Journal of Robotics and Control (JRC)
Volume 6, Issue 3, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i3.25881

Deep Learning-Based Continuous Sign Language Recognition

Lazzat Zholshiyeva [1], Tamara Zhukabayeva [2*], Azamat Serek [3], Ramazan Duisenbek [4], Meruert Berdieva [5], Nurshapagat Shapay [6]
[1, 2] Department of Information Systems, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
[3] School of Information Technology and Engineering, Kazakh-British Technical University (KBTU), Almaty, Kazakhstan
[2, 4] Department of Computer Engineering, Astana IT University, Astana, Kazakhstan
[5] Department of Medical Biophysics and IT, South Kazakhstan Medical Academy, Shymkent, Kazakhstan
[5] Department of Information Systems, SDU University, Kaskelen, Kazakhstan
Email: [1] lazzat.zhol.81@gmail.com, [2] zhukabaeva_tk@enu.kz, [3] a.serek@kbtu.kz
[4] ramazzan.duisenbek@gmail.com, [5] asssmeruert@gmail.com, [6] shiposha04@gmail.com
[*] Corresponding Author

***Abstract*—This study focuses on the development of a continuous sign language recognition system based on deep neural network models. A new Kazakh Sign Language (QazSL) dataset is created. DL models for continuous KazSL are developed, their accuracy and robustness under different environmental conditions are analyzed, and an optimized model algorithm to improve sign recognition processes are proposed. The main goal is to improve gesture recognition accuracy, account for gesture variability and environmental conditions, and promote the development of adaptive technologies for low-resource languages. This paper proposes a QazSL recognition system using an YOLOv8n and optimized 2DCNN models to improve accessibility for the hearing impaired. The optimized 2DCNN method includes optimal data preprocessing techniques and new training architecture, followed by model training and testing with precision, recall, and accuracy metrics. The proposed systems were trained using an open-course K-RSL dataset with 5 signers and a newly created QazSL dataset, recorded by 7 signers. The test accuracy of gesture recognition are 98.12% for Yolov8n and 98, 57% for 2DCNN, indicating the robustness and capability of the models for real-time application. Certain issues, such as background variation and gesture consistency, were found to affect recognition under different conditions. This research contributes to the development of AI-based assistive technology to facilitate social inclusion and access to communication for deaf and hard-of-hearing people. By addressing the challenges identified in gesture recognition, this study paves the way for more reliable interactions between users and technology. Future work will focus on optimizing the model further to enhance its performance in varied environments and to expand its applicability across different languages and sign systems.**

***Keywords*—Computer Vision; Sign Language; Real-Time Recognition; Deep Learning; 2DCNN; Yolo Optimization**

I. Introduction

People with high levels of hearing or profound hearing loss depend on sign language for communication. Similarly to spoken languages, sign languages have their phonology and syntax that enable complex and meaningful communication [1]. Spoken languages derive from sound signals, whereas sign languages derive from visual signals such as hand shape, movement, facial expression, and body orientation [2], [3]. A single sign can represent a word, a phrase, or an entire sentence [4]. In addition, sign languages have been recognized as natural languages, vital for the culture and identity of deaf people [5].

Sign language recognition (SLR) is performed using computer technology, so it is one of the most significant advances for the employment of hearing-impaired individuals. Automated gesture translation systems have the potential to improve communication between deaf and hearing individuals immensely [6]. For that, the improved recognition rates and enable real-time performance, deep learning (DL) and machine learning (ML) techniques are frequently employed [7]–[9]. Computer vision uses artificial intelligence (AI) to automate sign language detection through sophisticated image processing methods [10].

Among various AI-based techniques, the "You Only Look Once" (YOLO) algorithm has emerged as a good real-time object detection model. It supports quick gesture recognition with whole-image processing in one pass through the network, in contrast to other approaches that require several iterations [11], [12].

Variability in hand gesture, inconsistency of user execution, and the ambient physical environment, including lighting and noise in the background, are some of the major limitations of YOLO-based systems, whose robust performance has been established in real-time sign language recognition [13]. Also, the absence of diversity in current datasets lowers the generalizability of the model, and since there are over 300 different sign languages spoken across the world, each having a unique syntax

and grammar, developing a generalized recognition system remains a big challenge [14]–[16]. These issues are especially applicable to Kazakh Sign Language (KazSL), which lacks extensive research and big annotated datasets; these must be resolved to improve recognition accuracy and provide real-time KSL translation applications.

Increased progress in computational power, particularly employing graphics processing units (GPUs), has further advanced real-time image processing ability [17].

The YOLO model, enhanced with DL methods, provides real-time processing with high accuracy [18]. Despite its high accuracy, there are certain challenges the model faces, one of which is the performance of the dataset and the consistency of gestures performed by individual users [19]. If such factors as hand position, manner of performing gestures, and environmental conditions change, the recognition of gestures might easily fail [20]. These problems are similar to challenges that arose in processing texts of the Kazakh language [21]. Even if it is not without its problems, the YOLO-based systems exhibit colossal promise in real-time SLR; very prominent issues remain around data limitations, the need for optimizing algorithms, and the improvement of accuracy and quality of visualization in varying conditions [22], [23].

New systems not only recognize sign language but also convert text to sign language in real-time, easing communication between deaf individuals and other individuals who are not proficient in sign language [24]. YOLO-based systems effectively monitor and classify hand gestures, mapping them to appropriate words and phrases, thus improving the quality of interaction [25].

LSTM are effective in dynamic gestures recognizing because they are capable of learning sequential dependencies. On the other hand, 2DCNN models are more effective with spatial data. However, while research conducted on ordinary computers provided high accuracy and speed, the duration of training and data preparation remains an actual issue [26].

While all these advances enhance communication for the deaf significantly, there remains a gap in the achievement of widespread usage and understanding of sign language among the general population. Further research needs to be conducted to bridge this gap and enable inclusivity.

CSLR is a sequence-to-sequence task, in which the aim is to translate a sequence of visual gestures into a fluent sign language translation. This task normally needs the application of state-of-the-art DL models [27].

The solution to these challenges will be crucial for the successful integration of such systems for the everyday communication of deaf and hard-of-hearing communities. This research suggests the YOLOv8n, 2DCNN models for KSL recognition in real-time with high accuracy.
The research contribution:

- We present the YOLOv8n and optimized 2DCNN models for recognition of Continuous Kazakh Sign Language.
- We create own dataset.
- We quantify the precision and robustness of the models across different environmental conditions, which allows us to optimize SLR through DL.
- We propose the optimized algorithms of proposed models.

This paper aims to develop a real-time Continuous KazSL recognition system via DL models in an attempt to increase the recognition rate, address issues related to changes in gesture and environmental conditions.

The structure of the paper is as follows: Section 1 presents the relevance, contributions of the research, objectives, and structure of the paper. Section 2 discusses previous studies. Section 3 describes methodology. Section 4 presents the results and discussion. Section 5 concludes the paper.

II. Related Work

Static, dynamic and continuous sign language recognitions are different categories of sign language. Several kinds of DL, ML algorithms have been proposed for image-based emotion, activity, and sign language recognition systems [28].

YOLO, used for real-time SLR, involves sophisticated object detection technologies that enhance the accuracy and efficiency of gesture recognition [29], [30]. Some literature is now available reporting a variety of approaches that resort to YOLO and other methods to address problems related to overlapping and segmentation for continuous gesture recognition [31]–[33].

Every researcher knows that YOLO is best at real-time processing. This makes it suitable for applications that require instant feedback, such as sign language translation [34]. There have also been many indications that instructor gesture detection works well in YOLO, often achieving very encouraging accuracy rates with varied datasets [35]. Much research has been accomplished through different versions of YOLO based on American Sign Language (ASL) [36], Indonesian Sign Language [37], and Arabic Sign Language [38] within varying contexts. Integrating deep learning technologies with the YOLO sign language recognition approach shows good promise for communication improvements for people with hearing impairments [39], [40]. For instance, the various efforts at combining YOLO with CNN and RNN could enhance the appreciation of gestures and other context cues that can enhance the translation quality for sign language to text [41].

The integration of YOLO with other models, such as LSTM and CNN, contributes to accurate and efficient sign language gesture recognition by solving issues with overlaps and variations in gesture length [42], [43]. The combination of YOLO with LSTM allows the capture of both spatial and temporal features, improving the recognition of dynamic gestures [44].

Systems using YOLO have shown significant improvements in accuracy, achieving recognition rates of over 94% [45].

The processing speed makes YOLO architecture suitable for real-time applications. For instance, YOLOv8 has been successfully used in other fields, achieving over 80% accuracy in real-time video analysis [46]. The integration of YOLO with multi-threaded neural networks can further enhance gesture recognition by simultaneously analyzing hand movements, facial expressions, and body postures [47].

Sign language includes complex gestures that may look similar in appearance. YOLOs ability to learn from different datasets may solve this issue by improving recognition accuracy through intensive training [48]. Effective sign language recognition systems require huge annotated datasets. The development of such datasets is needed for training YOLO models to accurately recognize different gestures [49], [50].

In the last few years, numerous researchers were capable of applying convolutional neural networks (CNNs) deep models to feature engineering. CNNs were capable of achieving very good results in object and speech recognition, image classification, and human activity recognition [51].

It is difficult to design and implement 2D CNN-based systems for sign language recognition because there are several limitations such as scarce computational resources, precision, lack of data, and accessibility. Training deep neural networks, especially processing video data, requires heavy GPU power and memory. For example, models used for real-time video processing, such as those proposed by [52], face latency and inefficiency issues in low-end devices.

Real-time processing is another major challenge. While 2D CNN models are appropriate for image-based recognition, their adaptation to video-based sign language recognition requires temporal modeling. For instance, the paper [53] proposes Time Distributed CNN models to solve this issue, but these models are computationally intensive. Maintaining both accuracy and speed is a real challenge, particularly for systems intended to classify dynamic gestures online [54].

The depth of CNN models also leads to issues. Even though deeper models can learn higher-level features, they require greater data and more computational power as well. For example, the InceptionResNetv2 model used in [55] is highly accurate but not necessarily deployable on edge devices. It is an area of research to optimize such models for deployment without affecting performance [56].

Real-time translation and feedback are essential for ensuring continuous communication between signers and non-hearing individuals [57]. That platform integrates the SLR system with speech-to-text models, offering a more inclusive communication tool. Table I and the articles outlined in this section suggest the active development of YOLO for real-time sign language recognition.

In recent research on KazSL recognition, there are noticeable differences in the approaches, particularly concerning dataset size, models used, and the scope of work (Table II).

III. Methodology

The methodology involves data collection, preprocessing, model selection, training, testing, evaluation and deployment. Fig. 1 depicts the whole research process.

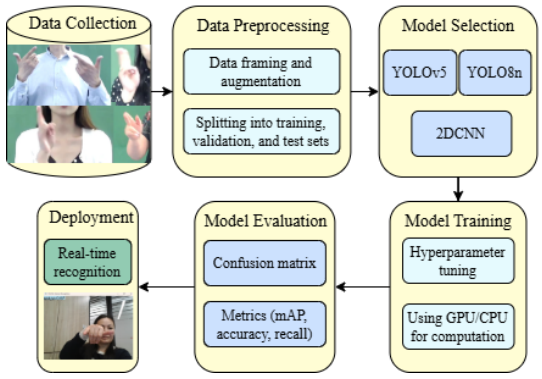

Fig. 1. Methodology

A. Data Description and Preprocessing

Interrogative signs may be accompanied by raised eyebrows, head tilt to the side or the back, or not at all. Eyebrow and head direction, therefore, and production serve independent roles in identifying signs: the former separates questions from statements, and the latter helps separate interrogative signs of various types. Accordingly, the datase, with question statmenets is chosen. A specific dataset was used in the experiment to separate comparable motions based on non-verbal features. The dataset comprises 5,200 video clips of Kazakh-Russian Sign Language (K-RSL) signs [74]. While hand movements are uniform, non-verbal signs are not, e.g., head direction, mouth positions, eyebrow lifting, and face expressions.

The dataset, which we chose, consists of video clips of individual signs extracted from complete sentences, divided into 20 folders (10 signs in statements and questions). Each folder contains approximately 260 videos (40 samples for 4 signers and 100 samples for 1 signer). Each video contains a single participant performing the same gesture multiple times.

All videos were taken with a solid green screen background to minimize distractions and were recorded indoors with stable lighting conditions to minimize variation. The videos are organized into subfolders, each named according to each gesture, such as "қалай?" - "How?", "қанша?" - "How much?" / "How many?", "не?" - "What?", "қашан?" - "When?", "қайда?" - "Where?", "қайсы?" - "Which?", "кім?" - "Who?", "неге?" - "Why?"(Fig. 2). The filming for all videos took place against a uniform green background to reduce distracting factors.

TABLE I. A SUMMARY OF STUDIES ON SLR WITH YOLO

Year	Key Results	Methodology	Main Ideas
2024 [58]	Combined YOLOv8 with NLP for robust real-time gesture recognition and translation	YOLOv8 for gesture detection, NLP for translation	Improved real-time bidirectional communication for sign language users
2024 [59]	Developed a YOLOv5-based system for real-time BISINDO alphabet recognition, achieving high detection accuracy	YOLOv5 algorithm for detecting BISINDO alphabet signs	Efficient sign recognition in Indonesian sign language, targeting real-time applications
2024 [60]	Developed a robust real-time detection system using YOLO for sign recognition, focusing on accessibility for DHH users	YOLO architecture tailored for sign language gestures	Aims to bridge communication gaps for the DHH community
2024 [61]	Comparative analysis of YOLOv5 and CNN on MSL dataset	YOLO architecture tailored for sign language gestures	Validates the effectiveness of YOLOv5 in sign language detection tasks
2024 [62]	Achieved high-speed detection of Indian Sign Language gestures using YOLO NAS	YOLO NAS model for fast and efficient gesture detection	Enhances real-time recognition for Indian Sign Language
2025 [63]	Highlighted the cost-effectiveness and accuracy of YOLO for detecting sign language gestures	YOLO algorithm for identifying sign language gestures	Focuses on accessible solutions for sign language recognition
2024 [64]	YOLO for gesture recognition, LSTM for text generation	Seamless integration of recognition and captioning for sign language	Seamless integration of recognition and captioning for sign language
2024 [65]	YOLO model provides better accuracy and precision	YOLO algorithm used for gesture identification and focus on accuracy, precision, and F1-Score measures	YOLO algorithm improves gesture identification accuracy and performance
2024 [66]	Achieves good accuracy with a small dataset size	YOLO-based sign recognition for hand gesture detection	YOLO-based system improves sign language recognition accuracy

TABLE II. STUDIES ON KAZSL RECOGNITION

Year	Purpose	Method	Accuracy	Advantages/Limitations
2020 [67]	Assessing the necessity of distinguishing similar signs differing only in non-manual elements	SVM, LR, RF, BayesNet	77%	Identified accuracy differences based on non-manual components, highlighting their importance in recognition
2021 [68]	Comparing EfficientNetB7 with a KazSL dataset and evaluating CNN models	CNN	99%, 99%, 64%, 96.7%	SHAP images revealed areas needing improvement, particularly in letters such as Gh, H, Ng, Ui
2023 [69]	Detecting all 42 dactyls of KazSL detection;	SVM, MediaPipe	99%	Effective for static gestures, not effective for dynamic gestures, highlights the potential of neural networks for improved accuracy
2023 [70]	Providing an in-depth analysis of recognition methods based on their objectives and efficiency	SVM, CNN	89%	Encountered issues with low-light conditions; camera sensitivity to angles and backgrounds affected recognition
2024 [71]	Developing an identification program with high precision for KazSL	RF, Extreme Gradient	88.6%, 98.68%, 98.54%	Computational constraints posed challenges; visibility and illumination quality were suboptimal
2023 [72]	Developing a system	CNN, Resnet50	94.4%, 84%	Created a dataset with 57,708 images for KazSLR
2024 [73]	To develop fast algorithms and optimize computational processes to detect real-time recognition	LSTM, RNN, Resnet50	85%	A combination of layers of LSTM and CNN models gave better performance in KazSL dactyls recognition.

Five individuals took part in data collection, including male and female signers from various age groups and linguistic backgrounds. To increase the model's resilience to environmental changes, videos were shot in a variety of indoor lighting scenarios. To record minute differences in execution style, each signer made motions several times. To ensure that every participant was included in each subset, the data was separated into training (70%), validation (15%), and test (15%) sets. By including a test set, generalizability to real-world situations is improved.

In the preprocessing phases, every video was divided into frames per second (FPS). The number of frames and labels varies depending on the duration of the video. Each frame label is assigned metadata that describes the content of the gesture and its position within the frame. The structure of the label looks as below:

$$\langle \text{class-id}, x_{\text{center}}, y_{\text{center}}, \text{width}, \text{height} \rangle$$

For increasing the robustness of the model, cross-validation

and data augmentation techniques were used to improve generalization, comprising brightness and contrast changes. The pixel values were also normalized into the range of [0,1] to achieve the best possible input for deep models.

Fig. 2. Dataset containing 8 classes with videos

All images and labels are saved in .jpg and .txt formats, respectively. An example of framing for class "қалай?" is shown below in Fig. 3.

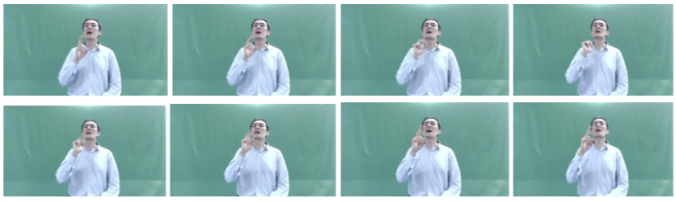

Fig. 3. Framing for class "қалай?" [74]

B. Model Selection and Justification

Though there are several YOLO versions, YOLOv8n was chosen since it maintains a compromise of accuracy and computational efficiency. Table III illustrates the comparison of YOLO models based on parameters, speed, and accuracy.

TABLE III. COMPARISON OF YOLO VARIANTS FOR SLR

Model	Parameters	FPS	mAP@0.5	Suitability
YOLOv5	7M	83	92%	Moderate
YOLOv6	6M	105	94%	High
YOLOv7	4.7M	125	94.5%	High
YOLOv8	3M	140	96%	Very high
YOLOv8 Nano	2M	170	92.2%	Optimal

YOLOv8n is ideal for real-time KazSL detection with an efficient, lightweight, yet strong architecture deployable in devices with poor computation capabilities.

Second dataset are KazSL Sentences like "Бүгін ауа райы өте жылы"-Bugin aya raiy ote zyly. "Достарыммен бірге кино көрдім"- Dostarymmen birge kino kordim, "Мен Қазақстанға келдім" - Men Kazakhstanga keldim, "Мен мектепке бара жатырмын" - Men mektepke bara zatyrmyn, "Не істеп жатырсың?" - Ne istep zatyrsyn?, - Men Kazakhstanga keldim, "Мен мектепке бара жатырмын" - Men mektepke bara zatyrmyn, "Қыста қар жауады" - Qysta qar zauady, "Сәлем! Қалың қалай?" - Salem! Kalyn kalai?. Seven signers participated in the collection of the dataset.

C. Model Architecture

1) Yolov8n: The YOLOv8n architecture is one of the lightweight versions of YOLOv8 that is designed for object detection, segmentation and classification with minimal computational resource usage [75], [76]. The YOLOv8n architecture can be described as a sequence of linear and nonlinear operations which transform input image X to output form containing detections. In our case, each video was converted into frames and each frame was input for the model. The input image size is

$$640 \times 640 \times 3$$

1. Input data

$$X \in \mathbb{R}^{640\times640\times3} \tag{1}$$

where 640×640 denotes the image resolution, and 3 is several channels (RGB).

2. Backbone

It is a core part of the model that extracts spatial and contextual features from the input image. The adapted version of CSP-Darknet (Cross Stage Partial Networks) is used in YOLOv8 as it optimizes computational efficiency.

Stem block

Stem is the initial convolutional block with Batch Normalization and activation (typically SiLu or Leaky ReLU). Each convolutional layer can be described as below:

$$Y_1 = \sigma(\text{Conv}(X, W_1)) \rightarrow \text{SiLU} \tag{2}$$

where W denotes the convolution kernel and

$$\sigma(\cdot)$$

represents the activation function.

C2f blocks

C2f Blocks are the main blocks that decrease computational complexity by splitting data paths. The input image splits into two paths, one going through convolutional layers, the other bypassing directly. The outputs are merged using concatenation. C2f divides the input tensor into two paths, such as the direct path

$$X_1 = X$$

and convolutional path

$$X_2 = \text{Conv}(X)$$

Their merge is performed through concatenation:

$$X_2 = \text{Conv}X \tag{3}$$

where

$X_{\text{direct}} denotes features bypassing through the direct path, X_{\text{conv}}$

and represents convolutionally processed features. Processing occurs at 3 levels: $64 \rightarrow 128 \rightarrow 256$

SPPF improves feature extraction at different scales. This architecture (Table IV) gradually reduces the image size while increasing the features.

TABLE IV. NETWORK STRUCTURE FOR IMAGE PROCESSING USING SPPF

Layer	Type	Channels	Kernel size	Output size
1	Conf	32	3x3	640Œ640
2	C2f	64	3x3	320Œ320
3	C2f	128	3x3	160Œ160
4	C2f	256	3x3	80Œ80
5	C2ff	512	3x3	40Œ40
6	SPPF	512	3x5	40Œ40

3. Neck (FPN+PANet)

The neck is responsible for 3 aggregating and processing feature maps with different resolution levels. It uses FPN and PANet. They work in multiscale feature representation. FPN merges low-level

$$X_{\text{low}}$$

and high-level

$$X_{\text{high}}$$

features using upsampling operations:

$$Y_{\text{PANet}} = \text{Conv}(Y_{\text{FPN}}) \tag{4}$$

PANet enhances the features through a downsampling mechanism:

$$Y_{\text{PANet}} = \text{Conv}(Y_{\text{FPN}}) \tag{5}$$

This module aggregates features extracted at various scales, enhancing the accuracy and reliability of object detection. Upsampling and Concatenation involve increasing the resolution of feature maps and subsequently merging them with corresponding feature maps from other layers.

C2F blocks optimize the network architecture by reducing redundancy in parameters and computations, leading to increased computational efficiency without compromising the quality of feature extraction. Upsampling increases the resolution, while C2f blocks optimize feature extraction (Table V).

4. Head

This module is responsible for object detection and classification and employs regression-based methods instead of predefined anchors. Predictions are made at resolutions of 80Œ80, 40Œ40, and 20Œ20. The head of the network is responsible for generating final predictions about the location of objects, their sizes, class probabilities and additional tasks (e.g., segmentation).

The network outputs include bounding boxes predicting coordinates of bounding boxes for objects, Confidence Scores estimating the probability of object presence within the bounding box, and Class Predictions determining the class of detected objects.

TABLE V. UPSAMPLE AND C2F LAYERS IN THE YOLOV8N

Layer	Type	Channels	Output size
7	Upsample	256	80
8	C2f	256	80
9	Upsample	128	160
10	C2f	128	160

During the detection phase, YOLOv8n forms an output tensor, which includes the coordinates of the bounding box (x, y, w, h), confidence score p, and class probabilities (c_1, c_2, ..., c_8).

$$Y_{\text{output}} = (x, y, w, h, p, c_1, c_2, \ldots, c_8) \tag{6}$$

Where each predicted anchor: (x,y) denotes the centre coordinates of objects, (w,h) is the width and height of the object, p represents the confidence score of an object being inside the bounding box, and c_i indicates class probabilities for eight classes. The layers in Table VI are used to make predictions at different scales, allowing the model to detect objects of various sizes. Each convolutional layer processes feature maps extracted in previous stages and generates final predictions for object detection.

TABLE VI. HEAD MODULE WITH OUTPUTS AT RESOLUTIONS 80Œ80, 40Œ40, AND 20Œ20

Layer	Type	Output size
11	Conv	80
12	Conv	40
13	Conv	20

Final model

The whole architecture can be presented as a composition of functions:

$$Y_{\text{output}} = f_{\text{head}}(f_{\text{neck}}(f_{\text{backbone}}(X))) \tag{7}$$

Where $f_{\text{backbone}}()$ denotes feature extraction, $f_{\text{neck}}()$ represents feature aggregation, $f_{\text{head}}()$ and indicates the generation of final predictions.

Loss function

YOLOv8 uses a combination of the next loss functions: - Coordinate loss (IoU Loss)

$$\text{LIoU} = 1 - \text{IoU}(B_{\text{pred}}, B_{\text{true}}) \tag{8}$$

where IoU represents a measure of overlap between predicted B_{pred} and true B_{true} regions.

$$L_{\text{cls}} = -(1 - p_t)\log(p_t) \tag{9}$$

- Loss classification (Focal Loss)

Where p_t denotes the probability of a correct class, and α, λ are hyperparameters.
- Object loss

$$L_{\text{obj}} = -\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \tag{10}$$

Final loss function:

$$L = \lambda_{\text{IoU}} L_{\text{IoU}} + \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{obj}} L_{\text{obj}} \tag{11}$$

2) Features of YOLOv8n: Unwavering, lightweight neural network for car detection By reducing the number of channels and convolutional filters, YOLOv8n can be implemented on devices with limited computer resources, such as mobile devices [77]. Major differences comprise the addition of a set of loss functions, including CIoU and Focal Loss, a further improved optimization process, and a multitasking detection, segmentation, and classification feature that allowed it to fit into several datasets. YOLOv8n is the most accurate and fastest model in the YOLOv8 family. Due to anchor-free detection, tremendous computation flexibility has been given to models. It is also very fast, having less than 4 million parameters, with an edge over other YOLO [78]. Integration of C2f and SPPF blocks to improve the efficiency of the features. The Fig. 4 demonstrates the algorithm of the proposed model.

3) 2DCNN: This model is used for continuous KazSL recognition. It is developed based on gesture images from videos and categorizes them into pre-defined classes. The model learns unique features of every gesture by utilizing convolutional layers for feature extraction. The model is then tested on diverse images after training to determine its accuracy.

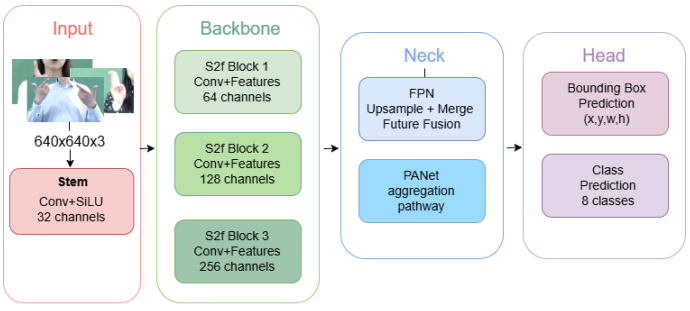

Fig. 4. Input Image Backbone (C2f Blocks) Neck (PANet) Head (Detection Layers)

Proposed 2DCNN architecture

Step 1. Preparing the Video for Processing
A sequence of frames is extracted from the input video.

$$X = x_1, x_2, \ldots, x_n \tag{12}$$

where

$$x_n \tag{13}$$

is an individual frame extracted from the video, and n is the total number of frames.). Step 2. Preprocessing Augmentation condition:

$$X_{\text{aug}} = \begin{cases} A(X), & \text{if } N < 600 \\ X, & \text{otherwise} \end{cases} \tag{14}$$

where: Here, $A(x)$ represents the augmentation methods: rotation, translation, brightness adjustment, and noise addition. Step 3. Data Split

$$X_{\text{train}}, X_{\text{val}}, X_{\text{test}}, X_{\text{aug}} \tag{15}$$

Step 4. Model Architecture Convolutional and Pooling Layers: Layer 1:

$$Z_1 = \text{ReLU}\left(\text{Conv2D}_{3\times3}(X)\right) \tag{16}$$

Layer 2:

$$Z_2 = \text{MaxPooling}_{2\times2}\,\text{ReLU}(Z_1) \tag{17}$$

Layer 3:

$$Z_3 = \text{Dropout}(0.25)(Z_2) \tag{18}$$

The Dropout method prevents overfitting in neural networks. This technique improves the overall generalization of the model by randomly disabling neurons during training.
2nd Convolutional Layer:

$$Z_4 = \text{ReLU}\left(\text{Conv2D}(3 \times 3, 64)(Z_3)\right) \tag{19}$$

3. MaxPooling and Dropout:

$$Z_5 = \text{MaxPooling}(2 \times 2)(Z_4), \quad Z_6 = \text{Dropout}(0.25)(Z_5) \tag{20}$$

The additional MaxPooling and Dropout layers help make computations more efficient and prevent overfitting. Step 5. Fully Connected Layers Flatten layer:

$$Z7 = \text{Flatten}(Z6) \tag{21}$$

Flatten layer converts two-dimensional data into a one-dimensional vector, making it suitable for classification. 1-Dense layer:

$$Z8 = \text{ReLU}(WZ7 + b1), \quad W \in \mathbb{R}^{128\times57600} \tag{22}$$

This layer is a fully connected layer. Here, W1 is the weight matrix, and b1 is the bias vector

$$y = \text{Softmax}(W_2 Z_9 + b_2), \quad W_2 \in \mathbb{R}^{7\times128} \tag{23}$$

Step 6. Loss Function

$$L(y, y) = -\sum_{i=1}^{n} y_i \log(y_i) \tag{24}$$

Step 7. Optimization:

$$\theta^* = \arg\min L(y, f(x); \theta) \tag{25}$$

Step 8. Evaluation Metric

$$\text{Accuracy} = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}(y_i = \hat{y}_i) \tag{26}$$

Step 9. Real-time recognition
The architecture of model is sown in Fig. 5.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
dropout (Dropout)	(None, 63, 63, 32)	0
conv2d_1 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
dropout_1 (Dropout)	(None, 30, 30, 64)	0
flatten (Flatten)	(None, 57600)	0
dense (Dense)	(None, 128)	7,372,928
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 7)	903

Total params: 7,393,223 (28.20 MB)
Trainable params: 7,393,223 (28.20 MB)
Non-trainable params: 0 (0.00 B)

Fig. 5. 2DCNN model architecture

D. Training and Model Summary

After framing and labeling all the training and validation videos, the Yolov8n model was trained in Google Colab using the Ultralytics package. Later, the dataset was uploaded to Google Drive, and a data.yaml file was created showing paths for both training and validation datasets with the class names. After training the model, testing was done.

1) Model Training Process: The YOLOv8n improved the effectiveness of the training process and allowed for effectiveness via transfer learning. The dataset was created with the data.yaml file. The training was conducted for 10 epochs, with the model parameters being optimized. An 8 batch size was set to enable effective utilization of memory and consistent training performance. Input image resolutions were set at 640 x 640 pixels, following the YOLOv8 architecture and ensuring computational efficiency.

As shown in Fig. 6, the precision, recall, and mAP for different classes demonstrate the model's good object detection efficiency, which can be inferred from the data analysis.

Model summary (fused): 168 layers, 3,007,208 parameters, 0 gradients, 8.1 GFLOPs

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
all	1008	1008	0.98	0.972	0.991	0.991
қалай	142	142	0.944	1	0.986	0.986
қанша	131	131	0.951	1	0.995	0.995
не	163	163	0.966	0.933	0.979	0.979
қашан	76	76	0.992	1	0.995	0.995
қайда	101	101	1	0.995	0.995	0.995
қайсы	176	176	0.998	1	0.995	0.995
кім	76	76	0.986	0.916	0.991	0.991
неге	143	143	1	0.931	0.995	0.995

Fig. 6. Yolov8n model Summary

The architecture with 168 layers, possessing 3,007,208 parameters and a computational complexity of 8.1 GFLOPs, can process in real-time and perceive objects falling into the following categories: "қалай?" - "How?", "қанша?" - "How much?" / "How many?", "не?" - "What?", "қашан?" - "When?", "қайда?" - "Where?", "қайсы?" - "Which?", "кім?" - "Who?", "неге?" - "Why?" somewhat lower recall values are recorded, probably evidence that further optimization of the model or growth of the training dataset is required to improve the generalization capability.

The training of the proposed 2DCNN model includes data preparation, neural network architecture and the optimisation process.

Data preparation Videos are converted into single frames. Data enhancement techniques (flipping, rotation, noise addition, brightness adjustment) are applied to increase the data set size and improve the generalisation ability of the model. The frames are then divided into training (80%), validation (10%) and test (10%) sets. The model is then trained on the prepared data with about 4000 images.

2) Testing: Since then, the first Yolov8n model has undergone testing with videos that do not form part of the training dataset. Manual testing reassessed the model for robustness and generalization.

Fig. 7 experiments with the model tested by manually uploaded videos. It will show how this model performs in real-life conditions, what its efficacy is, and it will indicate where further improvement is likely to take place. Fig. 8 shows the testing results of second 2DCNN model with testing accuracy - 98.57%, loss - 0.0572.

3) Model Evaluation: The Yolov8n model works with high precision (mAP50 $\approx$ 99%).

The Recall (R) is very high in all classes, i.e., all interrogative gestures were identified correctly.

Some interrogative gestures (e.g., "қайсы?" and "неге?") were recognized with 100% accuracy and sensitivity.

Fig. 7. Results of Yolov8n model testing [74]

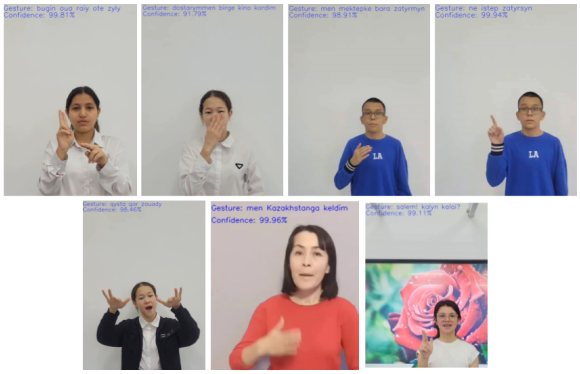

Fig. 8. Results of 2DCNN model testing.

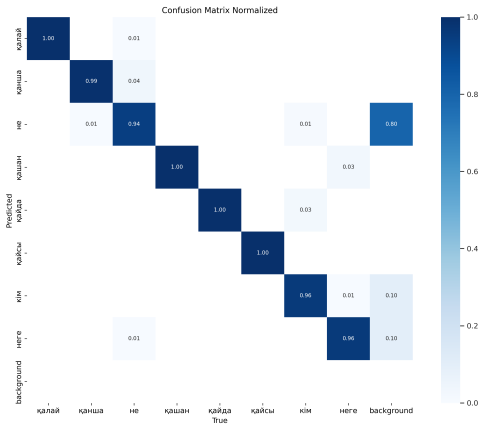

Fig. 9. Results of Yolov8n model confusion matrix

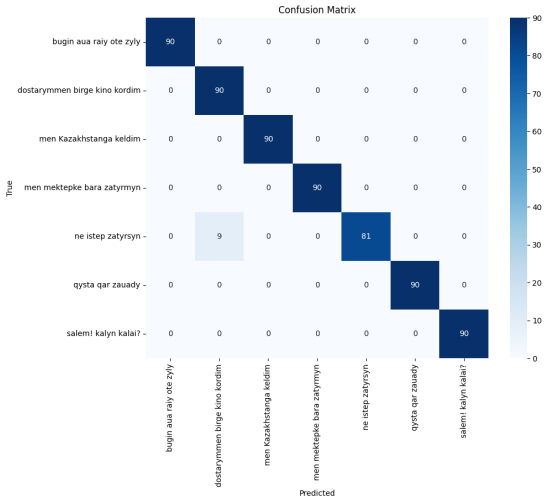

Fig. 10. Results of 2DCNN model confusion matrix.

Fig. 11. Real-time recognition results of eight KazSL question gestures

This shows that the YOLO model performs well and is appropriate for real-time application. The confusion matrix shows, that (Fig. 9).

The 2DCNN model performs well for most phrases. The majority of phrases are classified correctly, except for "ne istep zatyrsyn", which was misclassified 9 times as "dostarymmen birge kino kordim" (Fig. 10).

IV. Results and Discussion

A. Real-Time Recognition

The results for the real-time recognition of eight classes in YOLOv8n-based KazSL are shown below in Fig. 11. The confidence scores for the real-time detected signs ranged from 0.83 to 0.98, demonstrating strong model performance, though some challenges appeared. Variations in the background affected the recognition process, and the model had to identify signs from a tester who was not included in the training dataset, adding a layer of variability. Despite these challenges, the model demonstrated robust performance, highlighting its applicability for real-time sign language recognition.

The YOLOv8n model shows by high confidence values and an accuracy rate of 98.12%. The evaluation metrics indicate the effectiveness of the model. Nevertheless, the limited diversity of the dataset and the problems with complicated backgrounds indicate the need for further improvements. The optimized 2DCNN model of the Kazakh sign language recognition system is shown in Fig. 12.

B. Comparison by Other Models

The performance of the YOLO models was evaluated by using a confusion matrix for eight classes. omparison with earlier studies must be done. S. Tyagi et al. achieved an accuracy of 93.6% (Yolov5) and 96% (Yolov8), B. Steven et al. achieved an accuracy of 80% with Yolov8, B. Alsharif, et al. achieved map50-95 93% with Yolov8, while proposed Yolov8n method gives 98.12% accuracy (Table VII). The performance of 2DCNN significantly higher as 98.57% than other results. The accuracy was calculated by the following formula:

$$\text{Accuracy} = \frac{\text{True Positives}}{\text{Total Samples}} \times 100\% \quad (27)$$

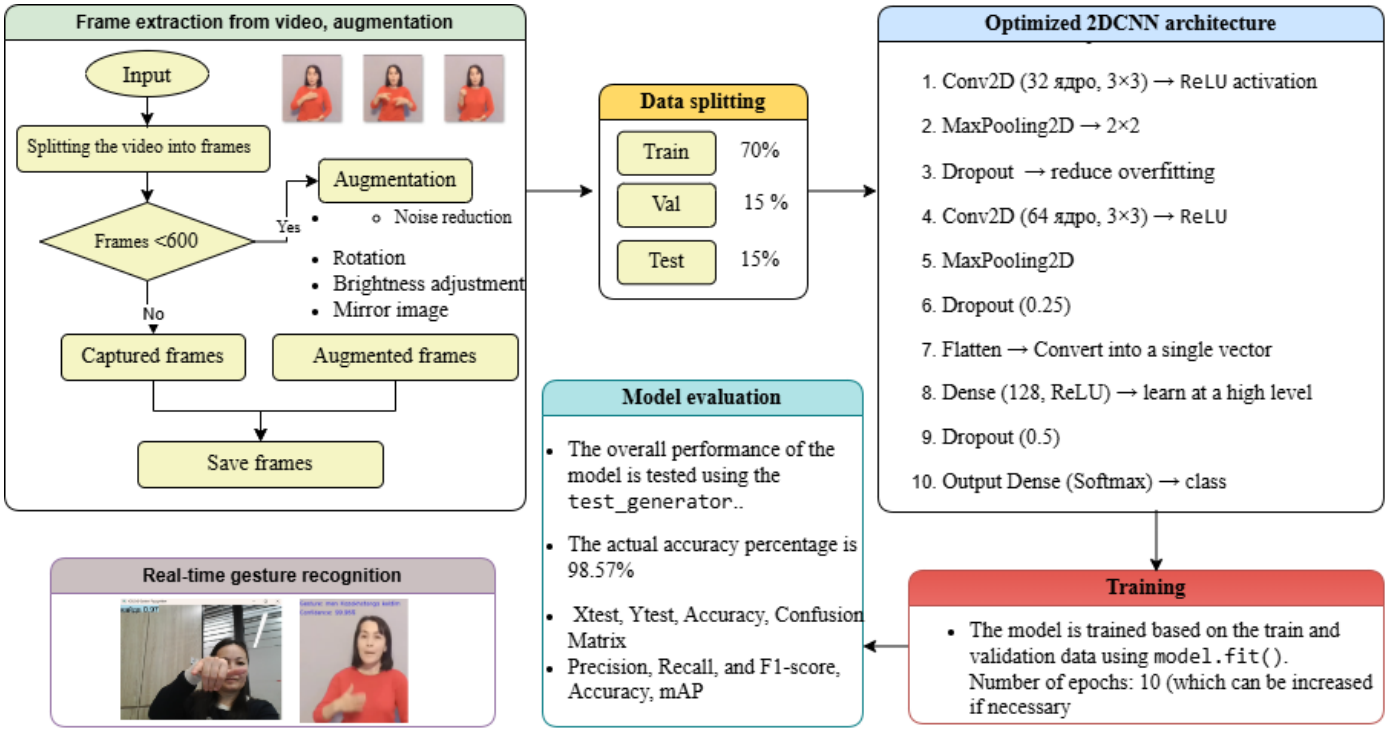

Fig. 12. Optimized 2DCNN Model algorithm

TABLE VII. COMPARISON THE PROPOSED MODEL WITH OTHER WORKS

Models	Authors	Dataset	Samples	Accuracy
Yolov5, Yolov8n	Tyagi et al. [79]	ASL	1733 images	93.6%, 96%
Yolov8	Steven et al. [80]	64	3x3	80%
Yolov8n	Alsharif, et al. [81]	128	3x3	93%
2DCNN, LSTM1024	Amangeldy et al. [82]	KSL	Sign words, sentences	85-91%
CNN+LSTM	Wang et al. [83]	LSA64	64 hand gestures	97%
CNN, LSTM	Noor et al. [84]	ArSL	20 different words	94.40%, 82.70%
CNN, LSTM, Yolov5	Zholshiyeva et al. [85]	QazSL	42 dactyls	98.87%, 85% , 99.98%
Yolov8n	Proposed	K-RSL	8 Question words	98.12%
2DCNN	Proposed	QazSL	Different sentences	98.57%

By comparing YOLOv5 and YOLOv8n models, it is noticeable that the last model has highly accurate results for distinguishing most Kazakh question words. YOLOv8n misclassified a few “кім”, “неге” and “не” words, while the oldest version struggles with “кім”, “не” and “қанша” words. Both models had failed some instances and were classified as the background category. The majority of validation dataset samples are correctly classified along the main diagonal for YOLOv8n, while the YOLOv5 shows fewer correct classifications on the main diagonal, demonstrating a higher rate of misclassification (Fig. 13).

The training curves show a steady decrease in box, classification, and DFL losses, suggesting effective learning. Precision, recall and mAP50-95 reach high values (Fig. 13). YOLOv5 training plots show decreases in losses (box_loss, obj_loss, cls_loss) and improvements in metrics (Precision, Recall, mAP), indicating model improvement. However, Loss and Recall may point to instability or detection issues. Overall, an improving trend is observed (Fig. 14).

C. Evaluation

The performance evaluation of the YOLOv8n model for real-time recognition of KazSL question words shows robust results, with high precision, recall, and F1 scores for all eight classes. According to the recall-confidence and precision-confidence curves, robust accuracy is maintained even at higher confidence thresholds. However, the precision-recall curve demonstrates a near-perfect balance between precision and recall with an mAP@0.5 value of 0.994.

The F1-confidence curve shows that the maximum performance for the optimal value of confidence threshold is around

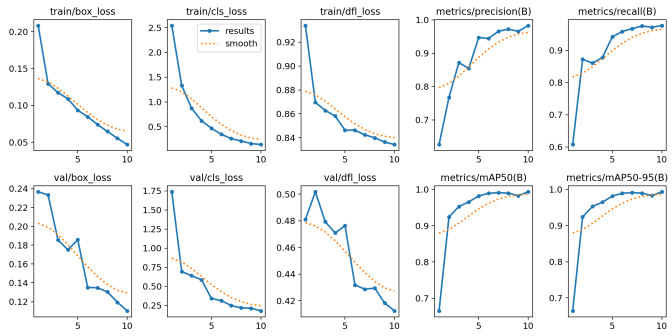

Fig. 13. Learning curves of Yolov8n

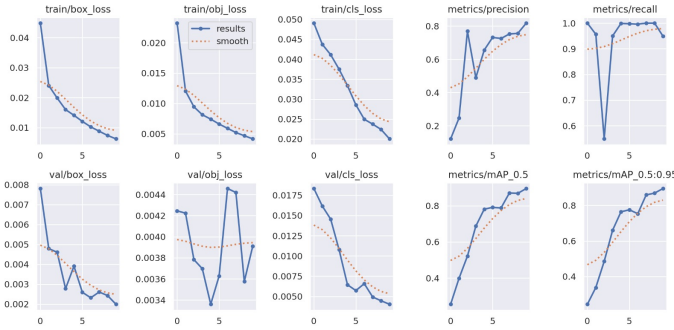

Fig. 14. Learning curves of Yolov5

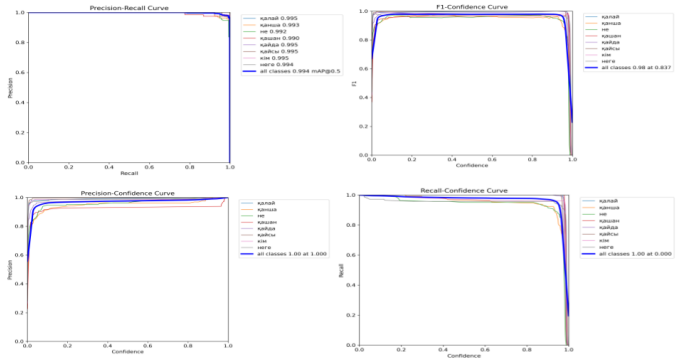

Fig. 15. Evaluation metrics of proposed model

0.83-0.85, where the F1 score achieves a value of 0.98 (Fig. 15). According to these metrics, the model is reliable in recognition applications in real-time since the difference in classification across words is minor.

This research highlighted the use of YOLOv8n and 2DCNN models for the identification of KazSL as effective. The findings reveal a high level of accuracy for the project in real time. But it is also significant to express some of the model's shortcomings.

The application of YOLOv8n has reached a balance between performance and accuracy, allowing this model to perform real-time recognition. In handling complex backgrounds, limitations are further required for algorithm improvement or an increase in the size of the training dataset. Fig. 11 shows the experiment results with the model tested by manually uploaded videos. That would show how the model performs in real-life conditions and what its efficacy is, which can indicate where further improvement is likely to take place. The YOLOv8n model was the one which effectively recognized question words at high confidence level. The questions about errors in complex background conditions were particularly discussing the necessity of system stability improvement in applying it in real time. As a countermeasure, the plans for the future include the improvement of video filtering, the application of adaptive thresholds, and the implementation of hybrid learning approaches.

The smallest data size is one of the main disadvantages of the study. The database consists of 400 and 4000 annotated videos, most of which were recorded in a controlled situation. While the current model provides some stability, it does not fully account for real-time lighting conditions, ambient noise, and variations among signers. Future work will therefore focus on expanding the dataset and recruiting more participants.

Practical Applicability and Model Generalization Evaluation

To assess the model's practical usability, tests were conducted on manually uploaded videos. The results indicated that while the model performs well under controlled conditions, its performance declines in uncontrolled environments.

Comparison with Other Research

Compared to previous studies, this research achieved an accuracy of 98%, marking an improvement. The use of the YOLOv8n model contributed to better performance than YOLOv5 and traditional CNN methods.

Major Findings

- The model achieved a mean average precision (mAP50) of 99%, confirming its reliability.
- A high recall rate was observed across all classes, demonstrating the model's ability to accurately recognize various signs.
- Certain characters, such as ``which" and ``why," were recognized with 100% accuracy, indicating clear differentiation.
- Complex backgrounds and challenging lighting conditions led to some errors.

Future Directions and Recommendations

To further enhance the model's effectiveness and practical usability, the following improvements are suggested:

- Expanding the dataset to include diverse backgrounds and a wider range of signers.
- Implementing advanced techniques to handle complex background conditions.
- Exploring hybrid models to improve recognition efficiency.

By adopting these recommendations, KazSL recognition technology can be further developed to improve communication accessibility for individuals with hearing loss.

V. Conclusion

The experimental results confirmed the efficiency of YOLOv8n in sentence recognition and the accuracy of optimized 2DCNN in sentence classification.

The proposed models is characterized by high accuracy, low latency, and minimal computational resource requirements, making it effective for use in real-time recognition systems.

Future research should focus on the application of transfer learning techniques or hybrid approaches could improve the adaptability of the models to different sign languages.

Finally, the integration of the developed technologies into real-world applications remains a crucial aspect. Potential applications in educational and communication tools for people with hearing impairments could significantly improve accessibility and usability.

Overall, this study lays the foundation for further advances in sign language recognition, promoting inclusivity and expanding technological opportunities for people with hearing impairments.

References

[1] K. Emmorey, "Ten Things You Should Know About Sign Languages," *Current Directions in Psychological Science*, vol. 32, no. 5, pp. 387394, 2023, doi: 10.1177/09637214231173071.

[2] F. M. Najib, "Sign language interpretation using machine learning and artificial intelligence," *Neural Computing and Applications*, vol. 37, no. 2, pp. 841857, 2024, doi: 10.1007/s00521-024-10395-9.

[3] L. Zulpukharkyzy Zholshiyeva, T. Kokenovna Zhukabayeva, S.Turaev, M. Aimambetovna Berdiyeva, and D. Tokhtasynovna Jambulova, "Hand Gesture Recognition Methods and Applications: A Literature Survey, *The 7th International Conference on Engineering*, pp. 18, 2021, doi: 10.1145/3492547.3492578.

[4] Y. Zhang, "National Institute on Deafness and Other Communication Disorders (NIDCD)," *Encyclopedia of Global Health*, vol. 4, pp. 1196–1196, 2008, doi: 10.4135/9781412963855.n840.

[5] D. Ferri, I. Tekuchova, and E. Krolla, "Between disability and culture: The search for a legal taxonomy of sign languages in the European Union," *International and Comparative Law Quarterly*, vol. 73, no. 3, pp. 669706, 2024, doi: 10.1017/s0020589324000253.

[6] R. K. Attar, V. Goyal, and L. Goyal, "State of the Art of Automation in Sign Language: A Systematic Review," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 4, pp. 180, 2023, doi: 10.1145/3564769.

[7] Y. Chen, S. Wang, L. Lin, Z. Cui, and Y. Zong, "Computer Vision and Deep Learning Transforming Image Recognition and Beyond," *International Journal of Computer Science and Information Technology*, vol. 2, no. 1, pp. 4551, 2024, doi: 10.62051/ijcsit.v2n1.06.

[8] A. Alayed, "Machine Learning and Deep Learning Approaches for Arabic Sign Language Recognition: A Decade Systematic Literature Review," *Sensors*, vol. 24, no. 23, 2024, doi: 10.3390/s24237798.

[9] K. Dakhare, V. Wankhede and P. Verma, "A Survey on Recognition and Translation System of Real-Time Sign Language," *2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI)*, pp. 1-6, 2024, doi: 10.1109/IDICAIEI61867.2024.10842738.

[10] S. Shanmugam and R. S. Narayanan, "An accurate estimation of hand gestures using optimal modified convolutional neural network," *Expert Systems with Applications*, vol. 249, 2024, doi: 10.1016/j.eswa.2024.123351.

[11] A. O. Hashi, S. Z. M. Hashim, and A. B. Asamah, "A Systematic Review of Hand Gesture Recognition: An Update From 2018 to 2024," *IEEE Access*, vol. 12, pp. 143599143626, 2024, doi: 10.1109/access.2024.3421992.

[12] P. Agrawal, R. Bose, G. K. Gupta, G. Kaur, S. Paliwal, and A. Raut, "Advancements in Computer Vision: A Comprehensive Review," *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)*, pp. 16, 2024, doi: 10.1109/icicet59348.2024.10616321.

[13] M. . Gündüz and G. Ik, "A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models," *Journal of Real-Time Image Processing*, vol. 20, no. 1, 2023, doi: 10.1007/s11554-023-01276-w.

[14] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 92439275, 2022, doi: 10.1007/s11042-022-13644-y.

[15] H. Bhuiyan, M.F. Mozumder, Md.R.I. Khan, Md. S. Ahmed, N.Z. Nahim, "Enhancing Bidirectional Sign Language Communication: Integrating YOLOv8 and NLP for Real-Time Gesture Recognition," *Translation*, 2024, doi: 10.48550/arxiv.2411.13597.

[16] S. Verma, V. Chandok, A. Gupte, S. S. Badhiye, P. Borkar and P. K. Agrawal, "Unlocking Communication: YOLO Sign Language Detection System," *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1-7, 2024, doi: 10.1109/ICCUBEA61740.2024.10775264.

[17] M. T. Patel, P. S. Kumar, A. R. De, and S. V. Raghavan, "YOLO Convolutional Neural Network Algorithm for Recognition of Indian Sign Language Gestures," *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pp. 18, 2023, doi: 10.1109/ACCAI58221.2023.10200524.

[18] T. T. H. Vu, D. L. Pham, and T. W. Chang, "A YOLO-based Real-time Packaging Defect Detection System," *Procedia Computer Science*, vol. 217, pp. 886894, 2023, doi: 10.1016/j.procs.2022.12.285.

[19] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.

[20] N. N. Herbaz, H. El Idrissi, and A. Badri, "Deep Learning Empowered Hand Gesture Recognition: Using YOLO Techniques," *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 2023, doi: 10.1109/sita60746.2023.10373734.

[21] A. Serek, A. Issabek, A. Akhmetov, and A. Sattarbek, "Part-of-speech tagging of Kazakh text via LSTM network with a bidirectional modifier," *2021 16th International Conference on Electronics, Computer and Computation (ICECCO)*, pp. 16, 2021, doi: 10.1109/ICECCO53745.2021.9774003.

[22] S. N. A. Mahmoud, A. Yousif, and M. H. Hassanein, "A Comparative Analysis of Machine Learning Algorithms for Sign Language Recognition," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 122130, 2021, doi: 10.14569/IJACSA.2021.0120717.

[23] J. S. Smith and R. K. Lee, "Deep Learning-based Sign Language Translation: A Survey," *Artificial Intelligence Review*, vol. 56, pp. 25672592, 2023, doi: 10.1007/s10462-023-10234-5.

[24] H. Chen, Y. Liu, and X. Zhang, "A Comprehensive Review on Hand Gesture Recognition Techniques Based on Computer Vision," *Pattern Recognition*, vol. 134, 2023, doi: 10.1016/j.patcog.2023.108655.

[25] M. Moradi, D. D. Kannan, S. Asadianfam, H. Kolivand and O. Aldhaibani, "A Review of Sign Language Systems," *2023 16th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 200-205, 2023, doi: 10.1109/DeSE60595.2023.10468964.

[26] N. Amangeldy, I. Krak, B. Kurmetbek, N. Gazizova, "A Comparison of the Effectiveness Architectures LSTM1024 and 2DCNN for Continuous Sign Language Recognition Process, *In Seventh International Workshop on Computer Modeling and Intelligent Systems*, vol. 3702, 2024.

[27] S. Alyami, H. Luqman, and M. Hammoudeh, "Reviewing 25 years of continuous sign language recognition research: Advances, challenges, and prospects, *Information Processing Management*, vol. 61, no. 5, 2024, doi: 10.1016/j.ipm.2024.103774.

[28] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif and M. A. Mekhtiche, "Hand Gesture Recognition for Sign Language Using 3DCNN," *in IEEE Access*, vol. 8, pp. 79491-79509, 2020, doi: 10.1109/ACCESS.2020.2990434.

[29] R. Chen, X. Tian, "Gesture Detection and Recognition Based on Object Detection in Complex Background," *Applied Sciences*, vol. 13, no. 7, 2023, https://doi.org/10.3390/app13074480

[30] S. K. Hussein, A. S. Ahmed, Z. Kul and A. M. Ashir, "Real- Time Hand Gesture Recognition for Home Automation: A YOLOv8-Based Approach with Identity Verification and Low-Resource Hardware Implementation," *2024 21st International Multi-Conference on Systems, Signals Devices (SSD)*, pp. 340-348, 2024, doi: 10.1109/SSD61670.2024.10548453.

[31] X. Wang and P. Wang, "Research and Analysis of Gesture Recognition Experimentation Based on YOLOv5, *2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 706711, 2024, doi: 10.1109/itnec60942.2024.10732916.

[32] M. Kang, C. M. Ting, F. F. Ting, and R. C. W. Phan, "ASF-YOLO: A novel YOLO model with attentional scale sequence fusion for cell instance segmentation, *Image and Vision Computing*, vol. 147, 2024, doi: 10.1016/j.imavis.2024.105057.

[33] A. Moryossef, Z. Jiang, M. Müller, S. Ebling, and Y. Goldberg, "Linguistically Motivated Sign Language Segmentation, *Findings of the Association for Computational Linguistics: EMNLP*, pp. 1270312724, 2023, doi: 10.18653/v1/2023.findings-emnlp.846.

[34] J. Wu, "A New Sign Language Translation System Based on Expert Model, *Applied and Computational Engineering*, vol. 81, no. 1, pp. 210218, 2024, doi: 10.54254/2755-2721/81/20241162.

[35] S. Feng and T. Yuan, "Sign language translation based on new continuous sign language dataset, *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2022, doi: 10.1109/icaica54878.2022.9844468.

[36] A. N. manov and A. B. Aben, "Investigation and Application of Various Algorithms used in Object Detection and Classification in Image Data, *A Iasaý atynday Halyqaralyq qazaq-túrk ýnverstetn habarlary (fzka matematka nformatka serasy)*, vol. 30, no. 3, pp. 6274, 2024, doi: 10.47526/2024-3/2524-0080.10.

[37] B. K. Pratama, Sri Lestanti, and Yusniarsi Primasari, "Implementasi Algoritma You Only Look Once (YOLO) untuk Mendeteksi Bahasa Isyarat SIBI, *ProTekInfo(Pengembangan Riset dan Observasi Teknik Informatika)*, vol. 11, no. 2, pp. 714, 2024, doi: 10.30656/protekinfo.v11i2.9105.

[38] N. Herbaz, H. El Idrissi and A. Badri, "Deep Learning Empowered Hand Gesture Recognition: using YOLO Techniques," *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1-7, 2023, doi: 10.1109/SITA60746.2023.10373734.

[39] A. Imran, M.S. Hulikal, H.A.A. Gardi, "Real Time American Sign Language Detection Using Yolo-v9," *arXiv.Org*, 2024, doi: 10.48550/arxiv.2407.17950

[40] B. Steven, C. Vannia Nathalie, C. Gerry Johanes, and P. Andam Suri, "Comparison Research: YOLO (You Only Look Once) Model for Indonesian Sign Language Detection Reducing Communication Inequalities, *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)*, pp. 16, 2024, doi: 10.1109/ictiia61827.2024.10761360.

[41] P. Patel, S. Pampaniya, A. Ghosh, R. Raj, D. Karuppaih and S. Kandasamy, "Enhancing Accessibility Through Machine Learning: A Review on Visual and Hearing Impairment Technologies," in *IEEE Access*, vol. 13, pp. 33286-33307, 2025, doi: 10.1109/ACCESS.2025.3539081.

[42] O. K. T. Alsultan and M. T. Mohammad, "A Deep Learning-Based Assistive System for the Visually Impaired Using YOLO-V7," *Revue d'Intelligence Artificielle*, vol. 37, no. 4, pp. 901906, 2023, doi: 10.18280/ria.370409.

[43] K. J. Jaiswal, A. Khan, H. Budhdev, and S. Gandhi, "Deep Learning-Driven Sign Language Recognition: A Multimodal Approach for Gesture-to-Text Translation Using CNN-RNN Architectures," *International Journal of Advanced Research in Science, Communication and Technology*, 2024, doi: 10.48175/ijarsct-19946.

[44] C. M. N. Kumar, A. Vanitha, N. Y. Lavanya, N. C. Lekhana, R. Tasmiya, and L. D. Nisarga, "Deep Learning-Based Recognition of Sign Language," *2024 Second International Conference on Data Science and Information System (ICDSIS)*, pp. 16, 2024, doi: 10.1109/ICDSIS61070.2024.10594011.

[45] H. Bhuiyan, M. F. Mozumder, Md. R. I. Khan, Md. S. Ahmed, and N. Z. Nahim, "Enhancing Bidirectional Sign Language Communication: Integrating YOLOv8 and NLP for Real-Time Gesture Recognition and Translation," *arXiv*, 2024, doi: 10.48550/arxiv.2411.13597.

[46] M. Alaftekin, I. Pacal, and K. Cicek, "Real-time sign language recognition based on YOLO algorithm," *Neural Computing and Applications*, vol. 36, pp. 7609–7624, 2024, doi: 10.1007/s00521-024-09503-6.

[47] D. Fan, M. Yi, W. Kang, Y. Wang, and C. Lv, "Continuous sign language recognition algorithm based on object detection and variable-length coding sequence, *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-78319-0.

[48] S. Ahmed, S. R. Revolinski, P. W. Maughan, M. Savic, J. Kalin, and I. C. Burke, "Deep learningbased detection and quantification of weed seed mixtures, *Weed Science*, vol. 72, no. 6, pp. 655663, 2024, doi: 10.1017/wsc.2024.60.

[49] M. Maruyama, S. Singh, K. Inoue, P. Pratim Roy, M. Iwamura, and M. Yoshioka, "Word-Level Sign Language Recognition With Multi-Stream Neural Networks Focusing on Local Regions and Skeletal Information, *IEEE Access*, vol. 12, pp. 167333167346, 2024, doi: 10.1109/access.2024.3494878.

[50] H. ZainEldin *et al*., "Silent no more: a comprehensive review of artificial intelligence, deep learning, and machine learning in facilitating deaf and mute communication, *Artificial Intelligence Review*, vol. 57, no. 7, 2024, doi: 10.1007/s10462-024-10816-0.

[51] J. Shin, A. S. M. Miah, K. Suzuki, K. Hirooka and M. A. M. Hasan, "Dynamic Korean Sign Language Recognition Using Pose Estimation Based and Attention-Based Neural Network," in *IEEE Access*, vol. 11, pp. 143501-143513, 2023, doi: 10.1109/ACCESS.2023.3343404.

[52] S. Parab and Mr. C. Bhattacharjee, "Empowering Accessibility: Bridging Communication Gap through Sign Language Detection Systems using Convolution Neural Network, *International Journal for Research in Applied Science and Engineering Technology*, vol. 13, no. 1, pp. 712, 2025, doi: 10.22214/ijraset.2025.66011.

[53] S. Renjith and R. Manazhy, "Real Time Recognition of ISL by Time Distributed CNN model using ISL video dataset, *Research Square*, 2023, doi: 10.21203/rs.3.rs-3046559/v1.

[54] D. Ezra, S. Mastitz, and I. Rabaev, "Signsability: Enhancing Communication through a Sign Language App, *Software*, vol. 3, no. 3, pp. 368379, 2024, doi: 10.3390/software3030019.

[55] Prof. V. M. Dilpak, Rewa S. Joshi, and Harshada K. Sonje, "SignSense: AI Framework for Sign Language Recognition, *International Journal of Advanced Research in Science, Communication and Technology*, pp. 372385, 2024, doi: 10.48175/ijarsct-17257.

[56] A. E. M. Ridwan *et al*., "Network-Based Sign Language Recognition: A Comprehensive Approach Using Transfer Learning with Explainability," *arXiv*, 2024, doi: 10.48550/arxiv.2409.07426.

[57] S. J, S. R. M, M. Vespa M and K. R, "Sign Language Translation and Voice Impairment Support System using Deep Learning," *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, pp. 558-563, 2024, doi: 10.1109/ICDICI62993.2024.10810957.

[58] S. Saxena, A. Paygude, P. Jain, A. Memon, and V. Naik, "Hand Gesture Recognition using YOLO Models for Hearing and Speech Impaired People, *2022 IEEE Students Conference on Engineering and Systems (SCES)*, 2022, doi: 10.1109/sces55490.2022.9887751.

[59] S. Al Ahmadi, F. Mohammad, and H. Al Dawsari, "Efficient YOLO-Based Deep Learning Model for Arabic Sign Language Recognition, *Journal of Disability Research*, vol. 3, no. 4, 2024, doi: 10.57197/jdr-2024-0051.

[60] H. J. Bhuiyan *et al*., "Enhancing Bidirectional Sign Language Communication: Integrating YOLOv8 and NLP for Real-Time Gesture Recognition Translation," *arXiv*, 2024, doi: 10.48550/arXiv.2411.13597.

[61] A. Munandar, Z. Yunizar, and S. Retno, "Indonesian Sign Language (BISINDO) Alphabet Detection System Using YOLO (You Only Look Once) Algorithm, *Proceedings of Malikussaleh International Conference on Multidisciplinary Studies (MICoMS)*, vol. 4, 2024, doi: 10.29103/micoms.v4i.952.

[62] S. Ghodke, A. Jadhav, R. Kakde, P. Navgare, and S. Khiani, "Sign Language Detection for Deaf and Hard of Hearing (DHH) Community, *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 15, 2024, doi: 10.1109/iccubea61740.2024.10775143.

[63] A. H. A. Halim, A. A. A. Rahim, N. A. Rozaini, S. L. M. Hassan, I. S. A. Halim, and N. E. Abdullah, "Malaysian Sign Language (MSL)

Detection: Comparison of YOLOv5 and CNN," *2024 IEEE 15th Control and System Graduate Research Colloquium (ICSGRC)*, pp. 2934, 2024, doi: 10.1109/icsgrc62081.2024.10691273.

[64] R. Raj, R. Sreemathy, M. Turuk, J. Jagdale, and M. Anish, "Indian Sign Language Recognition in Real Time using YOLO NAS," *2024 3rd International Conference for Advancement in Technology (ICONAT)*, pp. 18, 2024, doi: 10.1109/iconat61936.2024.10774832.

[65] N. Swapna, P. Shivani, D. Madhav Karthik, and G. Shivateja, "An Effective Real Time Sign Language Recognition using Yolo Algorithm, *SSRN Electronic Journal*, 2025, doi: 10.2139/ssrn.5083947.

[66] U. Jana, S. Paul, and D. Bhandari, "Real-Time Caption Generation for the American Sign Language Using YOLO and LSTM,"*2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, pp. 14, 2024, doi: 10.1109/iciteics61368.2024.10625098.

[67] M. Mukushev, *et. al*., "Evaluation of Manual and Non-manual Components for Sign Language Recognition, " *European Language Resources Association (ELRA)*, pp 6075-6080, 2020.

[68] C. Kenshimov, S. Mukhanov, T. Merembayev, and D. Yedilkhan, "A comparison of convolutional neural networks for Kazakh sign language recognition," *Eastern-European Journal of Enterprise Technologies*, vol. 5, no. 2, pp. 4454, 2021, doi: 10.15587/1729-4061.2021.241535.

[69] L. Zholshiyeva, T. Zhukabayeva, Sh. Turaev, M. Berdieva, and R. Sengirbayeva, "Real-time Kazakh Sign Language using MEDIAPIPE and SVM, *News of the National Academy of Republic of Kazakhstan. Series physico-mathematical*, vol. 1, no. 345, pp. 8293, 2023, doi: 10.32014/2023.2518-1726.170.

[70] S. Mukhanov *et al*., "Gesture recognition of machine learning and convolutional neural network methods for Kazakh sign language," *Scientific Journal of Astana IT University*, pp. 85100, 2023, doi: 10.37943/15lpcu4095.

[71] C. Kenshimov, Z. Buribayev, Y. Amirgaliyev, A. Ataniyazova, and A. Aitimov, "Sign language dactyl recognition based on machine learning algorithms," *Eastern-European Journal of Enterprise Technologies*, vol. 4, no. 2, pp. 5872, 2021, doi: 10.15587/1729-4061.2021.239253.

[72] N. Amangeldy, A. Ukenova, G. Bekmanova, B. Razakhova, M. Milosz, and S. Kudubayeva, "Continuous Sign Language Recognition and Its Translation into Intonation-Colored Speech," *Sensors*, vol. 23, no. 14, 2023, doi: 10.3390/s23146383.

[73] Y. Amirgaliyev, A. Ataniyazova, Z. Buribayev, M. Zhassuzak, B. Urmashev, and L. Cherikbayeva, "Application of neural networks ensemble method for the Kazakh sign language recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 5, pp. 32753287, 2024, doi: 10.11591/eei.v13i5.7803.

[74] https://krslproject.github.io/krsl20/

[75] https://habr.com/ru/articles/710016/

[76] M. Bakirci, "Real-Time Vehicle Detection Using YOLOv8-Nano for Intelligent Transportation Systems," *Traitement du Signal*, vol. 41, no. 04, pp. 17271740, 2024, doi: 10.18280/ts.410407.

[77] W. Fang and W. Chen, "TBFYOLOv8n: A Lightweight Tea Bud Detection Model Based on YOLOv8n Improvements," *Sensors*, vol. 25, no. 2, 2025, doi: 10.3390/s25020547.

[78] M. Yue, L. Zhang, J. Huang, and H. Zhang, "Lightweight and Efficient Tiny-Object Detection Based on Improved YOLOv8n for UAV Aerial Images," *Drones*, vol. 8, no. 7, 2024, doi: 10.3390/drones8070276.

[79] S. Tyagi, P. Upadhyay, H. Fatima, S. Jain, and A. K. Sharma, American Sign Language Detection using YOLOv5 and YOLOv8, *Research Square*, 2023, doi: 10.21203/rs.3.rs-3126918/v1.

[80] B. Steven, C. Vannia Nathalie, C. Gerry Johanes and P. Andam Suri, "Comparison Research: YOLO (You Only Look Once) Model for Indonesian Sign Language Detection Reducing Communication Inequalities," *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)*, pp. 1-6, 2024, doi: 10.1109/ICTIIA61827.2024.10761360.

[81] B. Alsharif, E. Alalwany, and M. Ilyas, "Transfer learning with YOLOV8 for real-time recognition system of American Sign Language Alphabet, *Franklin Open*, vol. 8, 2024, doi: 10.1016/j.fraope.2024.100165.

[82] N. Amangeldy, I., Krak, B. Kurmetbek, N. Gazizova, "A Comparison of the Effectiveness Architectures LSTM1024 and 2DCNN for Continuous Sign Language Recognition Process, " *Seventh International Workshop on Computer Modeling and Intelligent Systems*, 2024.

[83] J. Wang, "Isolated Sign Language Recognition Based on Deep Learning," *In Electronic Engineering and Informatics, IOS Press*, pp. 263-272, 2024.

[84] T. H. Noor *et al*., "Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model, *Sensors*, vol. 24, no. 11, 2024, doi: 10.3390/s24113683.

[85] L. Zholshiyeva, T. Zhukabayeva, D. Baumuratova, A. Serek, "Design of QazSL Sign Language Recognition System for Physically Impaired Individuals, " *Journal of Robotics and Control (JRC)*, vol. 6 no. 1, pp. 191-201, 2025, doi: 10.18196/jrc.v6i1.23879.