Comparative Study of Single-Stage vs. Two-Stage Detectors for ASL Gesture Recognition

Xiang Fang and Pablo Rivas

Department of Computer Science, Baylor University, Texas, USA
{Xiang_Fang1,Pablo_Rivas}@Baylor.edu

Abstract. American Sign Language (ASL) enables vital communication for deaf and hard-of-hearing individuals, yet automated recognition of its gestures remains challenging. Here, we assess three leading object detection frameworks, YOLOv11, Faster R-CNN and RT-DETR, on a moderately sized, extensively augmented ASL dataset. By comparing average precision across a range of overlap thresholds (mAP 50–95), as well as measuring inference latency and computational cost, we show that YOLOv11-l strikes the best compromise between speed and accuracy for real-time use. In contrast, RT-DETR-x attains the highest overall precision but demands substantially greater resources. Our results clarify how each model's trade-offs affect performance and lay the groundwork for refining ASL gesture detection. This work brings us closer to practical, responsive systems that can seamlessly interpret sign language in everyday settings.

Keywords: Computer Vision · Object Detection · RT-DETR · YOLO · Faster R-CNN · ASL Recognition

1 Introduction

Object detection has emerged as a pivotal area in computer vision, facilitating a wide range of applications from autonomous vehicles to security surveillance systems. Unlike traditional classification tasks that label an image as a whole, object detection not only identifies objects but also delineates their positions using bounding boxes. This dual capability makes it invaluable in scenarios such as medical imaging, robotics, and real-time monitoring. Early methods relied on handcrafted features, but recent advances involve sophisticated deep learning architectures that leverage convolutional neural networks (CNNs) for enhanced accuracy and efficiency [14,20,16]. Notable frameworks like YOLO (You Only Look Once), Faster R-CNN, and DETR (DEtection TRansformer) have demonstrated remarkable performance across various datasets and real-world conditions [15,3].

Among the many specialized object detection applications is the recognition of ASL gestures, which involves identifying specific hand signs corresponding to letters or words. ASL is a visual language relying on precise hand shapes, orientations, and movements; this complexity poses unique challenges for detection algorithms. For instance, the variability in hand configurations and the

influence of lighting or cluttered backgrounds can significantly impact detection performance. Moreover, some ASL gestures are visually similar, leading to potential misclassifications, while real-time requirements demand low-latency solutions [6,9]. Although frameworks such as Faster R-CNN incorporate region proposals to enhance speed and accuracy, they may still struggle with rapid or subtle hand movements [15,8]. YOLO, known for real-time processing, can find it difficult to differentiate between visually similar gestures if the bounding box predictions do not capture nuanced differences [7]. DETR employs a transformer-based detection approach that shows promise for complex scenes but may require adaptations to handle the temporal aspects inherent to ASL signs [3].

Despite notable progress in general object detection methods [11,4,2,18,17], there is a significant gap in their application to the dynamic and subtle task of ASL gesture recognition. Current literature has explored CNN-based strategies for hand gesture detection, but comprehensive evaluations comparing multiple cutting-edge models specifically tailored for ASL remain scarce [6,9]. In this paper, we focus on systematically assessing state-of-the-art object detection frameworks, including YOLO, Faster R-CNN, and DETR variants, to improve the accuracy and robustness of ASL detection. By leveraging datasets specifically curated for ASL and examining how these models handle variability in gestures and environmental factors, we aim to shed light on effective approaches that bridge the gap between general detection capabilities and the nuances of ASL.

The paper makes the following key contributions:

- A comprehensive comparison of modern object detection models (e.g., YOLO, Faster R-CNN, DETR) for ASL gesture recognition.
- An investigation into how these methods address challenges such as gesture similarity, background complexity, and real-time requirements.
- Practical insights into model adaptations and configurations that can enhance accuracy and robustness in ASL detection tasks.

2 Background and Related Work

Over the years, object detection has progressed from rudimentary classification tasks to more sophisticated frameworks that integrate segmentation and multimodal strategies. Central to these advances is the distinction between anchor-based and anchor-free approaches. Anchor-based methods, exemplified by Faster R-CNN and YOLOv3/v4 [14], rely on a predefined set of bounding boxes (or anchors) distributed at multiple scales and aspect ratios across an image. The model then refines these anchors by comparing them against ground truth objects using the Intersection over Union (IoU) metric, adjusting coordinates to more precisely capture each target. Although such strategies handle varying object sizes effectively, they can be computationally demanding due to the volume of anchors that must be evaluated.

In contrast, anchor-free approaches, including YOLOv8 and DETR [16], bypass predefined anchors by predicting object centers or key points directly. This mechanism reduces computational overhead and often proves beneficial for objects that exhibit extreme aspect ratios or appear in crowded scenes. Moreover, models in this category typically focus on a more streamlined detection pipeline, allowing them to maintain robust accuracy without the burden of extensive anchor generation.

Beyond the anchor-based and anchor-free dichotomy, object detection frameworks are frequently classified by the number of inference stages. Single-stage detectors, such as YOLO and RetinaNet, merge region proposal and classification within a unified pipeline, favoring real-time performance and simpler deployment scenarios. However, these systems may encounter limitations when detecting small or overlapping objects compared to more elaborate architectures. Two-stage detectors, including Faster R-CNN and Mask R-CNN [20], separate region proposal from subsequent refinement, with the first stage identifying candidate bounding boxes and the second stage focusing on classification and bounding box adjustment. While this paradigm often achieves higher accuracy, it introduces additional computational demands.

Among the prominent models in the literature, YOLO partitions an image into grids to simultaneously learn bounding box predictions and class probabilities, using non-maximum suppression to consolidate overlapping detections [14]. Faster R-CNN adopts a two-stage structure: it employs a Region Proposal Network (RPN) to identify likely object regions, followed by classification and bounding box fine-tuning through ROI pooling and fully connected layers [20]. DETR [16] applies a transformer-based encoder-decoder mechanism that identifies objects in a single feed-forward pass, dispensing with anchors entirely and relying on learned queries to capture and refine target locations. Collectively, these methods underscore the diverse strategies for object detection, each tailored to address specific challenges related to speed, accuracy, and the complexities of real-world imagery.

3 Dataset and Preprocessing

Preparing high-quality training data is essential for achieving robust and accurate detection. In this section, we describe the dataset composition, as well as the preprocessing and augmentation strategies used to diversify the training samples and improve model generalization.

3.1 Dataset Details

The dataset employed in this study consists of 1,728 original images, split into 1,512 training images, 144 validation images, and 72 test images [5]. Each image was uniformly resized to 384×384 pixels, a step that ensures consistent input dimensions across different models and simplifies image scaling considerations during training.

3.2 Data Augmentation

To augment the dataset and mitigate overfitting, over 20 augmentation techniques were applied, expanding the training set to 15,110 images and the validation set to 2,801. These techniques included rotations and flips, thereby exposing the models to varied orientations. Adjustments to the RGB channels and hue-saturation shifts were introduced to emulate different lighting and color conditions, while coarse dropout simulated occlusions by randomly masking out small regions of the image. Additionally, random cropping resized subregions back to 384×384, enhancing the models' ability to detect objects in diverse spatial contexts. Examples of the original and augmented images are illustrated in Fig. 1, highlighting the visual transformations achieved through these strategies.

4 Experiment Settings

This section describes the chosen detection architectures, their training protocols, the metrics used to evaluate performance, and the loss functions that guide model optimization. It concludes with a brief presentation of training curves, offering insights into convergence behavior and validation outcomes.

4.1 Models Used

Three primary detection frameworks, YOLOv11, Faster R-CNN, and RT-DETR, were examined in this study. YOLOv11, implemented through the Ultralytics package, provides three variants (n, l, x) that differ in size and computational demand [1,10]. This design integrates spatial attention mechanisms and multi-scale detection layers to handle densely cluttered scenes more effectively. The YOLO architecture is known for its inference-time efficiency. Faster R-CNN adopts a two-stage strategy by leveraging an RPN to generate candidate object regions, followed by classification and bounding box refinement in a subsequent phase [16,19]. This approach utilizes well-known backbones such as ResNet or VGG. Meanwhile, RT-DETR employs an encoder-decoder structure with hybrid feature extraction and attention-driven intra-scale as well as cross-scale fusion [20,10]. All models used in the experiments are pretrained on the COCO dataset to enable a fair comparison [12].

4.2 Training Setup

To accommodate varying memory requirements, models were trained on different GPU configurations using the AdamW optimizer [13] for 15 epochs. In particular, YOLOv11-n, YOLOv11-l, Faster R-CNN, and RT-DETR-l were trained on an RTX-4060 GPU equipped with 8GB of memory, whereas YOLOv11-x and RT-DETR-x required the higher-capacity RTX-6000 GPU (48GB) for efficient training. These hardware allocations ensured that each model variant could be optimized without encountering memory limitations, thereby facilitating a balanced evaluation of performance and resource utilization.

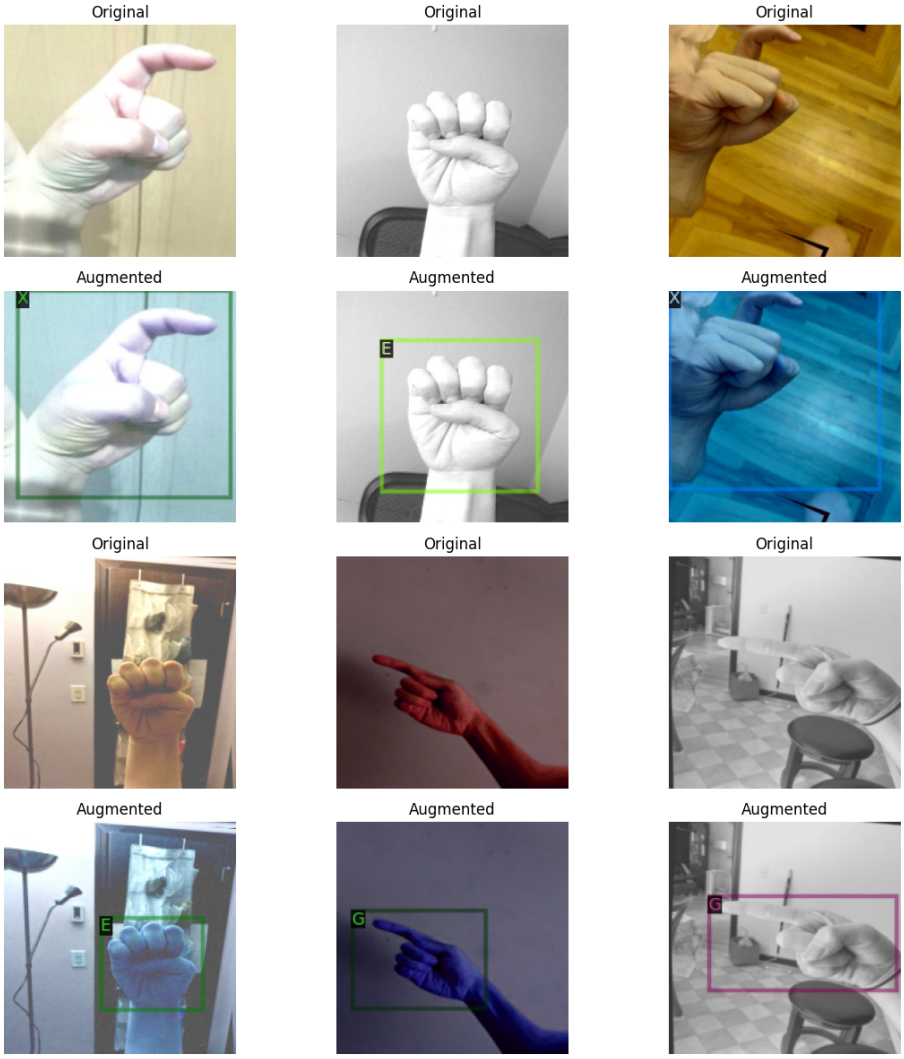


Fig. 1: Original and Augmented Images.

4.3 Evaluation Metrics

Model accuracy and detection capabilities were assessed using standard metrics in object detection. Average Precision (AP) measures precision across varying recall levels, while Mean Average Precision (mAP) reflects the average of AP values over all classes:

$$\text{AP} = \int_0^1 P(r)\,dr, \quad \text{mAP} = \frac{1}{N}\sum_{c=1}^{N} \text{AP}_c. \tag{1}$$

Intersection over Union (IoU) quantifies the overlap between predicted and ground truth bounding boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \tag{2}$$

In addition, mAP at a 50% IoU threshold (mAP 50) and the more stringent range-based metric (mAP 50–95) were employed. The latter captures performance under multiple IoU thresholds (from 0.50 to 0.95 in increments of 0.05), yielding a comprehensive overview of detection robustness across a variety of object sizes and spatial overlaps.

4.4 Loss Functions

Each model employs a specific loss function to reconcile predicted bounding boxes and class scores with ground truth annotations. Faster R-CNN optimizes a sum of losses from the RPN and the subsequent detection head:

$$L_{\text{Faster R-CNN}} = L_{\text{RPN}} + L_{\text{Detection}}, \tag{3}$$

where the RPN combines classification and Smooth L1 regression terms, and the detection branch refines bounding boxes identified by the RPN. YOLO focuses on bounding box regression, objectness prediction, and class probabilities using a combined loss:

$$L_{\text{YOLO}} = \lambda_{\text{box}} L_{\text{box}} + \lambda_{\text{obj}} L_{\text{obj}} + \lambda_{\text{cls}} L_{\text{cls}}, \tag{4}$$

where bounding box regression leverages a Complete IoU (CIoU) formulation that incorporates overlap, center distance, and aspect ratio. RT-DETR applies a multi-component loss composed of classification, L1 regression, and Generalized IoU (GIoU) terms:

$$L_{\text{RT-DETR}} = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{L1}} L_{\text{L1}} + \lambda_{\text{GIoU}} L_{\text{GIoU}}, \tag{5}$$

and penalizes non-overlapping regions by factoring in the smallest enclosing box that captures both the predicted and ground truth bounding boxes.

4.5 Training and Validation Plot

Model convergence was monitored by comparing training loss and validation mAP 50–95 over the course of training. As illustrated in Fig. 2, tracking these curves helps to identify underfitting or overfitting trends and informs decisions about hyperparameter tuning or early stopping, ultimately guiding model selection for the final evaluation.

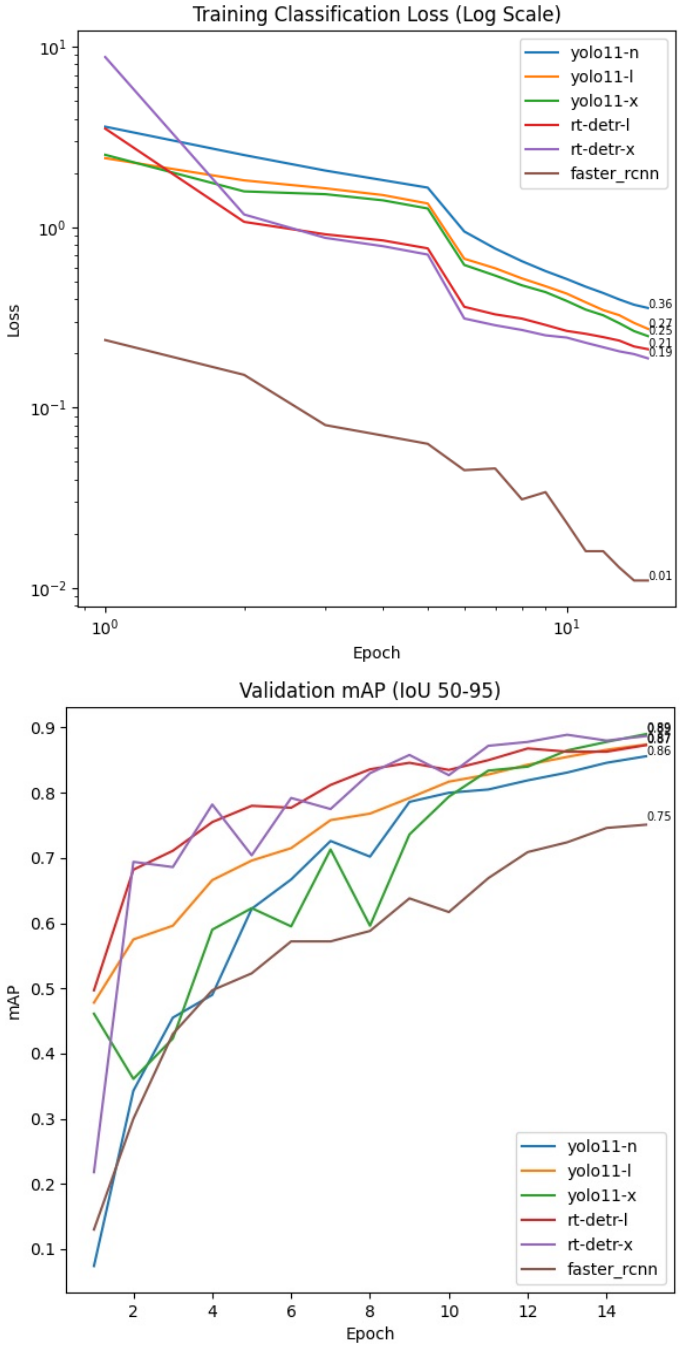


Fig. 2: Training loss and validation mAP50-95 performance.

Table 1: Performance comparison of object detection models. Bold values represent the best result in each column.

Model	Latency (ms)	mAP 50-95	Params (M)	Training Time (s)
YOLOv11-n	**3.32**	0.692	**2.50**	160
YOLOv11-l	7.49	0.777	25.3	170
YOLOv11-x	13.85	0.778	56.9	**140**
RT-DETR-l	9.50	0.753	32.9	400
RT-DETR-x	14.40	**0.793**	67.4	215
Faster R-CNN	20.69	0.555	32.3	440

5 Results and Discussion

This section analyzes the empirical outcomes for each model, examining both quantitative performance metrics and qualitative aspects such as misclassifications. Table 1 summarizes the trade-offs between latency, accuracy, and computational complexity, while Fig. 3 illustrates model-specific confusion patterns. An in-depth look at common error cases is presented next.

5.1 Quantitative Evaluation

As shown in Table 1, YOLOv11-n achieves the quickest inference times but at the expense of diminished accuracy, highlighting its suitability for scenarios that prioritize speed over precision. By contrast, YOLOv11-x exhibits notably higher mAP, yet demands more extensive computational resources during both training and inference. Notably, YOLOv11-l balances these factors, providing a strong compromise between speed and accuracy.

RT-DETR-l and RT-DETR-x also illustrate different performance trade-offs. RT-DETR-l offers moderate inference latency while attaining mAP scores competitive with YOLOv11-l, thereby serving as a plausible middle ground for real-time tasks. RT-DETR-x improves further on accuracy, delivering the highest mAP among all tested models, but requires substantially more computational overhead. Faster R-CNN, while achieving acceptable detection quality, shows the lowest mAP and the highest latency among the compared methods, which is largely attributable to its two-stage detection scheme. Additionally, the training time for Faster R-CNN is among the longest, aligning with expectations for two-stage detectors that generate and refine region proposals.

5.2 Qualitative Insights and Error Analysis

To gain deeper insights into model behavior, we examined both confusion matrices and individual misclassified samples. Fig. 3 depicts the confusion matrix for YOLOv11-l on the test set, revealing that most gestures are correctly identified

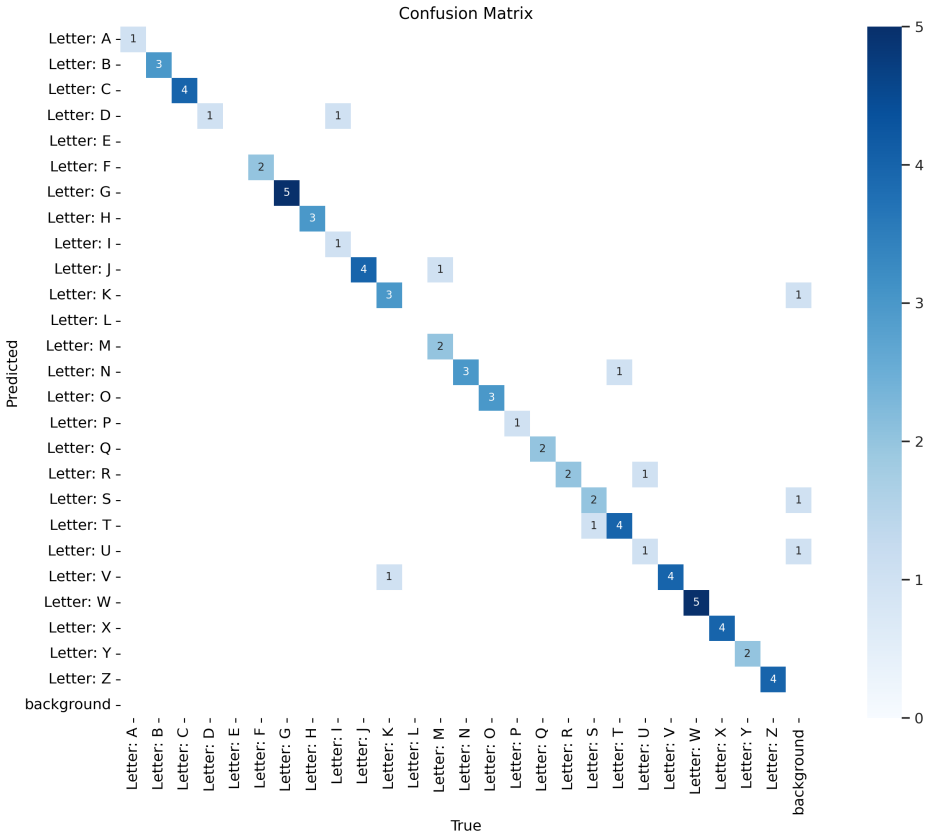
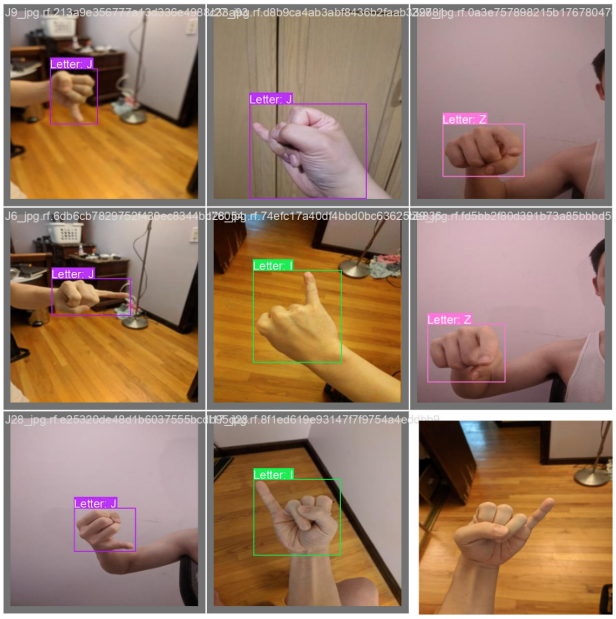


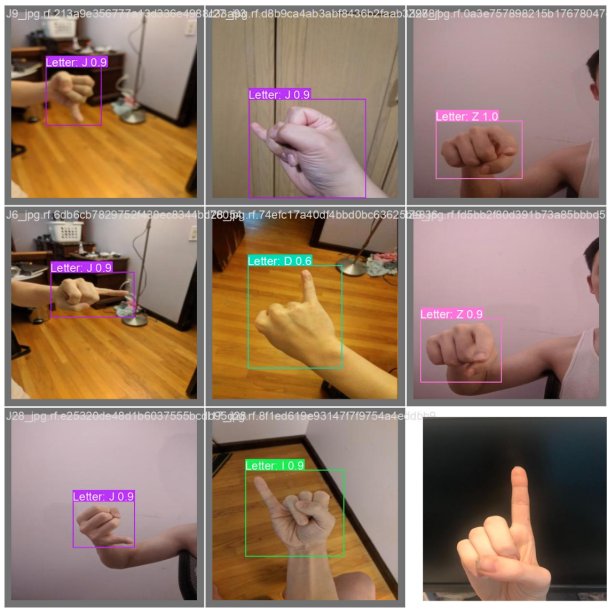
Fig. 3: Confusion matrix of YOLOv11-l on the test dataset.

with high confidence. However, certain categories display recurring confusion, underscoring the need for more training examples that capture visual nuances such as hand orientation or lighting conditions.

A closer look at specific errors is shown in Fig. 4, where the left grid indicates the ground truth and the right grid illustrates the predicted outputs. In one instance, the letter "I" is classified incorrectly as "D," likely due to the camera angle emphasizing the overall shape rather than the key differentiating feature (the position of the finger). The predicted confidence of 0.6 suggests model uncertainty. This issue hints that greater emphasis on subtle features (e.g., the front versus back of the hand) might be necessary. Accordingly, augmenting the dataset with more varied samples of visually similar letters or incorporating pose estimation components could help alleviate such misclassifications. More broadly, these observations imply that refined attention mechanisms and targeted data enrichment may enhance the model's sensitivity to fine-grained gesture differences.



(a) Ground Truth.



(b) Predicted Output.

Fig. 4: Comparison of ground truth (a) and predicted output (b).

Overall, the findings demonstrate that while high-accuracy models generally demand more computation, practical deployment may necessitate compromises among speed, resource constraints, and detection fidelity. Future work could explore specialized architectures or hybrid strategies that build on the strengths of each approach, potentially improving upon both real-time responsiveness and discriminative power in ASL gesture detection.

6 Conclusion and Future Work

This study has demonstrated the utility of advanced object detection architectures for recognizing ASL gestures. Our empirical results highlight the trade-offs between speed, accuracy, and resource consumption that define each model's suitability for specific deployment scenarios. Among the evaluated approaches, YOLOv11-l emerged as a strong candidate when balancing accuracy with low latency and parameter efficiency, making it well-suited for real-time applications on limited hardware. Meanwhile, RT-DETR-x attained the highest mAP 50–95 and thereby underscored the benefits of encoder-decoder structures coupled with attention mechanisms, albeit at a higher computational cost.

Moving forward, further investigation into multimodal strategies and more complex datasets could enhance model robustness and extend detection beyond isolated hand signs. Integrating techniques such as hand tracking, multi-object detection for overlapping gestures, and context-aware predictions may significantly improve generalization and resilience to challenging environments. Moreover, refining real-time performance, particularly for high-capacity models like RT-DETR-x, remains an essential objective in practical system deployments.

Finally, addressing continuous gesture representations, such as letters j and z that involve motion sequences, requires broader techniques that capture spatio-temporal dynamics. Future work could explore the capabilities of frameworks such as Spatio-Temporal Action Localization (e.g., YOWO) to detect and interpret gestures over multiple frames. These advances hold promise for a more comprehensive understanding of ASL gestures, paving the way for richer communication tools and enhanced accessibility for diverse user groups.

Acknowledgments. The authors thank the Rivas.AI Lab (`https://lab.rivas.ai`) for the support and helpful feedback throughout this project. Part of this work was funded by the National Science Foundation under grants CNS-2210091, OPP-2146068, CHE-1905043, and CNS-2136961; and by the Department of Education under grant P116Z230151.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alif, M.A.R.: Yolov11: A state-of-the-art object detection model. arXiv preprint arXiv:2410.22898 (2024), `https://arxiv.org/pdf/2410.22898`

2. Bejarano, G.M., Huamani-Malca, J., Cerna-Herrera, F., Alva-Manchego, F., Rivas, P.: Perusil: A framework to build a continuous peruvian sign language interpretation dataset. In: Proceedings of the LREC 2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources. pp. 1–8 (2022)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers pp. 213–229 (2020). https://doi.org/10.1007/978-3-030-58452-8_13
4. Das Jui, T., Bejarano, G.M., Rivas, P.: A machine learning-based segmentation approach for measuring similarity between sign languages. In: Proceedings of the LREC 2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources. pp. 94–101 (2022)
5. Datasets, A.: American sign language letters dataset (2024), https://public.roboflow.com/object-detection/american-sign-language-letters/1, accessed: 16-Feb-2025
6. Delforouzi, A., Pamarthi, B., Grzegorzek, M.: Training-based methods for comparison of object detection methods for visual object tracking. Sensors **18**, 3994 (2018). https://doi.org/10.3390/s18113994
7. Ding, L., Liu, G., Zhao, B., Zhou, Y., Li, S., Zhang, Z., Guo, Y., Li, A., Lu, Y., Yao, H., Yuan, W., Wang, G., Zhang, D., Wang, L.: Artificial intelligence system of faster region-based convolutional neural network surpassing senior radiologists in evaluation of metastatic lymph nodes of rectal cancer. Chinese Medical Journal **132**, 379–387 (2019). https://doi.org/10.1097/cm9.0000000000000095
8. Girshick, R.: Fast r-cnn (2015). https://doi.org/10.1109/iccv.2015.169
9. Hu, G., Yang, Z., Hu, L., Huang, L., Han, J.: Small object detection with multiscale features. International Journal of Digital Multimedia Broadcasting **2018**, 1–10 (2018). https://doi.org/10.1155/2018/4546896
10. Jocher, G., Qiu, J.: Ultralytics package: Yolov11 and beyond (2024), https://github.com/ultralytics/ultralytics, gitHub Repository, Licensed under AGPL-3.0
11. Lazo Quispe, C., Huamani Malca, J., Bejarano Nicho, G., Huaman Ramos, M., Rivas, P., Cerny, T.: Impact of pose estimation models for landmark-based sign language recognition. In: LXAI Workshop @ Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022) (2022)
12. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context. arXiv preprint arXiv:1405.0312 (2015), https://arxiv.org/pdf/1405.0312
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2019), https://arxiv.org/pdf/1711.05101
14. Redmon, J., Farhadi, A.: Yolov1: You only look once - unified, real-time object detection. arXiv preprint arXiv:1506.02640 (2016), https://arxiv.org/pdf/1506.02640
15. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. Ieee Transactions on Pattern Analysis and Machine Intelligence **39**, 1137–1149 (2017). https://doi.org/10.1109/tpami.2016.2577031
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497 (2015), https://arxiv.org/pdf/1506.01497v3

17. Rivas, P., Dand, D., Rivas, E., Velarde, O., Gonzalez, S.: A deep learning approach to sign language recognition using stacked sparse autoencoders. In: LXAI Workshop @ Neural Information Processing Society Conference (NeurIPS). p. 3 (2018)
18. Rivas, P., Rivas, E., Velarde, O., Gonzalez, S.: Deep sparse autoencoders for american sign language recognition using depth images. In: 21st International Conference on Artificial Intelligence (ICAI 2019) (2019)
19. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2: A pytorch-based object detection library (2019), `https://github.com/facebookresearch/detectron2`, gitHub Repository
20. Zhu, C., Du, Z., Wang, H.: Rt-detr: Real-time detection transformer. arXiv preprint arXiv:2304.08069 (2023), `https://arxiv.org/pdf/2304.08069`