Indian Sign Language Detection and Translation using Deep Learning and Text-to-Speech

MSc Research Project
Data Analytics

Aji Poovannapoikayil
Student ID: x22184431

School of Computing
National College of Ireland

Supervisor: Dr David Hamill

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Aji Vishwambharan Poovannapoikayil		
Student ID:	x22184431		
Programme:	MSc. Data Analytics	**Year:**	2023-2024
Module:	MSc. Research Project		
Supervisor:	Dr David Hamill		
Submission Due Date:	12/08/2024		
Project Title:	Indian Sign Language Detection and Translation using Deep Learning and Text-to-Speech		
Word Count:	9632 **Page Count:** 26		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Aji Poovannapoikayil

Date: 12th August, 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

MSc. Research Project

Indian Sign Language Detection and Translation using Deep Learning and Text-to-Speech

Your Name/Student Number	Course	Date
Aji Poovannapoikayil / x22184431	MSc. Data Analytics	12/08/2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
Grammarly	Grammar and Punctuation Checks	https://www.grammarly.com

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Grammarly	
Grammarly is used to rephrase sentences and check for grammar and punctuations.	

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

Indian Sign Language Detection and Translation using Deep Learning and Text-to-Speech

Aji Poovannapoikayil
x22184431

Abstract

This paper explores the integration of YOLOv10, a cutting-edge object detection model, for real-time static sign recognition and translation of Indian Sign Language (ISL) into regional Hindi text and speech. Motivated by the demand for more effective communication tools for the deaf community in India, particularly in non-English-speaking regions, this research compares the performance of YOLOv10 against the established YOLOv5 model. The study focused on key metrics such as accuracy, Mean Average Precision (mAP), precision, and inference time to assess the efficacy of YOLOv10 in ISL detection. The results demonstrate that YOLOv10 significantly improves upon the mAP@50:95 accuracy of YOLOv5, which was 89%, achieving 99% mAP@50:95 accuracy with 25 epochs for trained ISL words, along with superior precision and faster inference times. However, the false positive cases suggest potential overfitting, indicating the need for future work to refine the model. The findings further suggest that YOLOv10 offers enhanced real-time performance, making it a viable solution for improving accessibility and communication in both rural and urban areas of India. However, the research also identifies limitations, particularly related to the availability of diverse, high-quality data and the time-intensive nature of manual annotation. Future work will address these challenges by expanding the dataset to include a wider range of words and dynamic gestures, and by exploring the integration of Long Short-Term Memory (LSTM) networks to better capture complex sign language elements. This study not only advances the field of sign language recognition but also holds significant potential for commercial applications, particularly in developing assistive communication tools tailored to the Indian context.

1 Introduction

1.1 Background

Indian Sign Language is a sign language that those with speech and hearing impairments use to interact with others. A person’s ideas, feelings, and experiences can be expressed both manually and visually through sign language. The three main sign languages Pidgin Signed English, American Sign Language, and Signed Exact English each utilise English as their foundational language. Indian locals typically learn to speak their native languages, which are not English, and access to such education is limited in rural areas. Consequently, a language like English is unlikely to become a primary sign language for them. In India, accessibility is often overlooked, especially in tier 2 and tier 3 cities as well as rural areas. Most of the previous research has predominantly focused on American Sign Language, but this project will concentrate on Indian Sign Language (ISL). While the past studies have extensively covered the identification of letters and digits, the project aims to identify sign for words and phrases in ISL that would enhance communication (Bormane & Shirbahadurkar, 2023). Approximately 60 million people in India are disabled, with 42.5% of them being female. Notably, 75% of the disabled population lives in rural areas where English is not their primary language. Additionally, 20% of computer users have hearing impairments (Varshney, 2016).

Indian Sign Language (ISL) is a visual language that relies on hand movements, wrist orientations, and non-manual facial expressions to convey meaning. Signs are communicated through specific wrist and hand positions, each representing distinct signs and letters. Sign language gestures in ISL are divided into three groups: one-handed, two-handed, and non-manual signs. These groups are further classified into static and dynamic signs, as shown in Figure 1. Static signs involve no hand movement, while dynamic signs entail movement of the hands or body parts motion. (Dhanjal & Singh, 2022).

This research focused on manual static signs for various words like “doctor”, “afraid”, “bad”, “appreciate”, etc. rather than Alphabets. There is potential to improve the model in the future by including non-manual. Another goal is to bridge communication gaps and improve accessibility for rural populations by providing translations in their local language Hindi. A significant challenge in working with ISL is the lack of available data and the inconsistency when dealing with multilingual sign languages, which arises from the fact that ISL was adapted from American Sign Language (ASL) without standardization. To address this, it is essential to establish a common language for ISL, such as English, which can then be translated into other respective languages. The research opted to explore deep learning models and prioritize a vision-based approach over a glove-based one.

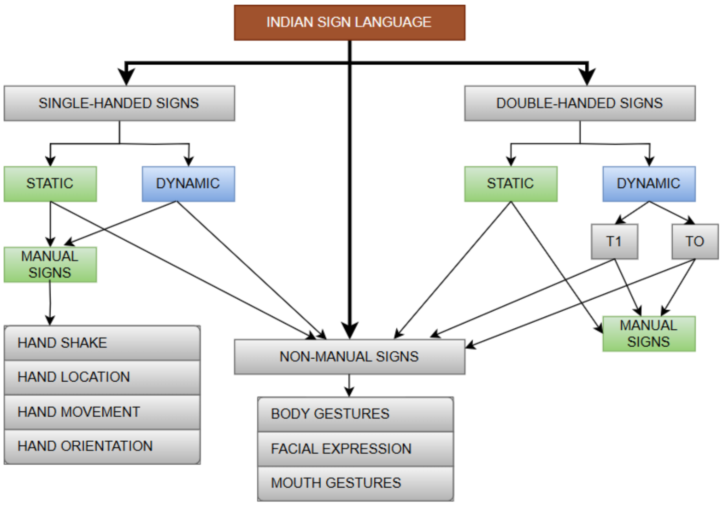


Figure 1: Hierarchical Framework of Indian Sign Language (Dhanjal & Singh, 2022)

1.2 Motivation

India’s status as a multilingual nation makes teaching the deaf particularly challenging. Hearing impairment is the second most prevalent disability in the country, yet reliable data from rural areas remain scarce. Despite a substantial population of individuals with auditory loss, there is a lack of sign-language consistency and accessibility in sign language across India. The nation’s hearing-impaired community has been overlooked by the Indian government, their priority is more focused on advancing technological resources and infrastructure. Several factors, such as an ageing population, lifestyle factors, and untreated ear infections, are expected to contribute to the increasing prevalence of hearing loss. However, assistive devices remain largely inaccessible because of their high cost and restricted availability.

While technology has rapidly advanced in this decade, with the internet and smartphones making digital communication more affordable and accessible, sign language should not be a barrier to communication, particularly for those committed to working in rural areas. Access to sign language is vital for deaf individuals to communicate with the outside world, as it is playing an essential role in their emotional, social, and linguistic development. This study provides insights into the diverse technologies that are available to support individuals who are deaf or hard of hearing. Additionally, it explores the obstacles associated with the advancement and implementation of these technologies, including issues related to accuracy, reliability, and usability.

The enhanced performance of word detection from static sign images and translating them into Hindi is the significance of the research. To ensure usability, the proposal seeks to improve accuracy by comparing existing object detection models with current technology. While numerous studies have focused on identifying letters in various sign languages, such as Tamil, Marathi, Kannada, etc. by employing computer vision and image augmentation techniques, this study will specifically target word recognition. By employing computer vision techniques with the YOLOv10 model, the research seeks to achieve faster object detection, thereby increasing both accuracy and speed.

1.3 Research Question

How well does integration of YOLOv10 affect the accuracy Mean Average Precision (mAP), accuracy, precision, and speed (inference time) of Indian Sign Language recognition and translation into regional Hindi text in comparison to existing YOLOv5 model?

Indian Sign Language is an essential interaction tool for the deaf and hard-of-hearing community in India. Traditional approaches to ISL recognition have focused primarily on detecting individual letters or simple gestures, often using models such as YOLOv4, YOLOv5, CNN, etc. While these models have shown promising results, there is a growing need for more accurate, faster, and reliable systems that can handle the recognition of entire words, which is essential for real-time application and translation system.

The impact of incorporating the most recent YOLOv10 model into the ISL recognition and translation process is investigated in this study. With its advancement over YOLOv5, YOLOv10 promises to improve on several aspects, including accuracy, precision, and overall performance in object detection. The goal is to determine whether YOLOv10 can provide significant enhancements, increasing the technology usefulness and efficacy for practical usage.

1.4 Proposed Solution and Contributions

The proposed solution involves a systematic comparison between the YOLOv10 and YOLOv5 models in the context of Indian Sign Language Detection and translation. The research will be conducted in different phases and detailed explanation is in Section 3. Research Methodology.

- Data Collection and Preprocessing
- Model Training
- Performance Evaluation
- Indian Sign Language translation Flask Web Application
- Analysis and Conclusion

This study intends to advance the field of sign language recognition by offering insights into how state-of-the-art object detection models can improve the accuracy, speed, and reliability of sign language translation.

1.5 Structure of Document

The content of this research paper is organized into the following sections. Section 2 presents a review of the related Literature. Section 3 covers the research methodology and its various aspects. Section 4 details the design specifications, and Section 5 briefly discusses the steps undertaken for implementation. Section 6 focuses on Evaluation, while Section 7 provides the conclusion and outlines future work.

2 Related Work

Sign languages around the world, like spoken languages, have many variations. Some, such as American Sign Language (ASL) and British Sign Language (BSL), are quite well-known. To investigate how existing research can aid in recognizing sign language in regional Hindi text, I have studied various sign language recognition systems, including ISL (Indian Sign Language). Much of the research focuses on detection of single character or numbers, which limits communication to just the alphabets or numbers. There is a need to explore computer vision-based methods capable of detecting both static and dynamic words or phrases, as taught by professional instructors in the deaf community. This inquiry focuses on ISL recognition and translation, with the goal of leveraging technology to advance ISL sign detection and translation. Let's explore the research conducted in this field.

2.1 Computer Vision-Based Translation

The reviewed study by (Zahid, et al., 2023), proposes adopting a computer vision-based system to recognise and classify Urdu Sign Language for differently-abled individuals. The system employs image processing techniques, including OpenCV, TensorFlow, and linear regression, to convert sign language to voice and written text. This model demonstrates a significant improvement in accuracy. However, the research also places additional emphasis on data diversity and the challenges of real-world application. Their future work focus on expanding the dataset and testing the system in various practical scenarios to ensure robustness.

The study conducted by (Vijitkunsawat, et al., 2023) addresses the lack of resources by introducing a novel video dataset for Thai Sign Language digit recognition. It compares the performance of four deep learning architectures CNN-Mode, VGG-Mode, CNN-LSTM, and VGG-LSTM under two scenarios: hand-cropped images and whole-body poses. The results show that the most accurate model, with an accuracy of 81.25% and an F1-score of 85.21%, is

VGG-LSTM. While models like CNN-Mode and VGG-Mode exhibit good training accuracy, they perform poorly in out-of-sample generalization. The study also employs YOLOv4 for recognizing hand signs, achieving a precision of 84.47% and a recall of 75.73%. The authors suggest that their future research will address the use of YOLOv10, which is faster and more accurate.

As a significant contribution to the field, the WASL dataset is introduced in the study by (Li, et al., 2020). This dataset is notable for its extensive vocabulary and large quantity of samples, addressing the issue of limited vocabulary in existing datasets and highlights the need for competent models for sign recognition. Collected from monocular RGB videos, the dataset ensures accessibility without requiring specialized equipment. Furthermore, the study introduces the novel Pose-TGNCN model, which captures both spatial and temporal dependencies in pose trajectories and compares holistic visual appearance. Their findings indicate that appearance-based and pose-based methods perform similarly, achieving an accuracy of up to 62.63% on 2000 words and glosses. To address the challenges of word-level sign language recognition, the authors employed advanced algorithms such as few-shot learning. However, the study lacks a thorough exploration of the trade-offs in each approach, which could provide deeper insights.

(Luong, 2023) investigates computer vision based expression and pose-based techniques for sign language recognition (SLR). The study address critical issues such as vocabulary size, subtle hand and body motions, and regional variations, while also suggesting deep learning as a potential remedy to these challenges. Traditional models, including HOG based features, as well as advanced techniques like CNNs, RNNs, and GCNs, have been utilized for SLR tasks. This paper reports an improvement in baseline accuracy on the WLASL dataset from 43.02% to 55.96%, showcasing progress while underscoring the need for further refinement. However, challenges such as overfitting and limited training data remain significant.

2.2 Overview of YOLO Object Detection Model

(Hussain, 2023) recent study provides an extensive analysis of the evolution of YOLO object detection models, from YOLOv1 to YOLOv8, highlighting their applicability in identifying industrial defects in digital manufacturing. The paper also emphasises the real-time performance and efficiency of various YOLO variants. According to his analysis, YOLOv8 surpasses YOLOv5 and YOLOv6 in computational efficiency. Although the research focuses on industrial applications, these models could also be used for detecting hand signs in video frames and for ISL, offering faster object detection and improved accuracy.

Research by (Sharma, et al., 2022) presents a real-time word-level sign language recognition system utilizing YOLOv4. While the use of deep learning is noteworthy, the paper neglects several critical aspects. It fails to address YOLOv4's limitations in recognizing intricate hand gestures, which are essential for sign language. Additionally, there is insufficient evidence supporting the system's ability to capture complex hand movements. The paper also lacks details on the diversity of signers and gesture variations in ISL. Moreover, the evaluation metrics, such as mean average precision (mAP), are inadequate as they do not consider the static and temporal dynamics intrinsic to sign languages.

An inquiry by (Alexey Bochkovskiy, 2020) introduces YOLOv4, a model designed to improve the speed and accuracy of object detection using CNNs. The study presents several new features and methods, including self-adversarial training, cross-stage partial connections, and weighted residual connections. However, it lacks clarity in explaining how these specific features enhance performance. While the study acknowledges the need for neural networks in production systems to operate at high speeds, the rationale behind selecting GPUs and VPUs is not adequately justified. The research compares these innovative features for object detection against other detectors like RFBNet and RetinaNet, motivating the adoption of the latest object detection model, YOLOv10, in an ongoing ISL project.

A very recent study by (Alif & Hussain, 2024) presents YOLOv10, a significant upgrade in YOLO family of real time object detection models. YOLOv10 introduces a more efficient backbone network, optimized training strategies, and advanced feature fusion methods. These enhancements result in improved accuracy and faster inherent times as compared to previous versions. The model achieved higher mean Average Precision(mAP) on benchmark datasets like COCO while maintaining real time performance. These could be particularly useful for real time applications due to its speed and efficiency. YOLOv10 offers a balanced solution with superior performance for hand sign detection (Alif & Hussain, 2024).

A study by (Ali, 2021) evaluates YOLO Algorithms YOLOv3, YOLOv4, and YOLOv5, by proposing a solution for automating sign language detection. YOLO, which employs CNNs for real-time object detection. YOLOv3 has lower performance metrics, with a 71% mAP, compared to YOLOv4 and YOLOv5, which have mAP values of 85% and 87% respectively. YOLOv5 outperforms the other models in additional metrics like precision and recall, suggesting it is the most effective for sign language detection. The author concludes that the choice of algorithm depends on a trade-off between accuracy and computational efficiency. Although this research demonstrates accuracy with a balanced trade-off, it is notable that performance could still be improved.

A very recent study by (Lakshmi & .D, 2024) reviews the use of advanced object detection techniques by comparing YOLOv5, YOLOv7, and YOLOv8 in terms of precision, recall, and mAP@0.5:0.95, using a dataset of 2216 images. YOLOv8 outperforms YOLOv5 and YOLOv7 with superior accuracy, achieving the highest mAP@0.5:0.95 scores. While YOLOv5 is effective, it shows lower precision (52.8%) and recall (56.4%) compared to YOLOv8. YOLOv7's performance is intermediate, with only negligible improvements over YOLOv5. It remains advantegous for its lower computational demands. Future research could further explore these trade-offs and assess the impact of newer YOLO versions on real-world applications.

A paper by (Biyani, et al., 2023) presents real-time sign language detection models based on the YOLOv5 algorithm, distinguishing between American Sign Language, Hindi/Marathi

Sign Language, and static gestures. Trained on custom dataset of 1892 images, these models achieved impressive real-time performance, with an average detection time of 0.05 seconds per frame. Compared to Faster R-CNN and Mask R-CNN, YOLOv5 models demonstrated superior accuracy and real-time efficiency. Despite these strengths, the study identifies issues with false negatives and false positives, particularly with ASL model. To enhance accuracy, the dataset size and training duration could be increased. While the models shows promise for improving communication for deaf and mute individuals, future research could address these inconsistencies with a more efficient YOLO model.

2.3 Overview of Other Deep Learning Models

Research by (Patil, et al., 2021) addresses the challenge of bridging the communication gap between individuals with hearing or speech disabilities and those without such disabilities through Indian Sign Language (ISL) recognition. The authors effectively highlight the unique difficulties of ISL, particularly its reliance on two-handed gestures, which complicates gesture recognition compared to single-handed system like ASL. The proposed system uses a Convolutional Neural Network (CNN) to classify hand gestures captured via webcam, achieves an accuracy of 95%. However, the paper could benefit from a thorough discussion of the dataset used, especially given the known scarcity of ISL datasets. Additional advancements could facilitate the recognition of words like " today" or " you," enabling basic conversational interactions rather than mere letter recognition.

This study by (Alahmadi, et al., 2023) presents an improved object detection system with a ResNet-101 backbone, based on modified YOLOv4 model, designed for visually impaired people. When compared to the conventional Darknet backbone, ResNet-101 provides better feature extraction, which helps the system to reach an impressive accuracy of 96.34% on the MS COCO dataset. One important feature is the integration of text-to-speech conversion, which gives immediate auditory feedback to help with object recognition and navigation. However, the study would benefit from a discussion on the model’s limitations, particularly in diverse environments. While the papers hints at future generation of aural and touch-based feedback. The study significantly advances the accuracy and usability of assistive technology for visually impaired.

This study by (Velmathi & Goyal, 2023) explores the communication challenges faced by the deaf community, particularly in interactions with those who do not know sign language. By contrasting the capabilities of Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), the research aims to create a system that can translate Indian Sign Language (ISL) into text or speech. According to results, CNNs perform better at identifying static sign language, catching individual letters and characters, while LSTMs excel at processing dynamic gestures, obtaining a 97% accuracy rate. With a training accuracy of around 85% for static signs, there is a room for improvement. The research underscores the need to select models based on the type of sign language, thereby improving accessibility for the deaf community.

The paper by (Khartheesvar, et al., 2023) proposes a method for recognizing isolated words in ISL using the MediaPipe holistic pipeline for feature extraction and Long-Short Term

Memory (LSTM) networks for classification. The method was tested on the INCLUDE dataset, and its subset, INCLUDE-50, achieving impressive accuracy rates of 94.8% on INCLUDE-50 and 87.4% on the full INCLUDE dataset. Additionally, macro averaged F1-scores of 93.5% and 86.6% were obtained for INCLUDE-50 and INCLUDE, respectively. While the results are promising, the method could be further criticized for its reliance on the specific dataset. Additonally, the paper does not address the challenges of scaling this approach on new words.

The Research paper by (Shenoy, et al., 2021) presents a robust system for real-time Indian Sign Language (ISL) hand pose and gesture recognition using grid-based extraction. By obtaining a high accuracy of 99% for static poses and 97.23% for gestures without the need for external hardware. The authors successfully address the major drawbacks of previous solutions, creating a system that is both accurate and user-friendly. Rapid processing times are made possible by the skilful application of Hidden Markov Models (HMM) for gestures and K-Nearest Neighbors (k-NN) for pose classification. However, the paper could benefit from an in-depth analysis of paper across varied lighting conditions and background. Additionally, while the potential for generalizing the system to other sign languages, the lack of detail on how it would adapt to different datasets is a shortcoming. A discussion on the challenges of adapting the system to recognize new words or phrases would strengthen the paper.

2.4 Summary

This research focuses on recognition and translation of Indian Sign Language (ISL), proposing the use of YOLOv10 to improve sign language translation and detection. A significant limitation of existing research is its emphasis on the detection of individual characters or numbers. This study will concentrate on word detection to improve digital communication for deaf community. Although various computer vision-based techniques, such as CNNs, LSTMs, and previous YOLO models, have been explored for sign language recognition, they have encountered challenges related to dataset quality and prediction accuracy.

YOLOv10, a more recent advancement, offers improved processing speed and accuracy through advanced fusion methods and optimized training strategies. This makes it a promising solution for addressing the shortcomings of previous models, particularly in recognizing hand signs and gestures in ISL. The aim of this study to demonstrate that YOLOv10 can offer a more accurate and efficient approach to Indian Sign Language detection compared to earlier YOLO models, potentially expanding the use of technology to assist the deaf community. The following section will briefly outline the proposed methodology and approach used to conduct this research.

3 Research Methodology

This section outlines the comprehensive research procedure used in developing and evaluating the Indian Sign Language (ISL) recognition and translation. The research methodology employed is based on the CRISP-DM framework (Parate, 2020). Refer to Figure.2 for the proposed methodology. This detailed approach provides a comprehensive overview of the process followed to develop and evaluate the ISL Recognition and translation system.

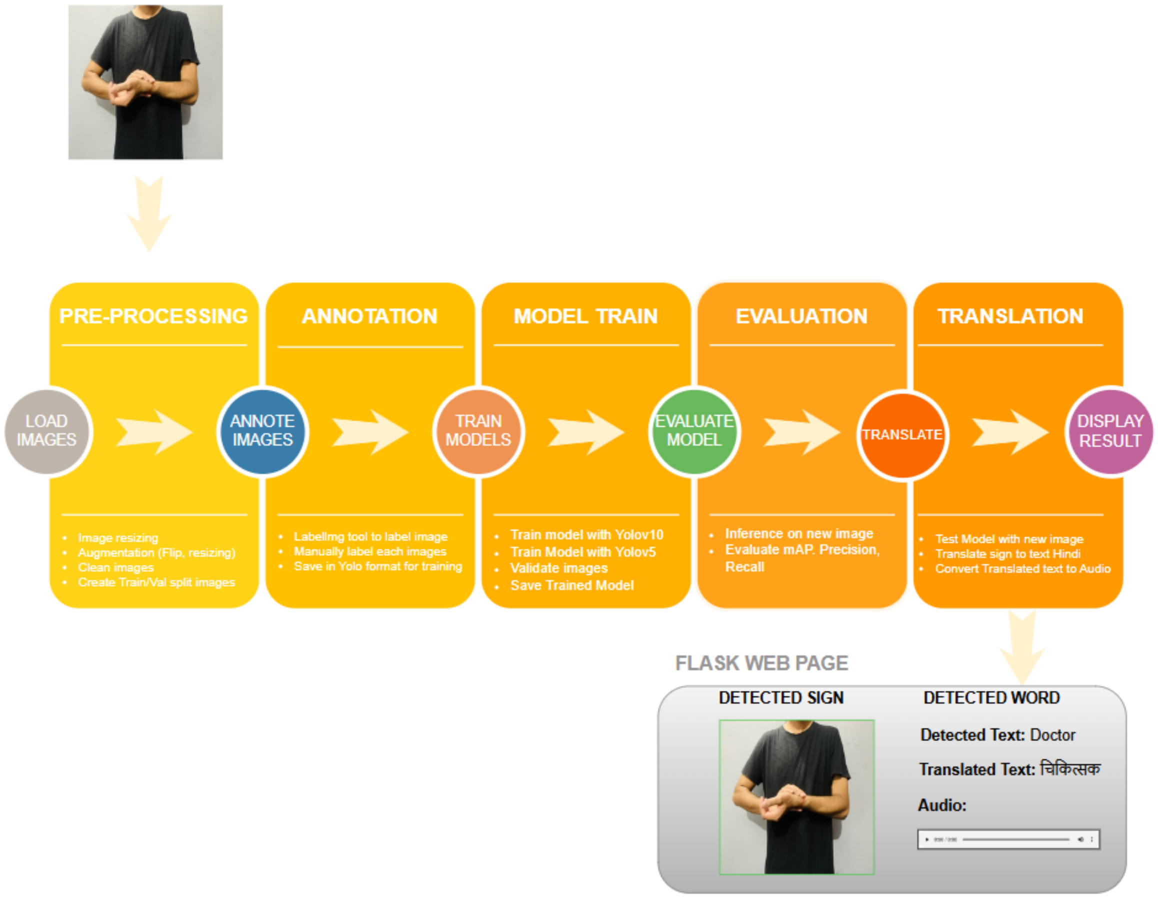


Figure 2: Proposed Methodology for ISL Recognition and Translation to text and audio

3.1 Business Understanding

This main goal of the study is to compare the performance of the most recent YOLOv10 object detection model with the top-performing YOLOv5 model and determine how this advanced technology could enhance the detection of Indian sign language in terms of Mean Average Precision (mAP), accuracy and precision. Although this study represents a relatively small step in contributing to the current body of work in this area, it employs a computer vision-based approach rather than glove-based technology to detect such signs. The aim is to assist people with disabilities in communicating more effectively in today's digital world.

3.2 Data Understanding

The dataset used for this research was curated from publicly available sources (Tyagi & Bansal, 2022). It consists of Twenty ISL words such as 'afraid', 'agree' etc., which are frequently used to express messages or ask for assistance in medical situations. These words are represented by RGB images of hand gestures. Images of eight people, six of whom were male and two were female, between the ages of nine and thirty were taken. In total, there are 18k jpeg images in the dataset. In this stage, identify any noise present and dimensions of the images.

3.3 Data Preparation

This stage involves preparing the raw data for training the model. This includes steps such as data pre-processing, annotation, and data splitting, which are detailed as follows.

3.3.1 Data Pre-processing

In this study, several pre-processing techniques were applied to the dataset to ensure consistency, enhance model performance, and improve generalization. These steps are essential in preparing the images for training the YOLO models and include the following:

- **Image Resizing**: Standardizing all the images to a uniform size to meet the input requirements for the Yolo Models. Python and OpenCV were used to resize all images to 200 x 200 pixels.
- **Data Augmentation**: In Machine Learning, especially computer vision, augmentation describes the process of artificially expanding the dataset by generating altered versions of the images. This method increases data diversity without collecting new data, thereby enhancing the model's capacity for generalization. By applying a series of transformations, this method enhances the model's ability to generalize to new, unseen data. The following data augmentation techniques were employed:

  1) **Blurring:** This technique applies a blur effect with a limit of 3 pixels and a probability of 20%. For instance, blurring was applied to simulate slightly out-of-focus images, making the image appear less sharp.
  2) **Random Brightness and Contrast Adjustment:** This technique adjusts the brightness and contrast of the image randomly with a 20% probability. These adjustments were used to account for different lighting conditions, ensuring that the model could recognize signs in both bright and dim environments.
  3) **Gaussian Noise**: Gaussian noise was added to the image with a probability of 20%, varying between 10.0 and 50.0. This noise creates a grainy or speckled appearance, simulating sensor noise from a camera.
  4) **Horizontal Flip:** This technique flips the image horizontally with a probability of 50%. For instance, an image of a person signing with their right hand would appear flipped, simulating a left-hand sign.
  5) **RGB Shift:** This technique shifts the RGB values of the image channels, altering the colors slightly. For instance, this shift could cause the image to have a reddish, greenish, or bluish tint, adding color variation to the dataset.
- **Image Cleaning**: To ensure data quality, a manual inspection of the images was conducted to identify and remove noisy or irrelevant images that did not accurately represent the target ISL signs. This included removing blurry, incorrect, or irrelevant images.
- **Train/Validation Split:** The dataset was divided into 80:20 ratio for training and validation sets to ensure unbiased evaluation. The sampling techniques were used to ensure that each class is represented proportionally in both sets**.**

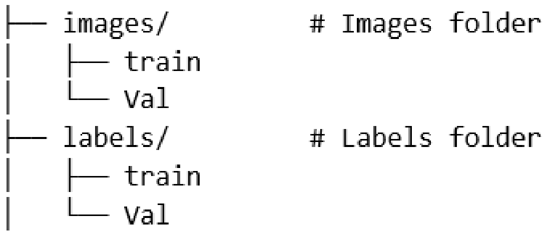
3.3.2 Annotation

Accurate image annotation is a crucial step in the dataset preparation process, as it directly influences the performance of the deep learning models used in this study. This phase involves labelling each image with the appropriate Indian Sign Language (ISL) sign, which is essential for training object detection models. During this step, bounding boxes are manually created around each sign using the graphical image annotation tool, LabelImg, to ensure precise annotation. Inaccurate annotations may lead to degraded model performance, necessitating additional rework.

Given that model training will utilize the YOLOv5 and YOLOv10 frameworks, it is imperative that all annotations adhere to the YOLO format. The following steps outline the detailed procedure for labelling images:

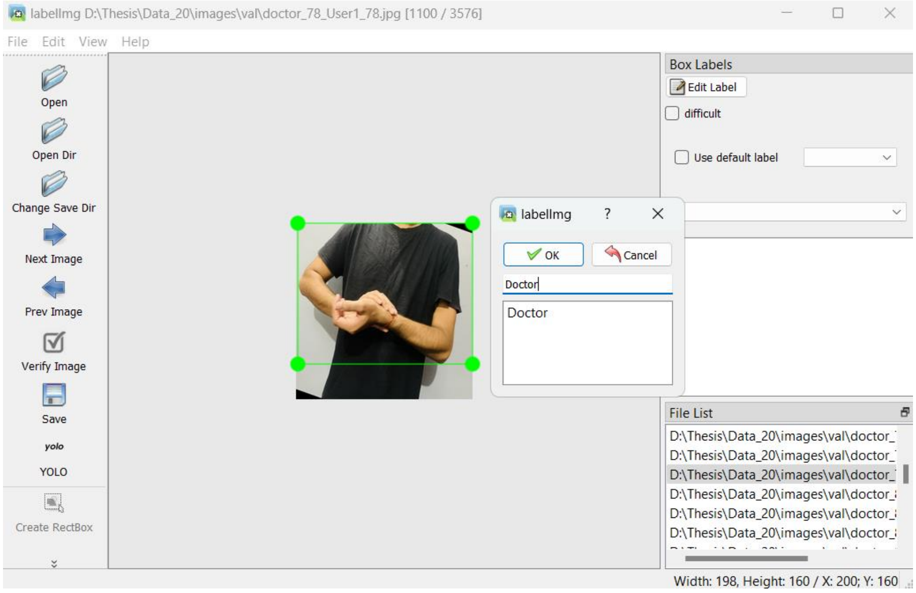
- **Installation of LabelImg:** Begin by downloading and installing the LabelImg software tool, which will be used to create the necessary annotations for both training and validation datasets.
- **Loading of Images:** Once LabelImg is installed, load the dataset images from the previously prepared training and validation sets into the tool.
- **Bounding Box Creation:** For each image, manually draw bounding boxes around each sign present. These bounding boxes must accurately enclose the sign, ensuring minimal overlap with over regions. After creating the bounding box, assign the corresponding label to the sign, referencing the class outlined in the datasets (see Figure 4). Precise annotation at this stage is essential for reducing errors during model training.
- **Saving Annotations in YOLO Format:** After assigning labels, annotations must be saved in the YOLO format, where each annotation is stored in a corresponding text file with the same filename as the image. The YOLO annotation format consists of the class index, x_center, y_center, width, and height, all normalized to the range [0, 1]. An example of the YOLO annotation format is provided below:

  class x_center y_center width height
  0 0.5 0.5 0.25 0.25
- **Validation of Annotations:** After saving the annotations, it is essential to verify that each image has a corresponding text file with accurate labels. Manual validation of the bounding boxes and their corresponding labels to ensure the integrity of the dataset.
- **Data Structure:** The final dataset should adhere to the structure illustrated in Figure 3, where all images and their corresponding annotation files are organized systematically. This structure critical for the next training phase.
- **Creation of** \`data.yaml\` **File:** A data.yaml file must be created to define the dataset configuration. This file includes the class names and the paths to the training and validation datasets, as shown in Figure 3. Proper configuration of this file ensures that the YOLO models can access the necessary data during the training process.



```
├── images/        # Images folder
│   ├── train
│   └── Val
├── labels/        # Labels folder
│   ├── train
│   └── Val
```

Figure 3: Folder Structure of processed data for training


labelImg D:\Thesis\Data_20\images\val\doctor_78_User1_78.jpg [1100 / 3576]

Doctor

D:\Thesis\Data_20\images\val\doctor_
D:\Thesis\Data_20\images\val\doctor_
D:\Thesis\Data_20\images\val\doctor_
D:\Thesis\Data_20\images\val\doctor_
D:\Thesis\Data_20\images\val\doctor_
D:\Thesis\Data_20\images\val\doctor_

Width: 198, Height: 160 / X: 200; Y: 160

Figure 4: LabelImg tool for Annotation

3.4 Model Training

To achieve optimal performance in Indian Sign Language (ISL) sign detection, this study employs two state-of-the-art object detection models: YOLOv5 and YOLOv10. These models were selected based on their demonstrated superiority in objection detection tasks. YOLOv5 is widely regarded as the best-performing models, known for its speed and accuracy in detecting objects in real-time applications. YOLOv10, a more recent version, introduces enhancements that reportedly offer improved accuracy and precision. To evaluate the comparative performance of these two models, both YOLOv5 and YOLOv10 are trained and evaluated on the annotated ISL dataset. This process involves fine-tuning pre-trained models using transfer learning, followed by hyperparameter optimization to ensure robust model performance. The specific steps for training are outlined below:

- **Fine-Tuning Pre-Trained Models:** Both the YOLOv5 and YOLOv10 models are initialized with pre-trained weights, which have been trained on large, generic datasets. These pre-trained models are then fine-tuned on the ISL dataset, allowing them to adapt to the specific task of sign language detection.
- **Hyperparameter Optimization:** Key hyperparameters, such as learning rate, batch size, and the number of epochs, are optimized to improve model performance. Grid search is employed to systematically explore combinations of these hyperparameters, with reference to the configurations outlined in Table 1. This ensures that the models achieve the best possible balance between accuracy and computational efficiency during training.

3.5 Evaluation

The evaluation phase is important in evaluating the model performance of the trained YOLOv5 and YOLOv10 models. This phase involves both the quantitative evaluation of the model performance and the qualitative analysis of inference results on new, unseen images. The procedures followed for evaluating the models are outlined below:

- **Performance Metrics**: Several key metrics are employed to measure and compare the performance of both models. These metrics include:

1) **Mean Average Precision(mAP):** This metric calculates the average precision scores at varying levels of recall and serves as the primary evaluation metric for object detection tasks.
2) **Accuracy:** Accuracy measures the overall percentage of correctly predicted signs, providing a straightforward assessment of model effectiveness
3) **Precision and Recall:** Precision is used to evaluate the proportion of correctly identified signs out of all predicted instances, while recall measures the proportion of true positive detections out of the total relevant instances. These metrics offers insights in terms of false positives and false negatives.
4) **Inference time:** To ensure real-time performance, the average inference time per image is calculated. This metric indicates the computational efficiency of the model.

- **Monitoring Training:** Throughout the training process, TensorBoard is utilized to monitor key metrics such as training loss, validation accuracy, and other relevant performance indicators. This real-time monitoring ensures that the models converge properly and that overfitting is avoided.
- **Results visualization:** To facilitate a deeper understanding of the model's performance, the results will be visualized using precision-recall curves and bar charts. The precision-recall curves would provide a detail analysis of the model's strengths and weaknesses, while the bar charts will summarize the performance of each model across various metrics.

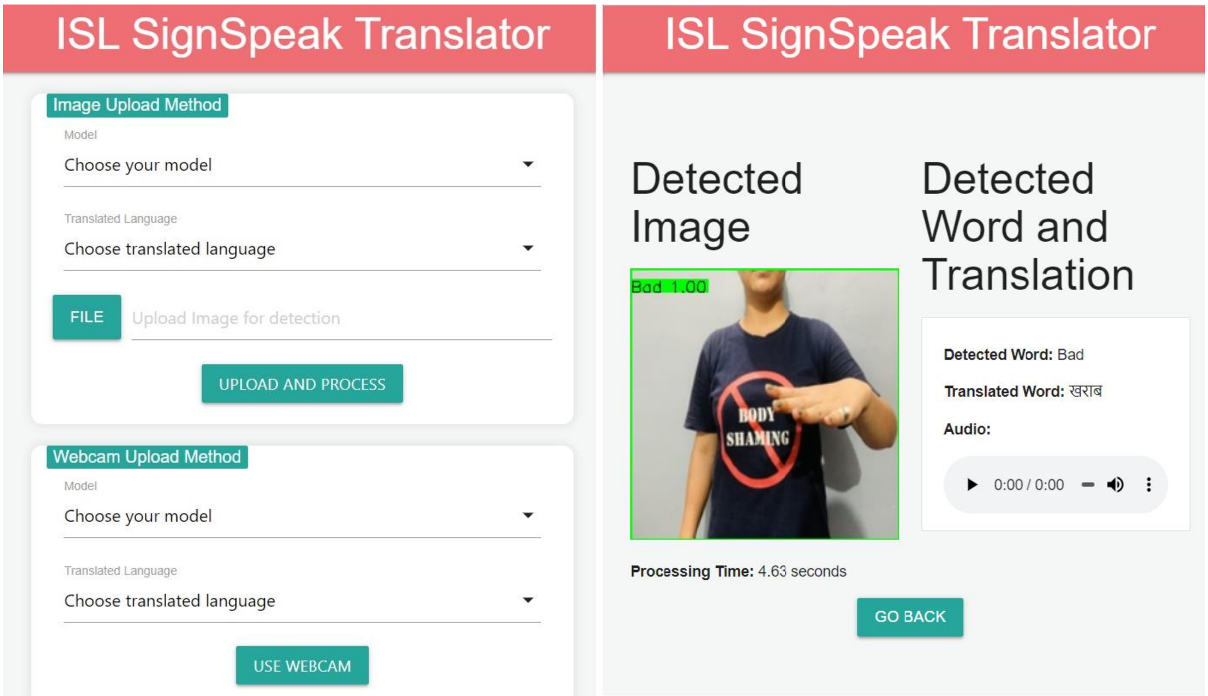
3.6 Deployment

To facilitate the demonstration and testing of the trained models, a Flask-based web application has been developed, providing a user-friendly interface for sign detection and translation. This application is deployed locally on localhost `http://127.0.0.1:5000` and will not be hosted on any external servers. As depicted in Figure 5, the application allows users to upload images containing signs, which are then processed by the trained models for detection. The user interface displays the results, including the translated text and voice output.

For the translation and speech integration functionality, the system integrates the Google translator API to convert detected signs into Hindi Text. This API ensures accurate and contextually relevant translations. Once the text is generated, it is converted into speech using gTTs (Google Text-to-Speech) library. The audio output is then played back to the user through the playsound library, providing a seamless, interactive experience. This web applications demonstrates the practical implementation of the trained models in real-time sign detection and translation, showing effectiveness of proposed methodology. Details of the modelling architecture used in this research are provided in Section 4.

Table 1: YOLO Hyperparameters

Hyperparameters	**Yolov5**	**Yolov10**
Image size	256	256
Epochs	25	25
Batch	8	8
Learning rate	0.001	0.001
Optimizer	AdamW	AdamW



Detected Word: Bad

Translated Word: खराब

Processing Time: 4.63 seconds

Figure 5: Flask Web Application for Indian Sign Language Recognition

4 Design Specification

This section provides a brief overview of the design specifications for the Indian Sign Language (ISL) Recognition and Translation to Text and Voice output system. It identifies and describes the model architecture underlying the implementation, and includes a description of the functionality of the models and algorithms used.

4.1 Evolution of YOLO

Software development firm Ultralytics is well-known for producing advanced computer vision tools, with an emphasis on object detection, and image recognition. They are the one who create YOLO (You Only Look Once) models, which are most popular because of their accuracy and speed for real-time object detection. This framework developed by Ultralytics, is widely recognized for its effectiveness in this field.

A significant breakthrough in real-time object detection has occurred with the evolution of YOLO (You Only Look Once), from its inception in 2016 (Terven, et al., 2023) to the latest YOLOv10 model. In contrast to more conventional techniques like Faster R-CNN and SSD, YOLO's single-shot detection framework transformed the field by enabling the model to predict bounding boxes and class probabilities directly from full images, thereby reducing computation time. Each iteration of YOLO has improved in speed, accuracy, and efficiency. For instance, YOLOv6 and YOLOv7 embedded anchor-free detection and sophisticated augmentation techniques for industrial and real-world robustness, while YOLOv5 introduced CSPDarknet53 for improved gradient flow. Further advancement in YOLOv8 and YOLOv10, which utilize an EfficientNet backbone and feature fusion techniques, have raised the bar for deep learning speed and accuracy.

4.2 Proposed Model Architecture of YOLOv10

The design of YOLOv10 incorporates several significant innovations to improve performance and efficiency, while building on the advantages of earlier YOLO models. This part describes the new components that YOLOv10 has integrated into his architecture (Ao Wang, 2024). The YOLOv10 model architecture consists of the following key components.

- **Backbone:** The extraction of features from the input images is handled by the backbone. The foundation of YOLOv10 is an improved version of CSPNet (Cross Stage Partial Network). By reducing computational redundancy and enhancing gradient flow, this improvement seeks to provide better feature extraction at lower computational cost.
- **Neck:** The YOLOv10 neck is designed to collate features from various scales and pass them to the head for final prediction. It incorporated layers from the Path Aggregation Network (PAN) to review multiscale feature fusion.
- **One-to-Many Head:** During training, the One-to-Many Head produces multiple predictions for each object. This component provides rich supervisory signals, which raises the model's learning accuracy.
- **One-to-One Head:** During inference, the One-to-One Head is intended to produce a single best prediction for each object. This design part lowers latency and boosts efficiency by doing away with the requirement for Non-Maximum Suppression (NMS).

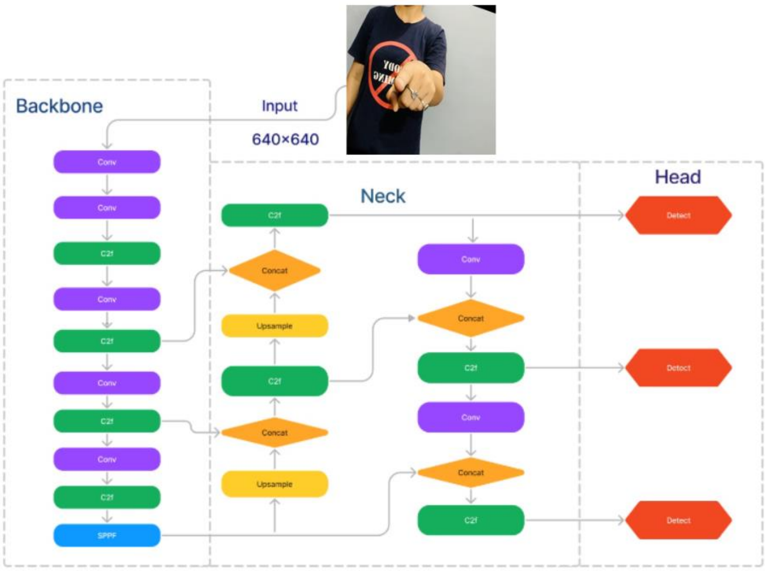


Figure 6: YOLOv10 Architecture Schema (Ao Wang, 2024)

4.3 Existing Model Architecture of YOLOv5

The advanced object detection model YOLOv5, created by Ultralytics, expands upon the original YOLO architecture with several enhancements in efficiency and accuracy. YOLOv5's architecture can be broken down into multiple essential components (Karthi, et al., 2021).

- **Backbone:** The feature extraction is handled by the backbone. YOLOv5 use CSPDarknet53, which enhances gradient flow and minimizes the computational redundancy by integrating Cross Stage Partial Networks (CSPNet). This component uses batch normalization, activation functions, and a sequence of convolutional layers to extract hierarchical features from the input image

- **Neck:** The neck combines features from various scales and passes them to the head for prediction. PANet facilitates multiscale feature fusion, guarantees that objects of different sizes are detected using both coarse-grained and fine-grained features.
- **Head:** The final predictions, such as bounding boxes, objectness scores, and class probabilities, are produced by convolutional layers at the top of YOLOv5. The final detected objects are obtained by processing the tensor that predicts bounding boxes, using anchor boxes and non-max suppression.

4.4 Differences between YOLO models and YOLOv10

The progression of YOLO object detection models from YOLOv5 to YOLOv10 demonstrates notable improvements in architecture, precision, and effectiveness (Terven, et al., 2023). Table 2 compares and summarizes the key features and improvements of each version.

Table 2: Comparison Table: YOLOv5 to YOLOv10

Feature	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**	**YOLOv10**
Backbone	CSPDarknet53	Modified YOLOv5 backbone	EfficientNet	CSPDarknet (Enhanced)	CSPDarknet (Further Optimized)
Neck	PANet	Improved PANet	SPP and PANet	Enhanced PANet with Feature Fusion	Enhanced PANet with efficiency optimizations
Head	NMS based detection head	Anchor free detection	NMS-based with Bag-of-Freebies	Anchor-free	Anchor-free with dual assignments
Data Augmentation	Standard	Improved for industrial applications	AutoAugment, CutMix, DropBlock	Cutout, Mosaic, Mixup	Advanced Augmentation Techniques
Speed (FPS)	Up to 140 FPS	Optimized for Industrial Settings	Real-time on various hardware	Fastest YOLO model to date	Fastest YOLO model to date
Accuracy (mAP)	High on COCO, PASCAL VOC	High in small object detection	State-of-the-art on COCO, PASCAL VOC	Highest rate of accuracy measured by COCO	Highest mAP on COCO
Parameters	Moderate	Similar to YOLOv5 with enhancements	Increased due to advanced techniques	Low number of parameters and FLOPs	Low number of parameters and FLOPs
Use Cases	General Purpose, Real-time Detection	Industrial Applications	General Purpose with advanced robustness	General Purpose, state-of-the-art detection	General Purpose, state-of-the-art detection

This study determined that YOLOv5, with its wide adoption, robust performance across multiple object detection tasks, and balance of speed and accuracy, is the best YOLO model to compare with YOLOv10. YOLOv5 is well known for its abilities, achieving real-time detection speeds of up to 140 FPS and demonstrating cutting-edge accuracy on benchmarks like COCO and PASCAL VOC. Moreover, the architecture of YOLOv5 is widely supported by the community, is simple to use, and has a wealth of documentation, making it a standard reference point for assessing new models. With respect to real-time object detection and overall model efficiency, in particular, YOLOv5's proven dependability and performance make it perfect benchmark to evaluate the improvements and innovations brought about by YOLOv10.

By overcoming the limitation of earlier YOLO models, YOLOv10 strikes an optimal balance between accuracy and efficiency, making it highly suitable for real-time detection of Indian sign Language. As a significant advancement in the YOLO series, YOLOv10 delivers improved latency, efficiency, and overall performance. The next section, Section 5, will discuss the implementation steps taken in the project.

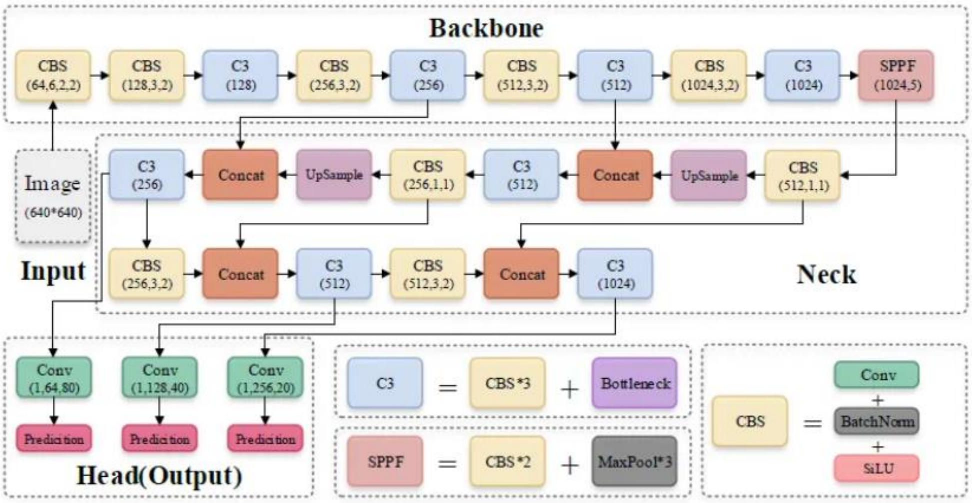


Figure 7: YOLOv5 Model Architecture (Xu, et al., 2021)

5 Implementation

The implementation of the Indian Sign Language (ISL) recognition and translation system involved multiple steps to evaluate and validate the proposed methodology. The experimental setup and specific requirements are described below.

5.1 Configure the environment

The experiment was conducted in two primary environments: a local system for preliminary development, pre-processing, Flask web application, and Google Colaboratory for extensive model training. The Configuration involved setting up a Python environment with essential libraries, including core deep learning frameworks such as Torch, Ultralytics, and torchvision, as well as OpenCV for image processing and modelling tasks. The setup details are as follows:

- The Local System's configuration was equipped with 500GB SSD, 8GB RAM, Intel Core i5 processor.
- A T4 GPU (16GB GDDR6, 2560 CUDA Cores) was used for initial training and hyperparameter tuning to balance speed and cost, while L4 GPU (24GB GDDR6, 7680 CUDA Cores) was employed for full dataset training, enabling faster processing and support for up to 50 epochs.

5.2 Data preparation and Annotation

The first step involved preparing the ISL dataset for training. Images were resized to 200 x 200 pixels to standardize the input dimensions. Each image in the dataset was manually annotated using the labelImg tool, as detailed in Section 3.3.2. The annotations were saved in the YOLO format to ensure compatibility with the models. The processed and annotated dataset is stored on Google drive for training purposes.

5.3 Model Training and Optimization

The subsequent step involved training the YOLOv5 and YOLv10 models on the annotated dataset. Both models were trained on a dataset, consisting of 14304 images for training and 3576 images for validation. The training procedures for each model are described below:

5.3.1 Model Training of YOLOv5

1) **Repository setup:** YOLOv5 was cloned directly from its official GitHub repository, which includes the pre-trained models, custom training scripts, and configuration necessary for fine-tuning the models.
2) **Dependency Installation**: Required dependencies were installed by navigating to the cloned repository. This official repository from Ultralytics is referenced in Section 4.1.
3) **Hyperparameter tuning:** The fine-tuned hyperparameters, include a learning rate 0.001, a batch size ranging from 8 and 16 depending on GPU memory, an image size set to 256, the optimizer 'AdamW', and the number of epochs, typically set between 8 and 50 for convergence.
4) **Transfer learning:** This is a machine learning approach that enhances model performance on a related task by leveraging knowledge obtained from a pre-trained model. To use this approach with YOLOv5, download the pre-trained model weights from their official website. Pre-trained weights including 'yolov5s.pt', 'yolov5m.pt', 'yolov5l.pt' and 'yolov5x.pt' were downloaded. The choice of pre-trained weights depends on factors like computational power and training parameters. In this project, the 'yolov5m.pt' weight were selected for training due to their balance of performance and computational efficiency.
5) **Training Execution:** The training was initially conducted for 8 and 25 epochs, followed by an additional 50 epochs to assess whether there was a reduction in loss and improvement in mAP50 and mAP50-95 accuracy.
6) **Training Results:** The training results are saved in the 'runs/train' folder within the directory of yolov5 repository. The trained 'best.pt' weights were stored for model inference and testing.

5.3.2 Model Training of YOLOv10

1) **Repository Setup:** YOLOv10 was cloned from the official GitHub repository, released in May 2024 by researchers from Tsinghua University.
2) **Dependency Installation**: Required dependencies were installed by navigating to the cloned repository. This official repository from Ultralytics is referenced in Section 4.1.
3) **Hyperparameter Tuning:** Fine-tuned hyperparameters include a learning rate 0.001, a batch size that varies between 8 and 16 depending on GPU memory, an image size of 256,

the ‘AdamW’ optimizer, and several epochs typically set between 8 and 50 for convergence.

4) **Transfer Learning:** Pre-trained weights, such as ‘yolov10s.pt’, ‘yolov10m.pt’, ‘yolov10b.pt’, and ‘yolov10x.pt’ were downloaded. The ‘yolov10m.pt’ weight was selected for training due to intermediate size and performance characteristics.
5) **Training Execution:** The training batch began with 8 epochs, followed by 25 epochs to observe reductions in loss and improvements in mAP50 and mAP50-95 accuracy.
6) **Training Results:** The training results, including model metrics, training and validation batch images and labels, are saved in the ‘runs/train’ directory of the yolov10 repository. The ‘best.pt’ weights were saved for subsequent inference and testing.

5.4 Model Inference and Testing

The third step was designed to evaluate the trained models on unseen data. This experiment validates their effectiveness. The models were tested in two stages:

1) **Inference Testing:** The models were tested using new images that were not included in the training dataset. This assessed the model ability to accurately detect and classify ISL manual signs.
2) **Performance Analysis:** The detailed comparison was conducted done based on precision-recall curves, validation loss, and precision, along with accuracy and inference speed. This analysis ensures the performance of YOLOv10 compared to YOLOv5 model.

5.5 Deployment and User Interaction Testing

The final step of the implementation involves deploying the trained model locally in a user-friendly simple Flask web application. The key steps are as follows:

1) A flask-based web application was developed that allow users to upload images for sign detection. The trained model was used in the application. When a new image is uploaded, the application renders a result page with the detected sign.
2) This process also includes testing the translation of detected signs into Hindi Text and audio. Recognized ISL gestures were translated into Hindi Text using the Google Translator API. The text was then converted to speech using the ‘gTTS’ library and the audio output was played back to the user, completing the translation process.

By adhering these structured implementation steps, Indian Sign Language was effectively converted into an ISL recognition system. The optimal results were achieved with 25 Epochs, a batch size of 8, and a learning rate of 0.001. Both models were trained using these parameters and comparable pre-trained weights for evaluation purpose. The results from the training process will be presented and discussed in detail in Section 6, Evaluation.

6 Evaluation

The main objective of this research is to compare the performance of YOLOv10 model for Indian Sign Language Recognition and translation with the well-established YOLOv5 model. When utilizing newly launched object detection models like YOLOv10, it is crucial to evaluate their performance against existing models to ensure accurate assessment. YOLOv5 is the closest to YOLOv10 in terms of architecture and efficiency, as discussed in Section 4.4. This study aims to assess the impact of integration YOLOv10 by analyzing its accuracy, precision, inference time, and Mean Average Precision(mAP), which are critical for real-time sign language recognition. The mAP50-95 metric measures the average precision of an object detection model across different IoU (Intersection over Union) thresholds, ranging from 0.5 to 0.95, while mAP50 specifically measures precision at an IoU threshold of 0.5. The findings may also contribute to enhancing the current system and improving overall performance. This section will present key results relevant to this research.

6.1 Results of YOLOv5 model

To assess the performance of the YOLOv5 model across 45 Epochs, training was halted when the patience parameter reached 10. This parameter means that if no significant gains is observed after 10 iterations, training would stop. Table 3 provides a summary of key metrics at different epoch checkpoints and illustrate how these metrics evolved as training progressed, offering insights into the model's improvement.

Referencing Table 3. and Figure 8, at epoch 0, the mean average precision (mAP) is 0.05, with a low precision of 3% and high recall of 98%. This indicates that while the model detects many objects, its accuracy is low. By epoch 8, mAP increases to 32%, precision rises to 27%, and recall decreases to 75%, showing progress but with some trade-offs.

By epoch 25, mAP reaches 88%, with precision and recall at 97% and 95%, reflecting enhanced object detection capabilities and high accuracy. By epoch 45, both precision and mAP show a slight increase of 1%, indicating that the models become highly effective at minimizing false positives, thus improving the reliability of its detections. Object loss, which starts at 0.02 at Epoch 0, decreases to 0.00 by epoch 25 and remains stable, confirming a reduction in deduction and localization errors.

Table 3: Summary of YOLOv5 Performance

Epochs	mAP (0.5)	mAP (0.5:0.95)	Precision	Recall	Object Loss
0	0.07	0.05	0.03	0.98	0.02
8	0.37	0.32	0.27	0.75	0.01
25	0.99	0.88	0.97	0.95	0.00
45	0.99	0.89	0.98	0.99	0.00

6.2 Results of YOLOv10 model

Based on the yolov10 performance data, the evaluation reveals significant improvements in detection accuracy and loss reduction over the training epochs. At epoch 0, the model exhibits relatively low performance, with a mean Average Precision (mAP) of 48% at 0.5 Intersection over Union (IoU) and 47% across IoU thresholds ranging from 0.5 to 0.95. At this initial stage, the model has a precision of 43% and recall of 53%, along with a high object loss of 0.83. This states that while the model has a basic foundational capability to detect objects, there is room for enhancement in both precision and recall.

By epoch 8, significant progress is evident. The mAP values increase to 99% at 0.5 IoU and 99% across multiple IoU thresholds, suggesting a significant enhancement in the model's capacity to accurately predict object locations. With a notable decrease in object loss to 0.04, precision has increased to 96% and recall to 97%. This indicates improved learning and decreased detection errors. By epoch 25, the performance metrics remains high, with mAP values holding steady at 99%, and slight increases in precision and recall to 99%, reflecting optimal performance. The model shows a robust learning trajectory with substantial improvements in detection accuracy, precision, and recall, alongside a reduction in object loss, as detailed in Table 4. And Figure 9.

Table 4: Summary of YOLOv10 Performance

Epochs	mAP (0.5)	mAP (0.5:0.95)	Precision	Recall	Object Loss
0	0.48	0.47	0.43	0.53	0.83
8	0.99	0.99	0.96	0.97	0.04
25	0.99	0.99	0.99	0.99	0.05

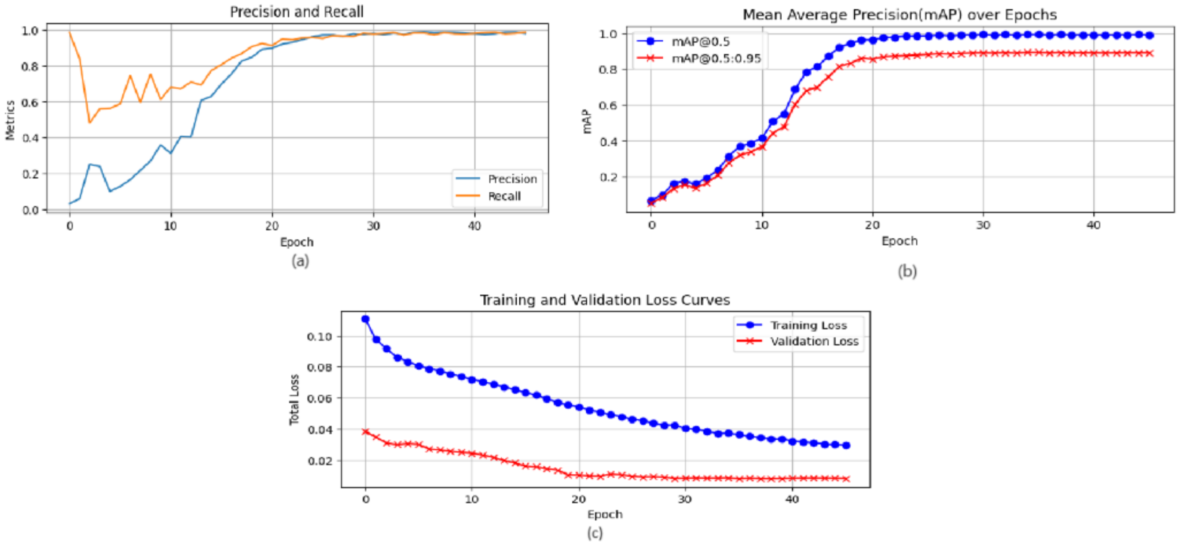


Figure 8: Plotting of YOLOv5 performance (a) Precision and Recall (b) Mean Average Precision(mAP), (c) Training and Validation loss

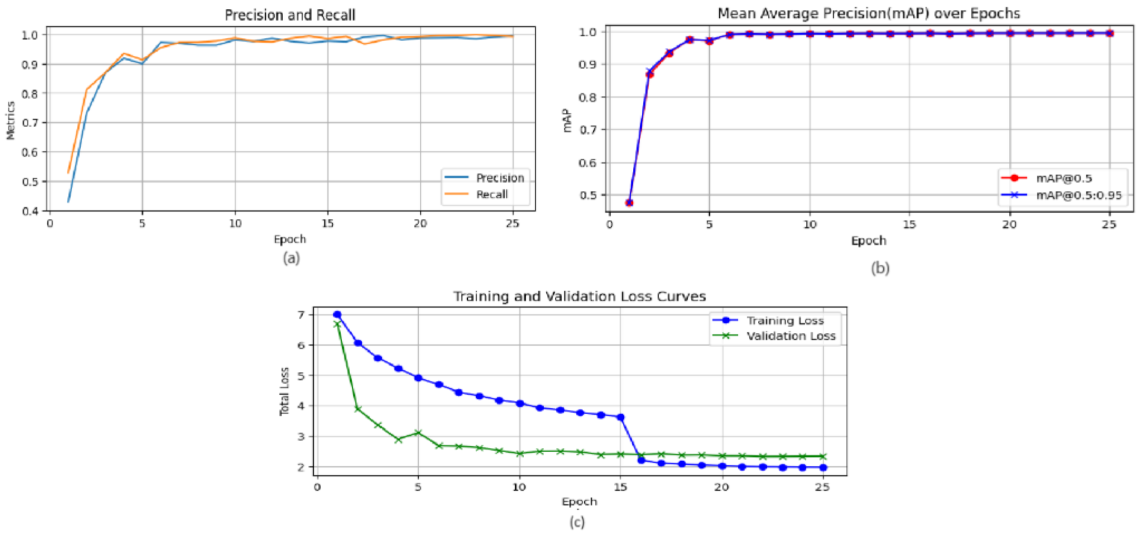


Figure 9: Plotting of YOLOv10 performance (a) Precision and Recall (b) Mean Average Precision(mAP), (c) Training and Validation loss

6.3 Comparison of performance between both models

Comparing YOLOv5 and YOLOv10, it is clear from Figure 10, that YOLOv10 consistently outperform the YOLOv5 across all epochs. The YOLOv10 starts with a higher mAP@50:95 (47%) and achieves near-perfect detection accuracy much earlier in the training process, maintaining these metrics to 99% mAP@50:95 by epoch 25. In contrast, YOLOv5 shows gradual improvement, reaching 89% mAP@50:95 accuracy, but never reaches the same level accuracy as YOLOv10. Inference speed testing was performed on 108 images for both models, with the average time taken calculated after multiple runs. The outcome of this test was completely dependent on the current machine's capacity. The results indicated inference speed of 48.37 seconds for YOLOv5 and 17.76 seconds for YOLOv10, demonstrating that YOLOv10 speed is faster based on these findings.

In terms of precision, as represented in Figure 11, YOLOv10 has a precision of (43%) and Recall (53%) compared to other model's Precision (3%) and Recall (98%) at the initial stage of training. As training progress, YOLOv10's Precision (99%) and Recall (99%) improves rapidly, while other model also improves Precision (98%) but trails slightly behind. The marginal difference in precision becomes critical in applications where minimizing false positives is critical.

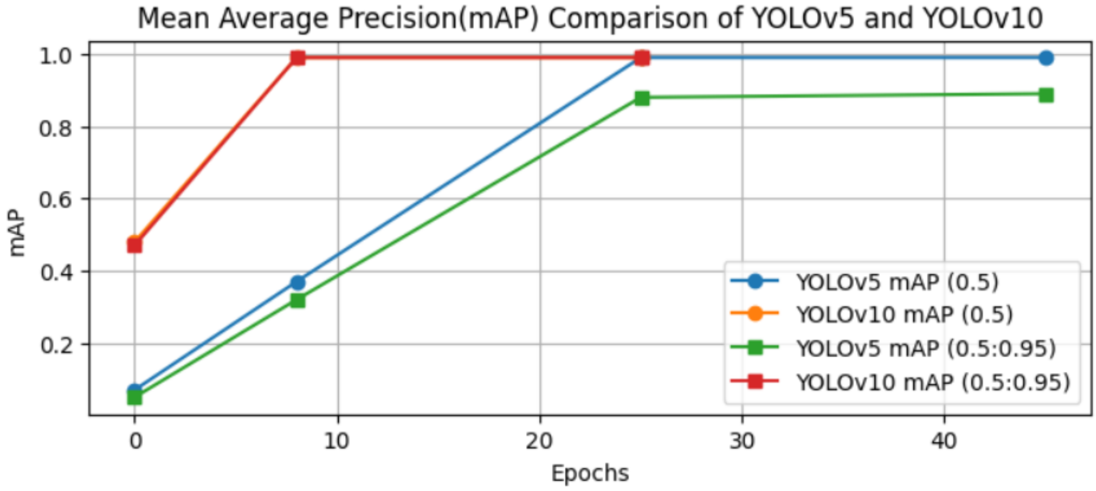


Figure 10: ISL Accuracy comparison plot of YOLOv5 and YOLOv10

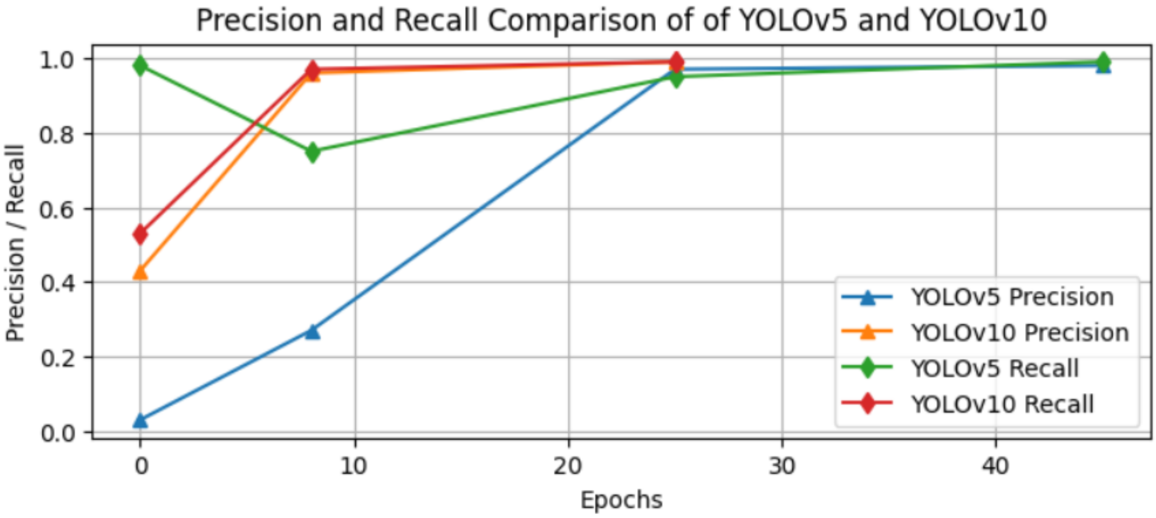
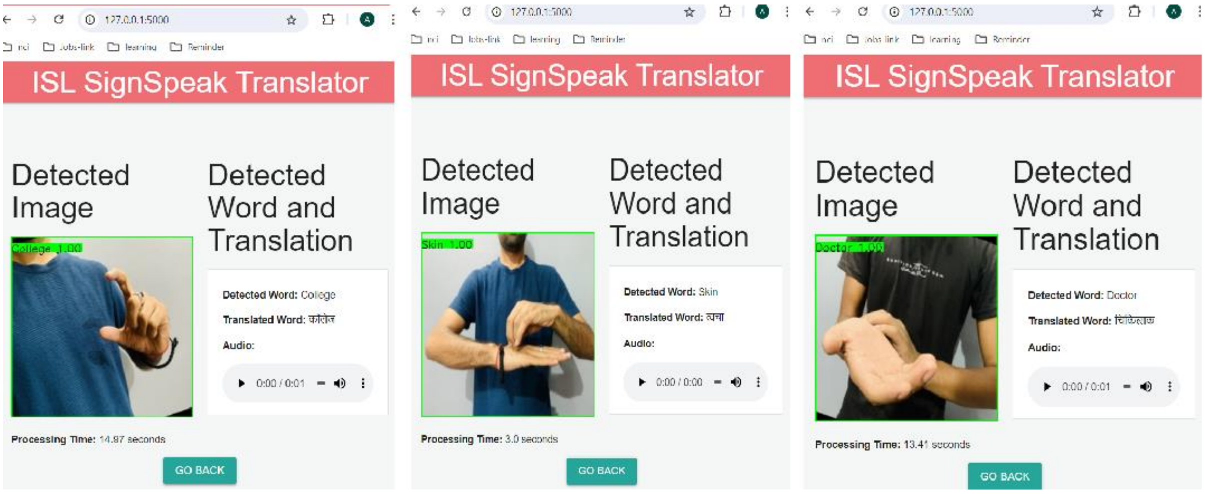


Figure 11: ISL Precision and Recall Curve

6.4 Flask web application ISL Predictions

The final evaluation of the model involved running inference using a webcam, which proved effective in detecting words with reasonable accuracy, as mentioned in Figure 13. Real-time sign detection performed adequately, though some incorrect detections were observed due to overfitting. The model was later fine-tuned, and validation images, comprising 20% of the dataset, were used to assess the model's performance, with detailed results provided in Section 6.1 and Section 6.2. Figure 13 presents example of inference results on new images captured via webcam. Although there were instances of overfitting in predictions of new images, the overall performance was satisfactory. Testing was conducted on both models, and evaluation also included prediction testing through a Flask web application, as illustrated in Figures 12 and 13. The Flask web application, as shown in Figure 13, processed sign images and returned both the translated text and audio.



ISL SignSpeak Translator

Detected Image

Detected Word and Translation

Detected Word: College

Translated Word: कॉलेज

Audio:

Processing Time: 14.87 seconds

ISL SignSpeak Translator

Detected Image

Detected Word and Translation

Detected Word: Skin

Translated Word: त्वचा

Audio:

Processing Time: 3.0 seconds

ISL SignSpeak Translator

Detected Image

Detected Word and Translation

Detected Word: Doctor

Translated Word: चिकित्सक

Audio:

Processing Time: 13.41 seconds

Figure 12: ISL Prediction and Translation with audio

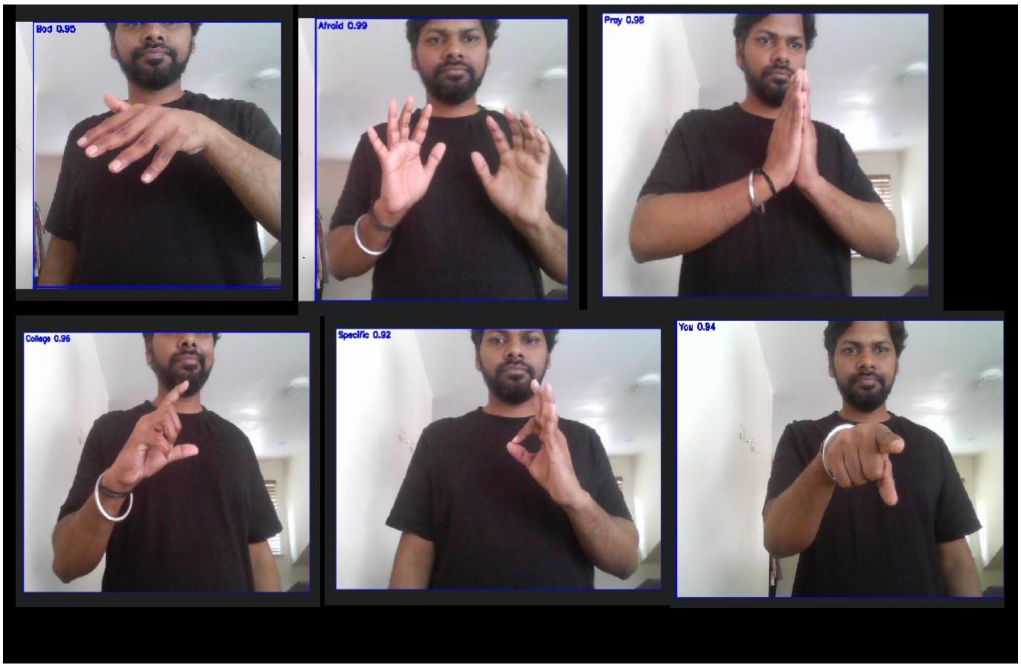


Figure 13: ISL Prediction on Webcam

6.5 Discussion

The research successfully implemented YOLOv10 for real-time detection and translation of Indian Sign Language (ISL) into Hindi text, marking a significant advancement in assistive technology for the deaf community in India. Introduced in 2024, YOLOv10 was chosen for its enhanced accuracy and faster processing times, which are crucial for real-time applications. The experiments demonstrated that YOLOv10 achieved an impressive 99% mAP@50:95 accuracy on trained ISL words, outperforming the baseline YOLOv5 model, which yielded 89% mAP@50:95 accuracy, as discussed in paper (Karthi, et al., 2021). These findings align with prior research on YOLO for sign language recognition. While earlier models like YOLOv5 have shown effectiveness, but as noted in the literature review, this study findings significantly improve the accuracy in detecting static sign for words and real-time performance.

YOLOv10's superior performance, reflected in its high Mean Average Precision (mAP), speed, and precision rates, underscores its potential as a more robust and reliable solution for ISL detection. The prediction accuracy for new images was fair with YOLOv10, while YOLOv5 struggled, achieving only 50% confidence on the same images. This discrepancy may indicate overfitting and highlights the needs for more diverse training data to improve the model's performance on unknown and varied images. Inference speed testing was conducted on 108 images for both YOLOv5 and YOLOv10, with the average time taken calculated after multiple runs. The outcome of this test was completely dependent on the current machine's capacity. The results showed an inference speed of 48.37 seconds for YOLOv5 and 17.76 seconds for YOLOv10, indicating that YOLOv10 offers a faster inference speed based on these findings.

This study contributes significantly to the field, particularly in the Indian context, where accurate translation of ISL into Hindi can bridge communication gaps for non-English-speaking populations. The focus on real-time applications, especially for users in rural and urban areas with disabilities, is particularly impactful. This aligns with prior studies emphasizing the need for accessible, efficient sign language detection systems tailored to specific regional languages ISL. The detection of words or phrases is essential for enabling reliable communication between signers and non-signers through digital applications.

Overall, the findings suggest that YOLOv10 not only improves upon existing models in terms of efficiency and latency but also holds promise for broader applications in enhancing communication for the deaf community in India. Despite demonstrating a 99% mAP@50:95 accuracy, the model still shows some false positive cases, indicating a need for further fine-tuning of data annotation. The challenges encountered, which have hindered the progress of the research, will be discussed in Section 6.6. Future work could explore further optimization of this system, including expanding the vocabulary, dynamic signs, facial gestures, and testing in varied real-world scenarios to ensure its robustness across different user contexts.

6.6 Limitations of the Work

One of the major limitations of this research is the availability of diverse real-time data. The dataset used was publicly available (Tyagi & Bansal, 2022), but it had limited words. Initially, data training was conducted using the ISLR corpus data (ISLRTC, n.d.), which consists of 100 words of signs available for public use. However, the image quality and dimesnions of this dataset was poor, and the annotation process was extremely time-consuming, ultimately yielding unsatisfactorily results. Creating a custom datatset and perform the necessary pre-processing would have taken more then a month, which was beyond the project's timeframe.As a result, the decision was made to limit the dataset to 20 words and focus model training on these words alone.

Another challenge was related to label annotation work for YOLO models, which had to be done manually using tools like LabelImg or Roboflow. This process is time-consuming, and many discrepancies in annotation can lead to further delays, requiring rework due to the project timeline. The accuracy of the sign detection heavily relies on the quality of the Annotation work. However, as demonstrated by this research, if the dataset is of high quality and annotations are accurate, the model could achieve greater accuracy and precison in ISL detection and translation to Hindi Text and speech.

Additionally, the lack of testing on real-world applications and diversified datasets further limits the validation of model's performance. Without exposure to varied and practical scenarios, the robustness and applicability of the model cannot be fully assessed, which could affect its reliability in real-world use cases.

6.7 Ethical Considerations

During this research, the priority was given to ethical considerations and cultural sensitivity with regards to how it will be represented in the project. The images used in this study is of researcher and publicly available data. The next section would provide the conclusion of work and outlines potential directions for future research.

7 Conclusion and Future Work

This study sought to determine how the integration of YOLOv10 affects the accuracy, Mean Average Precision (mAP), precision and speed of Indian Sign Language (ISL) recognition and translation into regional Hindi Text Language and speech, comparing to the existing YOLOv5 baseline model. The study achieved these objectives by demonstrating in evaluation, that YOLOv10 significantly improves upon YOLOv5 in all key metrics. The YOLOv10 starts with a higher mAP@50:95 (47%) and achieves near-perfect detection accuracy much earlier in the training process, maintaining these metrics to 99% mAP@50:95 through epoch 25. In contrast, YOLOv5 shows gradual improvement to 89% mAP@50:95 accuracy but never reaches the same level accuracy as YOLOv10. YOLOv10 has a precision (43%) and Recall (53%) compared to other model's Precision (3%) and Recall (98%) at the initial stage of training. As training progress, YOLOv10's Precision (99%) and Recall (99%) improves rapidly, while other model also improves Precision (98%) but trails slightly behind. The inference speed of the YOLOv10 model was three times faster than that of YOLO5 model.

By improving the accuracy and speed of ISL detection and translation, the developed system can facilitate better communication for deaf community, particularly in rural and tier 3 cities where English is not widely spoken. This advancement could enhance accessibility and support social integration for individuals with hearing impairments. However, the research faced limitations, including data scarcity and quality of data with diverse environments and the time-consuming nature of manual label annotation. There were instances of false positive cases while testing the application with new images, which requires more investigation. This issue highlights the need for improved quality in annotation for future work.

The future work should focus on expanding the dataset to include a more diverse range of words and phrases necessary for constructing meaningful sentences. Additionally, testing the model in real-world scenarios and training it with a broader variety of data would facilitate more effective communication and reduces barriers in interaction with non-signers. Incorporating dynamic signs, gestures, and motion into the model, potentially by combining the YOLOv10 with Long Short-Term Memory (LSTM) networks, could further enhance its capability to handle complex sign language elements. Evaluating the model under real-world condition and exploring its scalability can provide further insights. This research holds commercial potential, especially in developing assistive devices or applications tailored to the Indian market.

References

Alahmadi, T. J., Rahman, A. U., Alkahtani, H. K. & Kholidy, H., 2023. Enhancing Object Detection for VIPs Using YOLOv4_Resnet101 and Text-to-Speech Conversion Model. *Multimodal Technologies and Interaction,* Volume 7.

Alexey Bochkovskiy, C.-Y. W. H.-Y. M. L., 2020. *YOLOv4: Optimal Speed and Accuracy of Object Detection,* New York: arXiv.

Alif, M. A. R. & Hussain, M., 2024. YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain.

Ali, S. M., 2021. Comparative Analysis of YOLOv3, YOLOv4 and YOLOv5 for Sign Language Detection. *International Journal Of Advance Research And Innovative Ideas In Education,* 7(4), pp. 2395-4396.

Ao Wang, H. C. L. L. K. C. Z. L. J. H. G. D., 2024. *YOLOv10: Real-Time End-to-End Object Detection,* s.l.: arXiv.org.

Biyani, D., Doohan, N. V., Rode, M. & Jain, D., 2023. Real Time Sign Language Recognition Using Yolov5. *World Conference on Applied Intelligence and Computing (AIC),* pp. 582-588.

Bormane, P. D. & Shirbahadurkar, S. D., 2023. Indian Sign Language Recognition: A Comparative Study. *Intelligent Computing and Networking,* pp. 173-183.

Dhanjal, A. S. & Singh, W., 2022. An automatic machine translation system for multi-lingual speech to Indian sign language. *Multimedia Tools and Applications,* Volume 81, pp. 1-39.

Dhanjal, A. S. & Singh, W., 2022. An automatic machine translation system for multi-lingual speech to Indian sign language. *Multimedia Tools and Applications,* Volume 81, pp. 1-39.

Hussain, M., 2023. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines,* Volume 11, p. 677.

ISLRTC, n.d. *https://www.pinterest.com/pin/576390452317167827/.* [Online].

Karthi, M. et al., 2021. Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset. In: *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES).* s.l.:IEEE, pp. 1-6.

Khartheesvar, G., Kumar, M., Yadav, A. K. & Yadav, D., 2023. Automatic Indian sign language recognition using MediaPipe holistic and LSTM network. *Multimed Tools Appl,* Volume 83, p. 58329–58348.

Lakshmi, M. S. & .D, S., 2024. AN ANALOGY ANALYSIS OF THE OBJECT DETECTION ALGORITHMS USING YOLOV5, YOLOV7, AND YOLOV8. *Artificial Intelligence-Object Recognition.*

Li, D., Rodríguez, C., Li, H. & Yu, X., 2020. Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison. pp. 1448-1458.

Luong, S., 2023. *Video Sign Language Recognition using Pose Extraction and Deep Learning Models,* San Jose, California: SJSU ScholarWorks.

Parate, A., 2020. *Integrating Crisp DM Methodology for a Business Using Tableau Visualization,* s.l.: s.n.

Patil, R., Patil, V., Bahuguna, A. & Datkhile, G., 2021. Indian Sign Language Recognition using Convolutional Neural Network. *International Conference on Automation, Computing and Communication 2021 (ICACC-2021),* 40(ITM Web of Conferences), p. 5.

Sharma, S. et al., 2022. Real-Time Word Level Sign Language Recognition Using YOLOv4. *2022 International Conference on Futuristic Technologies (INCOFT),* pp. 1-7.

Shenoy, K., Dastane, T., Rao, V. & Vyavaharkar, D., 2021. Real-time Indian Sign Language (ISL) Recognition.

Terven, J., Córdova-Esparza, D.-M. & Romero-González, J.-A., 2023. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction,* Volume 5, pp. 1680-1716.

Tyagi, A. & Bansal, S., 2022. *Indian sign Language-Real-life Words.* [Online]
Available at: https://data.mendeley.com/datasets/s6kgb6r3ss/2

Varshney, S., 2016. Deafness in India. *Indian Journal of Otology,* pp. 73-76.

Velmathi, G. & Goyal, K., 2023. Indian Sign Language Recognition Using Mediapipe Holistic. *Computer Science.*

Vijitkunsawat, W., Racharak, T., Nguyễn, C. & Nguyen, L. M., 2023. Deep Multimodal-based Number Finger Spelling Recognizer for Thai Sign Language. In: *2023 22nd International Symposium on Communications and Information Technologies (ISCIT).* s.l.:IEEE, pp. 99-104.

Xu, R., Lin, H., Lu, K. & Cao, L., 2021. A Forest Fire Detection System Based on Ensemble Learning. *Forests,* Volume 12, p. 217.

Zahid, H. et al., 2023. A Computer Vision-Based System for Recognition and Classification of Urdu Sign Language Dataset for Differently Abled People Using Artificial Intelligence. *Mobile Information Systems,* pp. 1-17.