectiveness [5].

Hu et al. introduced a hand model recognition framework that considers the distinctive characteristics of hand joints for recognizing isolated sign language gestures [11]. Zhou et al. proposed an innovative method to address motion blur issues in video sequences using (2+1) D and 3D ResNet, incorporating blur detection to improve sign language recognition [9]. Yang et al. introduced the KSL dataset, a valuable resource for researchers interested in Korean sign language recognition, thereby contributing to the field with a large-scale dataset [13].

In wearable-based systems, Zhang et al. harnessed a multimodal CNN and bidirectional LSTM to achieve end-to-end continuous American Sign Language (ASL) recognition. Their MyoSign system demonstrated high accuracy at both word and sentence levels, leveraging data from Myo armband sensors [14]. Saunders et al. introduced an innovative approach for sign language generation (SLP) using the Progressive Transducer architecture. They converted spoken language sentences into continuous 3D multi-channel sign pose sequences, setting new benchmarks in this area [15]. Several recent studies have also proposed new datasets for sign language recognition. Fink et al. presented two new sign language datasets, LSFB-CONT and LSFB-ISOL, for continuous and isolated sign language recognition in French Belgian Sign Language [16]. Das et al. proposed a hybrid model for Bangla Sign Language recognition and evaluated it on two new datasets, Ishara-Bochon and Ishara-Lipi, providing a valuable resource for the Bengali sign language research community [17]. Rajalakshmi et al. proposed a hybrid neural network architecture for recognizing Indian and Russian sign language and created a new multi-signer dataset for evaluation, promoting research in these underrepresented sign languages [18].

Finally, some recent studies have proposed novel techniques for sign language recognition based on shape trajectories analysis, attention mechanisms, and batch normalization. Fakhfakh et al. developed a shape trajectories analysis approach based on deep learning for recognizing isolated word sign language, demonstrating the effectiveness of shape trajectory analysis in recognizing dynamic gestures [19]. Fang et al. proposed an adversarial multi-task deep learning framework that incorporated multiple modalities and solved the problem of signer independent SLR, improving the generalizability of the recognition system [20]. Luqman et al. proposed an efficient two-stream network for isolated sign language recognition using accumulative video motion, capturing both static and dynamic gesture features for improved recognition [21]. Sarhan and Frintrop proposed a novel method using spatial attention for isolated SLR, enabling the model to focus on relevant regions of the input images for better recognition performance [22]. Takayama et al. proposed a masked batch normalization technique to improve tracking-based sign language recognition using graph convolutional networks, addressing the issue of variable-length sequences in sign language recognition [23]. Wang et al. proposed a framework for multimodal sign language recognition under small sample conditions based on key-frame sampling. This approach combined spatial and temporal features for effective recognition in scenarios with limited training data, providing a viable solution for under-resourced sign languages [24]. Boukdir et al. proposed a deep learning approach for isolated video-based recognition of Arabic sign language, specifically Moroccan sign language, using 2D convolutional recurring neural networks (2DCRNN) and 3D convolutional neural networks (3DCNN) to classify video sequences with high accuracy [25]. Pariwat et al. developed a multi-stroke Thai finger-spelling sign language recognition system using deep learning with vision-based techniques, achieving impressive average accuracies for one, two, and three-stroke alphabets [26]. Furthermore, Rajalakshmi and colleagues proposed a hybrid deep neural network for recognizing Indian and Russian sign gestures, utilizing a combination of spatial feature extraction, attention-based Bi-LSTM, and a modified autoencoder for feature extraction, demonstrating better performance than existing frameworks [27]. These advancements in sign language recognition methods showcase the continuous improvement and innovation in the field, with various techniques and approaches being explored for effective and accurate sign language recognition systems.

In summary, recent research in sign language recognition has covered a wide range of techniques and approaches, including deep learning models, novel architectures, attention mechanisms, and the development of new datasets. These studies demonstrate the ongoing progress in the field and the potential for further advancements in

sign language recognition technology. In our study, we performed sign language recognition on a large newly published landmark-based data set. We conducted a study that will contribute to the literature with our CNN model and our accuracy rate on the new data set.

3. Material and Methods

3.1. Dataset

This study used Isolated Sign Language Recognition (ISLR) Corpus (version 1.0) dataset. The ISLR dataset is published by the Georgia Institute of Technology, the dataset consists of approximately 100,000 isolated signs performed by 21 deaf users using American Sign Language (ASL). Users performed cues from a 250-word vocabulary with hand, face, and pose landmarks obtained via MediaPipe (version 0.9.0.1). The ISLR dataset was primarily designed to be used in educational games aimed at helping deaf children, and their families learn ASL [28].

Videos represent the beacons in the dataset, each taken from a different user and shot in various environments. These videos were collected via smartphone apps without any review for accuracy or quality. Each video contains the coordinates of key points identified in the hands, faces, and poses of the users during the execution of the signs. This landmark data is input into machine-learning models to analyze and recognize signs. Landmark data is represented as x, y, and z coordinates.

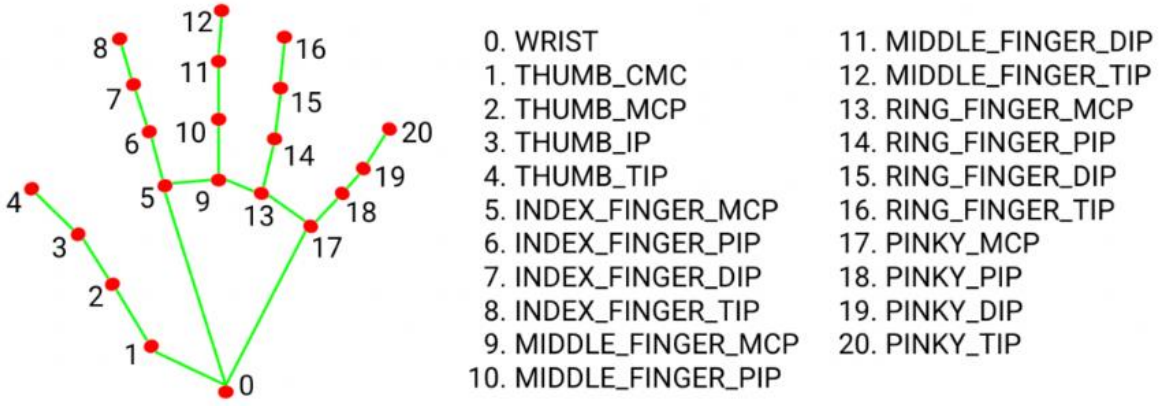

Fig. 1. MediaPipe landmarks for hand [29].

Each frame in the videos contains a total of 543 landmarks. These landmarks are distributed as follows: face (468), left hand (21), pose (33), and right hand (21). Example landmarks of the hand are presented in Fig. 1. Videos in the ISLR dataset vary in the number of frames that describe each sign. The maximum number of frames is about 330, the minimum is 4, and the average is 38. This study aims to contribute to the development of more effective and user-friendly educational tools for deaf individuals by taking advantage of the ISLR dataset.

3.2. Preprocessing of the ISLR Dataset

In the ISLR dataset, the number of frames per video varies, necessitating a preprocessing step to ensure compatibility with the deep learning network used in this study. To facilitate training, a reference frame count of 6 was selected. In this preprocessing stage, videos with more than 6 frames were down sampled by averaging

consecutive frames to reduce the total number of frames to the reference value. Conversely, for videos with fewer than 6 frames, zero-padding was employed by adding frames consisting of zeros until the target frame count was reached. This approach standardized all videos to have 6 frames, thereby enabling efficient training of the deep learning network on the ISLR dataset.

Additionally, from the original 543 landmark points in each frame, 118 dominant landmarks were identified based on their importance in sign language communication. These dominant landmarks predominantly include key points from hands, lips, and pose, which are known to carry significant information in sign language. By focusing on these 118 landmarks, the computational complexity of the deep learning model is reduced while maintaining the essential features required for accurate sign recognition. This refined set of landmarks further enhances the efficiency and effectiveness of the deep learning network's training on the ISLR dataset.

In the final stage, the processed data for each sign consists of 6 frames, each containing 118 landmarks and their corresponding 3-dimensional (x, y, z) coordinates. This standardized data representation ensures consistent input for the deep learning model, enabling accurate and efficient sign recognition in the ISLR dataset.

3.3. Proposed Method

In this study, a novel architecture based on CNN is proposed for recognizing isolated sign language (see Fig. 2). The architecture consists of four total blocks, each containing a Conv1D layer, a Batch Normalization layer, and a ReLU activation function.

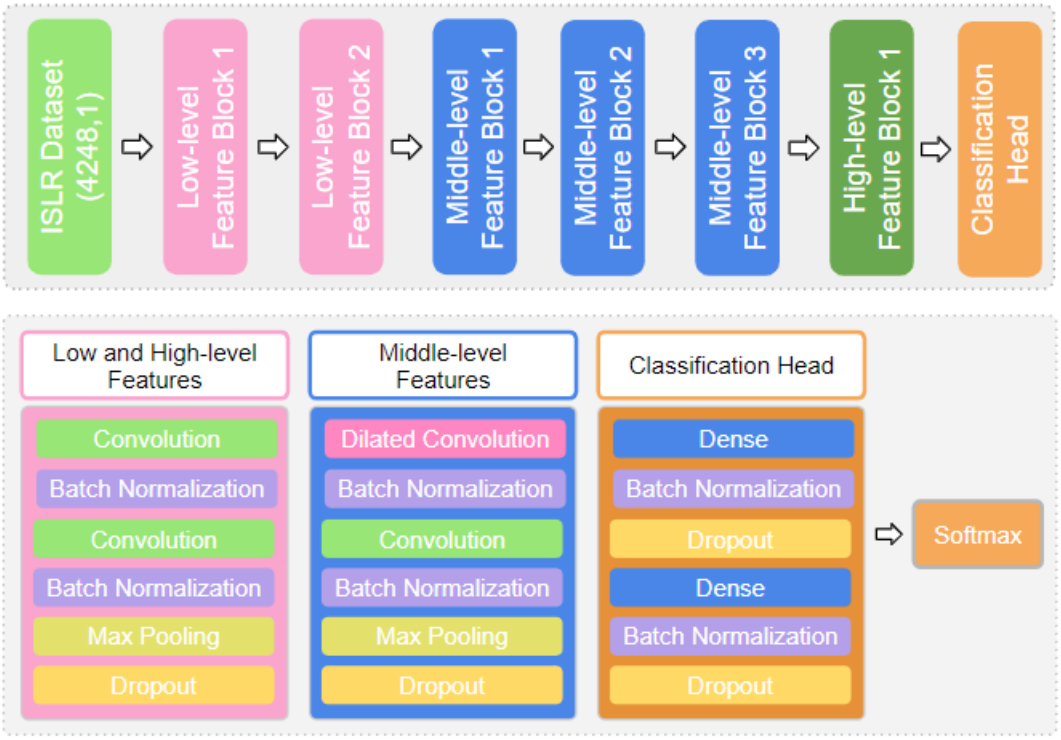

Fig. 2. Proposed model.

In our proposed model, dilation convolution is used in certain blocks. Dilation convolution is a technique that modifies traditional convolutional operations. In regular convolution, the kernel computes with a fixed stride size over the data. In this case, the kernel size is the same as the stride size. However, gaps are inserted between steps on the kernel in dilation convolution. This gap allows the convolution operation to have a wider receptive field. The dilation factor determines the size of this gap. Specifically, dilated convolutions are used in the third and fourth blocks to create an expanded receptive field. This allows the model to capture longer-range dependencies and larger-scale features [30]. Our input data consists of 94477 samples, each comprising 6-frames. Each frame contains 3-dimensional location data (x, y, z) for 118 individual landmarks and their derivative information. Therefore, the input data size is 6 x 118 x 6 = 4248. The total number of parameters in our deep learning model is 10,806,746.

The model blocks are described below:

Low-Level Feature Block 1: At this level, two Conv1D layers with 32 filter sizes and the following batch normalization layers were used. Batch Normalization stabilizes the training process by normalizing the data. Then, the MaxPooling layer with 3 filter sizes and the dropout layer, which deactivates neurons by 30%, were applied. Dropout helps reduce overfitting and helps the model learn more generalized features.

Low-Level Feature Block 2: Similarly, two Conv1D layer with 64 filter size is added, followed by the Batch Normalization layer. The MaxPooling layer with size 3 is applied. The Dropout layer deactivates 40% of the neurons randomly.

Middle-Level Feature Block 1,2, and 3: These blocks use dilated convolutions to expand the receptive field. The dilation convolution is performed by leaving gaps between steps on the kernel, with a dilation_rate=2. Each block includes two Conv1D and two Batch Normalization layers. The filters applied to the conv1d layers are 128,256 and 384, respectively. The MaxPooling1D layer implements a pooling operation of size. The Dropout layer deactivates 50% of the neurons randomly.

High-Level Feature Block 1: The layers in this section consist of two Conv1D layers with 512 filters, two Batch Normalization layers, a MaxPooling1D layer, and a Dropout layer. The MaxPooling1D layer implements a pooling operation of size 3. The Dropout layer deactivates 50% of the neurons randomly.

Classification Head: The Flatten layer flattens the 1D outputs. Then, a two-layer, fully connected structure is added. Each layer consists of 1024 and 512 neurons, respectively. Batch Normalization and Dropout layers are added after each fully connected layer. Finally, a Dense layer with the number of output classes (250) is added. The SoftMax activation function is used in this layer. The model performs convolution and pooling operations with consecutive blocks and then performs classification with fully connected layers.

The ReLU (Rectified Linear Unit) activation function was used in all model layers except the last layer. ReLU is a simple activation function where negative values equal zero and positive values remain unchanged. This function helps the network learn nonlinear features. The SoftMax activation function was used in the last layer, Dense (250). SoftMax is an activation function used in multiclass classification problems. It obtains a probability distribution representing the class probabilities at the output. This allows the model to make the most probable prediction across multiple classes.

3.4. Hyper-parameter Settings

AdamW optimizer, a derivative of the Adam optimizer with weight decay support, was applied. Weight decay controls the magnitude of weight parameters while updating the network's loss to optimize. This can encourage weight parameters for generalization and prevent overfitting. Essentially, it incorporates weight decay into the update steps of the Adam algorithm by adding a term. The term for weight decay is typically multiplied by a factor of the sum of squares of weight parameters and added to the total loss function [31].

Below is the description of the parameters used in the optimizer initialization:

- learning rate: This parameter determines the step size used in each optimization step. It controls how quickly the optimizer adjusts the model weights based on the computed gradients. A learning rate of 1e-3 (0.001) is used in this case.
- weight decay: This parameter determines the strength of weight decay regularization. It is a multiplying coefficient for the weight decay term added to the loss function. A weight decay of 1e-5 (0.00001) is used in this case.
- clip norm: This parameter is used to limit the gradients during optimization. It prevents them from growing too large by capping the norm (magnitude) of the gradients. In this case, a clip-norm value of 1.0 is used.

The Sparse Categorical Cross Entropy Loss function was combined with label smoothing to address the problem [32]. Label smoothing aims to correct harsh label classification errors, allowing the model to generalize better and reduce the risk of overfitting. Sparse Categorical Cross Entropy Loss is a commonly used loss function in multi-class classification problems. However, it can encounter issues when working with complex labels. For example, in some cases, the model may make harsh classification errors in overly confident classes, which can negatively impact the training process. Label smoothing is a technique used to reduce such harsh classification errors by smoothing the labels. Instead of using a label value for each class, this technique represents the actual labels using one-hot encoding, and low but unequal probabilities are assigned to all classes. This allows the model to learn a more balanced class distribution and make more generalized classifications. This loss function calculates the cross-entropy between the predicted probabilities and the true labels, measuring how well the predicted classes match the true labels. The cross-entropy is then averaged across all the samples to give the total loss. The goal during model training is to minimize the loss value. The equation for sparse categorical cross entropy loss is:

$$Loss = \frac{-1}{N} \sum_{i}^{N} \sum_{j}^{M} y_{ij} \log y'_{ij} \quad (1)$$

Here, N represents the total number of samples, M represents the number of classes, y_{ij} represents the jth class of the ith sample's true label, and y'_{ij} represents the predicted probability of the jth class for the ith sample. The negative sign indicates that the goal is to minimize the loss. During the optimization process, the model's parameters are updated iteratively to minimize the loss function.

A learning rate scheduling strategy called Linear Warmup and Cosine Decay was applied. This strategy gradually increases the learning rate at the beginning of model training and slowly decreases it from a specific turning point. This practice helped the model achieve a better balance and achieve better results. The model was trained for 300 epochs.

4. Experimental Results

The proposed CNN architecture was evaluated for recognizing isolated sign language using landmark data. Table 1 presents the evaluation metrics for the proposed model, including Sparse Categorical Accuracy, Sparse Top-5 Accuracy, and Sparse Top-10 Accuracy. Sparse Categorical Accuracy measures the percentage of correct predictions for a given sample set. It is called "sparse" because the predicted classes and actual tags are encoded as integers. To calculate Sparse Categorical Accuracy, the number of correct predictions is divided by the total number of samples.

Table 1. Proposed models benchmark results.

Model Properties	Model Type	Loss	Sparse Accuracy	Top 5	Top 10
	Batch Size 64	2.4967	0.8290	0.9350	0.9516
	Batch Size 96	2.4942	0.8310	0.9325	0.9516

Half Dilated Conv Frame Size 6 Kernel Size 7	Batch Size 128	2.5132	0.8220	0.9312	0.9485
	Batch Size 196	2.5368	0.8118	0.9272	0.9462
	Batch Size 256	2.5427	0.8098	0.9263	0.9453
	Batch Size 384	2.5699	0.8022	0.9218	0.9427
	Batch Size 512	2.5762	0.7986	0.9231	0.9440
Half Dilated Conv Frame Size 6 Batch Size 96	Kernel Size 5	2.5109	0.8249	0.9332	0.9508
	Kernel Size 7	2.4942	0.8310	0.9325	0.9516
	Kernel Size 9	2.5085	**0.8332**	0.9320	0.9476
	Kernel Size 11	2.5156	0.8262	0.9295	0.9454
Kernel Size 9 Frame Size 6 Batch Size 96	Normal Conv	2.4963	0.8289	0.9317	0.9472
	Half Dilated Conv	2.5085	**0.8332**	0.9320	0.9476
	Full Dilated Conv	2.5249	0.8229	0.9292	0.9449
Half Dilated Conv Kernel Size 9 Batch Size 96	Frame Size 4	2.6125	0.8021	0.9185	0.9389
	Frame Size 6	2.5085	**0.8332**	0.9320	0.9476
	Frame Size 8	2.532	0.8198	0.9245	0.9428

Sparse Top-5 Accuracy represents a versatile evaluation metric compared to Sparse Categorical Accuracy. It offers the flexibility to deem a prediction correct if the true label falls within the top five predicted classes. This metric accommodates scenarios where the actual label might not be the most apparent or direct choice. To compute Sparse Top-5 Accuracy, we tally the count of correct predictions (where the accurately guessed class resides among the top five) and divide it by the total number of samples. Similarly, Sparse Top-10 Accuracy gauges the percentage of instances where the accurately guessed class ranks within the top ten predicted classes. This metric enhances the model's predictive adaptability further. To calculate Sparse Top-10 Accuracy, we assess the number of correct predictions (where the accurately guessed class is among the top ten) and divide it by the overall sample count.

The model was trained using different batch sizes, kernel sizes, and convolutions, as shown in the table. Based on the test results, the highest accuracy was achieved using a batch size of 96, a kernel size of nine, and dilated convolutions in the middle layers.

In the first stage, we carried out training in different batch sizes (64,96,128,196,256,384,512) with the addition of semi-dilated conv, that is, dilatation factor, to the intermediate layers, with Kernel size seven and frame size six determined. As a result of this training, we found the highest sparse accuracy rate in test data at 83.10% in 96 batch sizes. While the batch size was 96, we also tried different kernel sizes (7,9,11), and we reached 83.32% accuracy with nine kernel sizes. We observed that the accuracy rate did not increase when tested by applying normal conv to the layers. We applied and tested dilated conv (full dilated conv) on all layers, and as a result, we found that the

accuracy rate was lower than half dilated conv. We trained by changing the number of frames in the dataset to four and eight and achieved maximum accuracy when the frame size was six.

As a result of these comparisons, our model with the best sparse categorical accuracy rate (83.32%) was obtained by applying kernel size 9, batch size 96, frame size six, and half dilation convolution. Additionally, we found the top 5 accuracies at 93.20% and the top 10 accuracies at 94.76%.

5. Conclusions

In conclusion, this study introduced a CNN architecture for recognizing isolated sign language using landmark data. The model was evaluated on a dataset consisting of 94,477 video clips, each containing 118 landmarks with x, y, and z coordinate information. The results demonstrated good accuracy on the validation set, with a sparse categorical accuracy of 83.32%. The top-5 accuracy achieved 93.20% on the validation set, while the top-10 accuracy reached 94.76%.

As the dataset used in this study is newly shared, no existing studies are found in the literature that specifically applied this dataset. Therefore, there is a potential to enhance the model's performance by exploring different preprocessing techniques tailored to this dataset. Additionally, future work could consider selecting different hand, face, and pose landmarks and experimenting with varying the number of frames in the video clips to improve accuracy rates. It could be tested on different sign language datasets to evaluate the proposed model further. Since the dataset used in this study involves many classes, the observed increase in accuracy rate needed to be more substantial. This may be attributed to the need for a well-established standard due to the novelty of the dataset.

This study provides valuable insights into the challenges and potential solutions for recognizing sign language using landmark data. The proposed model and findings can serve as a reference for future research in this field, contributing to the advancement of sign language recognition techniques.

Acknowledgements

This study did not receive any specific funding or financial assistance from governmental, commercial, or non-profit organizations.

References

[1] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian and B. B. Chaudhuri, "A modified LSTM model for continuous sign language recognition using leap motion" *IEEE Sensors Journal*, vol. 19, no. 16, pp. 7056-7063, Apr. 2019.

[2] S. Aly and W. Aly, "DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition", *IEEE Access*, vol. 8, pp. 83199-83212, Apr. 2020.

[3] O. M. Sincan and H. Y. Keles, "Autsl: A large scale multi-modal Turkish sign language dataset and baseline methods", *IEEE Access*, vol. 8, pp. 181340-181355, Aug. 2020.

[4] R. Rastgoo, K. Kiani and S. Escalera, "Sign language recognition: A deep survey" *Expert Systems with Applications*, vol. 164, pp. 113794, Feb. 2021.

[5] N. Aloysius, & M. Geetha, "Understanding vision-based continuous sign language recognition" *Multimedia Tools and Applications*, vol. 79, no. (31-32), pp. 22177-22209, May 2020.

[6] A. Wadhawan and P. Kumar, "Sign language recognition systems: A decade systematic literature review" *Archives of Computational Methods in Engineering*, vol. 28, pp. 785-813, Dec. 2021.

[7] M. De Coster, M. Van Herreweghe and J. Dambre, "Sign language recognition with transformer networks" in *12th international conference on language resources and evaluation*, May 2020, pp. 6018-6024.

[8] R. Rastgoo, K. Kiani and S. Escalera, "Video-based isolated hand sign language recognition using a deep cascaded model" *Multimedia Tools and Applications*, vol. 79, pp. 22965-22987, Jun. 2020.

[9] R. Rastgoo, K. Kiani and S. Escalera, "Hand pose aware multimodal isolated sign language recognition" *Multimedia Tools and Applications*, vol. 80, pp. 127-163, Sep. 2021.

[10] S. Sharma, R. Gupta and A. Kumar "Continuous sign language recognition using isolated signs data and deep transfer learning" *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 1-12, Aug. 2021.

[11] H. Hu, W. Zhou and H. Li, "Hand-model-aware sign language recognition" in *Proc. AAAI conference on artificial intelligence*, May 2021, vol. 35, no. 2, pp. 1558-1566.

[12] Z. Zhou, K. S. Kui, V. W. Tam and E. Y. Lam, "Applying (3+ 2+ 1) D residual neural network with frame selection for Hong Kong sign language recognition*"* in *2020 25th International Conference on Pattern Recognition (ICPR)*, Jan. 2021, pp. 4296-4302.
[13] S. Yang, S. Jung, H. Kang and C. Kim, "The Korean sign language dataset for action recognition" in *International conference on multimedia modelling*, Dec. 2019, pp. 532-542.
[14] Q. Zhang, D. Wang, R. Zhao, & Y. Yu, "MyoSign: enabling end-to-end sign language recognition with wearables" in *Proc. of the 24th international conference on intelligent user interfaces*, Mar. 2019, pp. 650-660.
[15] B. Saunders, N. C. Camgoz and R. Bowden, "Continuous 3d multi-channel sign language production via progressive transformers and mixture density networks" *International journal of computer vision*, vol. 129, no. 7, pp. 2113-2135, Mar. 2021.
[16] J. Fink, B. Frénay, L. Meurant and A. Cleve, "LSFB-CONT and LSFB-ISOL: Two new datasets for vision-based sign language recognition" in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1-8.
[17] S. Das, M. S. Imtiaz, N. H. Neom, N. Siddique, and H. Wang, "A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier*" Expert Systems with Applications*, vol. 213, pp. 118914, Mar. 2023.
[18] E. Rajalakshmi, R. Elakkiya, A. L. Prikhodko, M. G. Grif, M. A. Bakaev, J. R. Saini, ... and V. Subramaniyaswamy, "Static and dynamic isolated Indian and Russian sign language recognition with spatial and temporal feature detection using hybrid neural network" *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no.1, pp. 1-23, Nov. 2022.
[19] S. Fakhfakh and Y. B. Jemaa, "Deep Learning Shape Trajectories for Isolated Word Sign Language Recognition" *Int. Arab J. Inf. Technol*., vol. 19, no. 4, pp. 660-666, Jul. 2022.
[20] Y. Fang, Z. Xiao, S. Cai and Ni L., "Adversarial multi-task deep learning for signer-independent feature representation" *Applied Intelligence*, vol. 53, no. 4, pp. 4380-4392, Jun. 2023.
[21] H. Luqman, "An Efficient Two-Stream Network for Isolated Sign Language Recognition Using Accumulative Video Motion" *IEEE Access*, vol. 10, pp. 93785-93798, Sep. 2022.
[22] N. Sarhan and S. Frintrop, "Sign, Attend and Tell: Spatial Attention for Sign Language Recognition" in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, Dec. 2021, pp. 1-8.
[23] N. Takayama, G. Benitez-Garcia and H. Takahashi "Masked batch normalization to improve tracking-based sign language recognition using graph convolutional networks" in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, Dec. 2021, pp. 1-5.
[24] J. Wang, J. Chen and Y. Cai, "A framework for multimodal sign language recognition under small sample based on key-frame sampling" *In Fifth International Workshop on Pattern Recognition*, vol. 11526, pp. 46-52, Jun. 2020.
[25] A. Boukdir, M. Benaddy, A. Ellahyani, O. E. Meslouhi and M. Kardouchi, "Isolated video-based Arabic sign language recognition using convolutional and recursive neural networks*" Arabian Journal for Science and Engineering*, pp. 1-13, Sep. 2021.
[26] T. Pariwat and P. Seresangtakul , "Multi-stroke thai finger-spelling sign language recognition system with deep learning" *Symmetry*, vol. 13, no.2, pp. 262, Feb. 2021.
[27] E. Rajalakshmi, R. Elakkiya, V. Subramaniyaswamy, L. P. Alexey, G. Mikhail, M. Bakaev, ... and A. Abraham, "Multi-Semantic Discriminative Feature Learning for Sign Gesture Recognition Using Hybrid Deep Neural Architecture*" IEEE Access*, vol. 11, pp. 2226-2238, Jan. 2023.
[28] Deaf Professional Arts Network and the Georgia Institute of Technology, Kaggle ASL dataset, https://www.kaggle.com/competitions/asl-signs/overview (accessed June 12, 2023).
[29] MediaPipe Solutions, Mediapipe hand landmarks, (n.d.). https://developers.google.com/mediapipe/solutions/vision/hand_landmarker (accessed June 12, 2023).
[30] N. N. Arslan, D. Ozdemir and H. Temurtas, "ECG heartbeats classification with dilated convolutional autoencoder" *Signal, Image and Video Processing*, pp. 1-10, Sep. 2023.
[31] R. Llugsi, S. El Yacoubi, A. Fontaine and P. Lupera, "Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito*"* in *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, Oct. 2021, pp. 1-6.
[32] T. Andrei-Alexandru and D. E. Henrietta, "Low-cost defect detection using a deep convolutional neural network" in *2020 IEEE International conference on automation, quality and testing, robotics (AQTR)*, May 2020, pp. 1-5.