Efficient YOLO Based Deep Learning Model for Arabic Sign Language Recognition

Saad Al Ahmadi  
King Saud University

Farah Mohammad  
fnazar@ieee.org  
King Saud University

Haya Al Dawsari  
King Saud University

Article

Keywords: Sign Language, Deep Learning, YOLO, Sign Language Detection, ArSL

Posted Date: March 28th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4006855/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Journal of Disability Research on January 1st, 2024. See the published version at https://doi.org/10.57197/JDR-2024-0051.
Efficient YOLO Based Deep Learning Model for Arabic Sign Language Recognition

Saad Al Ahmadi 1, Farah Mohammad 1,*, and Haya Al Dawsari1

1 Department of Computer Science, College of Computer & Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia;
Correspondence: fnazar@ieee.org

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, or nonverbal. Consequently, these individuals depend on sign language to communicate with others. Sign Language is a complex system of gestures and visual cues that facilitate the inclusion of individuals into vocal communication groups. In this manuscript a novel technique proposed using deep learning to recognize the Arabic Sign language (ArSL) accurately. Through this advanced system, the objective is to help in communication between the hearing and deaf community. The proposed mechanism relies on advanced attention mechanisms, and state-of-art Convolutional Neural Network (CNN) architectures with the robust YOLO object detection model that highly improves the implementation and accuracy of ArSL recognition. In our proposed method, we integrate the self-attention block, channel attention module, spatial attention module, and cross-convolution module into the features processing, and the ArSL recognition accuracy reaches 98.9%. The recognition accuracy of our method is significantly improved with higher detection rate. The presented approach showed significant improvement as compared with the conventional techniques with a precision rate of 0.9. For the mAP@0.5, the mAP score is 0.9909 while for the mAP@0.5:0.95 and the results tops all the state-of-the-art techniques. This shows that the model has the great capability to accurately detect and classify complex multiple ArSL signs. The model provides a unique way of linking people and improving the communication strategy while also promoting the social inclusion of deaf people in the Arabic region.

Keywords: Sign Language; Deep Learning; YOLO; Sign Language Detection; ArSL.

1. 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 sign language [1]. 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. There exist numerous sign languages globally, however, their quantity remains lower than that of spoken languages [2, 3]. Like other languages, sign languages undergo continuous evolution 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) is not a precise representation of 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 [4].

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 YOLO (You Only Look Once) technique has great potential for sign language recognition. YOLO object identification models are utilized in several fields such as surgical procedures for identifying organ locations, driverless vehicles, and detecting face masks. This technology has proven to be beneficial in various practical scenarios [5-7].

In visual applications such as radiography [8], CNNs are essential components of models like YOLO, employ a three-layered methodology. The convolution layer utilizes filters to extract crucial characteristics from images. By reducing the dimensions of these feature maps, the pooling layer effectively controls overfitting. As the data passes through the fully connected layer, it transforms into concrete image analysis conclusions. The remarkable increase in precision and efficiency in diagnostic imaging, achieved by implementing CNN architecture, truly showcases the powerful influence of Artificial Intelligence (AI) in the field of visual domains [9].

In the field of computer vision, the addition of attention mechanisms to CNNs has proved to be a breakthrough, leading to significantly enhanced performance. This has been demonstrated by notable advancements observed over the last decade [10]. Inspired humans to process visual information, CNN attention mechanisms adeptly filter out extraneous details and home in on specific targets or regions within intricate visual surroundings. This mimics our instinctual tendency to focus on key areas when processing visual scenes. In recent years, attention mechanisms have made incredible strides in improving our ability to identify crucial elements in images. By dynamically adjusting the importance of various channels and spatial regions (like spatial and channel attention), these mechanisms have significantly enhanced numerous computer vision applications, such as object detection and image classification. As a result, these tasks can now be performed more efficiently and effectively [11].
Recent research has enhanced the comprehension of attention mechanisms in the field of computer vision. These technologies are capable of dynamically and automatically evaluating the importance of data. There are two types of attention mechanisms: soft attention and hard attention [11]. Soft attention calculates a weighted average to generate the gradient context vector and may be used with traditional backpropagation training. On the other hand, hard attention relies on reinforcement learning and employs stochastic inputs, but it 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, the task of defining ‘attention’ in these systems continues to be intricate, with ongoing discussions regarding its essence, particularly when compared to human visual attention [12, 13].

In our proposed method we utilized the detection model for Arabic sign language detection using deep learning. The major contributions of our proposed method are given below.

- 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 attention and spatial attention modules.
- Additionally, a cross-convolution module with a constant parameter of vector in the feature extractor for two-way correlated matrices.
- Overall improved model robustness and recognition performance.

This paper is organized as: Section 2 will provide an extensive review of current techniques in the field. Section 3 will look at the fundamental methodology of Arabic Sign Language Recognition (ARSL), discussing its main principles. Section 4 covers the research's implementation and simulation. Section 5 comprises the discussion and comparison while Section 6 covers the conclusion of the proposed work.

2. Related Work

The incorporation of gesture recognition technology, namely in the domain of Arabic Sign Language (ArSL), has represented a notable advancement in enabling communication between those with speech impairments and computer systems [14]. This technological innovation is crucial for the identification and comprehension of ArSL, which possesses its distinct repertoire of gestures and facial expressions [15, 16]. Utilizing these strategies significantly reduces Deaf ASL users’ communication barriers and increases the ability for them to participate in a wide range of professional and social settings. There are a lot of benefits in the concept of integrating the Automatic Sign Language Recognition (ASLR) system into Arabic Sign Language (ArSL), the embodying of ASLR in ALS Malaysian, could be used to serve the deaf community, as well as social custom which will give a great opportunity and open
new phased to deaf community members about Arabic speaker. It also shows the importance of catering technology designed for every country which emphasizes linguistics and culture. However, there is a lack of a comprehensive database on only fingerspelling, isolated sign, and continuous sign which may be a challenging issue for the advancement of sign language recognition technology [17]. The Sign Language recognition system on manual alphabets has a image accuracy rate at 93.55% by using Adaptive Neuro-Fuzzy Inference System (ANFIS) and feature vector extraction [18], or the ArSL dataset, the ANFIS technique 122 outperform the polynomial classifier. The reason for polynomial classifier outperforms the ANFIS may be due to a lack of consistency in the training data, if the training data is more consistent and comprehensive, it may help to improve the sign language acknowledgment early research houses [19].

In this study, we introduce a novel technique to accurately recognize [20] hand signals given by basketball referees from game footage. Our technique exploits the performance of image segmentation algorithms and combines Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP) features together. By using LBP and a Support Vector Machine, the experimental results showed that the proposed technique can achieve a 95.6% success rate. To accurately track our hand and finger moves we used a Leap Motion device [21]. To be able to detect several gestures, we used our innovative-hidden Markov classification (HMC) algorithm. Additionally, we also used motion detection, gesture recognition, and data cleansing to assess the performance of the system, measuring success in Words Per Minute (WPM) and the Mistake Rate, using the minimum string distance (MSD).

CNN combined with Long Short-Term Memory (LSTM) or Bidirectional LSTM (LSTM) has become a popular technique of sign language recognition [16]. A highly effective technique of detecting Indian and Russian sign languages accurately using deep neural networks and computer vision is proposed in this paper. It can detect the meaning of both manual and non-manual components of the languages perfectly. The spatial information is extracted by the 3D Convolutional Neural Network (3D-CNN), which results in 92% accuracy by the 2D Convolutional Recurrent Neural Network (2D-CRNN) and 99% accuracy by the 3D-CNN [22]. Researchers obtained a 92% accuracy rate by utilizing a two-dimensional Convolutional Recurrent Neural Network (2D-CRNN) and a 99% accuracy rate with a three-dimensional Convolutional Neural Network (3D-CNN). These models were tested on a dataset consisting of 224 movies, where five signers performed 56 distinct signs [15].

This study [23] aims to increase sign language identification accuracy. This will be done by developing three advanced deep-learning models utilizing YOLOv5x and attention. 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 [24] in conjunction with the ArSL21L dataset to achieve this. The YOLOv7 medium model outperformed YOLOv5 variants in terms of mean Average Precision (mAP) ratings. More precisely, the YOLOv7 medium model achieved a score of 0.8306 for mAP0.5:0.95. Additionally, the YOLOv7-tiny model fared better than both the YOLOv5 small and medium models. The YOLOv5 tiny model achieved the lowest scores, with a mean average precision (mAP) of 0.9408 at an intersection over union (IoU) threshold of 0.5, and a mAP of 0.7661 within the IoU range of 0.5 to 0.95.

Researchers [25] presented the ArabSign dataset, which consists of 9,335 video clips from six persons. They also developed an encoder-decoder model for recognizing sign language sentences and achieved a word error rate (WER) of 0.50 on average. The authors [26] introduced a Transformer model based on stance, specifically tailored for the KArSL-100 dataset. This dataset consists of 100 different classes and is focused on recognizing sign videos and were able to attain 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 impose constraints on their practical implementation.

3. Proposed Method

A deep CNN-based detection model with an adjusted backbone and detection head is employed by the suggested method. The model employs a YOLO that has already been trained with modified layers to identify Arabic sign language. To extract features, our suggested model makes use of the attention module in conjunction with channel and spatial attention. All the parts of the newly built cross-convolution module share the same parameter vector. The feature extractor greatly benefits from this module, which is designed to handle matrices with two-way correlation. The initial parameter vector is used to carry out focused convolution. Such focused convolution allows interaction and correlation of the features in the incoming data; thus, having a high potential for capturing the complex relationships within the data effectively, as exemplified by the detection of ArSL. The model architecture of ArSL is shown in Figure 1 which includes the backbone and detecting head.
3.1. Dataset

The foundation of our research is the Arabic Sign Language Letters dataset (ArSL21L). This extensive universal letter signage collection is essential to our research. With 14,202 photos, each representing one of the 32 unique letter signs in Arabic Sign Language, this dataset is crucial to our research. Sample images from the dataset are visualized in Figure 2. The unique contribution of the ArSL21L dataset is the deployment of an extremely diverse group of contributors (50 individuals). This diversity of experts provides the dataset with a rich tapestry of viewpoints and origins. We are lucky to have such a comprehensive and productive instrument available to us for this research. It not only permits us to have a multitude of measurements to work with for a detailed analysis and firm results but also validates our research as viable as our participants are incredibly diverse in their orientation, cultures, and aspirations. You can locate the dataset at Mendeley’s repository, and it is freely accessible [27].
3.2. CNN Based Detection Model

Due to its incredible speed and accuracy, one method that is particularly prominent in real-time computer vision is “You Only Look Once” (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 very precisely all at once [28].

An ordinary YOLO model overlays a $s \times s$ 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

$$\text{Score}_{\text{confidence}} = \text{prob(obj)} \times \text{IoU}_{\text{actual,estimated}}$$

(1)

The object's presence probability is $\text{prob(obj)}$, which ranges from 0 to 1. Here, 0 means the object is absent and 1 means it's likely present. $\text{IoU}_{\text{actual,estimated}}$ estimated using the Intersection over Union measure, the estimated bounding box is compared 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 completely 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, 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 real boxes using center coordinates \( (a_{\text{center}}, b_{\text{center}}) \) at the start. The variable \( \zeta_{ij}^{\text{obj}} \) is 1 if an object is in the \( i^{\text{th}} \) predicted box in the \( i^{\text{th}} \) cell, and 0 otherwise. The adjusted equation shows that the box should predict the object with the highest IoU as shown in Eq. 2

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

(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 to 1, applying the square root function enhances the influence of inaccuracies in smaller boxes to a greater extent than in bigger ones expressed in Eq. 3

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

(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 \( \zeta_{ij}^{\text{obj}} \) 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 absent and 0 presented in Eq. 4.

\[
\text{Loss} = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \zeta_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \gamma_{\text{noobj}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \zeta_{ij}^{\text{noobj}} (X_i - \hat{X}_i)^2 + (C_i - \hat{C}_i)^2
\]

(4)

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

\[
\text{Loss}_{\text{class}} = \sum_{i=0}^{S^2} \sum_{\text{obj}} \sum_{c \in \text{Classes}} (p_1(c) - p_1\hat{c})^2
\]  

Redmon et al. [28] introduced YOLO. Multiple YOLO algorithm improvements occurred from further research. To improve accuracy and performance YOLO9000, a real-time object detection system that can recognize 9000 categories. YOLOv3 was introduced with improved gradually over its predecessors [28]. Bochkovskiy et al. [29] introduced YOLOv4, which improved object detection and GPU usage for the model. Zhu et al. [30] introduced YOLOv5 to improve GPU utilization. Later YOLOv6 and v7 were introduced [31, 32]. The most current version of YOLOv8 has been introduced [33].

The proposed technique incorporates a multi-scale feature extraction architecture to improve YOLO v8 object detection. A more advanced feature extractor including Convolutional (Conv) layers, Convolutional Block Attention Module (CBAM), and C3x layers has been added to the YOLO v8 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 can be mathematically expressed in Eq. 6.

\[
F_{\text{extracted}} = \text{CBAM(Convinput)} \oplus \text{SPPF(Convinput)}
\]  

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 up-sampling methods. Refined features are shown presented in Eq. 7.

\[
F_{\text{C3x}} = \text{Upsample}(\text{C3x}(F_{\text{extracted}}))
\]  

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 score, cls scores, and class probabilities. These detection heads are expressed in Eq. 8.

\[
B_i = H_{\text{bbox}}(F_D)
\]

Where, \( B_i \) The spatial coordinates that determine the position of object \( i \) in the image and \( H_{\text{bbox}} \) 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.

\[
P_i = \{\text{DFL}_i, \text{CLS}_i, \text{Prob}_i\} = H_{\text{prob}}(F_D)
\]
Where $H_{\text{prob}}$ refers to the function utilized by detection heads to construct the probability matrix, which includes DFL scores, classification results, and probabilities for each class.

4. Results

This section provides a concise description of the experimental results, their interpretation, as well as 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 $3\times10^{-4}$. The values we used for our parameters were 15 epochs, a batch size of 24, and an image size of $640 \times 640$. 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. The model performance was validated by a series of extensive experiments. For our research, we employed a training dataset consisting of 9927 samples and a separate validation dataset including 4247 distinct samples. In Figure 3 the confusion matrix of the proposed model has been presented.

![Confusion Matrix](image)

**Figure 3**: Confusion matrix of proposed model results

The confusion matrix of the dataset and after normalization are shown in Figure 4 which depicts that after the normalization miss detection rate reduced significantly.
In Figure 5, the 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 a little 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 precisely and completely when metrics/precision(B) reach 0.5 to slightly over 0.8 and metrics/recall(B) reach 0.4 to almost 0.9. Mean Average Precision metrics/mAP50(B) and mAP50-95(B) increase from 0.5 to 0.8 and 0.4 to over 0.7. These increases suggest the model can detect items with more ground truth overlap at various thresholds. Bottom-row validation loss numbers are more variable. Val/box_loss fluctuates but declines from 0.76 to 0.65, unlike its training counterpart. 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.
**Figure 5:** Proposed model training validation precision-recall and different types of loss

In Figure 6 the x-axis of this 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 6:** Proposed model for ArSL detection F1_Confidenc Curve

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

**Figure 7**: Precision confidence curve of proposed detection model for ArSL

In Figure 8 x-axis corresponds to recall, while the y-axis corresponds to precision. The objective in numerous models is to optimize both precision and recall, leading to a position near the upper-right quadrant of the graph. The various gray lines depict the trade-off between precision 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 a mean Average Precision (mAP) of 0.982 at an IoU (Intersection over Union) threshold of 0.5, a commonly employed criterion in object detection tasks. The mapped 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 8: Precision recall curve for the proposed ArSL identification method

Figure 9 shows that our method for ArSL identification is highly effective, validated using recall confidence score which highlights 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. Recall-Confidence score of 100% has been achieved by fine tuning and parameter settings. 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” boldly, we can see one of the factors allowing our method to be so successful: this single line makes up the entirety of the model’s recall, indicating that with high confidence all classes can be recalled. The gray lines, on the other hand, stand for different categories or variants, as multiple lines of different lengths and at different places can be seen. These visual depictions mean that the recall of the model in response to different confidence thresholds varies. This showcases the model's responsiveness and capacity to accurately recognize all relevant instances, even at a confidence threshold of zero. However, it should be noted that this does not necessarily reflect the precision of these predictions. Where a low confidence threshold is used to achieve 100% recall, it often leads to a significant number of false positives. This curve is vital for comprehending the balance between achieving a high recall and the level of confidence in the predictions, which is necessary for optimizing model performance according to the accurate detection of ArSL.
Figure 9: Recall confidence curve of the proposed ArSL detection approach

The proposed model detection performance across different signs has been presented in Figure 10. The results show that the model's correct detection rate is higher in the detection of most of the signs. The proposed detection model shows its robustness in terms of accuracy precision and recall with the lowest training and validation loss.
4.1. 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 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 zero to more than 0.6 and 0.2 to more than 0.7, respectively.
These changes result in more correct positive predictions and improved detection of all positive events. Mean Average Precision (mAP) scores, including mAP50(B) and mAP50-95b, improved significantly. The mAP50(B) has risen from 0 to over 0.8, while the mAP50-95(B) has risen from 0.1 to more than 0.6. IoU scores consistently improve in precision. The numerical patterns demonstrate the model's improved predicting and categorization abilities after training.

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 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.
Figure 12: F1-Confidence curve of the base detection model

In Figure 13 a Precision-Confidence Curve is depicted, illustrating the accuracy of a classification model at different levels of confidence thresholds.

Figure 13: Precision confidence curve of base model for ArSL detection
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 utmost accuracy achieved by the model when the confidence threshold is set to its highest level.

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 stands for the average performance across all classes. The phrase "all classes 0.786 mAP@0.5" stands for a mAP score of 0.786 at an IoU criterion of 0.5. This shows a high level of model performance, on average, across all classes.

Figure 14: Precision recall curve of detection rate of base model

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

A novel technique utilizing a deep CNN model YOLO has been utilized to robustly recognize ArSL. Our proposed method showed competitive performance as compared with the SOTA approaches. Recent researchers have attempted to concatenate different advanced recognition systems to get more promising identification rates for ArSL gestures especially. (ANFIS) proved its pioneering by many researchers; however, it is counteracted by its important weakness in adapting swiftly and performing intricate tasks of gesture recognition which is a very important criterion in any SL 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 [34]. An independent-user based technique employed using deep learning and vision-based techniques to interpret ArSL with 98% accuracy [35]. A YOLOv5 based approach presented [36] for the sign language recognition with mAP of 0.98 with precision of 95%. Researchers successfully recognized Arabic signs using DL based techniques on ArASL2018 dataset with accuracy of 94.46% [37]. YOLOv6 employed to recognize the ArSL using static and dynamic images with 96% accuracy of statics images and 92% accuracy on different continuous sign [38].

The modified model also improved precision and recall rate to 0.99 in recognition and detection of different signs of ArSL. The robustness of the proposed model showed signification improvement in detection and recognition of different of different signs. The model performance decreases the error rate with higher rate of accurate signs recognition. The YOLOv8-based model achieves significant 0.9909 and 0.8306 of mAP at 0.5 and mAP at 0.5:0.95 respectively. The performance on the ArSL21L
dataset of the pretrained model shows decline in recognition rate with higher error rate. We utilized pretrained model as well as with our custom module integration.

Different from the previous methods, this proposed technique contributes to specifying the key unique and discriminative features of the visual data which can minimize the most distracting information and maximize the identification process. In this sense, taking the key roles of visual data in computation, several attention mechanisms were incorporated such as 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 parts of the visual data. By emphasizing the importance of the unique features of the ArSL movements, the proposed technique can reach a better and more comprehensive level of understanding of the ArSL movements. Unlike the specified traditional works, this methodology researches the maximum capability of the prominent features of ArSL rather than using hardcoded features extraction techniques employed in previous studies.

6. Conclusions

In this study, we propose a newly groundbreaking deep learning strategy to effectively separate different sign languages such as American Sign Language (ArSL), thereby narrowing down the language divide between deaf hearing. To explain our method, we built a powerful architecture-intensive CNN combined with YOLO a famous object detection model that authorized the construction of the best working passage in object detection and enhanced using attention mechanics. These are our contributing assets to increase the precision, accuracy, and recognition speed of gesture recognition. ArSL gestures are much higher than any existing approaches. The integration of the self-attention modules, usage of the channel attention and spatial attention module to represent compress and decompress in feature; using the cross-convolution module to mathematically process for split three-way matrix efficiently. We have pushed the state of the art in recognition rates and opened the best way to attempt broad-range applications for ArSL-extracted deep learning in the future. The validation of the model using the ArSL21L highlights its efficacy in accurately finding a diverse assortment of ArSL gestures. This research shows a foundation for future advancements in the domain of sign language recognition, offering the potential for improved inclusiveness and social integration for the deaf community in Arabic-speaking areas and beyond.

Author Contributions: “Conceptualization, F.M. and S.A.; methodology, F.M.; software, F.M.; validation, F.M., H.A. and S.A.; formal analysis, S.A.; investigation, S.A.; resources, S.A.; data curation, F.M.; writing—original draft preparation, F.M.; writing—review and editing, F.M. and H.A;
visualization, F.M.; supervision, S.A.; project administration, S.A.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.”

**Funding:** “This research was funded by King Salman Center for Disability Research, grant number KSRG-2023-512.

**Data Availability Statement:** The dataset utilized in this research is publicly available ("https://data.mendeley.com/datasets/f63xhm286w/1") accessed on (05-Jan-2024)

**Acknowledgments:** The authors extend their appreciation to the King Salman Center for Disability Research for funding this work through Research Group no KSRG-2023-512.

**Conflicts of Interest:** “The authors declare no conflicts of interest.”

**References**


