




Efficient YOLO-Based Deep Learning Model for Arabic Sign Language Recognition

Saad Al Ahmadi¹, Farah Mohammad^{1,*}  and Haya Al Dawsari¹

¹Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia[✉]

Correspondence to:
Farah Mohammad*, e-mail: fnazar@ieee.org, Tel.: +1 619 374 0127

Received: February 4 2024; Revised: April 19 2024; Accepted: April 20 2024; Published Online: May 7 2024

ABSTRACT

Verbal communication is the dominant form of self-expression and interpersonal communication. Speech is a considerable obstacle for individuals with disabilities, including those who are deaf, hard of hearing, mute, and nonverbal. Sign language is a complex system of gestures and visual signs facilitating individual communication. With the help of artificial intelligence, the hearing and the deaf can communicate more easily. Automatic detection and recognition of sign language is a complex and challenging task in computer vision and machine learning. This paper proposes a novel technique using deep learning to recognize the Arabic Sign Language (ArSL) accurately. The proposed method relies on advanced attention mechanisms and convolutional neural network architecture integrated with a robust You Only Look Once (YOLO) object detection model that improves the detection and recognition rate of the proposed technique. In our proposed method, we integrate the self-attention block, channel attention module, spatial attention module, and cross-convolution module into feature processing for accurate detection. The recognition accuracy of our method is significantly improved, with a higher detection rate of 99%. The methodology outperformed conventional methods, achieving a precision rate of 0.9 and a mean average precision (mAP) of 0.9909 at an intersection over union (IoU) of 0.5. From IoU thresholds of 0.5 to 0.95, the mAP continuously remains high, indicating its effectiveness in accurately identifying signs at different precision levels. The results show the model's robustness in accurately detecting and classifying complex multiple ArSL signs. The results show the robustness and efficacy of the proposed model.

KEYWORDS

sign language, deep learning, YOLO, sign language detection, ArSL

INTRODUCTION

Individuals with disabilities and underrepresented minority populations have faced enduring societal marginalization. Despite notable progress in integrating those with hearing impairments into society, there still exists a persistent barrier to properly connecting with other community members. The predominant means of communication within many deaf communities is the sign language (Leigh et al., 2022; Alyami et al., 2023). Sign language facilitates communication using manual gestures, oral movements, bodily positions, and facial cues. Each symbol can denote a single letter, a numerical value, or even an entire expression. Numerous sign languages exist globally; however, their quantity remains lower than spoken languages (Strobel et al., 2023a, b). Like other languages, sign languages continuously evolve and adhere to linguistic laws. However, they do not possess standardized written forms. Sign languages and spoken languages are fundamentally distinct. American Sign Language (ASL) does not precisely represent spoken American language. Many individuals with normal hearing lack interest in acquiring sign language skills, presenting difficulties communicating with sign language users. Deaf individuals face an additional

obstacle, namely the lack of support for sign language in most communication devices (Alaghband et al., 2023).

Hence, it is crucial to devise a technological solution that improves communication between individuals with normal hearing and the deaf community. The proposed solution should possess the ability to understand sign language and autonomously convert it into spoken or written text. Prior research on sign language recognition has utilized diverse methodologies. The You Only Look Once (YOLO) technique has excellent potential for sign language recognition. YOLO object identification models are used in several fields, such as surgical procedures for identifying organ locations, driverless vehicles, and detecting face masks. This technology has proven beneficial in various practical scenarios (Sarda et al., 2021; Wang et al., 2021; El-Alfy and Luqman, 2022).

In visual applications (Wu et al., 2022), convolutional neural networks (CNNs) are essential components of models like YOLO, which employ a three-layered methodology. The convolutional layer utilizes filters to extract crucial characteristics from images. By reducing the dimensions of these feature maps, the pooling layer effectively controls

overfitting. The data pass through the fully connected layer, transforming into accurate medical imaging recognition. The remarkable increase in precision and efficiency in diagnostic imaging, achieved by implementing CNN architecture, truly showcases artificial intelligence's powerful influence in visual domains (Mustafa and Nsour, 2023).

In computer vision (CV), adding attention mechanisms to CNNs has proved to be a breakthrough, leading to significantly enhanced performance demonstrated by notable advancements observed over the last decade (Mammeri et al., 2023). CNN attention mechanisms adeptly filter out extraneous details and home-specific targets or regions within intricate visual surroundings, which inspire humans to process visual information. This mimics our instinctual tendency to focus on critical areas when processing visual scenes. In recent years, attention mechanisms have made incredible strides in improving our ability to identify crucial image elements. By dynamically adjusting the importance of various channels and spatial attention mechanisms have significantly enhanced numerous CV applications, such as object detection and image classification. As a result, these tasks can now be performed more efficiently and effectively (Hussain et al., 2020; Ji et al., 2021).

Recent research has enhanced the comprehension of attention mechanisms in CV. These technologies are capable of dynamically and automatically evaluating the importance of data. Two types of attention mechanisms exist: soft and complex (Guo et al., 2022). Soft attention calculates a weighted average to generate the gradient context vector and may be used with traditional backpropagation training. On the other hand, intricate attention relies on reinforcement learning and employs stochastic inputs, which is not differentiable. Significantly, advancements such as HiLo attention have surfaced, effectively handling data with high and low frequencies to provide more refined processing. Nevertheless, defining "attention" in these systems continues to be intricate, with ongoing discussions regarding its essence, particularly when compared to human visual attention (Yang, 2020; Niu et al., 2021).

Due to the ongoing difficulties in communication experienced by people with hearing impairments, there is a crucial requirement for a precise and effective technical solution to facilitate smooth communication among the deaf community and individuals with normal hearing. Our proposed method utilizes state-of-the-art developments in CV and deep learning (DL) to improve sign language identification significantly. Utilizing CV and machine learning (ML) provides deaf individuals with more efficient communication options by automatically converting sign language into spoken or written text. In our proposed method, we utilized the detection model for Arabic Sign Language (ArSL) detection using DL. The significant contributions of our proposed method are as follows:

- The deep CNN-based features extractor has been modified using a self-attention module block.
- The attention module consists of features compression and decompression, with channel and spatial attention modules utilized to enhance feature representation.
- Additionally, a cross-convolution module with a constant vector parameter is used in the feature extractor

for two-way correlated matrices and complex feature derivation.

- Overall, the model is fine-tuned to achieve robust recognition performance.

This paper is organized as follows. The Related Work section provides an extensive review of current techniques in the field. The Proposed Method section looks at the fundamental methodology of ArSL, discussing its main principles. The Results section covers the research's implementation and simulation. The Discussion section comprises the discussion and comparison, while the Conclusion section covers the conclusion of the proposed work.

RELATED WORK

The incorporation of gesture recognition technology in the ArSL domain has represented a notable advancement in enabling communication between those with speech impairments and computer systems (Aly and Aly, 2020; Alnabih and Maghari, 2024). This technological innovation is crucial for identifying and comprehending ArSL, which possesses a distinct repertoire of gestures and facial expressions (Boukdir et al., 2021; Shanableh, 2023). Utilizing these ML, DL, and CV reduces communication barriers and increases their ability to participate in various professional and social settings. Integrating the automatic sign language recognition system into ArSL has many benefits. It also shows the importance of catering technology designed for every country, emphasizing linguistics and culture (Kahlon and Singh, 2023; Renjith et al., 2024).

However, there is a lack of a comprehensive database on only fingerspelling, isolated signs, and continuous signs, which may be a challenging issue for the advancement of sign language recognition technology (Sharma et al., 2023). The sign language recognition system on manual alphabets has an image accuracy rate of 93.55% by using an adaptive neuro-fuzzy inference system (ANFIS) and feature vector extraction (Al-Jarrah and Halawani, 2001) from the ArSL dataset; The ANFIS outperforms the polynomial classifier due to a lack of consistency in the training data, whereas the training data for the ANFIS are more consistent and comprehensive.

Žemgulys et al. (2020) introduced a novel technique to accurately recognize hand signals given by basketball referees from game footage. Our method exploits the performance of image segmentation algorithms and combines the features of histogram of oriented gradients (HOG) and local binary patterns (LBP). By using LBP and a support vector machine, the experimental results showed that the proposed technique can achieve a 95.6% success rate. Vaitkevicius et al. (2019) used a Leap Motion device to accurately track our hand and finger movements. The innovative hidden Markov classification algorithm detects several gestures. Motion detection, gesture recognition, and data cleansing are utilized to assess the system's performance. The presented technique was also validated using words per minute and the miss rate using the minimum string distance.

CNN combined with long short-term memory (LSTM) or bidirectional LSTM has become a popular technique of sign language recognition (Kumari and Anand, 2024). A highly effective method of detecting Indian and Russian sign languages accurately using deep neural networks and CV is proposed in this paper. It can ideally detect the meaning of the languages' manual and non-manual components. The spatial information is extracted by the two-dimensional convolutional neural network (2D-CNN), which results in 92% accuracy by the two-dimensional convolutional recurrent neural network (2D-CRNN) and 99% accuracy by the 3D-CNN (Rajalakshmi et al., 2023). Researchers obtained a 92% accuracy rate by utilizing a 2D-CRNN and a 99% accuracy rate with a 3D-CNN. These models were tested on a dataset consisting of 224 videos, where 5 signers performed 56 distinct signs (Boukdir et al., 2021).

The study conducted by Attia et al. (2023) aims to increase sign language identification accuracy by developing three advanced DL models utilizing YOLO5x with attention module. These models will recognize alphabetic and numeric hand movements by design. The models had 98.9% and 97.6% accuracies on the MU HandImages ASL and OkkhorNama: BdSL datasets, outperforming earlier models. Optimization for real-time ASL recognition makes these models ideal for edge-based solutions. The YOLOv7 algorithm is utilized (Mazen and Ezz-Eldin, 2024) to detect ArSL signs. The YOLOv7 medium model outperformed YOLOv5 variants regarding mean average precision (mAP) ratings. More precisely, the YOLOv7 medium model scored 0.8306 for mAP@0.5:0.95. Additionally, the YOLOv7 tiny model fared better than the YOLOv5 small and medium models. The YOLOv5 tiny model achieved the lowest scores, with an mAP of 0.9408 at an intersection over union (IoU) threshold of 0.5 and an mAP of 0.7661 within the IoU range of 0.5-0.95.

Luqman (2023) presented the ArabSign dataset comprising 9335 video clips from six persons. They also developed an encoder–decoder model for recognizing sign language sentences and achieved an average word error rate of 0.50. Alyami et al. (2023) introduced a transformer model based on stance, tailored explicitly for the KArSL-100 dataset. This dataset consists of 100 classes focused on recognizing sign videos. The model attained a 68.2% accuracy rate while using a signer-independent mode. The techniques entail thorough preprocessing, complex structures, and the utilization of Kinect sensors. Although they demonstrate high performance on tiny datasets, their intricate nature and dependence on sophisticated networks and sensors may constrain their practical implementation.

PROPOSED METHOD

Our presented strategy improves a deep CNN model designed exclusively for recognizing ArSL. Expanding on the YOLO framework, we have improved the main structure and the detection component by utilizing advanced methods like modified layers and attention processes to achieve better performance. The core of our technique lies in including attention modules in the feature extraction process. These modules consist of channel and spatial attention mechanisms, improving feature representation by selectively emphasizing essential parts of the input data. Furthermore, we provide a new cross-convolution module specifically built to efficiently handle matrices with two-way correlation. This module utilizes a shared parameter vector across all components, which enables concentrated convolution operations. The model architecture of ArSL is shown in Figure 1, which includes the backbone and detecting head.

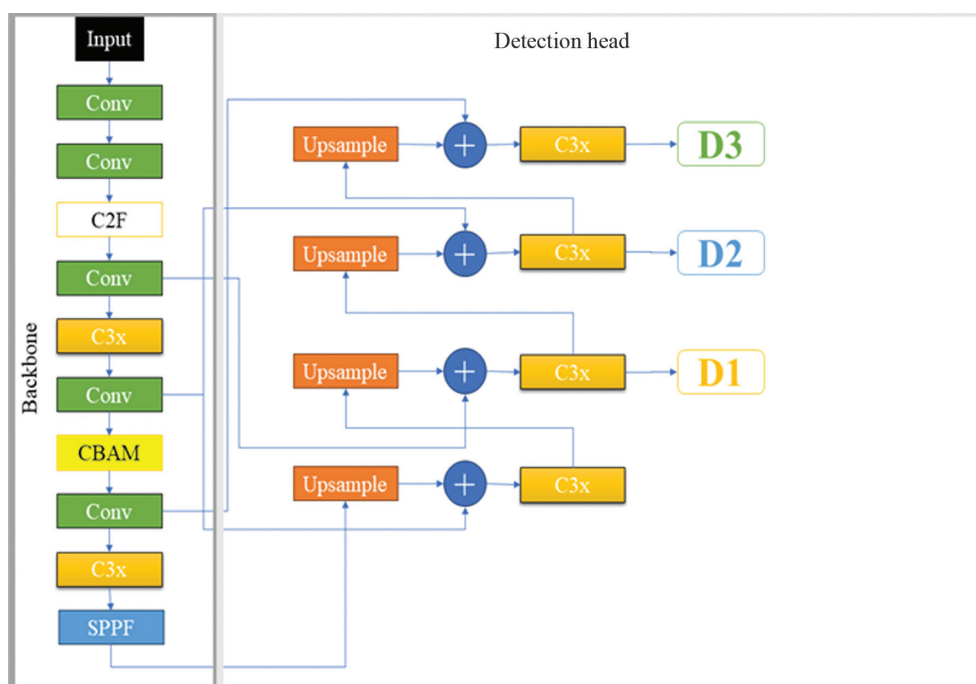


Figure 1: Proposed backbone and detection head. Abbreviations: CBAM, Convolutional Block Attention Module; SPPF, Spatial Pyramid Pooling Fusion.

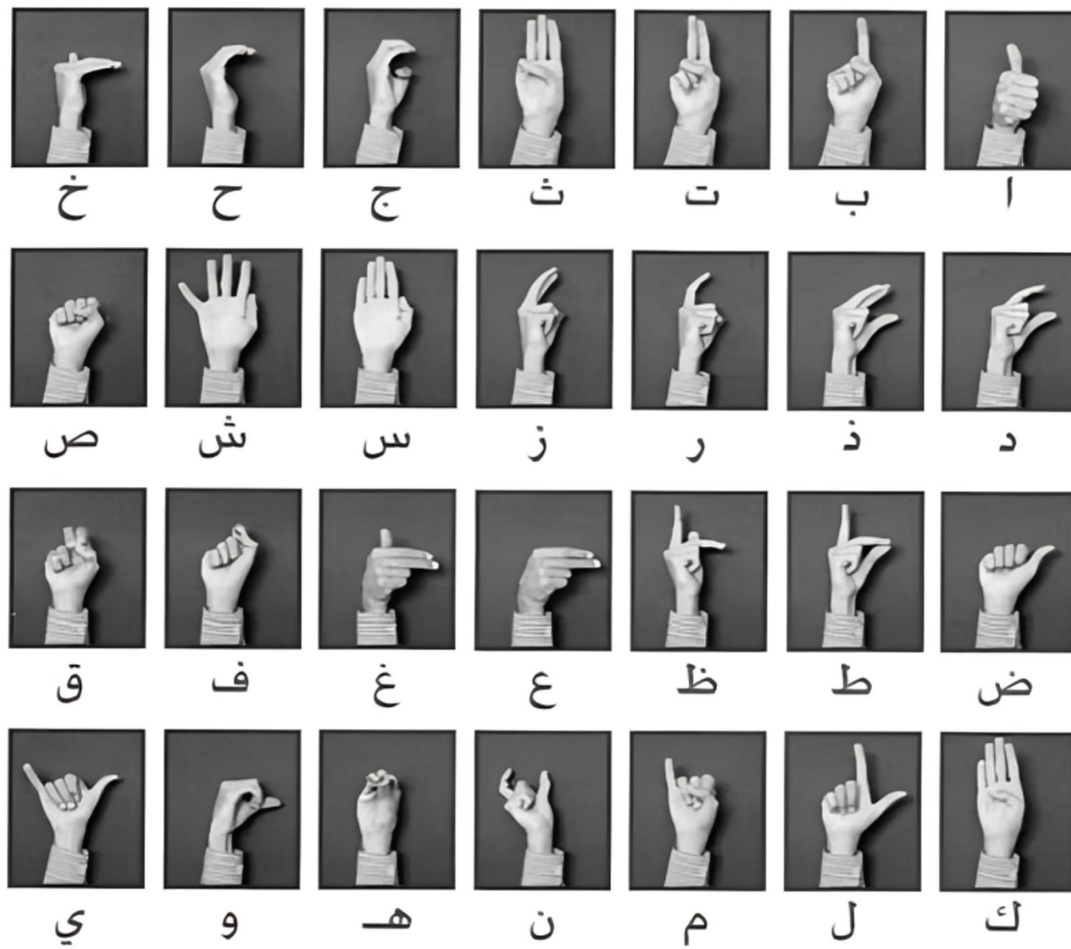


Figure 2: Sample image from the ArSL dataset. Abbreviation: ArSL, Arabic Sign Language.

Dataset

We utilized the Arabic Sign Language Letters dataset (ArSL21L) to validate the robustness of our proposed technique. ArSL is a challenging and benchmark dataset for sign language detection and recognition. The dataset contains 14,202 images, each representing one of the 32 unique letter signs in ArSL. Overall, 52 participants posed with different ArSL signs in different scenarios and conditions. Sample images from the dataset are visualized in Figure 2. All the signs have been captured in different conditions with variable light, angle, and saturation. Different environments make accurate sign detection and recognition a challenging task in the domain of CV. The dataset is publicly accessible and available in Mendeley's repository (Batnasan et al., 2022).

CNN-based detection model

Due to its incredible robustness and accuracy, one particularly prominent method in real-time CV applications is YOLO. This method relies on a neural network to quickly evaluate the input and identify objects. YOLO consumes the input image through a predetermined grid and assesses the chances of the target object that resides in each grid section. Essentially, YOLO performs regression to predict the image categories and positions precisely all at once (Redmon et al., 2016).

An ordinary YOLO model overlays an $s \times s$ grid on the image. Every grid cell predicts B bounding boxes and their confidence ratings, which indicate the likelihood of an object being there. The grid cell that detects an object's center detects it, whereas the other cells can ignore it. This method enhances item detection by precisely locating and classifying items using cell grids and bounding box estimations. The confidence score of the predicted is expressed in Eq. 1.

$$\text{Score_confidence} = \text{prob}(\text{obj}) \times \text{IoU}_{\text{actual,estimated}} \quad (1)$$

The object's presence probability, represented as $\text{prob}(\text{obj})$, ranges from 0 to 1. Here, 0 indicates the object is absent, and 1 indicates it is likely present. $\text{IoU}_{\text{actual,estimated}}$, which is calculated using the IoU measure, compares the estimated bounding box to the actual (ground truth) bounding box.

Five components define a bounding box: a , b , c , d , and confidence score. a and b represent the bounding box's center coordinates, while c and d represent its width and height. The final parameter, the confidence score, represents the likelihood of an object in the box.

Bounding boxes help YOLO and general object detection discover objects. Two bounding box vectors are required: b for ground truth and \hat{b} for expected. In YOLO, non-maximum suppression (NMS) handles multiple bounding boxes for absent or identical objects. NMS rejects overlapping predicted boxes with an IoU below a threshold.

The original Darknet-based YOLO had two versions. Two ultimately linked layers followed 24 convolutional layers in the standard model. The simplest Fast YOLO included nine convolutional layers and fewer filters. Both versions used GoogLeNet's inception module-inspired 1×1 convolutional layers to reduce feature space.

To deter incorrect bounding box predictions, the authors assigned different weights: $\gamma_{\text{coord}} = 5$ for boxes with objects and $\gamma_{\text{noobj}} = 0.5$ for empty boxes. The loss function integrates all bounding box parameters and calculates the loss between anticipated and actual boxes using center coordinates (a_{center} , b_{center} at the start). The variable ζ_{ij}^{obj} is 1 if an object is in the j^{th} predicted box in the i^{th} cell, and 0 otherwise. The adjusted equation shows that the box should predict the object with the highest IoU, as shown in Eq. 2.

$$\gamma_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^B \zeta_{ij}^{\text{obj}} [(a_i - \hat{a}_i)^2 + (b_i - \hat{b}_i)^2] \quad (2)$$

The subsequent component of the loss function computes the discrepancy in the estimated width and height of the bounding box. Contrary to the previous component, faults in larger boxes have a diminished effect compared to smaller ones. By standardizing the width and height to a range of 0-1, applying the square root function enhances the influence of inaccuracies in smaller boxes to a greater extent than in bigger ones, as expressed in Eq. 3.

$$\gamma_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^B \zeta_{ij}^{\text{obj}} [(a_i - \hat{a}_i)^2 + (b_i - \hat{b}_i)^2] \quad (3)$$

The loss function calculates confidence score discrepancy based on the object's existence or absence in the bounding box. When the predictor determines the bounding box, object confidence errors are penalized. The variable ζ_{ij}^{obj} is set to 1 if an object is present in the cell and 0 otherwise. Alternatively, $\zeta_{ij}^{\text{noobj}}$ evaluates objects as 1 when presented and 0 when absent and 0, as presented in Eq. 4.

$$\begin{aligned} \text{Loss} = & \sum_{i=0}^{s^2} \sum_{j=0}^B \zeta_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \gamma_{\text{noobj}} \sum_{i=0}^{s^2} \sum_{j=0}^B \zeta_{ij}^{\text{noobj}} (X_i - \hat{X}_i)^2 + (C_i - \hat{C}_i)^2 \end{aligned} \quad (4)$$

Like the conventional classification loss, the last component of the loss function calculates the loss in the probability of the class (c). Nevertheless, this computation includes ζ_{ij}^{obj} , which modifies the loss depending on the presence or absence of an object within the bounding box, as stated in Eq. 5.

$$\text{Loss}_{\text{class}} = \sum_{i=0}^{s^2} \zeta_{ij}^{\text{obj}} \sum_{c \in \text{Classes}} (p_i(c) - \hat{p}_i)^2 \quad (5)$$

Redmon et al. (2016) introduced YOLO. Multiple YOLO algorithm improvements occurred from further research. YOLO9000 is a real-time object detection system that can recognize 9000 categories to improve accuracy and performance. YOLOv3 was introduced and gradually enhanced over its predecessors. Bochkovskiy et al. (2020) introduced YOLOv4, which improved the model's object detection and GPU usage. Zhu et al. (2021) introduced YOLOv5 to improve GPU utilization. Later, YOLOv6 and v7 were introduced (Li et al., 2022; Wang et al., 2023). The most current version of YOLOv8 has been introduced (Ultralytics, 2024).

Attention module

The attention module, essential for the deep CNN-based feature extractor, utilizes complex approaches to improve feature representation. By utilizing compression and decompression techniques, this procedure condenses input features to optimize processing efficiency while retaining crucial information. The channel and spatial attention modules enhance the discriminative power of features by selectively emphasizing informative channels and geographical regions, respectively. By using a cross-convolution module enhanced with a constant vector parameter, the process of feature derivation is enhanced as it captures intricate patterns and interconnections within the data. The module identifies the interconnections between different feature dimensions by utilizing two-way correlation matrices to create a rich feature map.

The proposed technique incorporates a multi-scale feature extraction architecture to improve YOLOv8 object detection. A more advanced feature extractor, including Convolutional (Conv) layers, convolutional block attention module (CBAM), and C3x layers, has been added to the YOLOv8 backbone to increase detection efficiency. This improvement uses attention processes and spatial pyramid pooling to improve the model's feature extraction from input images. The increased feature extraction procedure is mathematically expressed in Eq. 6.

$$F_{\text{extracted}} = \text{CBAM}(\text{Conv}_{\text{input}}) \oplus \text{SPPF}(\text{Conv}_{\text{input}}) \quad (6)$$

Here, \oplus represents the fusion of characteristics retrieved using attention processes and spatial pooling layers. The architecture neck enhances and perfects details after extraction. This stage merges features at different scales using "C3x" layers and upsampling methods. Refined features are presented in Eq. 7.

$$F_{\text{C3x}} = \text{Upsample}(\text{C3x}(F_{\text{extracted}})) \quad (7)$$

This technique effectively combines characteristics of unusual sizes, improving model recognition at varied resolutions.

The model detects small-sized, medium-sized, and large-sized decompression nodes that are perfectly integrated with a pre-trained multi-scale module. These heads predict bounding box coordinates for various objects in the image frame with the probability matrix. The probability matrix includes dfl scores, cls scores, and class probabilities. These detection heads are expressed in Eq. 8.

$$B_i = H_{\text{bbox}}(F_D) \quad (8)$$

where B_i denotes the spatial coordinates that determine the position of object i in the image and H_{bbox} denotes the function applied by detecting heads to anticipate bounding. F_D denotes the decompression nodes that have been identified for each object presented in Eq. 9.

$$P_i = \{\text{DFL}_i, \text{CLS}_i, \text{Prob}_i\} = H_{\text{prob}}(F_D) \quad (9)$$

where H_{prob} refers to the function utilized by detection heads to construct the probability matrix, which includes distributional focal loss (DFL) scores, classification results, and probabilities for each class.

RESULTS

This section concisely describes the experimental results, their interpretation, and the experimental conclusions.

The proposed model is trained and evaluated using the YOLO PyTorch framework for object detection. The utilized pre-trained model employed the Adam optimizer with a learning rate of $3e-4$. The values we used for our parameters were 15 epochs, a batch size of 24, and an image size of 640×640 . The enhanced feature extractor backbone and three-channel detectors are utilized to estimate the class probability of the ArSL dataset. The collection consists of Arabic signs. A series of extensive experiments validated the model's performance. We employed a training dataset of 9927 samples and a separate validation dataset of 4247 distinct samples for our research. In Figure 3, the confusion matrix of the proposed model is presented.

The confusion matrix of the dataset after normalization is presented in Figure 4, which shows that the miss detection rate was significantly reduced after normalization.

In Figure 5, the top row training loss metrics show that `train/box_loss` has decreased from 0.9 to 0.5, indicating better-bounding box predictions. `Train/cls_loss` declines from

4 to slightly above 0, suggesting better-predicted box object categorization. Model performance on this composite loss parameter improves when `train/df1_loss` drops from 1.3 to 0.9. Performance measurements reveal that the model predicts class "B" more wholly and precisely when `metrics/precision(B)` reaches 0.5 to slightly over 0.8 and `metrics/recall(B)` reaches 0.4 to almost 0.9. `mAP metrics/mAP50(B)` and `mAP50-95(B)` increase from 0.5 to 0.8 and 0.4 to over 0.7, respectively. The results show that the model can detect signs with more ground truth overlap at various thresholds with higher validation loss. Unlike its training counterpart, `val/box_loss` fluctuates but declines from 0.76 to 0.65. Like training, `val/cls_loss` drops from 2 to 0.5. Starting around 1.15, `val/df1_loss` drops but fluctuates. This variation in validation losses implies improving the model on unseen data to avoid overfitting and improve consistency.

In Figure 6, the x-axis of the graph shows confidence levels, while the y-axis represents the F1 score. The bold blue line, labeled "all classes 0.95 at 0.547," shows that when the model's confidence threshold is set at around 0.547, the F1 score for all classes combined approaches 0.95. This high score signifies excellent model performance.

Figure 7 shows the confidence threshold, the model's assessed probability of forecast correctness. The y-axis

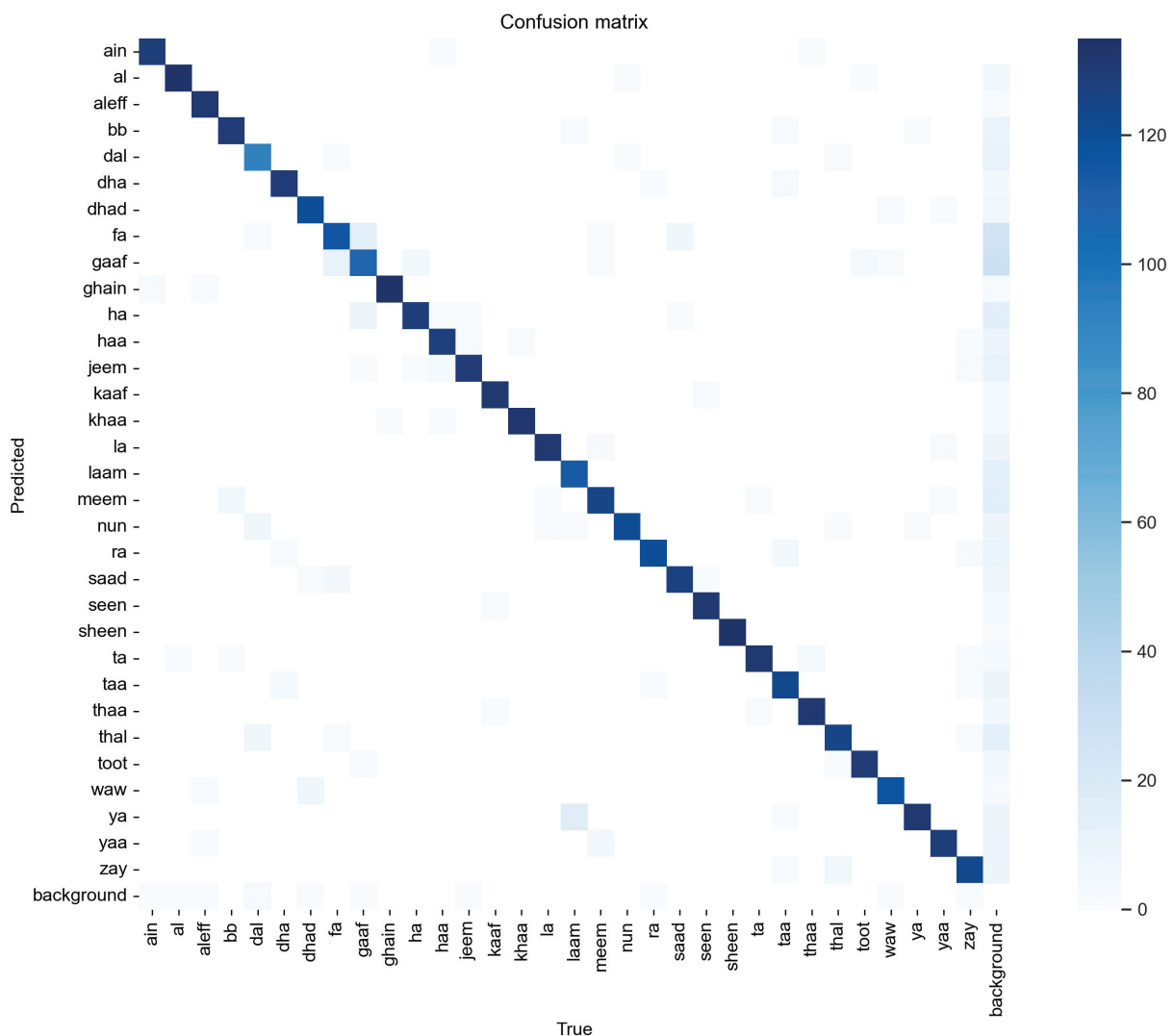


Figure 3: Confusion matrix of proposed model results.

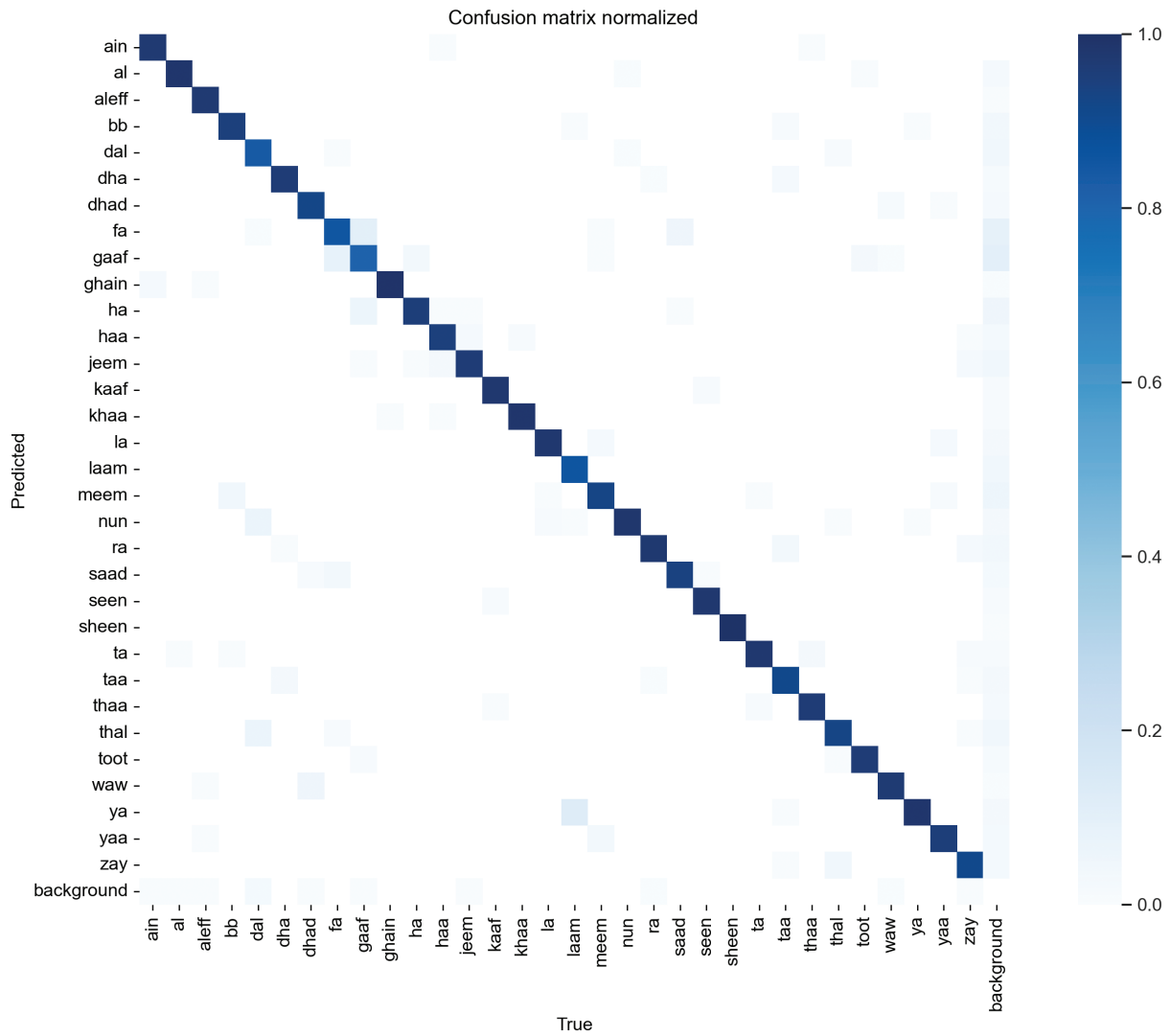


Figure 4: Confusion matrix of proposed model detection results after data normalization.

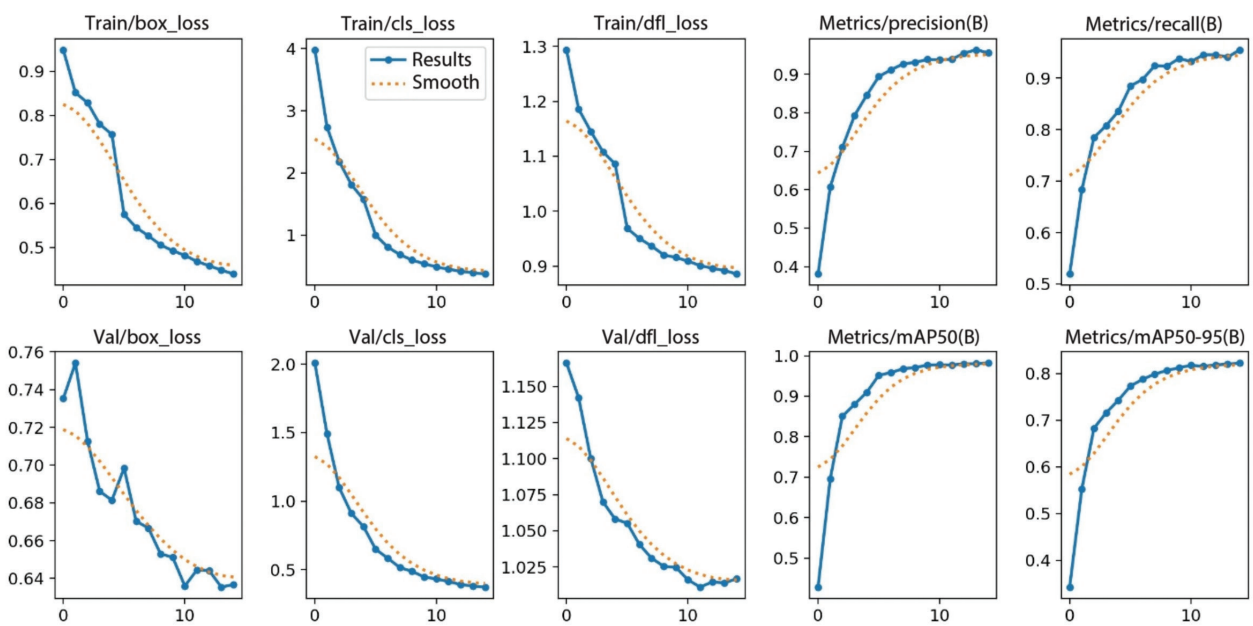


Figure 5: Proposed model training validation precision–recall and different types of loss. Abbreviation: mAP, mean average precision.

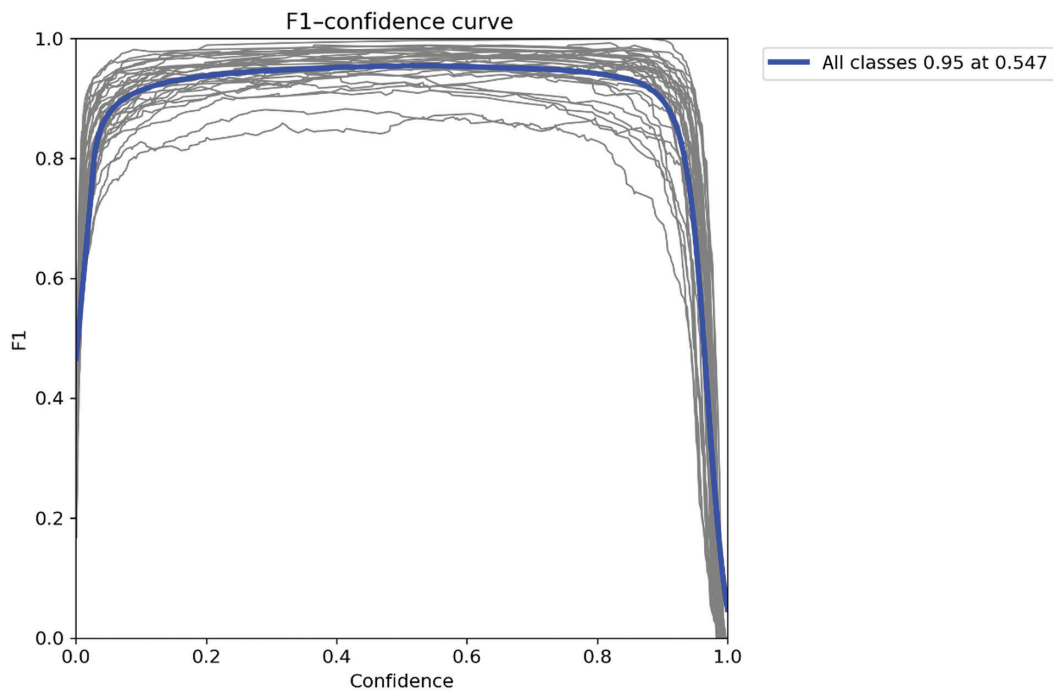


Figure 6: Proposed model for ArSL detection F1–confidence curve. Abbreviation: ArSL, Arabic Sign Language.

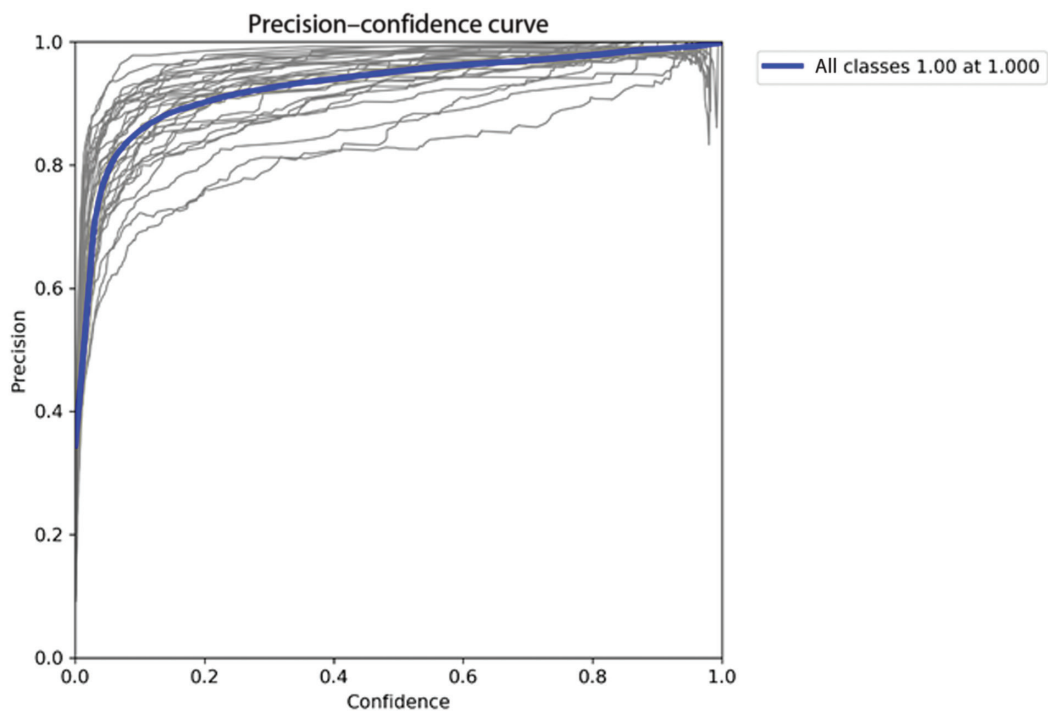


Figure 7: Precision–confidence curve of proposed detection model for ArSL. Abbreviation: ArSL, Arabic Sign Language.

shows model accuracy at each confidence level. The bold blue line, “all classes 1.00 at 1.000,” shows the model’s precision of 1.00 (or 100%) for all classes when it forecasts 100% accurately. This model is ideal since it predicts with certainty and is accurate. However, choosing high confidence thresholds may cause the model to miss many true positives when it lacks the confidence to foresee them, reducing recall.

In Figure 8, several experiments and iterations aim to optimize precision and recall, leading to a position near the upper-right quadrant of the graph. The various gray lines depict the trade-off between accuracy and recall for different classes or runs of the model. The blue line, denoted as “all classes 0.982 mAP@0.5,” indicates that the model achieves an mAP of 0.982 at an IoU threshold of 0.5, a commonly employed criterion in object detection tasks. The mapped

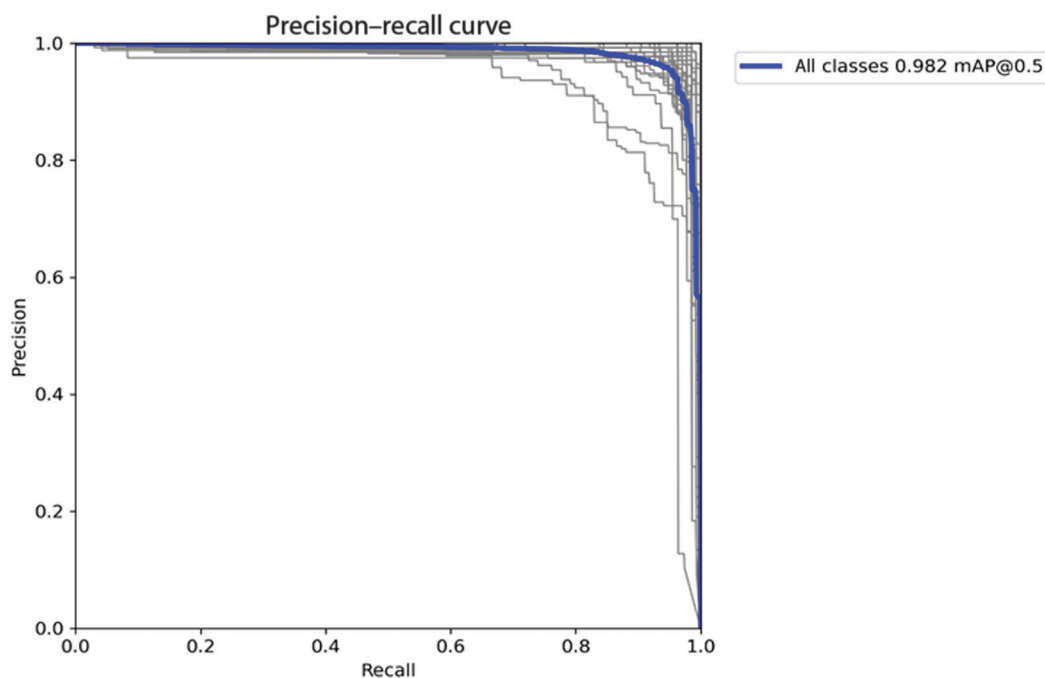


Figure 8: Precision–recall curve for the proposed ArSL identification method. Abbreviations: ArSL, Arabic Sign Language; mAP, mean average precision.

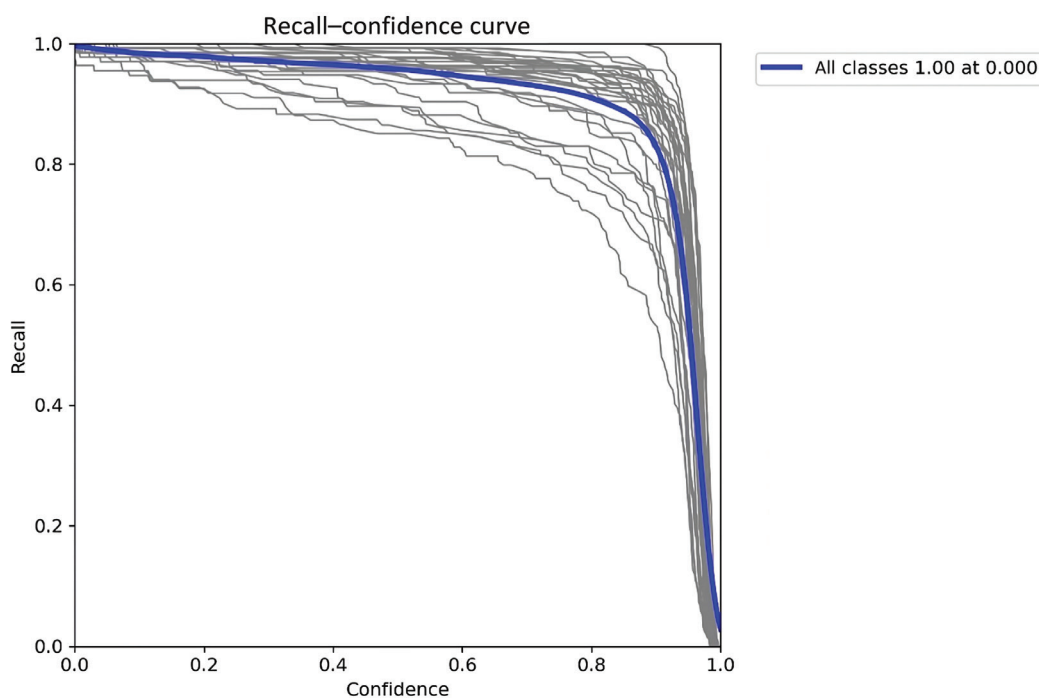


Figure 9: Recall–confidence curve of the proposed ArSL detection approach. Abbreviation: ArSL, Arabic Sign Language.

value is significantly high, suggesting that the model exhibits strong performance across all classes in terms of both precision and recall at this threshold.

Figure 9 shows that our method for ArSL identification is highly effective and validated using a recall–confidence score, highlighting the model’s effectiveness. The y-axis is the recall, representing the model’s ability to identify as many relevant observations as possible in each category. The

x-axis gives a confidence level. As mentioned earlier, the ability of our model is confirmed in terms of ArSL detection. Fine-tuning and parameter settings have achieved a recall–confidence score of 100%. Then, the method at hand is considered the best for classifying ASL signature images under that confidence level. Through the thick blue line with the label “all classes 1.00 at 0.000,” we can see one of the factors allowing our method to be so successful: this single line



Figure 10: Proposed model detection results on different signs.

makes up the entirety of the model's recall, indicating that all classes can be recalled with high confidence. The gray lines, on the other hand, stand for different categories or variants, as multiple lines of different lengths and at various places can be seen. These visual depictions mean that the recall of the model in response to different confidence thresholds varies, which showcases the model's responsiveness and capacity to accurately recognize all relevant instances, even at a confidence threshold of 0. During experiments, it was noted that a low confidence threshold was used to achieve 100% recall. This curve is vital for comprehending the balance between achieving a high recall and the confidence level in the predictions, which is necessary for optimizing model performance according to the accurate detection of ArSL.

The proposed model detection performance across different signs is presented in Figure 10. The results show that

the model's correct detection rate is higher in detecting most signs. The proposed detection model shows robustness regarding accuracy, precision, and recall, with the lowest training and validation loss.

Detection results on base model

Model training progress is summarized in Figure 11. The training and validation losses for bounding box prediction (box_loss), class prediction (cls_loss), and direction/feature learning (dfl_loss) all decreased significantly over time. Train/box_loss begins above 2.5 and decreases to 0.5. Similarly, val/box_loss decreases from 2 to 0.5, implying improved item detection. Similarly, train/cls_loss starts around 5 and falls below 0.5. Val/cls_loss decreases from 4

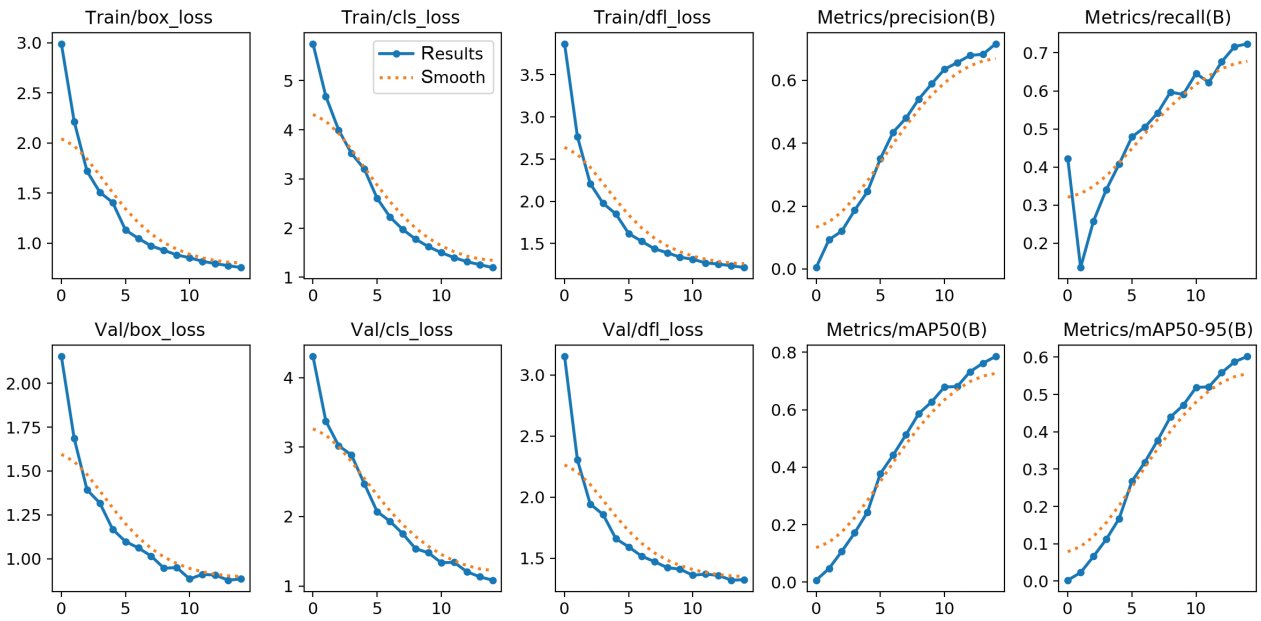


Figure 11: Model training validation precision–recall and loss of YOLOv8. Abbreviations: mAP, mean average precision; YOLO, You Only Look Once.

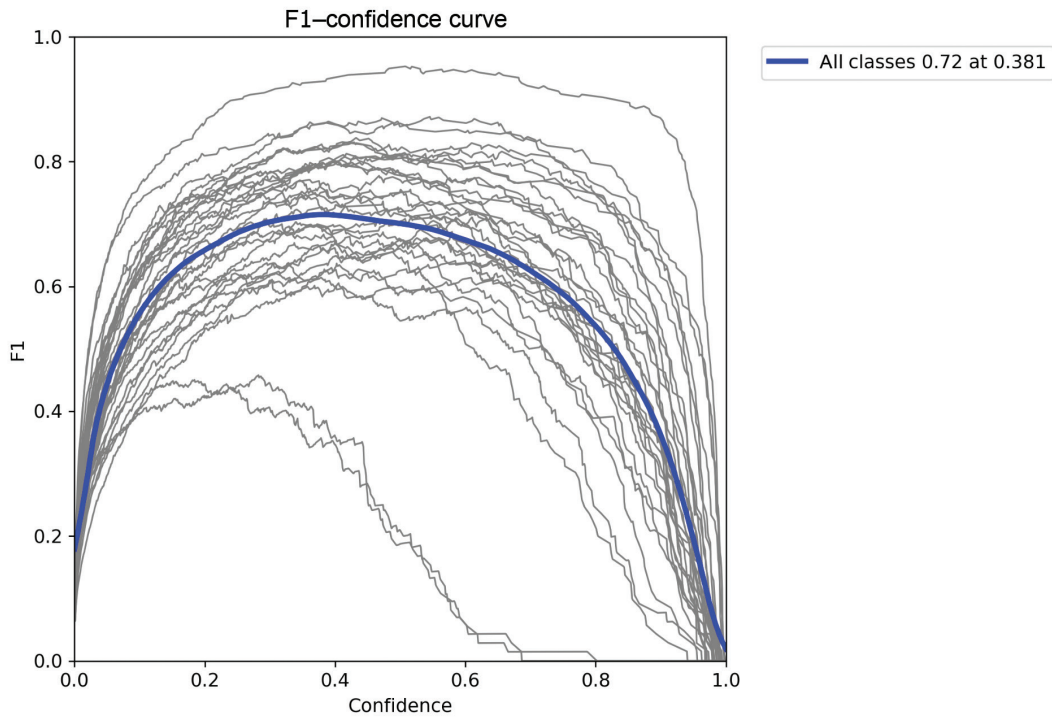


Figure 12: F1–confidence curve of the base detection model.

to slightly above 0.5, indicating improved item categorization accuracy. The train/dfl_loss and val/dfl_loss ratios drop from more than 3.5 to 0.5 and 3 to slightly above 0.5, respectively, showing feature or directional learning improvement. Precision and recollection increase steadily. Class B precision and recall measures increase from 0 to more than 0.6 and 0.2 to more than 0.7, respectively.

These changes result in more correct predictions and improved detection of all signs. mAP scores improved

significantly, including mAP50(B) and mAP50-95b. The mAP50(B) has risen from 0 to over 0.8, while the mAP50-95(B) has increased from 0.1 to more than 0.6. IoU scores consistently improve in precision. After training, the numerical patterns demonstrate the model’s improved predicting and categorization abilities.

The F1–confidence curve shows the link between the F1 score and the classification model confidence threshold in Figure 12. While the broad blue line shows overall

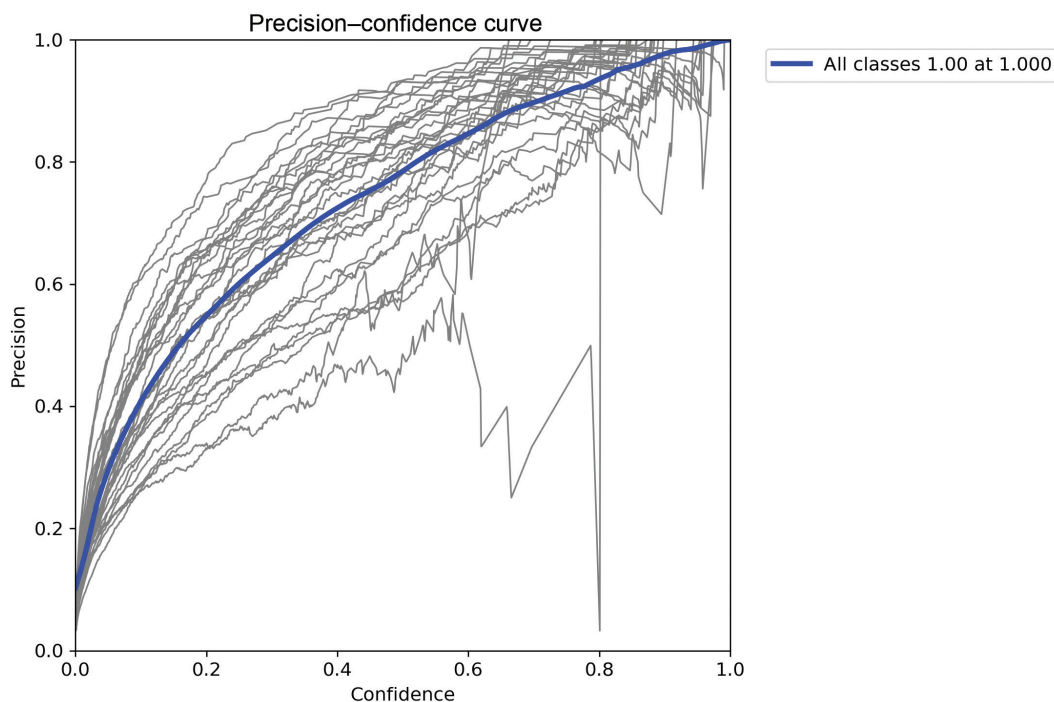


Figure 13: Precision–confidence curve of the base model for ArSL detection. Abbreviation: ArSL, Arabic Sign Language.

performance across all courses, the gray lines likely show F1 scores for various classes at varying confidence criteria. The model’s highest F1 score for all classes is “all classes 0.72” at a confidence threshold of 0.381.

In Figure 13, a precision–confidence curve is depicted, illustrating the accuracy of a classification model at different levels of confidence thresholds.

Each gray line refers to a distinct class, whereas the bold blue line reflects the overall precision encompassing all

classes. The phrase “all classes 1.00 at 1.000” signifies the accuracy achieved by the model when the confidence threshold is set to its highest level.

A precision–recall curve is presented in Figure 14, which assesses the effectiveness of a classification model. The gray lines depict the trade-offs between precision and recall for each class, while the blue line represents average performance across all classes with 0.786 mAP@0.5 and an mAP score of 0.786 at an IoU criterion of 0.5. The results show

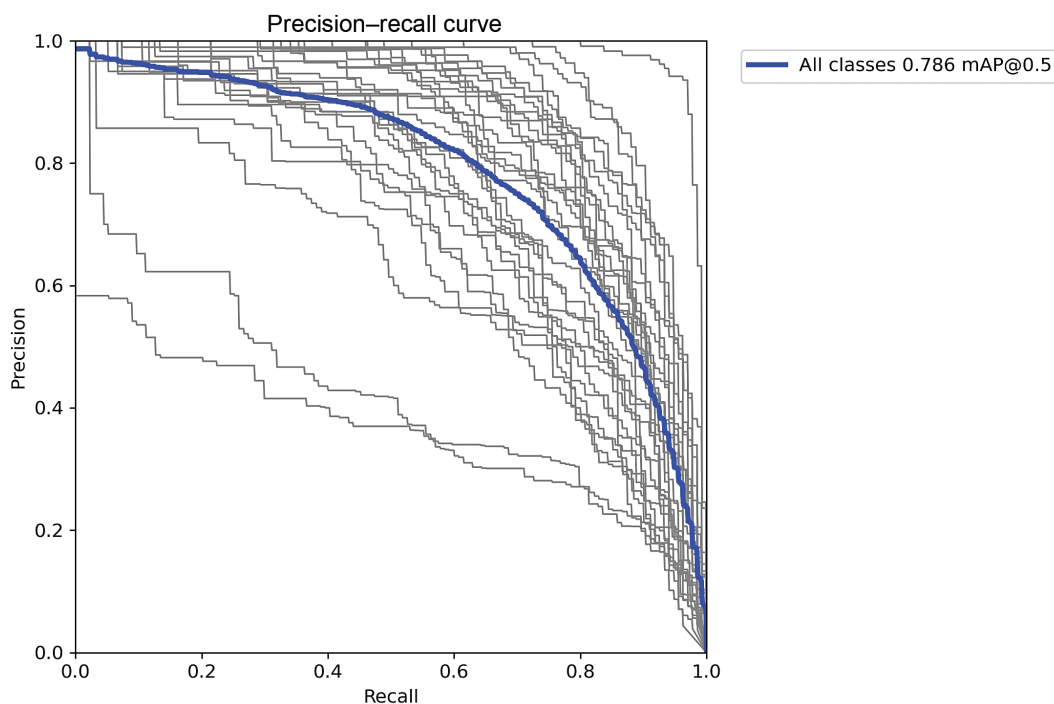


Figure 14: Precision–recall curve of detection rate of the base model. Abbreviation: mAP, mean average precision.

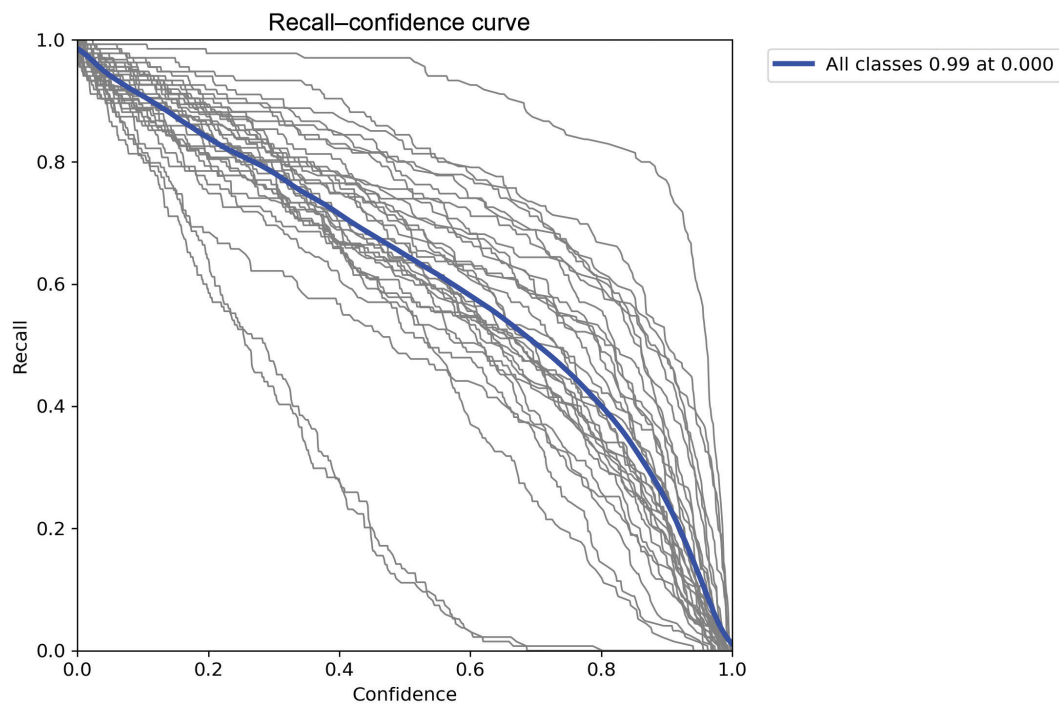


Figure 15: Recall–confidence curve of the base model for ArSL detection. Abbreviation: ArSL, Arabic Sign Language.

a high level of model performance, on average, across all classes.

Figure 15 illustrates a recall–confidence curve, which demonstrates the fluctuation of recall at various confidence thresholds for a classification model. The gray lines represent the recall at different confidence levels for each class, while the average recall for all classes is indicated by the blue line. All classifications (0.99 at 0.000) signify that the model achieves almost perfect recall when using a confidence threshold of 0, indicating a strong ability to detect true positives across all classes.

DISCUSSION

In this paper, we proposed a technique that utilizes a deep CNN model integrated with YOLO to detect ArSL signs. Our proposed method showed competitive performance as compared with the state-of-the-art (SOTA) approaches. Unlike the previous methods, this proposed technique contributes to specifying the visual data’s key unique and discriminative features, which minimizes redundant information and maximizes the detection rate. In our proposed method, taking the critical roles of visual data in computation, two attention mechanisms were incorporated, channel and spatial attention, which used self-attention module blocks and cross-convolution modules. The mechanism of attention techniques, which has some differences, is highlighted in the most significant features of the ArSL images. By emphasizing the importance of the unique features of the ArSL signs, the proposed technique achieved a better and more comprehensive understanding of the ArSL gestures.

Recent researchers have attempted to concatenate different advanced recognition systems to get more promising

identification rates for ArSL gestures. ANFIS has been proven pioneering by many researchers; however, it is counteracted by its significant weakness in swiftly adapting and performing intricate tasks of gesture recognition, an essential criterion in any sign language recognition system. As a result, ANFIS has achieved a high precision of 86.69%, but its performance and adaptation ability to intricate gesture recognition tasks remain limited (Podder et al., 2023). A detailed comparison of the proposed model with SOTA approaches is presented with an accuracy of 94.46% (Aldhahri et al., 2023) presented in (Table 1).

An independent-user-based technique employed DL and vision-based techniques to interpret ArSL with 98% accuracy (Balaha et al., 2023). A YOLOv5-based approach was presented (Dima and Ahmed, 2021) for sign language recognition with an mAP of 0.98 with a precision of 95%. Aldhahri et al. (2023) successfully recognized Arabic signs using DL-based techniques on the ArASL2018 dataset with an accuracy of 94.46%.

YOLOv6 was employed to recognize the ArSL using static and dynamic images with 96% accuracy on static images and

Table 1: Comparison of the proposed model with SOTA techniques.

Reference	Year	Model	Recognition rate (%)
Podder et al. (2023)	2023	ANFIS	86.69
Balaha et al. (2023)	2023	Vision-based DL	98
Dima and Ahmed (2021)	2021	YOLOv5	95
Aldhahri et al. (2023)	2023	DL model	94.46
Buttar et al. (2023)	2023	YOLOv6	92
	2024	Proposed	99

The bold is the proposed model with highest recognition rate. Abbreviations: ANFIS, adaptive neuro-fuzzy inference system; DL, deep learning; YOLO, You Only Look Once.

92% accuracy on different continuous signs (Buttar et al., 2023).

The modified model also improved the precision and recall rate to 0.99 in recognizing and detecting different signs of ArSL. The robustness of the proposed model showed significant improvement in detecting and recognizing different signs. The model performance decreases the error rate with a higher rate of accurate sign recognition. The YOLOv8-based model achieves significant 0.9909 and 0.8306 for mAP@0.5 and mAP@0.5:0.95, respectively. The utilization of a pre-trained DL model shows a decline in the recognition rate with a higher error rate. We utilized a pre-trained model as well as our custom module integration.

CONCLUSION

In this paper, we proposed a novel DL-based approach using YOLOv8 to detect and recognize the ArSL signs effectively. The robust object detection model YOLOv8 is used as a baseline to develop an attention-aware feature descriptor for feature engineering. The integration of the self-attention modules with channel attention and spatial attention modules utilized to compress and decompress in features, and the implementation of the cross-convolution module to mathematically process for split three-way matrix efficiently. These are our contributing resources to increase the precision, accuracy, and recognition speed of gesture recognition. ArSL sign detection rates are much higher than those of any existing approaches. We have driven SOTA recognition rates and opened the best way to attempt broad-range applications for ArSL recognition and extraction of DL features. The validation of the model using the ArSL21L dataset highlights its efficacy in accurately finding a diverse assortment of ArSL gestures. Unlike the specified traditional approaches, the proposed technique derived the most relevant features from ArSL images, rather than using hardcoded feature extraction techniques employed in previous studies. The proposed model is robust in sign detection and recognition, but it is trained on limited data, and more complex data and environments will hamper its performance. A more robust DL-based technique will be developed in the future, and the model will be trained on more diverse datasets to detect and classify

diverse Arabic signs. Model generalization and interoperability will be enhanced for real-world applications.

AUTHOR CONTRIBUTIONS

Conceptualization: FM and SAA; methodology: FM; software: FM; validation: FM, HAD, and SAA; formal analysis: SAA; investigation: SAA; resources: SAA; data curation: FM; writing—original draft preparation: FM; writing—review and editing: FM and HAD; visualization: FM; supervision: SAA; project administration: SAA; funding acquisition: SAA. All authors have read and agreed to the published version of the manuscript.

FUNDING

This research was funded by the King Salman Center for Disability Research, grant number KSRG-2023-512 (funder ID: <http://dx.doi.org/10.13039/501100019345>).

DATA AVAILABILITY STATEMENT

The dataset utilized in this research is publicly available (<https://data.mendeley.com/datasets/f63xhm286w/1>), accessed on January 5, 2024.

ACKNOWLEDGMENTS

The authors extend their appreciation to the King Salman Center for Disability Research (funder ID: <http://dx.doi.org/10.13039/501100019345>) for funding this work through Research Group no. KSRG-2023-512.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

- Alaghaband M., Maghroor H.R. and Garibay I. (2023). A survey on sign language literature. *Mach. Learn. Appl.*, 14, 100504.
- Aldahri E., Aljuhani R., Alfaidi A., Alshehri B., Alwadei H., Aljojo N., et al. (2023). Arabic Sign Language recognition using convolutional neural network and mobilenet. *Arab. J. Sci. Eng.*, 48(2), 2147-2154.
- Al-Jarrah O. and Halawani A. (2001). Recognition of gestures in Arabic Sign Language using neuro-fuzzy systems. *Artif. Intell.*, 133(1-2), 117-138.
- Alnabih A.F. and Maghari A.Y. (2024). Arabic Sign Language letters recognition using vision transformer. *Multimed. Tools Appl.*, 1-15.
- Aly S. and Aly W. (2020). DeepArSLR: a novel signer-independent deep learning framework for isolated Arabic Sign Language gestures recognition. *IEEE Access*, 8, 83199-83212.
- Alyami S., Luqman H. and Hammoudeh M. (2023). Isolated Arabic Sign Language recognition using a transformer-based model and landmark keypoints. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 23, 1-19.
- Attia N.F., Ahmed M.T.F.S. and Alshewimy M.A. (2023). Efficient deep learning models based on tension techniques for sign language recognition. *Intell. Syst. Appl.*, 20, 200284.
- Balaha M.M., El-Kady S., Balaha H.M., Salama M., Emad E., Hassan M., et al. (2023). A vision-based deep learning approach for independent-users Arabic Sign Language interpretation. *Multimed. Tools Appl.*, 82(5), 6807-6826.
- Batnasan G., Gochoo M., Otgonbold M.-E., Alnajjar F. and Shih T.K. (2022). Arsl211: Arabic Sign Language letter dataset benchmarking

- and an educational avatar for metaverse applications. In: *2022 IEEE Global Engineering Education Conference (EDUCON)*; pp. 1814-1821, IEEE.
- Bochkovskiy A., Wang C.-Y. and Liao H.-Y.M. (2020). YOLOv4: optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Boukdir A., Benaddy M., Ellahyani A., Meslouhi O.E. and Kardouchi M. (2021). Isolated video-based Arabic Sign Language recognition using convolutional and recursive neural networks. *Arab. J. Sci. Eng.*, 1-13.
- Buttar A.M., Ahmad U., Gumaei A.H., Assiri A., Akbar M.A. and Alkhamees B.F. (2023). Deep learning in sign language recognition: a hybrid approach for the recognition of static and dynamic signs. *Mathematics*, 11(17), 3729.
- Dima T.F. and Ahmed M.E. (2021). Using YOLOv5 algorithm to detect and recognize American Sign Language. In: *2021 International Conference on Information Technology (ICIT)*; pp. 603-607, IEEE.
- El-Alfy E.-S.M. and Luqman H. (2022). A comprehensive survey and taxonomy of sign language research. *Eng. Appl. Artif. Intell.*, 114, 105198.
- Guo M.-H., Xu T.X., Liu J.J., Liu Z.-N., Jiang P.-T., Mu T.-J., et al. (2022). Attention mechanisms in computer vision: a survey. *Comput. Vis. Media*, 8(3), 331-368.
- Hussain N., Khan M.A., Sharif M., Khan S.A., Albeshier A.A., Saba T., et al. (2020). A deep neural network and classical features based scheme for objects recognition: an application for machine inspection. *Multimed. Tools Appl.*, 1-23.
- Ji Y., Zhang H., Zhang Z. and Liu M. (2021). CNN-based encoder-decoder networks for salient object detection: a comprehensive review and recent advances. *Inf. Sci.*, 546, 835-857.
- Kahlon N.K. and Singh W. (2023). Machine translation from text to sign language: a systematic review. *Univers. Access Inf. Soc.*, 22(1), 1-35.
- Kumari D. and Anand R.S. (2024). Isolated video-based sign language recognition using a hybrid CNN-LSTM framework based on attention mechanism. *Electronics*, 13(7), 1229.
- Leigh I.W., Andrews J.F., Miller C.A. and Wolsey J.-L. A. (2022). *Deaf People and Society: Psychological, Sociological, and Educational Perspectives*, Routledge.
- Li C., Li L., Jiang H., Weng K., Geng Y., Li L., et al. (2022). YOLOv6: a single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.
- Luqman H. (2023). ArabSign: a multi-modality dataset and benchmark for continuous Arabic Sign Language recognition. In: *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*; pp. 1-8, IEEE.
- Mammeri S., Amroune M., Haouam M.-Y., Bendib I. and Corrêa Silva A. (2023). Early detection and diagnosis of lung cancer using YOLO v7, and transfer learning. *Multimed. Tools Appl.*, 1-16.
- Mazen F. and Ezz-Eldin M. (2024). A novel image-based Arabic hand gestures recognition approach using YOLOv7 and ArSL21L. *Fayoum Univ. J. Eng.*, 7(1), 40-48.
- Mustafa Z. and Nsour H. (2023). Using computer vision techniques to automatically detect abnormalities in chest X-rays. *Diagnostics*, 13(18), 2979.
- Niu Z., Zhong G. and Yu H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, 452, 48-62.
- Podder K.K., Ezeddin M., Chowdhury M.E.H., Sumon M.S.I., Tahir A.M., Ayari M.A., et al. (2023). Signer-independent Arabic Sign Language recognition system using deep learning model. *Sensors*, 23(16), 7156.
- Rajalakshmi E., Elakkiya R., Subramaniaswamy V., Prikhodko Alexey L., Mikhail G., Bakaev M., et al. (2023). Multi-semantic discriminative feature learning for sign gesture recognition using hybrid deep neural architecture. *IEEE Access*, 11, 2226-2238.
- Redmon J., Divvala S., Girshick R. and Farhadi A. (2016). You Only Look Once: unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; pp. 779-788.
- Renjith S., Rashmi M. and Suresh S. (2024). Sign language recognition by using spatio-temporal features. *Procedia Comput. Sci.*, 233, 353-362.
- Sarda A., Dixit S. and Bhan A. (2021). Object detection for autonomous driving using YOLO algorithm. In: *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*; pp. 447-451, IEEE.
- Shanableh T. (2023). Two-stage deep learning solution for continuous Arabic Sign Language recognition using word count prediction and motion images. *IEEE Access*, 11, 126823-126833.
- Sharma S., Gupta R. and Kumar A. (2023). Continuous sign language recognition using isolated signs data and deep transfer learning. *J. Ambient Intell. Humaniz. Comput.*, 1-12.
- Strobel G., Schoormann T., Banh L. and Möller F. (2023a). Artificial intelligence for sign language translation—a design science research study. *Commun. Assoc. Inf. Syst.*, 52(1), 33.
- Strobel G., Schoormann T., Banh L. and Möller F. (2023b). Artificial intelligence for sign language translation—A design science research study. *Commun. Assoc. Inf. Syst.*, 53(1), 22.
- Ultralytics. YOLOv8. <https://github.com/ultralytics/yolov8>. Accessed 24 January 2024.
- Vaitkevicius A., Taroza M., Blažauskas T., Damaševičius R., Maskeliūnas R. and Woźniak M. (2019). Recognition of American Sign Language gestures in a virtual reality using leap motion. *Appl. Sci.*, 9(3), 445.
- Wang Y., Sun Q., Sun G., Gu L. and Liu Z. (2021). Object detection of surgical instruments based on YOLOv4. In: *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*; pp. 578-581, IEEE.
- Wang C.-Y., Bochkovskiy A. and Liao H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; pp. 7464-7475.
- Wu P., Li H., Zeng N. and Li F. (2022). FMD-Yolo: an efficient face mask detection method for COVID-19 prevention and control in public. *Image Vis. Comput.*, 117, 104341.
- Yang X. (2020). An overview of the attention mechanisms in computer vision. *J. Phys. Conf. Ser.*, 1693(1), 012173.
- Žemgulys J., Raudonis V., Maskeliūnas R. and Damaševičius R. (2020). Recognition of basketball referee signals from real-time videos. *J. Ambient Intell. Humaniz. Comput.*, 11, 979-991.
- Zhu X., Lyu S., Wang X. and Zhao Q. (2021). TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*; pp. 2778-2788.