Deep Learning for Sign Language Detection and Caption Translation

Hamsalakshmi Ramachandran, Sugandha Chauhan, Himani Shah,
Manjot Singh, Saqib Chowdhury, Kush Bindal

Department of Applied Data Sciences
San Jose State University

***Abstract*—The study offers a whole deep learning framework that can realize detection of sign language and caption translation and combine the cutting-edge computer vision and natural language processing technology. From this system, semantic hand segmentation and gesture classification are integrated with multilingual translation to establish a reliable basis for ASL glosses recognition and interpretation. For hand region extraction to be accurate, we utilized the EgoHands and WLAS. ResNet-18 was used to classify the hand regions after segmentation, which lead to accurate recognition of 50 different ASL glosses. For improving usability, the system rendered the guessed glosses into Hindi using the Helsinki-NLP MarianMT translation service. The assessment showed the system's strengths in segmentation in hand regions, glosses identification and translation, with remarkable results in segmentation (according to F1-score), classification, and quality of the translation (accordingly to BLEU score). Our framework provides a scalable and flexible system that enables continuous sign language interpretation in an educational context, accessibility, and overcoming linguistic barriers.**

***Index Terms*—Sign Language Recognition, Deep Learning, Computer Vision, Hand Segmentation, Action Recognition, ASL, WLASL, EgoHands, ResNet-18, Transformer, Translation.**

I. Introduction

Deaf and Hard-of-Hearing communities highly depend on sign language as the dominant means of communication with distinguishable visual-gestural expressions used for conveyance and reception of messages. A language with a very wide use in the North American region, ASL has a special grammar system, is characterized by agile movements of the signs, and it calls to use the spatial positioning setting it apart from other sign languages. While sign language is a critical part of the overall process of bridging the divide between ASL fluent people and non-signers, this requires ASL recognition and translation technologies continuously.

The developments of deep learning and computer vision have brought to light the development of automated systems that are geared to identify and translate sign gestures. Unfortunately, current systems tend to continuously experience difficulties in satisfying the performance requirement and reliably generalize under lighting variations, signer variations, and complex backgrounds. The overall objective of the present work is to design an end-to-end deep learning system which facilitates recognition of ASL and caption translation using sophisticated segmentation, classification as well as translation techniques. We begin by performing hand segmentation using six state-of-the-art models, namely U-Net, E-Net, HRNet, Mask R-CNN, DeepLabV3, and YOLOv8m-Seg, to train on the EgoHands dataset for training the model to correctly extract hands. After segmentation of the hands on the WLASL dataset, isolated areas are classified by a ResNet-18 model, thereby allowing the successful differentiation of 50 different ASL gestures. The selected glosses were then translated into another language – Hindi – using the Helsinki-NLP MarianMT model, which thus provided for multilingual interaction and increased accessibility for the users of sign language. Using sophisticated hand segmentation, credible gesture identification, and functional linguistic translation, this system enables ASL communicators to communicate with non-signers, addressing educational, accessible, and transcultural communication needs.

II. Related Work

Research on automated sign language recognition and translation has evolved through distinct waves of innovation—beginning with rule-based, multidisciplinary systems in the mid-2000s and advancing to today's deep-learning–driven, real-time, multilingual frameworks. The foundational work by Parton, B. S. (2006) introduced a hybrid AI pipeline combining linguistic rules with computer-vision features to segment and classify static hand shapes . Although capable of parsing simple alphabetic signs, these early systems struggled with signer variability and dynamic gestures. Camgoz, N. *et al.*(2020) then leveraged self-attention–based transformers for end-to-end sign recognition and translation, achieving both gloss prediction and sentence-level translation within a single architecture.

Deep CNNs soon became the core of static sign classifiers. Wadhawan, A., & Kumar, P. (2020) showed that a straightforward CNN architecture could outperform handcrafted-feature systems on isolated ASL words. Papastratis *et al.*(2021) provided a comprehensive survey of sensor-based and vision-based AI techniques for sign language, highlighting enduring challenges in dataset diversity and signer generalization. Yadav *et al.* (2021) built a browser-based audio-to-sign system using Flask and YOLO.

El Zaar *et al.*. (2022) extended this approach across multiple national sign languages, underscoring CNNs' scalability. Web-deployed, real-time systems quickly followed. Samonte *et al.*(2022) showcased a full translator that combined YOLO for hand detection with a lightweight classifier to generate text subtitles. Wahane *et al.* (2022) demonstrated real-time recognition for small vocabularies. Kothadiya *et al.* proposed DeepSign, a robust multi-language detector using ensemble CNNs.

To capture dynamic gestures, researchers began fusing spatial and temporal streams. Triwijoyo *et al.* (2023) combined image features with hand-landmark sequences using multi-headed CNNs, while Buttar *et al.* (2023) integrated temporal convolutions and residual blocks to handle continuous signing. Strobel *et al. (2023)* then applied design-science research methods to assess usability and deployment barriers of AI translation systems in real-world contexts.

At the same time, efforts to unify detection and classification intensified. Jana *et al.* (2024) fused YOLO with LSTM-based captioning to generate running translations of ASL sequences. Zhang, Y., & Jiang, X. (2024) surveyed the latest deep-learning breakthroughs in sign language recognition, while Zhang *et al.*(2024) introduced EvSign, an event-based framework for low-latency translation using neuromorphic cameras.

Translation-focused studies have also gained traction. Sharma *et al.*(2024) developed an Indian Sign Language web translator leveraging sequence-to-sequence models, and Tian *et al.* (2024) outlined an inclusive communication framework to bridge AI and sign language gaps. Sharma *et al.* (2024) also launched a public-facing deep-learning awareness portal for sign language.

Two cautionary notes have emerged. Pathan *et al.* (2023) was retracted for methodological flaws in multimodal fusion approaches, highlighting the need for rigorous validation. Meanwhile, Najib, F. M. (2025) proposed novel sequence-to-sequence models for sign interpretation but awaits independent replication.

Recent studies have effectively applied deep-learning segmentation models such as DeepLabV3 and DeepLabV3+ for isolating hand regions in sign language recognition tasks. The work by Bchir (2020) employed DeepLabV3+ to segment Arabic Sign Language alphabet gestures, achieving high pixel-wise accuracy and IoU, which supported precise gesture localization. Similarly, Aly and Aly (2020) integrated DeepLabV3+ into a signer-independent Arabic Sign Language pipeline, where segmented hand masks were used to improve temporal gesture modeling with Bi-LSTM. In another application, Rajan and Rajendran (2021) utilized DeepLabV3 for hand segmentation in American Sign Language and combined deep and handcrafted features, achieving high recognition accuracy under varying backgrounds.

Chanda and Nyeem (2022) first demonstrated that U-Net segmentation on the NUS Hand Posture Dataset II (2000 images, 10 classes) could boost static sign recognition to 97.15% by feeding binary hand masks into CNNs such as Inception V3, VGG16/19, and ResNet50. Chung *et al.* (2022) then combined U-Net with an ensemble of VGG19, ResNet-50, and MobileNet—fused via two fully connected layers—and reached 99.86% accuracy on real-time ASL alphabet translation. Most recently, Md. Shaheenur Islam Sumon *et al.* (2024) added temporal modeling by applying a DenseNet-backboned U-Net to each frame of a custom 30-word ASL video dataset and feeding the segmented sequences into an LRCN (CNN+LSTM), achieving 93.66% on segmented "pose" videos (92.66% on raw), confirming that precise segmentation plus motion context yields the strongest sign-gesture recognition.

Several prior works have explored sign language recognition using Mask R-CNN model, though they differ significantly from the objectives and methodology of the present study. Sewwantha, R. R., Ginige, T. (2021), propose a Mask RCNN-based model for classification of ASL alphabet signs. Their work is based on static image signs unlike ours which is based on videos. While their system achieves respectable accuracy on isolated hand signs, it lacks translation capabilities. The study by Alawwad, R. A. *et al.*(2021) introduces a Faster R-CNN framework to recognize Arabic sign gestures by detecting hands using bounding boxes. Unlike our approach, which employs Mask R-CNN for pixel-level segmentation and supports gesture classification and multilingual translation, the authors focus solely on detection without end-to-end interpretation. The work by Hoque, O. B. *et al.* employs Faster R-CNN for detecting Arabic sign gestures from images and bounding boxes, distinguishing it from our pipeline, which integrates Mask R-CNN–based segmentation, ResNet-18 gesture classification, and language translation using Helsinki-NLP.

Recent advancements in segmentation models have been significantly influenced by the introduction of ENet, a lightweight and efficient model proposed by Paszke et al. (2016) , designed for real-time semantic segmentation, ENet employs an optimized encoder-decoder architecture, achieving 18× faster inference with 75× fewer FLOPs than conventional models, making it a benchmark for real-time applications. It was extensively tested on Cityscapes, CamVid, and SUN RGB-D, demonstrating robust performance with minimal computational cost. Building on this foundation, Yun and Park (2022), developed a hybrid approach combining ENet for segmentation and YOLO for object detection, tailored for lightweight embedded devices . This integration not only improved detection accuracy but also maintained efficiency, aligning with the need for real-time applications. In the medical domain Lau F et al. (2017), leveraged ENet's precision for tissue segmentation in 3D confocal microscopy, specifically analyzing collagen structures in rat hearts. Their study revealed critical structural differences between healthy and hypertensive hearts, demonstrating how precise segmentation can uncover vital insights into tissue health and function. These works collectively underscore ENet's versatility and impact across various domains, from efficient object detection to high-precision medical imaging.

As deep learning for sign language processing continues to evolve, the fusion of multimodal technologies, real-time captioning, and scalable translation frameworks promises to

revolutionize communication for the deaf and hard-of-hearing communities. The progress in this field attests to the power of deep learning and the advancements in artificial intelligence technologies in breaking communication barriers and enhancing inclusivity.

Recent advancements in sign language recognition have increasingly leveraged deep learning and object detection models to improve accuracy and real-time performance. A 2023 study titled *Performance Evaluation of ResNet Model on Sign Language Recognition* by Agangiba *et al*. evaluated multiple ResNet architectures (ResNet18 to ResNet152) on both American and Indian Sign Language datasets, concluding that deeper networks like ResNet152 offered superior accuracy, especially for complex two-handed gestures. Building on this, *American Sign Language Detection using YOLOv5 and YOLOv8* by Tyagi *et al.,*compared YOLOv5 and YOLOv8 models and found that YOLOv8 achieved the highest precision (95%), recall (97%), and mAP (96%) on a custom ASL dataset, while YOLOv7 showed higher recall, indicating better coverage in gesture recognition. Finally, the paper *Transfer Learning with YOLOv8 for Real-Time Recognition of American Sign Language Alphabet* by Alsharif *et al.* proposed a real-time ASL recognition system that combines MediaPipe with YOLOv8 and uses transfer learning. The fine-tuned model achieved outstanding performance (98% precision and recall, 99% F1-score, 98% mAP), demonstrating strong potential for assistive technologies supporting the hearing-impaired.

III. Datasets

This project utilizes two datasets: the EgoHands dataset, which provides pixel-level hand annotations for training segmentation models, and the WLASL (Word-Level American Sign Language) dataset, which offers labeled video samples of ASL glosses for gesture recognition and multilingual translation.

A. EgoHands

The EgoHands dataset, developed by Bambach et al., is a first-person video benchmark aimed at enhancing hand segmentation and comprehension in egocentric scenarios. The collection comprises 48 video clips, all filmed in 720×1280 resolution (HD) with the use of Google Glass. Each video has a set length of 90 seconds. The dataset is organized into three primary elements:

- All 48 clips in MP4 format,
- All retrieved frames (JPEG format),
- Annotated data, which includes the manually segmented ground-truth labels.

Moreover, the dataset encompasses four unique interactive tasks–card games, chess, jenga, and puzzle solving–offering a range of activities and contextual differences in hand gestures. This variety enhances generalization for subsequent tasks such as sign language gesture segmentation, where hands may be in different positions and settings. The dataset captures diverse hand poses and skin tones under varied lighting and activity contexts, providing a robust foundation for training hand segmentation models.

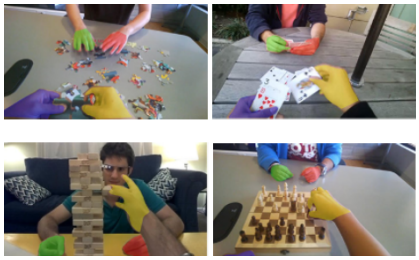

Fig. 1. Pixel-level hand annotations in the EgoHands dataset across four activity scenarios: puzzle solving, card playing, Jenga, and chess.

Fig. 2. Sample cropped hand instances from the EgoHands dataset.

For this project, we utilized exclusively the 'Labeled Data' which offers 100 annotated frames for each video, resulting in a total of 4,800 frames. These annotations are available in MATLAB `.mat` files and comprise pixel-level masks that differentiate various hand types: left or right hand and egocentric (wearer's) or allocentric (other participant's) hand. Every annotated frame in the labeled data includes the original image in JPEG format (720×1280px), along with a per-frame structure that records metadata regarding the linked video and activity. The segmentation masks representing the ground truth utilize polygon coordinates that define the contour of each hand. These masks deliver excellent annotations even with difficulties like occlusions, motion blur, and differing lighting conditions, common in actual human activity videos. The reason for selecting the 'Labeled data' subset instead of the complete video or all extracted frames is due to the necessity for high-precision annotations essential for training supervised segmentation models.

B. WLASL

The WLASL dataset serves as an extensive benchmark aimed at Word-Level American Sign Language recognition, prioritizing dynamic hand movements and temporal context rather than isolated gestures. The version used in this project includes around 12,000 edited video clips, showcasing 2,000 unique glosses where a gloss signifies the standard English equivalent of an ASL sign (e.g., book', 'help', 'run'). Every video in the dataset is a brief, realistic capture of a signer

executing a single, standalone ASL word(see Fig. 3). The usual length of a clip falls between 1 and 3 seconds, while frame rates and resolutions differ among various sources. The dataset comprises various signers and recording configurations to enhance resilience to variations in lighting, body stance, background, and hand movements.

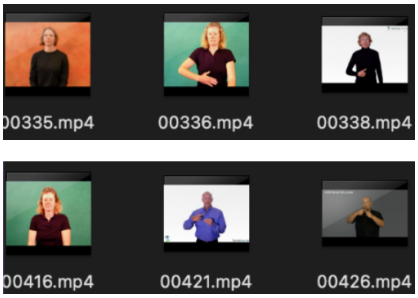

Fig. 3. Sample video files from the WLASL dataset. Each video clip captures a signer performing a distinct word-level American Sign Language (ASL) gloss.

The dataset contains a centralized metadata JSON file, called `WLASL_v0.3.json`, to handle the variety and guarantee reproducibility. This JSON document (see Fig. 4) comprises a collection of entries, with each entry representing a distinct video sample and featuring organized fields like:

- Each JSON entry corresponds to a `gloss` (an ASL word), such as "book".
- The `instances` array contains multiple labeled video samples for that gloss.
- Each instance includes:
  - `bbox`: bounding box coordinates [x1, y1, x2, y2] enclosing the signer's hand or upper body.
  - `fps`: frames per second (typically 25).
  - `frame_start` and `frame_end`: define the usable frame range.
  - `instance_id`: unique index within the gloss.
  - `signer_id`: unique ID for the signer, supporting signer-specific analysis or exclusion.
  - `source`: the data origin (e.g., aslbrick, aslsignbank)
  - `url`: direct video link for download or reference
  - `split`: official dataset partition (e.g., train, val, test)
  - `video_id` and `variation_id`: unique identifiers for each video clip and its variant


```
"gloss": "book",
"instances": [
    {
        "bbox": [
            385,
            37,
            885,
            720
        ],
        "fps": 25,
        "frame_end": -1,
        "frame_start": 1,
        "instance_id": 0,
        "signer_id": 118,
        "source": "aslbrick",
        "split": "train",
        "url": "http://aslbricks.org/New/ASL-Videos/book.mp4",
        "variation_id": 0,
        "video_id": "69241"
    },
```


Fig. 4. Sample structure of the WLASL metadata JSON file.

A key feature of the dataset is the allocation of samples for each gloss. Although common signs like 'go', 'come', or 'help' can have numerous video examples (up to 50), less frequent signs are represented by just a handful of samples.

For this project, a manually chosen filtered subset of 50 glosses was created to strike a balance between training practicality and vocabulary breadth. This narrowed focus enabled efficient benchmarking while alleviating severe class imbalance. Additionally, only the middle 60% of frames from every video were kept, minimizing noise from transitional frames and ensuring that the recorded gesture accurately represents the sign's main articulation.

In general, the WLASL dataset offers a diverse, authentic collection for examining temporal dynamics, variations among signers, and linguistic expressiveness in sign language. When paired with organized annotations and gloss labels, it promotes strong supervised learning and aids in subsequent tasks like translation and the creation of multimodal captions.

IV. Methodology

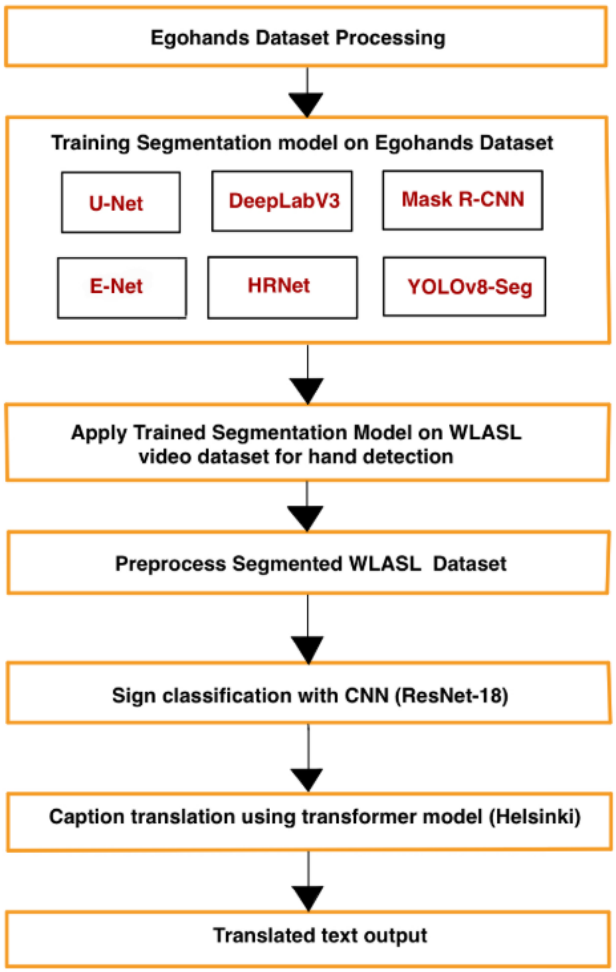

Fig. 5. End-to-end pipeline for sign language detection and translation.

A. Dataset Preparation

We began with the EgoHands dataset, which provides first-person video frames and precise polygonal hand annotations. Using OpenCV's fillPoly, we converted each polygon into an 8-bit binary mask and paired it with its corresponding RGB frame. All images and masks were then resized to 256 × 256 pixels, normalized using ImageNet's mean and standard deviation, and split into an 80/20 training/validation set to ensure no frame appeared in both.

B. Segmentation Model Training

Rather than rely on a single backbone, we trained six architectures: U-Net, ENet, HRNet-W18, Mask R-CNN, DeepLabV3-ResNet101, and YOLOv8-Seg, on the processed EgoHands data. Each model was trained (or fine-tuned from ImageNet) for 50 epochs under a unified recipe: cross-entropy loss, the Adam optimizer (lr = 1×10^{-4}, weight decay = 1×10^{-5}), spatial dropout for regularization, and simple augmentations (random flips and ±10° rotations). This side-by-side approach allowed us to directly compare segmentation quality across architectures.

C. Inference & Hand Crop Extraction

With segmentation backbones trained, we applied them to the WLASL sign-language videos. From each selected clip, 20 frames evenly spaced through the middle 60% were sampled. Each frame was passed through the segmentation network to yield a 256 × 256 hand mask, which was applied to the resized RGB image. Crops containing fewer than 5000 hand pixels were discarded to avoid noisy or empty results, and the remaining clean hand images were saved into gloss-named folders (e.g. . . . /segmented_unet/action/. . .) for classification.

D. Gloss Classification with ResNet-18

We then trained a ResNet-18 classifier to map each hand crop to its ASL gloss. Starting from ImageNet weights, we replaced the final fully connected layer with a Dropout(0.5) → Linear block matching our number of glosses. Using RandomResizedCrop(224), random horizontal flips, and the same normalization, we ran 50 epochs of Adam (lr = 1×10^{-4}, weight_decay = 1×10^{-5}), monitoring both loss and multiclass AUC-ROC. By evaluating classification accuracy on crops from each segmentation backbone, we identified the model whose masks produced the highest recognition performance, making it our chosen hand-detector for the full pipeline.

E. One-to-One Gloss Translation

Rather than embedding glosses in full sentences, we performed direct, word-level translation using a Helsinki-NLP MarianMT English→Hindi model. Each predicted gloss (e.g. "action", "angel", "candy") was translated in isolation, and the outputs were compared against a human-verified Hindi reference list. We reported BLEU-1 scores per gloss and an overall average to quantify translation quality.

This streamlined pipeline, from polygon conversion through segmentation, classification, and translation, ensures that our hand-segmentation models are directly optimized for ASL recognition and that every step contributes measurably to end-to-end sign-to-text performance.

V. Experimental Design

- The primary objective of this study is to develop an end-to-end system for hand gesture segmentation and gloss-level sign language classification. The experiments were conducted to evaluate the performance of the proposed segmentation and classification pipeline across cross-domain hand gesture data. The system was implemented using the PyTorch deep learning framework and trained on GPU hardware for efficiency.
- This study investigates the performance of multiple semantic segmentation models across a curated subset of 50 sign language gesture classes. All experiments were conducted using the PyTorch deep learning framework. The dataset was divided following an (80:10:10) split ratio for training, validation, and testing, respectively. To ensure temporal diversity and reduce redundancy, frames were sampled from the central 60% segment of each video. Each model was trained for 50 epochs.
- Data augmentation techniques such as horizontal flipping and rotation were applied during training to enhance model robustness and prevent overfitting. The segmentation models evaluated include U-Net, DeepLabV3, Mask R-CNN, ENet, HRNet, and YOLOv8m-Seg.
- The training pipeline incorporated early stopping with a patience value of 3, dropout regularization, and weight decay to further combat overfitting and improve generalization. Experiments were run on a variety of hardware platforms, including Kaggle P100 GPUs, Google Colab A100 GPUs, and Nvidia RTX 4060 GPUs.
- In addition to segmentation, the framework also included classification and translation modules. A ResNet-18 convolutional neural network was used to predict gesture labels, and a Helsinki-NLP transformer model was used for English-to-Hindi label translation.
- Performance evaluation for the classification model was conducted using standard classification metrics (e.g., accuracy, precision, recall), while translation quality was assessed using the BLEU score.

VI. Evaluation Metrics

For the segmentation models, evaluation was conducted using standard pixel-wise metrics including accuracy, precision, recall, and F1-score, where the F1-score corresponds to the Dice coefficient. Specifically for the YOLOv8m-Seg model, additional object detection metrics were applied, such as mean Average Precision at 50% Intersection over Union (mAP@50 IoU), along with mean average precision and recall across the validation set. For the ResNet-18 CNN classifier used in gesture recognition, performance was assessed using standard classification metrics including accuracy, precision, recall, and F1-score, all macro-averaged across the 50 ASL gloss classes. In the translation stage, the output of the classifier

was translated from English to Hindi using the Helsinki-NLP model, and evaluated using the BLEU (Bilingual Evaluation Understudy) score. In addition, custom translation accuracy was calculated based on exact string match and fuzzy matching using the Levenshtein similarity metric, where a similarity score greater than or equal to 0.8 was considered acceptable. The segmentation models were evaluated using pixel-level accuracy, precision, recall, and Dice-based F1-score, while YOLOv8m-Seg was further assessed with mAP@50 IoU alongside mean average precision and recall on the validation dataset. For the ResNet-18 gesture classifier, we presented macro-averaged accuracy, precision, recall, and F1-score for all 50 ASL glosses, and assessed translation quality utilizing BLEU along with custom exact-match and Levenshtein-derived accuracy.

VII. Segmentation Models

A. Mask R-CNN

Mask R-CNN (Mask Region-based Convolutional Neural Network) is an enhancement of the Faster R-CNN architecture, created by He et al., aimed at instance segmentation tasks. Although Faster R-CNN carries out object detection by estimating class labels and bounding boxes, Mask R-CNN includes an additional branch to produce pixel-level segmentation masks for every identified object instance. The model is created to identify various object instances and concurrently produce: Bounding boxes, category labels and segmentation masks. Mask R-CNN is ideal for hand segmentation tasks as it can identify several hands in an image, even when they overlap, and delivers intricate spatial data, which is essential for subsequent gesture classification. The design of Mask R-CNN includes multiple essential elements:

- **Backbone CNN**: Usually a ResNet50 or ResNet-101 architecture combined with a Feature Pyramid Network (FPN) for obtaining multi-scale feature maps from the input image.
- **Region Proposal Network (RPN)**: Analyzes the feature maps and suggests regions of interest (ROIs) that could include objects (e.g., hands).
- **ROI Align**: Accurately extracts features from every proposed region.
- **Detection Head**: The model predicts for every ROI- a label for a class, an enclosing box and a segmentation mask. The segmentation branch operates separately from class prediction, enabling the model to create a binary mask for each object instead of masks specific to classes. In this project, the Mask R-CNN model was initially trained on the EgoHands dataset, which offers polygon-level annotations for hands. These annotations were transformed into the COCO JSON format necessary for Mask R-CNN training workflows. After training, the model produced reliable segmentation masks delineating hand regions even in cluttered or occluded scenes(see Fig 6). The model was developed to differentiate between hand (foreground) and background pixels.

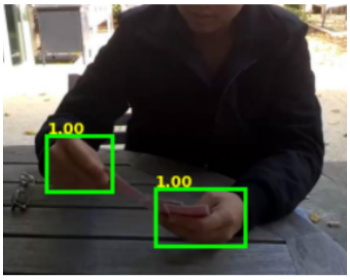

Fig. 6. Detection of multiple hands using Mask R-CNN on a frame from the EgoHands dataset.

Following preliminary training on EgoHands, the model underwent fine-tuning using a chosen subset of frames taken from WLASL sign language videos. To process this dataset, frames were extracted from approximately WLASL video clips corresponding to 50 manually selected glosses. The selection ensured a balanced class distribution suitable for initial classification experiments. The extracted frames were then passed through the trained Mask R-CNN model, which includes bounding boxes highlighting hand regions (see Fig. 7). Parallely, metadata from the WLASL_v0.3.json file was parsed to associate each frame and its parent video with the correct gloss label.

Fig. 7. Hand localization using a bounding box generated by the Mask R-CNN model on a WLASL video frame.

B. YOLOv8m-seg

Recent advancements in instance segmentation have been significantly influenced by the YOLO (You Only Look Once) family of models. In general, modern YOLO variants such as YOLOv8 extend beyond object detection to include instance segmentation. The YOLOv8m-seg model integrates a CSP-Darknet backbone for robust feature extraction, a PAN-FPN neck for feature fusion across scales, and two parallel output heads—one for object detection and another for segmentation. The segmentation head adopts a prototype-based approach, generating a fixed set of shared masks and per-instance coefficients, which are linearly combined to produce object-specific binary masks. Final segmentation results are refined through bounding box cropping and post-processing steps such as non-maximum suppression (NMS).

In our specific implementation, we utilized the YOLOv8m-seg model for hand segmentation tasks using customized datasets derived from EgoHands and WLASL. Initially, polygonal annotations available in MATLAB .mat files from the

EgoHands dataset were parsed and visualized. Each annotated polygon was converted into a bounding box and exported into YOLO-compatible .txt annotation files. This enabled the training of YOLO on non-native formats by bridging the gap between polygonal annotations and YOLO's bounding-box-based training requirements.

During inference, the trained segmentation model was deployed on WLASL video frames. Importantly, the original bounding boxes provided by the WLASL dataset were not used. Although COCO-format JSON files were available, they contained bounding boxes corresponding to entire human figures and not the hands performing sign gestures. In our pipeline, we intentionally ignored these pre-supplied annotations and instead applied our EgoHands-trained YOLOv8m-seg model to detect and segment hands directly. The bounding boxes were dynamically derived from the predicted segmentation masks. Each segmentation mask was thresholded, resized to fit the original frame dimensions, and overlaid with alpha blending for visual clarity. Detection metadata, including model-generated bounding boxes and confidence scores, was saved to JSON files for each frame.

This project-specific pipeline demonstrates how a general-purpose segmentation model like YOLOv8m-seg can be tailored to domain-specific applications such as gesture and hand activity recognition. By repurposing YOLO's segmentation output and disregarding potentially imprecise human-level bounding boxes in WLASL, our model delivered refined and accurate instance-level hand annotations. The approach supports efficient batch inference, visualization, and the generation of structured outputs for further training or analysis.

During early experimentation, we observed that our trained YOLOv8m-seg model often detected each hand separately, resulting in two distinct bounding boxes per frame. This introduced ambiguity in downstream classification, where each hand could incorrectly be assigned a different gloss label. To resolve this, we implemented logic that computes a combined bounding box encompassing all detected hand masks per frame. This merged bounding box was then used to extract a unified region of interest (ROI), ensuring that both hands contributing to a single sign were included in the input. The resulting images, with mask overlays, was passed to the ResNet classifier.

Fig. 8. YOLO-based sign detection for the WLASL "drama" class, showing (a) the input video frame, (b) raw detection output, (c) localized bounding box, and (d) class activation heatmap(GradCam).

C. E-Net

E-Net is a rapid and effective image segmentation deep learning model. Developed by Paszke et al., it's unique because it's lightweight and it's perfect for those applications where speed is key. Unlike other segmentation models which are highly concerned with accuracy and as such very slow, E-Net achieves a balance between being fast and good results all due to the encoder-decoder style.

The E-Net model is an encoder-decoder network for fast, accurate segmentation of images. It is coupled with a ResNet18-based classifier to increase the segmentation and recognition of gesture of hands. The architecture of E-Net guarantees efficient feature extraction; ResNet18 is good for strong classification. `Initial Block`: Convolutional and max-pooling layers extract elementary details of the input image.

`Bottleneck Modules`: Residual blocks with dilated convolutions enable the model to obtain multi-scale information without an increased computation. Asymmetric Design: The encoder has more layers compared to decoder in order to maintain model speed without limiting on accuracy.

`Efficient Skip Connections`: The direct connections between the encoder and decoder maintain fine details in the segment.

`BCE + Dice Loss`: Dice loss alongside Binary Cross-Entropy (BCE) during training camouflages the pixel-wise accuracy and overall region accuracy.

`ResNet18 for Classification`: After its segmentation the ResNet18 model labels the hand gestures. Its robust and accurate classification is guaranteed by residual block structure.

`Integrated Pipeline`: Direct transmission of segmented hand images from E-Net into ResNet18 forms an efficient segmentation-classification pipeline.

The hand segmentation process was integrated using E-Net and ResNet-18 used for gesture classification in training process. E-Net was trained on the EgoHands dataset and ResNet-18 was trained on isolated hand parts present in the WLASL dataset. Adam optimizer with an initial learning rate of 0.001 used a ReduceLROnPlateau scheduler for learning rate to reduce if validation loss plateaued. For pixel-wise segmentation E-Net used BCE + Dice Weighted Loss and for multi-class classification task, Cross-Entropy Loss was applied in ResNet-18. Some augmentation techniques used by E-Net are random scaling, horizontal flipping, and brightness former and random rotations and color jitter for ResNet-18. E-Net was trained using a batch size of 16 for 100 iterations and early stopping (10 iterations of time before improvement) , ResNet-18 in terms of a batch size of 32 on 50 epochs. The most optimal models were saved by validation loss (E-Net) and validation accuracy (ResNet-18).The hand regions first partitions were made by E–Net which was later applied to ResNet-18 for classification purposes.

After training on the EgoHands dataset, we fine-tuned the E-Net model using frames from WLASL (Word-Level American Sign Language). This helped the model adjust to a wide variety of lighting conditions, different signers, and various hand shapes seen in WLASL videos. During inference, the model processed each WLASL frame to generate a binary mask that highlighted the hand regions. These segmented hand regions were then used to create a clean, hand-focused

dataset, making it easier for the classifier to focus solely on recognizing gestures. Thanks to E-Net's lightweight design. The combination of Binary Cross-Entropy (BCE) and Dice loss during training also ensured that the model learned to accurately identify hands without mistakenly segmenting the background.

Fig. 9. The performance of the E-Net model on a sample frame from the WLASL dataset.

D. DeepLabV3

We used a standard encoder–decoder architecture, selecting DeepLabV3 with a ResNet-18 backbone due to its balance of spatial precision and context-aware feature extraction. The model was pre-trained on ImageNet, and we fine-tuned it using our dataset of masked sign language images. These images were annotated with binary masks indicating hand regions, allowing the model to learn hand segmentation explicitly.

Input frames were resized to 224×224 and normalized using ImageNet statistics. During training, we applied standard augmentation techniques like horizontal flipping and random rotation to increase data variability and reduce overfitting. We also tried early stopping with weight decay with different learning rates. But the result was the same. Also, the loss function used was a pixel-wise binary cross-entropy loss, optimized using Adam. We added a threshold to only save frames where the segmented hand region is sufficiently large (more than 5000 pixels) to avoid saving empty or low-confidence masks. This ensures that only meaningful, clearly visible hand frames are stored for training the classification model.

In the inference phase, we passed each RGB frame through the trained DeepLabV3 model to obtain a binary mask. This mask was applied to the input image to extract the segmented region, which was then forwarded to the classification module. To improve visualization and interpretability, we also generated Grad-CAM overlays, showing which parts of the image contributed most to the classification prediction.

E. HRNet

High-Resolution Network (HRNet) has proved to be a robust architecture for different dense prediction tasks such as semantic segmentation Ke Sun *et al.* (2019), Wang *et al.* (2020). Different from traditional networks that recover high-resolution representations from low-resolution outputs produced by a high-to-low resolution stream, HRNet keeps high-resolution representations throughout the whole process. It parallelly links high-to-low resolution subnetworks and iteratively passes information between resolutions which results in rich high-resolution representations, which are vital for accurate localization.

For hand segmentation, we used the hrnet_w18 variant from the timm library that was pre-trained on ImageNet. The main points of our model based on the HRNet are:

`Backbone`: hrnet_w18 was used with features only true to obtain multi-scale feature maps. We specifically used the feature map output at index 0 of the list of feature maps returned by the backbone, which had 64 at 1/2 of input resolution (e.g., 160x320 for a 320x640 input).

`Segmentation Head`: A simple SimpleSegmentationHead was connected to the chosen HRNet feature map. This head is made up of one single 1x1 convolutional layer that transforms the 64 input feature channels into NUM_CLASSES (3 in our case: background, user's hand, other's hand).

`Upsampling`: The output logits of the segmentation head, which has a lower resolution than the input image, were upsampled back to the original input size in (320x640) via nn.functional.interpolate with bilinear mode prior to loss calculation or final mask generation.

The HRNet segmentation model was trained exclusively on the EgoHands dataset. Polygonal annotations from the MATLAB .mat files were first converted into binary masks representing three classes: background, the wearer's own hand, and the other participant's hand. For data augmentation, incorporating transformations such as horizontal flipping (with a probability of 0.5) and normalization using ImageNet statistics. The model was optimized using the Adam optimizer with an initial learning rate of 1×10^{-4}, and a ReduceLROnPlateau scheduler was applied to dynamically adjust the learning rate based on the validation loss. Cross-entropy loss was used as the objective function to guide training Yuan *et al.* (2020) .

Fig. 10. Sample HRNet segmentation results on the EgoHands validation set. (Left: Input Image, Middle: Ground Truth Mask, Right: Predicted Mask).

The HRNet model, after being trained on the EgoHands dataset, was applied to frames extracted from the WLASL dataset for hand segmentation. For each frame, the model generated segmentation masks that identified hand regions. These masks were then resized back to the original dimensions of the WLASL frames to preserve spatial alignment.

F. U-Net

U-Net presented by Ronneberger et al. (2015) is a fully convolved network for accurate image segmentation. Its symmetric 'U' shape has a contracting path, which uses successive 3 by 3 convolutions (BatchNorm + ReLU) and 2 by 2 max-pooling, to capture context (and doubles the

channel depth from 64 to 512, while halving spatial size), and the restoring path that uses 2 by 2 transposed convolutions to reinflate the original 256 by 256 resolution. Importantly, skip connections transfer high-res features directly from each encoder block to the decoder, combining global context with fine detail. Our U-Net's structure unfolds in a series of clearly defined stages: `Encoder (Down-Sampling)`: Four blocks, each with two 3×3 convolutions (neuron-level BatchNorm and ReLU), alternating with spatial Dropout (p=0.2), followed by a 2×2 max-pool to cut resolution by half (256-128-64-32-16) and double channels (64-128-256-512).

`Bottleneck`: At resolution 16×16, we use 2 three-by-three ConvLayers with 1 024 channels (BatchNorm+ReLU), followed by a heavier Dropout (p=0.3) to promote good regularization in most compressed features.

`Decoder (Up-Sampling)`: For each of the four stages, there is a 2×2 transposed convolution at the start to double spatial size (16-32-64-128-256), concatenation with its encoder skip connection, 2 layers of 3×3 Conv-BatchNorm-ReLU each to perfect details.

`Prediction Layer`: Finally, a final 1×1 convolution maps 64 channels to two values (background vs. hand), and one pixel softmax yields the 256x256 probability mask.

We trained U-Net for 50 epochs on the EgoHands dataset, pairing each 256×256 RGB frame with its binary hand mask. Augmentations included random horizontal flips and ±10° rotations to simulate varied viewpoints. Normalization used ImageNet mean/std to align input distributions. Optimization was via Adam (lr = 1 × 10, weight decay = 1 × 10). Spatial Dropout in every convolution block (p = 0.2) and in the bottleneck (p = 0.3) further prevented overfitting. We monitored pixel-wise cross-entropy loss, overall pixel accuracy, and hand-class AUC-ROC on a held-out split, and performed save–reload checks to ensure exact reproducibility of outputs.

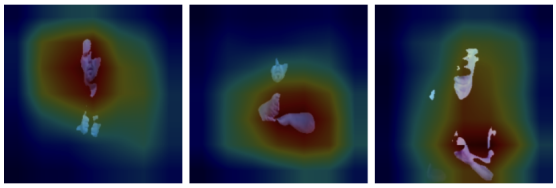

Fig. 11. Grad-CAM overlays on validation frames showing where U-Net concentrates its hand activations.

VIII. Gesture Classification

Following hand segmentation using each of the trained segmentation models , the isolated hand regions from WLASL frames were used to train a **gesture classification model based on ResNet-18**. This CNN was chosen for its balance between performance and computational efficiency. The model was trained on cropped hand images labeled with ASL glosses, leveraging the structural annotations from WLASL_v0.3.json. The ResNet-18 classifier learned to recognize 50 selected ASL glosses by analyzing spatial patterns in the segmented hand regions, achieving reliable classification performance across multiple signers and conditions.

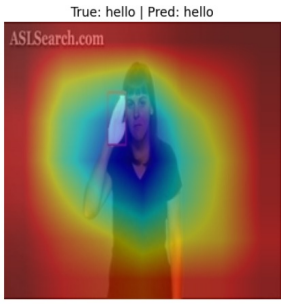

Fig. 12. Attention heatmap overlay for the ASL sign "hello," showing the model's true and predicted label both as "hello," with peak activation focused on the signer's hand region.

IX. Experimental Results

In this section we will discuss the results of our gesture classification.

TABLE I
Gesture classification results across different models.

Metric	U-Net	YOLO	Mask R-CNN	HRNet	E-Net	DeepLab
Accuracy	0.63	0.89	0.53	0.86	0.94	0.96
Precision	0.81	0.92	0.89	0.49	0.96	0.90
Recall	0.64	0.89	0.38	0.40	0.94	0.91
F1 Score	0.69	0.88	0.54	0.44	0.94	0.90

A. Mask R-CNN

Although the Mask R-CNN model reached a high precision (0.899), its overall accuracy (0.534) and recall (0.389) were relatively low. This difference can be linked to two main factors. Initially, the precise segmentation provided by Mask R-CNN generally results in closely cropped hand areas that omit contextual elements like the wrist, forearm, or objects involved in interaction. These surrounding elements—frequently preserved in bigger bounding boxes—could hold significant spatial details that ResNet-18 unconsciously depends on for gesture classification. The absence of this context probably affected the model's capacity to generalize among sign variations. The Region Proposal Network (RPN), RoIAlign, and segmentation branches need multi-stage structure of Mask R-CNN to operate reliably. Even slight discrepancies between the anticipated area of focus and the hand mask can result in uneven or distorted input crops, impacting the reliability of classification. Conversely, single-shot segmenters like U-Net or DeepLabV3 generate more consistent masks, which may render them more reliable for subsequent tasks in workflows with restricted data. Fig. 13 shows an example for misclassification ("hot" predicted instead of "have"), while the second and third examples illustrate correct predictions ("night" and "cemetery") with focused attention on key spatial areas.

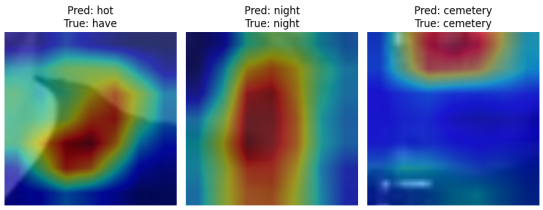

Fig. 13. Grad-CAM visualizations for ResNet-18 gesture classification after training on Mask R-CNN. Images show 2 correct predictions(middle and right) and 1 incorrect prediction(left)

B. YOLOv8m-seg

The trained sign language recognition model achieved a test accuracy of 88.96% on a held-out sample. The evaluation used standard classification metrics, including precision, recall, and F1-score. The macro-averaged F1-score across all classes was 0.8587, and the weighted F1-score was slightly higher at 0.8844, reflecting stronger performance on more frequent classes.

C. E-Net

The E-Net model achieved an accuracy of 0.94, with a precision of 0.96, recall of 0.94, and F1-score of 0.94. While it performed consistently across most gestures, it struggled slightly with classes like ”contact” and ”paper,” likely due to aggressive downsampling and limited context. Overall, E-Net excelled in recognizing most gestures with high precision and recall, making it effective for real-time segmentation.

D. DeepLabV3

In terms of results, the training loss dropped significantly from 0.22 to 0.02, and the validation loss decreased to approximately 0.03. This indicates that the segmentation model was able to learn the task effectively. However, we observed that the validation loss occasionally increased while the training loss continued to decline — a classic symptom of overfitting. Despite this, the visual quality of the segmentation masks remained satisfactory and effectively isolated hand regions for use in the classification stage.

For the classification task, we used a ResNet-18 model to predict gloss labels from the segmented images. The model reached 100% training accuracy by epoch 5, and validation accuracy quickly improved to around 95.68%, after which it plateaued. Although the classification accuracy was high, we still encountered overfitting. Techniques such as dropout, early stopping, and data augmentation were applied, but the model still tended to memorize the training data. We suspect the primary cause is the limited number of samples available for many gloss classes — with some classes represented by only one or two examples — making it difficult for the model to generalize reliably. We will improve it in future work.

E. HRNet

Although the final validation loss of the HRNet model was low (0.1817) on the EgoHands dataset, its associated mean Intersection over Union (mIoU) was 0.4120. This implies that although the model became good at classifying individual pixels accurately (evidenced by the low cross-entropy loss), its capacity at precisely segmenting the whole hand regions relative to the ground truth (mIoU) was relatively modest. This performance can be traced to a couple of factors. Firstly, HRNet's architecture that is intended to keep high-resolution representations is excellent at representing fine-grained details. Nevertheless, this sensitivity may cause segmentations in which, although pixel-precise, occasionally fragment complex hand poses or fail on heavy occlusions, which affect the final IoU. Secondly, an mIoU of 0.4120 indicates that the masks produced for the WLASL dataset while generally isolating hands may sometimes incorporate small background fragments or fail to capture delicate areas of the hand. Such impurities in the input to the following ResNet-18 gesture classifier may introduce noise or remove discriminative features which may influence the robustness of the classifier.

F. U-Net

Using U-Net's hand masks, we extracted 224×224 hand crops and trained a ResNet-18 to recognize 41 ASL glosses. After 50 epochs with identical augmentations and optimizer settings, the classifier achieved 63% accuracy on 1 072 validation images. Its macro-average F1 was 0.69, reflecting high performance on distinctive signs, chapter (F1 = 0.91) and contact (F1 = 0.80) and challenges on subtle gestures like candy (F1 = 0.30) and schedule (F1 = 0.50). Finally, we did a word-to-word English-Hindi translation of every predicted gloss. In contrast to our human verified Hindi references, the model provided 56.1% exact match (23/41) and average BLEU-1 score at 0.52, showing a mediocre level of success, but underlining the necessity of context cues when translating individual words.

X. Translation

For the translation component, we used a pretrained MarianMT-based English-to-Hindi model available from Helsinki-NLP (`Helsinki-NLP/opus-mt-en-hi`). This model was used to convert gloss-level predictions into Devanagari Hindi translations, which were then rendered alongside classification output to improve end-user interpretability. The model was selected due to its relatively lightweight architecture (approximately 298MB in size) and its seamless integration with Hugging Face's `pipeline` API, enabling fast, on-the-fly translation during inference.

During implementation, each predicted gloss was tokenized and passed through the Helsinki model to generate the corresponding Hindi phrase. The translated output was visually embedded within the Grad-CAM visualizations, allowing us to validate both the correctness of the classification and the semantic clarity of the translation. Although this setup was effective for rapid prototyping, we observed limitations in translation fluency and contextual accuracy, particularly for abstract or domain-specific glosses. These limitations are likely due to the general purpose nature of MarianMT training, which is not specifically tuned for sign language inputs. After

implementation we applied BLEU score to check the accuracy. We understood that it is totally relative to that particular word. But after every experiment, we got an average of 0.623 accuracy. If there is a total match, then the accuracy gets higher, but if there are any synonyms or different verbs as meaning, then it predicts different word. The reason behind this is the lack of semantic vocabulary.

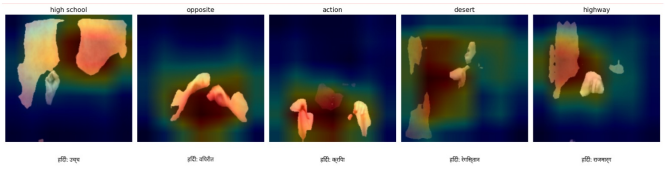

Fig. 14. Translation output

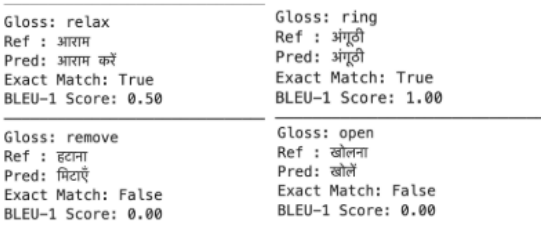

Fig. 15. BLEU score output

XI. Novelty

- Unlike prior works that use bounding boxes (including those for full human detection in WLASL), our method performs the same along with true instance segmentation of hands using polygon-level training data from EgoHands.
- Our segmentation models, trained on still images from EgoHands, generalizes effectively to signer-centric video frames in WLASL without requiring re-annotation.
- We constructed a gloss-labeled dataset from mask-overlaid hand images and trained a ResNet classifier to recognize specific sign glosses — a step beyond simple alphabet classification.
- Our approach eliminates dependence on hand keypoint estimation frameworks like MediaPipe. Instead, we rely solely on learned visual segmentation from annotated data, improving generalizability and system simplicity.
- To resolve the ambiguity caused by separate detections of left and right hands, we introduced a bounding box merging strategy that combines all hand detections into a unified gesture region, improving classification reliability.

While similar components have been used in isolation within the research community, the integration of these elements into a cohesive pipeline for gloss-level sign language recognition represents a novel contribution. This integrated approach has not been explicitly documented in the existing literature, indicating its uniqueness in the field.

XII. Conclusion

In this project, we explored and implemented multiple segmentation models for hand-based sign language understanding, leveraging the EgoHands and WLASL datasets. The study focused on six architectures—U-Net, YOLOv8m-seg, DeepLabV3, ENet, HRNet, and Mask R-CNN—each selected based on its architectural strengths, such as precision, speed, or high-resolution retention. We preprocessed and filtered the data carefully, including the selection of the middle 60% of frames to capture the most relevant gesture content. The segmented hand regions were further used to train a ResNet18 classifier on 50 glosses, demonstrating the effectiveness of combining pixel-level understanding with high-level recognition. Throughout the pipeline, visualization techniques such as Grad-CAM++ were employed to improve model interpretability. We then used a natural language translation model to convert sign into Hindi language and evaluate using the score method. Overall, the system presents a robust modular framework for gesture-level sign language recognition and translation, balancing segmentation accuracy, classification reliability, and computational efficiency.

XIII. Future Work

There are several directions we intend to explore based on the current findings. First, although the pipeline was built with real-time readiness in mind (e.g., using lightweight models like ENet), actual real-time execution was not implemented; future iterations will include model optimization through ONNX and integration with a live camera feed. Secondly, an end-to-end multi-task model that combines segmentation and classification (e.g., YOLOv8-seg with gloss-level supervision) could reduce latency and simplify the pipeline. Third, We were facing overfitting in few methods, we will try to improve that.

In addition to segmentation and classification, a key next step will be integrating natural language translation for the recognized gloss sequences. We plan to experiment with the ai4bharat/indictrans2-en-indic-1B model, a state-of-the-art multilingual translation model capable of translating English to Indian languages. To make the deployment feasible on resource-constrained devices, we will apply quantization techniques to reduce the model size and improve inference speed. This step is critical to enable real-time translation of recognized glosses into spoken or written Hindi, which aligns with the broader goal of making the system accessible to native signers and non-signers in India.

Additionally, expanding the dataset- both in terms of the number of gloss labels and environmental diversity- will help improve the system's robustness. We also intend to explore temporal modeling techniques (e.g., LSTMs or temporal CNNs) to capture the motion dynamics of gestures better. Finally, developing a lightweight user interface and performing in-depth error analysis, particularly under challenging conditions like hand occlusion or gesture overlap, will be important to ensure real-world usability and interpretability.

XIV. CRediT authorship contribution statement

Hamsalakshmi Ramachandran: Initial methodology design, Implementation and training of Mask R-CNN for hand segmentation, Gesture Classification using ResNet-1, Translation, Report writing.

Sugandha Chauhan: Implementation and training of YOLOv8m-Seg for hand detection and segmentation, Initial methodology design, Gesture Classification using ResNet-18, Report writing.

Himani Shah: Implementation and training of DeepLabV3 for semantic hand segmentation, Integration and evaluation of Helsinki-NLP for translation, Gesture Classification using ResNet-18 , Report writing.

Manjot Singh: Implementation and training of U-Net for hand segmentation, Preprocessing of WLASL dataset, Gesture Classification using ResNet-18, Report writing.

Saqib Chowdhury: Implementation and training of E-Net for efficient hand segmentation, Gesture Classification using ResNet-18, Report writing.

Kush Bindal: Implementation and training of HRNet for high-resolution hand segmentation, Gesture Classification using ResNet-18, Report writing.

References

[1] B. S. Parton, "Sign language recognition and translation: A multidisciplined approach from the field of artificial intelligence," *J. Deaf Stud. Deaf Educ.*, vol. 11, no. 1, pp. 94–101, 2006. [Online]. Available: https://academic.oup.com/jdsde/article-abstract/11/1/94/410770

[2] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10023–10033. [Online]. Available: https://arxiv.org/abs/2003.13830

[3] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 7957–7968, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s00521-019-04691-y

[4] A. El Zaar, N. Benaya, and A. El Allati, "Sign language recognition: High performance deep learning approach applied to multiple sign languages," in *E3S Web Conf.*, vol. 351, 01065, 2022. [Online]. Available: https://www.e3s-conferences.org/articles/e3sconf/abs/2022/18/e3sconf_icies2022_01065

[5] I. Papastratis *et al.*, "Artificial intelligence technologies for sign language," *Sensors*, vol. 21, no. 17, p. 5843, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/17/5843

[6] B. K. Triwijoyo, L. Y. R. Karnaen, and A. Adil, "Deep learning approach for sign language recognition," *J. Ilm. Tek. Elektro Komp. dan Informatika*, vol. 9, no. 1, pp. 12–21, 2023. [Online]. Available: https://www.researchgate.net/profile/Ahmat-Adil/publication/367461727

[7] A. M. Buttar *et al.*, "Deep learning in sign language recognition: A hybrid approach for the recognition of static and dynamic signs," *Mathematics*, vol. 11, no. 17, p. 3729, 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/17/3729

[8] G. Strobel, T. Schoormann, L. Banh, and F. Möller, "Artificial intelligence for sign language translation–A design science research study," *Commun. Assoc. Inf. Syst.*, vol. 53, no. 1, pp. 42–64, 2023. [Online]. Available: https://aisel.aisnet.org/cais/vol53/iss1/22

[9] M. J. C. Samonte *et al.*, "Using deep learning in sign language translation to text," in *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, 2022, pp. 7–10. [Online]. Available: https://ieomsociety.org/proceedings/2022istanbul/758.pdf

[10] A. Yadav *et al.*, "Audio to sign language translator web application," in *2021 Int. Conf. Comput. Perform. Eval. (ComPE)*, 2021, pp. 321–326. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9751857

[11] A. Wahane *et al.*, "Real-time sign language recognition using deep learning techniques," in *2022 IEEE 7th Int. Conf. Convergence Technol. (I2CT)*, 2022, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9825192

[12] D. Kothadiya *et al.*, "Deepsign: Sign language detection and recognition using deep learning," *Electronics*, vol. 11, no. 11, p. 1780, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/11/1780

[13] O. B. Hoque, M. I. Jubair, M. S. Islam, A. F. Akash, and A. S. Paulson, "Real time Bangladeshi sign language detection using Faster R-CNN," in *Proc. 2018 Int. Conf. Innovation in Engineering and Technology (ICIET)*, Dhaka, Bangladesh, Dec. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8660780

[14] U. Jana, S. Paul, and D. Bhandari, "Real-time caption generation for the American Sign Language using YOLO and LSTM," in *2024 IEEE Int. Conf. Inf. Technol., Electron. and Intell. Commun. Syst. (ICITEICS)*, 2024, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10625098

[15] Y. Zhang and X. Jiang, "Recent advances on deep learning for sign language recognition," *CMES-Comput. Model. Eng. Sci.*, vol. 139, no. 3, 2024. [Online]. Available: https://openurl.ebsco.com/EPDB%3Agcd%3A13%3A13422765/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A176091289&crl=c&link_origin=scholar.google.com

[16] P. Zhang *et al.*, "EvSign: Sign language recognition and translation with streaming events," in *European Conf. Comput. Vis.*, 2024, pp. 335–351. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-72652-1_20

[17] P. Sharma *et al.*, "Indian sign language recognition and translation: Text to sign language using deep learning technique," in *AIP Conf. Proc.*, vol. 3112, no. 1, 020027, 2024. [Online]. Available: https://pubs.aip.org/aip/acp/article-abstract/3112/1/020027

[18] Y. Tian, J. Su, L. Ni, and Y. Fang, "Bridging the gap: AI and sign language recognition–A path toward inclusive communication," *Int. J. Artif. Intell. Robots. Res.*, 2401003, 2024. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S2972335324010038

[19] A. Sharma *et al.*, "Promoting sign language awareness: A deep learning web application for sign language recognition," in *Proc. 2024 8th Int. Conf. Deep Learn. Technol. (ICDLT)*, 2024, pp. 22–28. [Online]. Available: https://dl.acm.org/doi/full/10.1145/3695719.3695723

[20] R. A. Alawwad, O. Bchir, and M. M. B. Ismail, "Arabic sign language recognition using Faster R-CNN," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 3, 2021. [Online]. Available: https://www.researchgate.net/profile/Ouiem-Bchir/publication/350529742_Arabic_Sign_Language_Recognition_using_Faster_R-CNN/links/645377815762c95ac36fc3b1/Arabic-Sign-Language-Recognition-using-Faster-R-CNN.pdf

[21] R. K. Pathan *et al.*, "*RETRACTED ARTICLE*: Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network," *Sci. Rep.*, vol. 13, no. 1, 16975, 2023. [Online]. Available: https://www.nature.com/articles/s41598-023-43852-x

[22] R. R. Sewwantha and T. Ginige, "Mask region-based convolutional neural networks (R-CNN) for Sinhala sign language to text conversion," in *CS & IT Conference Proceedings*, vol. 11, no. 14, Sep. 2021. [Online]. Available: https://csitcp.org/paper/11/1114csit18.pdf

[23] F. M. Najib, "Sign language interpretation using machine learning and artificial intelligence," *Neural Comput. Appl.*, vol. 37, no. 2, pp. 841–857, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s00521-024-10395-9

[24] O. Bchir, "Hand segmentation for Arabic sign language alphabet recognition," in *Proc. Comput. Sci. Inf. Technol. (CSIT)*, 2020, pp. 65–74. [Online]. Available: https://doi.org/10.5121/csit.2020.100701

[25] S. Aly and W. Aly, "DeepArSLR: A signer-independent framework for Arabic sign language recognition," *IEEE Access*, vol. 8, pp. 83199–83212, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2990699

[26] R. G. Rajan and P. S. Rajendran, "Fusing handcrafted and CNN features for ASL classification," *Rev. Int. Geogr. Educ. Online*, vol. 11, no. 7, pp. 1168–1177, 2021. [Online]. Available: https://rigeo.org/menu-script/index.php/rigeo/article/view/1626

[27] K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5693-5703). Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Sun_Deep_High-Resolution_Representation_Learning_for_Human_Pose_Estimation_CVPR_2019_paper.html

[28] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., ... & Xiao, B. (2020). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *43*(10), 3349-3364. Available: https://ieeexplore.ieee.org/abstract/document/9052469

[29] Yuan, Y., Chen, X., & Wang, J. (2020). Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16* (pp. 173-190). Springer International Publishing. Available: https://link.springer.com/chapter/10.1007/978-3-030-58539-6_11

[30] Paszke, A.Chaurasia, S.Kim, & E.Culurciello(2016) "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," *arXiv preprint arXiv:1606.02147*, 2016. [Online]. Available: https://arxiv.org/abs/1606.02147

[31] Yun, Y. & Park, J. (2022) "Efficient Object Detection Based on Masking Semantic Segmentation Region for Lightweight Embedded Processors," *Sensors*, vol. 22, no. 1, pp. 1–14. doi:10.3390/s22010014. [Online]. Available: https://www.mdpi.com/1424-8220/22/22/8890

[32] Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E. (2016). "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," *arXiv preprint arXiv:1606.02147*. Available: https://arxiv.org/abs/1606.02147

[33] Alsharif, B., Alalwany, E., & Ilyas, M. (2024). *Transfer learning with YOLOV8 for real-time recognition system of American Sign Language Alphabet*. Franklin Open, 8, 31. https://doi.org/10.1016/j.fraope.2024.100165

[34] Tyagi, S., Upadhyay, P., Fatima, H., Jain, S., & Sharma, A. (2023). *American Sign Language Detection using YOLOv5 and YOLOv8*. https://doi.org/10.21203/rs.3.rs-3126918/v1

[35] Agangiba, M., Ezekiel, M., Agangiba, W., & Appiah, O. (2023). *Performance Evaluation of ResNet Model on Sign Language Recognition*. International Journal of Computer Applications, 184(43), 22–27. https://doi.org/10.5120/ijca2023922534

[36] Lieman-Sifry, J., Le, M., Lau, F., Sall, S., and Golden, D. (2017). "FastVentricle: Cardiac Segmentation with ENet," in *Proc. International Conference on Image Analysis and Processing (ICIAP)*, Catania, Italy, pp. 376–387. doi:10.1007/978-3-319-59448-4_13. Available: https://link.springer.com/chapter/10.1007/978-3-319-59448-4_13

XV. Appendices

The below link contain all project related files:

- **GitHub Repository**: https://github.com/hamsaram14/DeepLearningProject-SignLanguageDetection