A Comparison of the Effectiveness Architectures LSTM1024 and 2DCNN for Continuous Sign Language Recognition Process

Nurzada Amangeldy[1], Iurii Krak[2,3] , Bekbolat Kurmetbek[1] and Nazerke Gazizova[1]

[1] *L.N. Gumilyov Eurasian National University, 2, Satpayev str., Astana, 010000, Republic of Kazakhstan*
[2] *Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska str. Kyiv, 01601, Ukraine*
[3] *Glushkov Institute of Cybernetics, 40 Glushkov ave., Kyiv, 03187, Ukraine*

Abstract

This study advances the field of real-time sign language recognition by assessing the efficacy of deep learning architectures, specifically those based on recurrent neural networks (RNNs) and convolutional neural networks (CNNs), for the intricate task of continuous sign language interpretation. Our research highlights the RNN-based model's superior capability in processing temporal gesture sequences, indicating its suitability for predicting dynamic sign language with high accuracy. In contrast, the CNN-based model demonstrates remarkable proficiency in identifying spatial patterns within gestures, proving its utility in visually complex situations. Employing advanced data processing techniques, such as real-time pose estimation and sophisticated array manipulation for gesture classification, this work signifies a leap forward in developing AI-powered assistive technologies. The study leverages a diverse dataset to ensure the models' robust performance, underscoring deep learning's role in promoting communication inclusiveness. This scientific exploration into the capacities of deep learning methodologies underscores our commitment to leveraging cutting-edge technology for fostering more inclusive communication solutions. This research is focusing on creating automated interpretation systems for Kazakh sign language, aiming at broader inclusivity and technological innovation in communication aids.

Keywords

Deep learning; sign language recognition; inclusive communication; human pose detection

1. Introduction

In recent years, interest in sign language recognition systems has grown significantly due to their potential to reduce communication barriers for people with hearing and speech impairments. Continuous sign language recognition is a complex task due to the high degree of variability in movements and the need for precise identification and analysis of multiple parameters, including hand, body, lip movements, and facial expressions.

In this study, we compare two architectures we previously proposed, based on Long Short-Term Memory LSTM1024 [1] and 2DCNN [2], designed to enhance the Continuous Sign Language Recognition (CSLR) process. Our approach begins with the initiation of the video capture process, during which either a specially installed camera or a standard video file can be utilized.

Using the MediaPipe Holistic model, our system detects and tracks key hand points in each video frame, creating a sequence necessary for further analysis. These sequences are then processed using either the LSTM1024 or 2DCNN models to predict specific gestures based on hand movements and overall body posture.

One of the key aspects of our approach is its ability to dynamically respond to input data: - if the probability of the predicted gesture exceeds the threshold we have established, that gesture is automatically integrated into the sentence being formed.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

nurzadaamangeldy@gmail.com (N. Amangeldy); yuri.krak@gmail.com (Iu. Krak); bigboss990930@gmail.com (B. Kurmetbek); g.nazerke2755@gmail.com (N. Gazizova)
0000-0002-4669-9254 (N. Amangeldy); 0000-0002-8043-0785 (Iu. Krak); 0009-0001-7510-2445 (B. Kurmetbek); 0009-0007-7278-4202 (N. Gazizova)

All of this occurs in real-time, with feedback for the user via a demonstration window that displays the current sentence, the most recently recognized gesture, and the corresponding probability of its accuracy of recognition.

Through the comparative tests conducted, our system demonstrates significant progress in the field of CSLR, promising more efficient and inclusive communication for individuals using sign language. This research underscores not only a technological breakthrough but also the social importance of enhancing communication means for individuals with hearing impairments.

2. Relevant works review

Deep neural networks (DNNs) have been effectively used for sign language recognition, facilitating communication between individuals who use sign language and those who may not understand it. Sign language recognition involves translating sign language gestures or signs into text or spoken language.

CNNs, widely utilized for image-based sign language recognition, are proficient in extracting spatial information from gesture images or video frames, with architectures like VGG, ResNet, and Inception being particularly effective [1,3,4]. A novel approach is demonstrated in a study [5] that employs a sequential CSLR mechanism specifically for Japanese sign language. This method involves a two-step process: firstly, tense segments are identified automatically via a Random Forest algorithm, and secondly, a CNN performs word categorization within these segments. The system analyzes detailed 2D angular trajectories from the presenter's body, hands, and face, using data captured through the OpenPose framework. Tested under various conditions and with different word class labels, this mechanism has achieved notable accuracy rates, reaching 0.92 for segmentation and 0.91 for classification.

Visual Geometry Group (VGG) is known for its simplicity and effectiveness. VGG networks typically feature a uniform architecture with multiple convolutional layers followed by max-pooling layers. These networks have been adapted for sign language recognition by modifying the output layer to accommodate the specific number of sign language classes [6]. ResNet (Residual Networks) is notable for its deep network architecture with skip connections or residual blocks. These skip connections help combat the vanishing gradient problem, allowing the training of very deep networks. Sign language recognition models based on ResNet often leverage this architecture's ability to capture fine-grained spatial information in sign gestures [7].

One innovative approach is the SignBERT system [8], which tackles these challenges by utilizing high-quality video clips and intelligently selecting key frames. It employs a (3+2+1)D ResNet for visual feature extraction and a pre-trained BERT model for language modeling, incorporating partially obscured videos from isolated sign language datasets. The advanced SignBERT version includes hand images as additional input and uses an iterative learning strategy to fully leverage the system's potential on available datasets. The results demonstrate a significant enhancement in continuous sign language recognition (CSLR).

Inception (GoogLeNet): The Inception architecture, also known as GoogLeNet, introduced the inception modules that use multiple filter sizes. This architecture is particularly suitable for sign language recognition as it enables the model to capture spatial information at various scales. By adapting Inception, features from sign gesture images with different spatial resolutions can be efficiently extracted [9].

RNNs, including Long Short-Term Memory (LSTM)s [2] and Gated recurrent units (GRU)s [10], are pivotal for sign language recognition in videos, capturing temporal dynamics inherent in gestures. The Bi-ST-LSTM-A system [11] enhances this by eliminating sequence segmentation, utilizing dual-stream CNNs for comprehensive motion analysis, and merging data using ST-LSTM. Attention-focused Bi-LSTMs and bidirectional networks improve accuracy by emphasizing key frames and considering contextual information. These advancements underscore the importance of sophisticated temporal-spatial analysis and real-time data processing in translating sign language video sequences into text. Hybrid Models: the combination of CNNs and RNNs is prevalent in sign language recognition [12]. CNNs are employed for feature extraction from video frames, while RNNs are used to model the temporal dependencies between sequential frames. An alternative approach [13] presents a sequential CSLR system.

Another strategy [14] employs a cross-modal learning approach that utilizes textual information to discern sign language states and their tense boundaries from minimally annotated video sequences. This method uses a multimodal transformer to simulate intra-class dependencies, improving the network's recognition accuracy. This technique has proven to surpass contemporary methods on various CSLR datasets, including RWTH-Phoenix-Weather-2014 [15], RWTH-Phoenix-Weather-2014T [16], and CSL [17].

3D Convolutional Neural Networks (3D CNNs): These networks, explicitly designed to capture both spatial and temporal information simultaneously, are particularly advantageous for video-based sign language recognition. 3D CNNs take into account the progression of sign gestures over time, rendering them highly suitable for this application [18]. Another innovative approach is the sign language recognition generative adversarial network (SLRGAN) [19], which utilizes a generative adversarial network framework. This system comprises a generator, which discerns sign language annotations by extracting spatial and temporal cues from video sequences, and a discriminator, which evaluates the quality of the generator's outputs by analyzing textual information related to sentences and glosses.

In conclusion, Deep Neural Networks (DNNs) have significantly propelled the field of sign language recognition forward, helping to bridge the communication divide between sign language users and those who do not understand it. This technological advancement plays a critical role in promoting inclusivity and accessibility in communication. The adaptation and implementation of various DNN architectures highlight the complexity and diversity inherent in interpreting sign language, underscoring that it is not merely a series of static gestures but a dynamic language rich in spatial and temporal components.

After a comprehensive review of various methodologies and deep learning architectures employed in sign language recognition, it becomes evident that the dynamic and complex nature of sign language poses unique challenges that demand innovative solutions. The exploration of different neural network models, including CNNs, LSTMs, and hybrid models, underscores the diversity of approaches in addressing spatial and temporal aspects of sign language. These models, each with their strengths in processing spatial information (like images) and temporal data (like sequences), have shown considerable promise in improving the accuracy and reliability of sign language recognition systems.

Based on this fundamental understanding, the problem statement un this work is the use and detailed comparative analysis of two neural networks - LSTM and CNN with known architectures for gesture recognition from a continuous video stream. The selection of these specific networks is driven by their proven effectiveness in our previous works for processing sequential data and image data, respectively, both of which are integral components of sign language communication.

3. Methodology

3.1. Data acquisition

Our approach commences with the real-time collection of video data via an integrated or external camera connected to the system. We employ high-resolution streaming video to enhance detection accuracy, targeting a frame rate of 60 frames per second for gestural words and 30 frames for the finger-spelling alphabet to ensure smoothness and minimize latency.

3.2. Video processing and pose detection using MediaPipe

Each frame from the real-time video feed undergoes processing using MediaPipe, a cutting-edge computer vision library, for human pose detection. MediaPipe offers pre-trained machine learning models capable of identifying various body landmarks including hands, facial features, and overall body posture (Figure 1).

We leverage this technology not only to pinpoint and visualize these key points on every processed video frame but also to capture comprehensive motion data with the aim of generating dynamic gestures in the future.

This extensive data acquisition is foundational for our system's ability to understand and interpret a more nuanced and natural form of sign language, which includes a combination of static poses and fluid, continuous motions.

3.3. Key points extraction

After the key points are identified on the frames, we extract the coordinates of these points, especially those related to hands and gestures (e.g., wrist position, finger placement, etc.). These data are structured in a format suitable for input into a machine learning model and used as features for gesture recognition.

The extract_keypoints function in the script processes data from MediaPipe's pose detection, methodically extracting and structuring key points from various human body components into a unified NumPy array. It selectively retrieves *x*, *y*, *z*, and visibility values for general body pose landmarks, and, if specified, facial landmarks, alongside the coordinates for landmarks on both hands. These values are then flattened and concatenated. In scenarios where certain landmarks are undetected, the function intelligently fills in these gaps with zeros to maintain data consistency, crucial for subsequent machine learning operations. This comprehensive approach ensures a rich, uniform dataset, poised for advanced analyses and interpretations, particularly in gesture recognition applications.

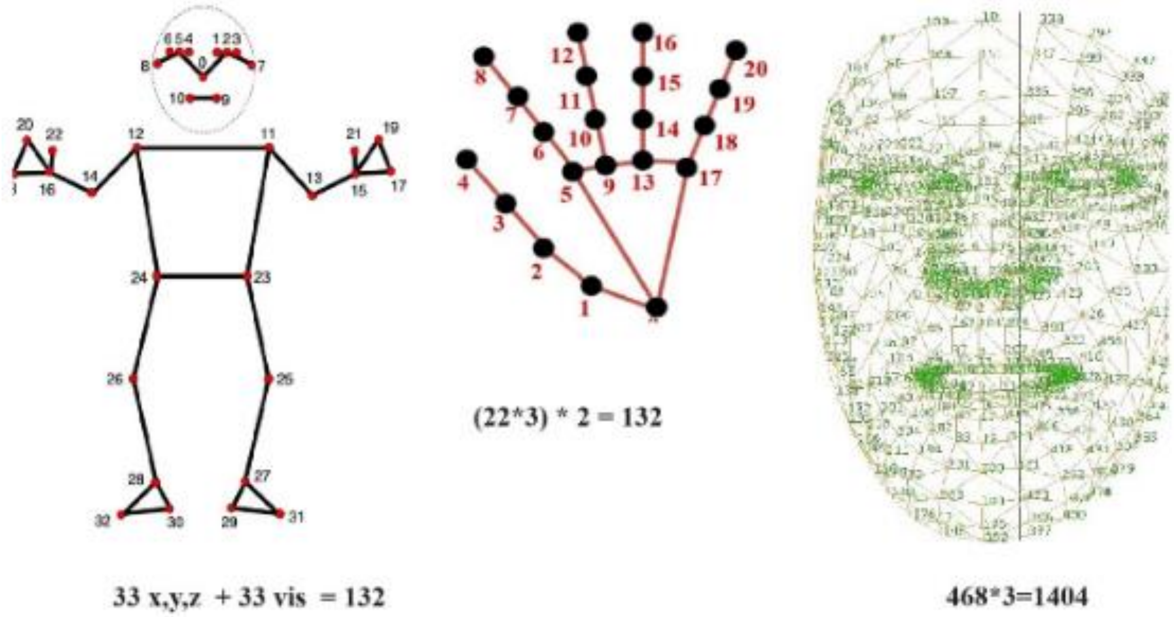

Figure 1: Various body landmarks

3.4. Gesture language recognition

In the enhanced explanation [1], the Convolutional Neural Networks (CNNs) designed for dynamic gesture recognition intricately balance the convolutional layers, kernel sizes, and pooling strategies to adeptly extract relevant features. This balance is pivotal in preventing the model from overfitting, a common challenge when the model complexity exceeds the necessity for accurate feature representation. The architecture incorporates layers specifically arranged to process and classify sign language gestures, employing a sequential approach that includes convolutional layers for feature extraction and dense layers for classification. The introduction of dropout techniques within the densely connected layers further mitigates overfitting by randomly ignoring a subset of neurons during training, enhancing generalization to unseen data.

The model's performance is critically analyzed using the categorical cross-entropy loss function, a standard in multiclass classification tasks, which quantifies the difference between the predicted probabilities and the actual distribution of the classes. This metric is instrumental in fine-tuning the model's parameters to minimize prediction error. The training process, supported by a substantial dataset comprising 30,960,000 data points across 40 dynamic gestures, demonstrates the model's ability to learn and generalize from the complex, high-dimensional input space. The dataset includes a comprehensive range of gestures, each represented by multiple frames and parameters, offering a rich basis for the model to learn the nuanced patterns of sign language.

Upon evaluation, the model exhibits remarkable accuracy in both validation and testing phases, confirming its robustness and the effectiveness of the chosen architecture and training regimen. This precision underscores the potential of 2DCNNs in bridging communication gaps for the deaf and hard-of-hearing community, providing a powerful tool for real-time sign language interpretation.

To delve deeper into the methodology [2], the model's LSTM layer, tailored with 1024 units, is pivotal for processing sequences inherent in sign language gestures, capturing temporal dependencies essential for dynamic gesture recognition. The dataset, meticulously curated from 1495 gesture samples, translates into 89,700 frames, providing a comprehensive foundation for model training. The inclusion of a Dropout layer, set at a 0.5 rate, strategically combats overfitting by nullifying half of the neuron activations, thereby forcing the model to learn more robust features.

The architecture concludes with a softmax-activated Dense layer, optimized for class distribution across gesture categories. This system, fine-tuned with the Adam optimizer and the cross-entropy loss function, undergoes rigorous evaluation to ensure accuracy and generalizability across diverse datasets, demonstrated by high performance metrics in training, testing, and validation phases. The methodology's thoroughness is further exemplified by the adoption of a Leave-One-Out cross-validation strategy, enhancing the model's reliability and applicability in real-world sign language interpretation scenarios.

The culmination of this research underscores the integration of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks as a robust methodology for dynamic gesture recognition in sign language. By balancing architectural elements to prevent overfitting and employing dropout and cross-entropy optimization, the models demonstrate exceptional ability to generalize from extensive training data. This approach not only achieves high accuracy in recognizing a wide range of gestures but also highlights the potential for real-time communication assistance for the deaf and hard-of-hearing community.

The meticulous dataset preparation, alongside rigorous validation strategies like Leave-One-Out cross-validation, further validates the models' effectiveness and applicability in practical scenarios, marking a significant advancement in assistive communication technologies.

3.5. Visualization and user interaction

The system is engineered for seamless real-time operation, ensuring the continuous intake of video input and delivering instant feedback to the user. It is meticulously structured around the following foundational steps, which form the core of its operational algorithm:

- Data acquisition: collect video data using an integrated or external camera, ensuring high-resolution and appropriate frame rates for capturing gestural words and finger-spelling alphabet.
- Video processing and pose detection: process each video frame with MediaPipe for human pose detection, identifying various body landmarks including hands, facial features, and body posture.
- Key points extraction: extract coordinates of key points related to hands and gestures from the video frames, structuring them in a format suitable for machine learning models.
- Gesture language recognition: use the extracted features to classify gestures employing LSTM1024 (for sequence data) and 2DCNN (for spatial structures in images) models, trained on a comprehensive dataset.
- Visualization and user interaction: display recognized gestures on the user interface alongside the video feed for real-time feedback, allowing users to interact with the system via a graphical interface.

This algorithm emphasizes real-time processing, dynamic response to input data, and inclusivity for individuals using sign language, leveraging advanced neural network architectures for accurate gesture recognition. Recognized gestures are displayed on the user interface alongside the video feed, providing an intuitive interaction for the user (Figure 2).

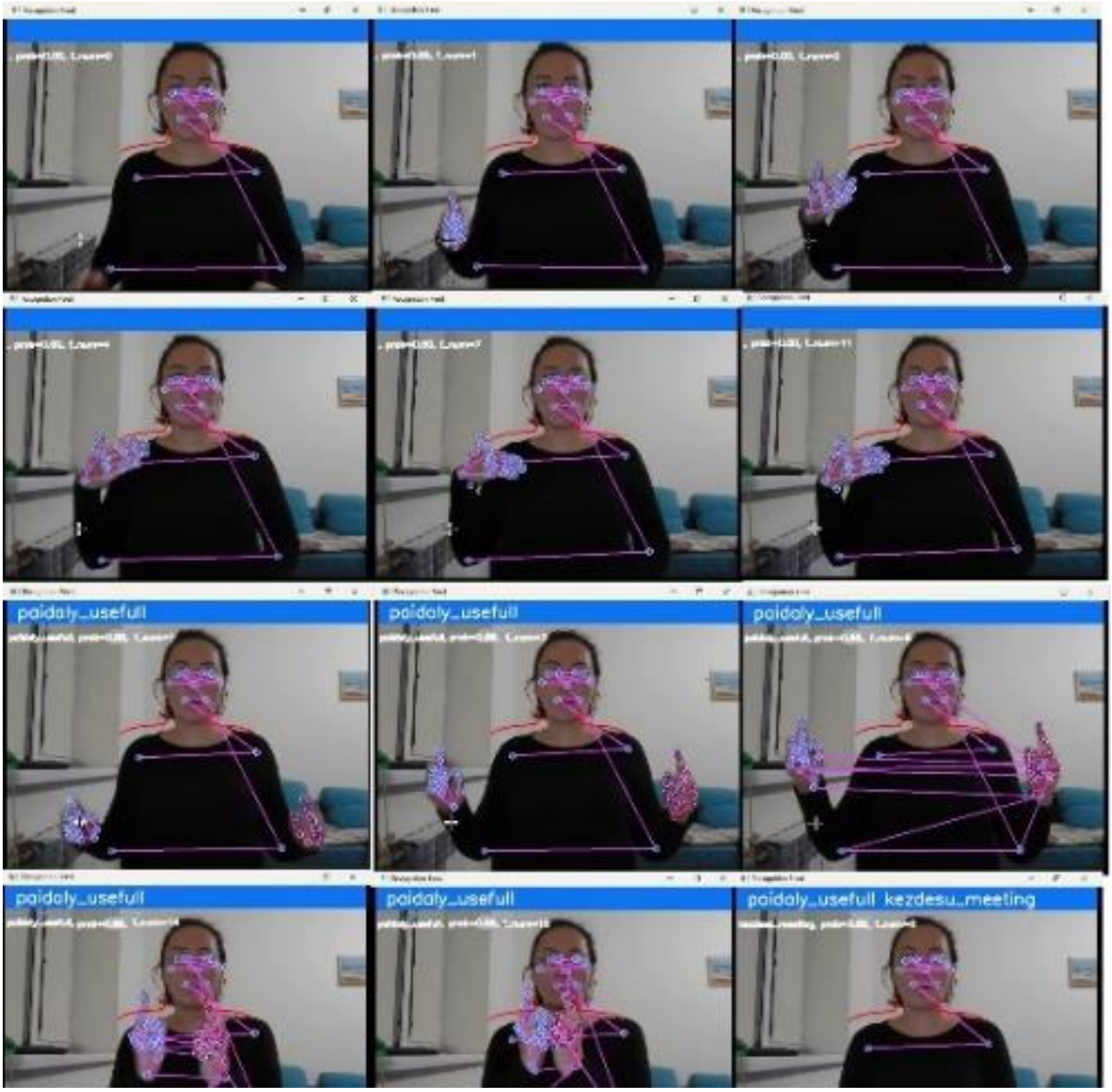

Figure 2: Current sentence consisting of words in the root form

Users can initiate or halt the gesture recognition process at their discretion and interact with the system through a simple graphical interface. The system's design prioritizes real-time operation, ensuring continuous video input and immediate feedback, as detailed in the algorithm below. This approach facilitates dynamic, instantaneous response to sign language gestures, offering an inclusive, accessible communication tool for the deaf and hard-of-hearing community.

Through a sophisticated process involving data acquisition, video processing, key points extraction, gesture language recognition, and interactive visualization, the system utilizes advanced neural network architectures to achieve precise gesture recognition.

This methodology underscores the importance of real-time processing and user interaction, showcasing the technology's capability to translate sign language gestures into text or speech instantaneously, providing an intuitive, user-friendly interface for effective communication.

4. Comparing methods for continuous gesture recognition: LSTM vs CNN

The performance variation between LSTM1024 and 2DCNN across the two datasets [1,2] suggests that the models respond differently to different data distributions. This is a common scenario in machine learning, where a model that performs well on one dataset may not necessarily perform as well on another due to inherent differences in data characteristics (such as context, phrasing, language style, etc.).

It should be noted that the experiments were conducted on the INVIDIA FeForse RTX 3050 Laptop GPU, where training took from 5 to 10 minutes, and namely, the extraction of characteristics from the video stream and the preparation of data for the input of the neural network takes about 40 minutes.

The compare the performance of two different machine learning models, LSTM1024 and 2DCNN, on sign language words recognition tasks it tp got based on data from Table I and Table II.

Table 1

Kazakh sign language (KSL) dataset

No.	Sign word	LSTM1024			2DCNN		
		P	*R*	*F1-sc*	*P*	*R*	*F1-sc*
0.	Bring	1.00	1.00	1.00	1.00	1.00	1.00
1.	Mother	1.00	1.00	1.00	0.75	1.00	0.86
2.	Grand mother	0.75	1.00	0.86	1.00	1.00	1.00
3.	Affect	1.00	0.80	0.89	1.00	1.00	1.00
4.	Children	0.75	1.00	0.86	1.00	0.67	0.80
5.	Belt	0.83	1.00	0.91	1.00	0.60	0.75
6.	Room	1.00	1.00	1.00	1.00	0.80	0.89
7.	Medications	1.00	1.00	1.00	0.75	1.00	0.86
8.	Good	1.00	1.00	1.00	1.00	1.00	1.00
9.	Light	1.00	0.80	0.89	1.00	1.00	1.00
10.	Nephew	0.62	1.00	0.77	0.67	0.40	0.50
11.	Meeting	1.00	1.00	1.00	0.71	1.00	0.83
12.	Rejoice	0.83	1.00	0.91	1.00	0.80	0.89
13.	Blanket	1.00	1.00	1.00	1.00	1.00	1.00
14.	Neighbor	1.00	1.00	1.00	1.00	1.00	1.00
15.	Sit down	0.75	0.60	0.67	0.60	0.60	0.60
16.	Usefull	1.00	0.80	0.89	0.62	1.00	0.77
17.	Teapot	0.80	0.80	0.80	0.80	0.80	0.80
18.	Call	1.00	0.33	0.50	0.75	1.00	0.86
19.	Fast	1.00	0.80	0.89	0.75	0.60	0.67
accuracy		0.89			0.85		
macro avg		0.92	0.90	0.90	0.87	0.86	0.85
weighted avg		0.92	0.90	0.90	0.87	0.85	0.85

Each row represents a different sign word, and the models' performance is measured in terms of:

Precision P ($P = TP/(TP + FP)$),

Recall R ($R= TP/(TP + FN)$),

F1-score (F1-sc = $2 * P * R/(P + R)$).

Here $TP -$ precision ration of true positive; $TP -$ precision ration of faulse positive; $FN -$ precision ration of faulse negative.

Both models exhibit strong performance across most sign words, with high scores in precision, recall, and F1-score. However, there are variances on a per-sign-word basis.

In the Table I, LSTM1024 has higher average precision, recall, and F1-score than 2DCNN, indicating it may be better at recognizing those particular sign words.

In the Table II, both models are mostly comparable, though 2DCNN has a slightly higher overall accuracy.

Certain words like "Nephew" in the first table and "Continuation" in the second table show significant differences in performance between the two models, indicating that one model may have learned the features for these signs better than the other. The "Call" sign in the first table is a clear weak point for the LSTM1024 model, with a recall of 0.33, indicating it frequently misses this sign.

The overall val_accuracy is high for both models, but remember, accuracy might not be a sufficient measure when the classes are imbalanced.

For both models, the performance metrics for both training and testing are 0.99. This demonstrates exceptional accuracy and indicates that the models have adapted superbly to the provided data. Within the experiment outlined in articles [3] and [9], we encountered overfitting.

However, by employing an early stopping strategy, we achieved the results reflected in the Table I.

When applied to continuous recognition tasks, the LSTM model exhibited superior performance, achieving the highest values in average probabilities compared to other models. This underscores the model's robustness and advanced capability in handling sequential data, enhancing its reliability for real-time, continuous sign language recognition [2].

Table 2

Continuous Kazakh Sign Language (KSL) Dataset

No.	Sign word	LSTM1024			2DCNN		
		P	*R*	*F1-sc*	*P*	*R*	*F1-sc*
1	Pity	1.00	1.00	1.00	1.00	1.00	1.00
2	Bring	1.00	1.00	1.00	1.00	0.80	0.89
3	Drag	0.62	1.00	0.77	0.62	1.00	0.77
4	Influence	1.00	1.00	1.00	1.00	1.00	1.00
5	Skirt	1.00	0.60	0.75	1.00	0.80	0.89
6	Continuation	1.00	0.40	0.57	1.00	0.80	0.89
7	Nephew	0.83	1.00	0.91	1.00	1.00	1.00
8	Forgive	1.00	0.80	0.89	0.80	0.80	0.80
9	Husband	1.00	0.80	0.89	0.75	0.60	0.67
10	Loading	0.83	1.00	0.91	1.00	1.00	1.00
11	Bring	0.50	1.00	0.67	1.00	1.00	1.00
12	Mother	1.00	1.00	1.00	0.75	1.00	0.86
13	ter	0.75	1.00	0.86	0.75	1.00	0.86
14	Disease	1.00	1.00	1.00	1.00	1.00	1.00
15	Children	1.00	1.00	1.00	1.00	1.00	1.00
16	Medicine	1.00	1.00	1.00	1.00	0.67	0.80
17	Meat	1.00	1.00	1.00	1.00	1.00	1.00
18	Neighbor	1.00	1.00	1.00	1.00	0.67	0.80
19	The invitation	1.00	0.33	0.50	1.00	1.00	1.00
20	Dating	1.00	1.00	1.00	0.75	1.00	0.86
accuracy		0.89			0.90		
macro avg		0.93	0.90	0.89	0.92	0.91	0.90
weighted avg		0.93	0.89	0.89	0.92	0.90	0.90

1. "Көрші шәйнекті алып келді." ("The neighbor brought a kettle.") - accuracy 0.88
2. "Анам шаттанды." ("My mother was happy.") -accuracy 0.91
3. "Балалар жарық бөлмеде отыр." ("The children are sitting in a bright room.") - accuracy 0.89
4. "Апам күйеу баласын қуана шақырды." ("My mother happily invited her son-in-law.") - accuracy 0.85
5. "Тәтеме дәрі тез әсер етті." ("The medicine affected my aunt quickly.") - accuracy 0.89
6. "Апам жақсы белдемшені алып келді." ("My mother brought a nice skirt.") - accuracy 0.88
7. "Анам балалардың көрпесін әкелді." ("My mother brought the children’s blankets.") - accuracy 0.82
8. "Апам жиенін кездесуге апарды." ("My mother took her nephew to the meeting.") - accuracy 0.90

5. Conclusion

In summary, the performance of LSTM1024 and 2DCNN models varies with specific sign words, indicating that the nature of these models influences their effectiveness. LSTM networks excel in recognizing dynamic gestures due to their ability to learn order dependence in sequence prediction problems, a crucial aspect of continuous sign language recognition.

Conversely, 2D CNNs, primarily utilized for image data, show heightened effectiveness with static signs or in datasets rich in spatial features.

Despite these differences, the LSTM model stands out for its exceptional efficiency and precision in continuous sign language recognition, evident from its superior average probability metrics. Its proficiency in handling temporal dependencies makes it ideal for the nuanced task of interpreting the fluid nature of sign language.

This theoretical superiority is corroborated by practical tests, where the LSTM model consistently showed high prediction accuracy across various sentence examples.

Conducted research on ordinary computers allows to achieve fairly high accuracy and speed on test data without the use of special graphic tools. Disadvantages of using such methods on ordinary computers include a rather long time of training and data preparation for the neural network.

Future research will be aimed at expanding the set of Kazakh sign language sentences and increasing the accuracy of the proposed methods.

Research will also be conducted for other sign languages.

Acknowledgements

This work is part of the AP19175452 project funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, aimed at developing automated interpretation models for Kazakh sign language.

References

[1] N. Amangeldy, M. Milosz, S. Kudubayeva, A. Kassymova, G. Kalakova, L. Zhetkenbay. A Real-Time Dynamic Gesture Variability Recognition Method Based on Convolutional Neural Networks. Appl. Sci. (2023) 13, 10799. https://doi.org/10.3390/app131910799.

[2] N. Amangeldy, A. Ukenova, G. Bekmanova, B. Razakhova, M. Milosz, S. Kudubayeva. Continuous Sign Language Recognition and Its Translation into Intonation-Colored Speech. Sensors (2023) 23, 6383. https://doi.org/10.3390/s23146383.

[3] A. S. M. Miah, M. A. M. Hasan, J. Shin, Y. Okuyama, Y. Tomioka. Multistage Spatial Attention-Based Neural Network for Hand Gesture Recognition. Computers, (2023) 12(1). https://doi.org/10.3390/computers12010013.

[4] D. Satybaldina, G. Kalymova. Deep learning based static hand gesture recognition. Indonesian Journal of Electrical Engineering and Computer Science, (2021) 21(1). https://doi.org/10.11591/ijeecs.v21.i1.pp398-405.

[5] H. Brock, I. Farag, K. Nakadai. Recognition of non-manual content in continuous Japanese sign language. Sensors, (2020). 20(19). https://doi.org/10.3390/s20195621.

[6] B. Aksoy, O. K. M. Salman, Ö. Ekrem. Detection of Turkish Sign Language Using Deep Learning and Image Processing Methods. Applied Artificial Intelligence, (2021) 35(12). https://doi.org/10.1080/08839514.2021.1982184.

[7] D.R. Kothadiya, C.M. Bhatt, A. Rehman, F.S. Alamri, T. Saba. SignExplainer: An Explainable AI-Enabled Framework for Sign Language Recognition with Ensemble Learning. IEEE Access, (2023) 11. https://doi.org/10.1109/ACCESS.2023.3274851

[8] Z. Zhou, V. W. L. Tam, E. Y. Lam. SignBERT: A BERT-Based Deep Learning Framework for Continuous Sign Language Recognition. IEEE Access, (2021) 9. https://doi.org/10.1109/ACCESS.2021.3132668.

[9] K. Amrutha, P. Prabu, R. C. Poonia. "LiST: A Lightweight Framework for Continuous Indian Sign Language Translation". Information, (2023) 14, 2: 79. https://doi.org/ 10.3390/info14020079.

[10] M. S. Abdallah, G. H. Samaan, A. R. Wadie, F. Makhmudov, Y. I. Cho. Light-Weight Deep Learning Techniques with Advanced Processing for Real-Time Hand Gesture Recognition. Sensors, (2023) 23(1). https://doi.org/10.3390/s23010002.

[11] Q. Xiao, X. Chang, X. Zhang, X. Liu. Multi-Information Spatial-Temporal LSTM Fusion Continuous Sign Language Neural Machine Translation. IEEE Access, (2020) 8. https://doi.org/10.1109/ACCESS.2020.3039539.

[12] T. Witchuda, A. Wiranata, S. Maeda, C. Premachandra. Reservoir Computing Model for Human Hand Locomotion Signal Classification. IEEE Access, (2023) 11. https://doi.org/10.1109/ACCESS.2023.3247631.

[13] Y. Jiang, L. Song, J. Zhang, Y. Song, M, Yan. Multi-Category Gesture Recognition Modeling Based on sEMG and IMU Signals. Sensors, (2022) 22(15). https://doi.org/10.3390/s22155855.

[14] I. Papastratis, K. Dimitropoulos, D. Konstantinidis, P. Daras. Continuous Sign Language Recognition through Cross-Modal Alignment of Video and Text Embeddings in a Joint-Latent Space. IEEE Access, (2020) 8. https://doi.org/10.1109/ACCESS.2020.2993650.

[15] O. Koller, J. Forster, H. Ney. "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers", Comput. Vis. Image Understand., vol. 141, pp. 108-125, Dec. 2015. DOI: 10.1016/j.cviu.2015.09.013.

[16] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural sign language translation", Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 7784-7793, March 2018. DOI: 10.1109/CVPR.2018.00812.

[17] J. Huang, W. Zhou, Q. Zhang, H. Li and W. Li, "Video-based sign language recognition without temporal segmentation", Proc. 32nd AAAI Conference on Artificial Intelligence (AAAI-18): New Orleans, Louisiana, USA, February 2018. pp. 1-8. ArXiv:1801.10111v1 [cs.CV]30 Jan2018.

[18] S. Kondratiuk, I. Krak, V. Kuznetsov, A. Kulias. Using the Temporal Data and Three-dimensional Convolutions for Sign Language Alphabet Recognition. CEUR Workshop Proceedings 3137, CEUR-WS.org 2022. – P.78-87. DOI: 10.32782/cmis/3137-7.

[19] I. Papastratis, K. Dimitropoulos, P. Daras. Continuous Sign Language Recognition through a Context-aware Generative Adversarial Network. Sensors, (2021) 21(7) 2437. https://doi.org/10.3390/s21072437.