A

Project Report

on

Indian Sign Language Gesture Recognition Using Deep Learning Techniques

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Department of Computer Science and Engineering

By

Abhinav Singh (2100290100006)

Akshat Verma (2100290100017)

Disha Goel (2100290100057)

Aayush Kumar (2200290109001)

Under the supervision of

Mr. Umang Rastogi

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Abhinav Singh

Roll No.: 2100290100006

Date: 01-May-2025

Signature

Name: Disha Goel

Roll No.: 2100290100057

Date: 01-May-2025

Signature

Name: Akshat Verma

Roll No.: 2100290100017

Date: 01-May-2025

Signature

Name: Aayush Kumar

Roll No.: 2200290109001

Date: 01-May-2025

CERTIFICATE

This is to certify that Project Report entitled "**Indian Sign Language Gesture Recognition Using Deep Learning Techniques**" which is submitted by Abhinav Singh, Akshat Verma, Disha Goel, Aayush Kumar in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Mr. Umang Rastogi	**Dr. Vineet Sharma**
(Assistant Professor, CSE)	**(Dean CSE)**

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to **Mr. Umang Rastogi** (Assistant Professor), Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of **Dr. Vineet Sharma**, Dean of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature

Name: Abhinav Singh

Roll No.: 2100290100006

Date: 01-May-2025

Signature

Name: Disha Goel

Roll No.: 2100290100057

Date: 01-May-2025

Signature

Name: Akshat Verma

Roll No.: 2100290100017

Date: 01-May-2025

Signature

Name: Ayush Kumar

Roll No.: 2100290100006

Date: 01-May-2025

ABSTRACT

In today’s era of rapid technological advancement, the demand for accessible communication tools has become increasingly important, particularly for individuals with hearing and speech impairments. Despite being a primary mode of communication for the Indian deaf and mute community, Indian Sign Language (ISL) remains underrepresented in mainstream systems, creating a communication gap between signers and non-signers.

To address this gap, intelligent systems capable of real-time ISL interpretation into text or speech are essential. Recent progress in computer vision and deep learning, especially with object detection algorithms, offers promising solutions. This report presents the development of a real-time ISL recognition system utilizing YOLOv9, a state-of-the-art object detection model introduced in 2024. YOLOv9 builds upon its predecessors with advanced features such as transformer-based backbones, enhanced multi-scale feature aggregation, and anchor-free detection strategies, making it well-suited for real-time applications.

Despite its advanced architecture, YOLOv9 has seen limited application in sign language recognition since its release. This research explores its potential by evaluating YOLOv9’s performance on a curated dataset of ISL hand gestures, focusing on accurate detection and classification of both static and dynamic gestures in real-time. By comparing YOLOv9 with earlier YOLO versions, this study highlights its unique strengths and its potential to bridge communication gaps, fostering greater inclusivity for the hearing-impaired community.

TABLE OF CONTENT

List of Content Page No.

DECLARATION ii

CERTIFICATE iii

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF FIGURES x

LIST OF TABLES xi

CHAPTER 1: INTRODUCTION 1

1.1 Need for Sign Language Recognition 1

1.2 Previous Work in ISL Gesture Recognition 3

1.3 Overview of YOLO Framework 6

1.3.1 The YOLO Paradigm 6

1.3.2 Evolution of YOLO 7

1.4 YOLOv9 Architecture 9

1.4.1 Backbone (Feature Extraction) 11

1.4.1.1 Stem 11

1.4.1.2 RepC3ELAN Blocks 11

1.4.1.3 MPConv (Max Pooling Convolution) 12

1.4.1.4 Stages in the Backbone 12

1.4.2 Adown Block (Asymmetric Downsampling) 13

1.4.3 Neck: PANet Integration 13

1.4.3.1 Concat & Upsample 13

1.4.3.2 RepConv Layer 13

1.4.3.3 C2f Block 14

1.4.3.4 SPPFLA 14

1.4.4 Detection Head and Output 14

1.4.4.1 Detect Layers 15

1.4.4.2 Convolutional Layers 15

1.4.4.3 Anchor Boxes 15

1.4.4.4 Sigmoid Activation 15

1.4.5 Auxiliary Branch 16

1.4.5.1 Conv Layers 16

1.4.5.2 C2f Block 16

1.4.5.3 RepConv Layer 16

1.4.5.4 CBAM 17

1.5 Overall Architecture and Benefits 17

CHAPTER 2: LITERATURE REVIEW 20

2.1 Emerging Role of YOLOv9 in Gesture Recognition 20

2.2 Evolution of YOLO Architectures 20

2.3 YOLOv9 Innovations and Relevance 21

2.4 Datasets and Evaluation Metrics 21

2.5 Applications in Real-World Cases 22

2.6 Research Gaps and Future Directions 23

2.7 Conclusion: Insights for Future Work 23

CHAPTER 3: PROPOSED METHODOLOGY 25

3.1 Programmable Gradient Information (PGI) 27

3.1.1 Introduction to PGI 27

3.1.2 Mathematical Foundation 28

3.1.3 Main Branch Integration 29

3.1.4 Auxiliary Reversible Branch 29

3.1.5 Multi-Level Auxiliary Information 29

3.1.6 Comparative Analysis 29

3.1.6.1 PAN 30

3.1.6.2 RevCol 31

3.1.6.3 Deep Supervision 31

3.1.6.4 PGI (Proposed) 31

3.1.6.5 Legend and Visual Annotations 31

3.1.6.6 Summary 32
3.2 GELAN 32
3.2.1 Purpose and Design 32
3.2.2 Channel-wise Transformation 32
3.2.3 SPP 34
3.2.4 Multi-path Fusion 35
3.2.5 Comparative Study 35
3.2.5.1 CSPNet 36
3.2.5.2 ELAN 36
3.2.5.3 GELAN 37
3.2.5.4 Legend and Notations 37
3.2.5.5 Summary 37
3.3 Summary of Methodology 38
CHAPTER 4: RESULTS AND DISCUSSION 39
4.1 Experimental Setup 41
4.2 Quantitative Metrics 41
4.2.1 mAP 43
4.2.2 Precision and Recall 43
4.3 F1-Confidence Curve Analysis 43
4.4 Dataset and Anchor Analysis 47
4.5 Training and Validation Metrics 48
4.5.1 YOLOv9c Training 49
4.5.2 YOLOv9e Training 50
4.6 Loss Metrics 51
4.7 Qualitative Analysis 51
4.7.1 Detection Visualizations 51
4.7.2 Confusion Matrix 51
4.8 Performance Summary 53
4.9 Conclusion from Results 53

CHAPTER 5: CONCLUSION AND FUTURE SCOPE 54

5.1 Conclusion 55

5.2 Future Scope 55

REFERENCES 56

LIST OF FIGURES

Figure No.	**Description**	**Page No.**
1.1	YOLO Evolution Over the Years	9
1.2	YOLOv9 Architecture	19
3.1	PGI ARCHITECTURE	30
3.2	GELAN Architecture	36
4.1	P curve of YOLOv9c	44
4.2	P curve of YOLOv9e	45
4.3	R Curve of Yolov9c	45
4.4	R Curve of Yolov9e	46
4.5	F1 curve of YOLOv9c	46
4.6	F1 curve of YOLOv9e	47
4.7	Lables of YOLOv9c	48
4.8	Labels of YOLOv9e	48
4.9	Results of YOLOv9c	49
4.10	Results of YOLOv9e	50
4.11	Detection Results of YOLOv9c	52
4.12	Detection Results of YOLOv9e	52
4.13	Confusion Matrices of YOLOv9c	53
4.14	Confusion Matrices of YOLOv9e	53

LIST OF TABLES

Table. No.	Description	Page No.
4.1	Comparison of YOLOv9c and YOLOv9e Models	42
4.2	Comparative Study of both Models	43

CHAPTER 1

INTRODUCTION

1.1 Need for Sign Language Recognition

Sign language is the cornerstone of communication for individuals with hearing and speech impairments. Unlike spoken languages, which rely on auditory and vocal channels, sign languages use a rich tapestry of hand gestures, facial expressions, and body movements to convey meaning. These gestures are not arbitrary; they form a structured system with grammatical rules, idiomatic expressions, and linguistic complexity comparable to any spoken language. For millions around the world, sign language is not simply a tool—it is a primary language of expression, thought, and connection.

In India, Indian Sign Language (ISL) plays this pivotal role for the deaf and hard-of-hearing community. ISL is a complete language, featuring a grammar and syntax that are entirely independent of spoken Indian languages or other global sign languages. For example, ISL differs significantly from American Sign Language (ASL) and British Sign Language (BSL) in its signs, sentence structure, and idioms. Despite the fact that it is widely used by thousands across India, ISL remains relatively underrepresented in mainstream education and technology platforms. This has resulted in communication challenges, especially when interacting with those outside the deaf community.

According to recent data published by the World Health Organization (WHO), over 5% of the global population—approximately 430 million people—suffer from disabling hearing loss. This figure is expected to surge to over 700 million by the year 2050, equating to about 1 in every 10 individuals on the planet. Such staggering statistics underscore the urgency of creating inclusive tools and systems that can bridge the communication gap between hearing-impaired individuals and the broader population. However, one of the most significant obstacles in this regard is the scarcity of people fluent in any form of sign language, including

ISL. Without widespread understanding and adoption, deaf individuals often encounter substantial barriers in education, healthcare, employment, and daily life.

In an effort to overcome these communication barriers and foster greater inclusivity, researchers and technologists have been working on intelligent Sign Language Recognition (SLR) systems. These systems aim to interpret and translate sign language gestures into text or spoken language in real time, enabling seamless communication between deaf individuals and those unfamiliar with sign language. The rise of artificial intelligence (AI), particularly advancements in computer vision and deep learning, has been instrumental in driving this field forward. By using cameras and AI models, modern SLR systems can analyze video feeds to detect hand shapes, motion trajectories, facial expressions, and body posture, effectively converting complex visual data into coherent linguistic output.

Computer vision algorithms have matured significantly in recent years, enabling machines to understand and interpret visual input with remarkable accuracy. Deep learning architectures, especially convolutional neural networks (CNNs) and transformer-based models, have demonstrated state-of-the-art performance in image classification, object detection, and sequence modeling. These advances have naturally extended into the realm of gesture recognition, where they provide the computational foundation for identifying dynamic sign language gestures in real-time environments. With the introduction of more efficient training techniques, enhanced feature extraction methods, and GPU acceleration, real-time performance is now a reality rather than an ambition.

It is important to emphasize that sign language is not universal. In fact, linguistic studies estimate there are between 138 and 300 distinct sign languages around the world. Each of these languages has evolved within its cultural and geographic context, featuring unique sets of signs, regional variations, and idiomatic expressions. American Sign Language (ASL), for instance, is predominantly used in the United States and parts of Canada. French Sign Language (FSL), Brazilian Sign Language (Libras), and Indian Sign Language (ISL) are other prominent examples, each tailored to the cultural and communicative needs of their respective communities. This diversity means that a one-size-fits-all approach to sign language recognition is ineffective. Models must be designed with linguistic and cultural specificity in mind to ensure both technical accuracy and real-world usability.

This research centers on recognizing ISL gestures using the latest advancements in object detection through the YOLOv9 algorithm. YOLO, or “You Only Look Once,” is a deep learning-based approach that treats object detection as a single regression problem, mapping image pixels directly to bounding boxes and class probabilities in a single pass. YOLOv9 represents the latest evolution in this family of models, combining efficiency and accuracy in a lightweight architecture suitable for real-time deployment. Its ability to process complex scenes at high speed makes it particularly well-suited for the task of sign language recognition, where gestures must be identified and classified rapidly as they are performed.

The goal of this study is to design a system capable of recognizing ISL gestures with high precision and minimal latency, thereby enabling real-time communication aids for the deaf community in India. By focusing specifically on ISL, this work aims to address the unique challenges and linguistic structures present in the Indian context, which are often overlooked by generalized or globally-trained models. The successful deployment of such a system has the potential to revolutionize interactions for deaf individuals, enabling smoother integration into educational institutions, workplaces, public services, and everyday social scenarios.

As we move forward into an increasingly digitized and connected world, ensuring that no community is left behind becomes a moral and technological imperative. Real-time ISL recognition systems stand as a testament to the promise of AI when used for socially impactful innovations. By bridging the gap between the deaf and hearing communities, such tools not only facilitate communication but also empower individuals, affirming their linguistic identity and fostering a more inclusive society.

1.2 Previous Work in ISL Gesture Recognition

The application of the YOLO (You Only Look Once) object detection algorithm has garnered substantial attention in the field of sign language recognition, owing to its impressive balance between detection speed and accuracy. With each successive iteration—from YOLOv1 through to the more recent YOLOv9—the architecture has seen notable improvements in computational efficiency, model compactness, and the ability to generalize across complex visual tasks. These enhancements have enabled researchers to move beyond static image classification toward dynamic and real-time recognition systems, making YOLO a preferred choice for gesture recognition tasks such as Indian Sign Language (ISL) detection.

Early explorations in this domain employed versions like YOLOv3, which, despite being a significant leap at the time, struggled with the intricacies of ISL gestures, particularly in terms of inter-class variation and overlapping features in hand configurations. Studies using YOLOv3 on smaller ISL datasets, often limited to subsets of the alphabet, reported an average accuracy of around 82%. While promising, this performance exposed the model's difficulty in distinguishing between visually similar gestures and its sensitivity to lighting and occlusion—factors common in real-world scenarios.

The field quickly evolved with the introduction of YOLOv5, a more modular and scalable version of the architecture. A notable study that deployed YOLOv5x, the largest model variant within the YOLOv5 family, enhanced its architecture further by integrating attention mechanisms—specifically, channel and spatial attention blocks. These modifications allowed the model to focus more effectively on the hand region within noisy backgrounds and differentiate between fine-grained finger articulations. When tested on the MU HandImages ISL dataset, the system achieved an accuracy of 99.4%, suggesting that YOLO-based models could now feasibly run on edge devices like smartphones and embedded processors without sacrificing performance.

The emergence of YOLOv8 marked another leap forward. Leveraging improved backbone and neck modules, YOLOv8 offered higher precision and better generalization across diverse datasets. Researchers applying YOLOv8 to a Roboflow-preprocessed ISL dataset reported a precision of 98.46%, a recall of 98.78%, and a mean Average Precision (mAP@0.5) of 97.54%. These metrics not only surpassed earlier YOLO models but also demonstrated the utility of proper dataset augmentation and annotation pipelines. The success of YOLOv8 in this context underscored the importance of combining architectural advances with well-structured datasets to improve detection robustness across different skin tones, hand sizes, and gesture orientations.

Concurrently, alternative approaches explored the use of Neural Architecture Search (NAS) to automatically identify optimal configurations for ISL recognition. A YOLO NAS-based solution, trained on a custom ISL dataset, recorded a mean Average Precision of 95.68%, suggesting that model auto-tuning could further enhance performance while minimizing manual architecture tweaking. These models capitalized on dynamic feature routing, adaptive receptive fields, and scalable depth-width configurations, all of which contributed to their improved understanding of spatial hand dynamics in ISL.

These successive advancements collectively paint a compelling picture of YOLO's evolving capabilities in sign language recognition. As each iteration introduces more efficient layers, novel activation functions, and optimized training regimes, the practical feasibility of deploying such models in real-time applications becomes increasingly tangible. They enable a broad range of use cases, from mobile applications that assist in everyday communication to interactive systems in public services like banks, hospitals, and government offices.

Building upon this robust lineage of research, the current report presents an advanced gesture recognition system using YOLOv9, the latest and most sophisticated version in the YOLO family as of 2024. YOLOv9 incorporates enhanced backbone designs, generalized efficient layer aggregation, and refined anchor-free detection modules, all of which contribute to faster convergence and superior accuracy across multiple object categories. In this study, the YOLOv9 model is applied to a real-time ISL dataset comprising gestures for all 26 alphabets, offering an end-to-end pipeline from video input to gesture classification.

By rigorously evaluating YOLOv9’s performance on ISL recognition tasks, this work aims to push the boundaries of current capabilities in gesture recognition and provide a benchmark for future research. Real-time ISL recognition powered by YOLOv9 stands to significantly impact accessibility solutions, educational tools, and assistive technologies for the deaf and hard-of-hearing community in India, ensuring that advances in AI and computer vision are meaningfully translated into social good.

1.3 Overview of YOLO Framework

1.3.1 The YOLO Paradigm

Introduced in 2015 by Joseph Redmon and his collaborators, YOLO (You Only Look Once) marked a paradigm shift in the field of object detection. Prior to YOLO, most state-of-the-art detection systems relied on multi-stage pipelines, where region proposal networks first identified possible object locations, and then separate classifiers determined the category of the object within each region. This approach, though accurate, was computationally expensive and often unsuitable for real-time applications. YOLO addressed this limitation by reimagining object detection as a single, end-to-end regression problem.

Instead of isolating the tasks of localization and classification, YOLO processes the entire input image in a single forward pass through a convolutional neural network (CNN). It divides the image into a grid and, for each grid cell, simultaneously predicts bounding box coordinates, objectness scores, and class probabilities. This tightly coupled detection pipeline dramatically reduces inference time while maintaining competitive accuracy. The result is a system capable of detecting multiple objects in real-time, with a clear advantage in speed over traditional approaches like R-CNN and Fast R-CNN.

YOLO's architecture is highly scalable and has evolved through multiple versions, each introducing enhancements in speed, accuracy, and model size. From the initial YOLOv1 to the more recent YOLOv9, improvements have included better feature extraction backbones, more efficient bounding box prediction strategies, and the incorporation of techniques like anchor boxes, spatial pyramid pooling, and attention mechanisms. These innovations have allowed YOLO models to excel across a broad range of environments and hardware platforms, from high-performance GPUs to resource-constrained edge devices.

Due to its real-time detection capabilities and adaptable design, YOLO has become the algorithm of choice in a wide array of computer vision applications. In autonomous driving, it aids in quickly identifying pedestrians, vehicles, and traffic signs. In medical imaging, YOLO helps detect abnormalities in X-rays and MRIs with minimal latency. In the field of assistive technology, YOLO is extensively used for gesture and sign language recognition, enabling systems to translate visual gestures into actionable outputs almost instantaneously. This versatility, combined with its open-source nature, has contributed to YOLO's sustained popularity and continuous development within the research community and industry alike.

1.3.2 Evolution of YOLO

Since its groundbreaking debut, the YOLO (You Only Look Once) framework has undergone a series of transformative upgrades, each iteration building upon the strengths of its predecessor while introducing innovative components aimed at improving detection accuracy, computational efficiency, and adaptability across a range of vision tasks. These advancements have solidified YOLO's status as a cornerstone in the object detection landscape, evolving in parallel with the broader progress in deep learning and computer vision.

The journey began with **YOLOv1 to YOLOv3**, where the foundation of single-stage object detection was firmly laid. YOLOv1 introduced the concept of framing object detection as a regression problem, predicting both class probabilities and bounding box coordinates in one pass. YOLOv2 refined this by introducing batch normalization, high-resolution classifiers, and anchor boxes to better predict object locations. YOLOv3 took it further with multi-scale predictions using a deeper Darknet-53 backbone, allowing the model to handle both small and large objects more effectively while maintaining real-time inference speeds.

With **YOLOv4 and YOLOv5**, the framework saw a significant leap in performance, driven by both architectural and training optimizations. YOLOv4 incorporated advanced techniques such as Mosaic and CutMix data augmentation, Cross-Stage Partial (CSP) networks for efficient feature reuse, and SPP (Spatial Pyramid Pooling) modules to capture multi-scale features. YOLOv5, although not officially released by the original authors, was a community-driven iteration that gained immense popularity due to its ease of use, modular codebase, and competitive results. It featured scaled versions (s, m, l, x) and introduced auto-learning bounding box anchors, improved optimizer settings, and exportable ONNX/TFLite models for deployment on edge devices.

The evolution from **YOLOv6 to YOLOv8** marked a strategic shift towards modern design philosophies. These versions increasingly embraced **anchor-free detection**, which eliminated the need for pre-defined anchor boxes, reducing hyperparameter tuning and improving generalization. Attention mechanisms like CBAM (Convolutional Block Attention Module) and SE (Squeeze-and-Excitation) blocks were integrated to enable the model to focus more selectively on salient features. Transformer-based elements also began to appear, helping the models capture long-range spatial dependencies more effectively, particularly useful for recognizing small or overlapping objects. YOLOv8, in particular, emerged as a state-of-the-art model with high performance on a range of benchmarks, flexible export options, and user-friendly training pipelines.

YOLOv9, released in 2024, introduced one of the most substantial architectural overhauls in the YOLO family. It incorporated a **transformer-enhanced backbone** and a generalized efficient layer aggregation network (GELAN), leading to significant improvements in both feature extraction and contextual understanding. YOLOv9 also optimized real-time performance with an efficient anchor-free head design and adaptive spatial feature fusion, balancing inference speed with state-of-the-art accuracy. This version quickly became a preferred choice for tasks requiring high frame rates and precision, such as video surveillance, autonomous navigation, and real-time sign language recognition.

The most recent evolution, **YOLOv12**, announced in February 2025, further builds on the transformer-augmented foundation laid by YOLOv9. It incorporates **multi-query attention mechanisms**, advanced feature pyramid refinements, and dynamic inference pathways that adapt model depth and resolution based on scene complexity. YOLOv12 emphasizes not only speed and accuracy but also **scalability and energy efficiency**, aligning with the growing need for sustainable AI deployment on mobile and embedded platforms. This version supports even wider compatibility with real-world applications, from smart city infrastructure to advanced human-computer interaction systems.

Together, the development of the YOLO series reflects the trajectory of modern object detection research: toward models that are faster, more accurate, and more adaptable than ever before. As the framework continues to evolve, it remains at the forefront of real-time computer vision, offering researchers and practitioners an ever-improving toolkit for solving complex visual recognition tasks.

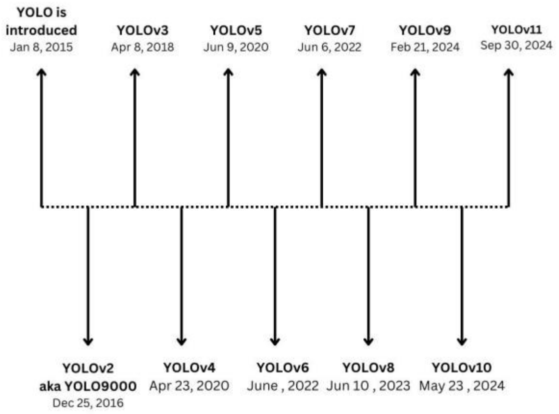


Figure 1.1 YOLO Evolution Over the Years

1.4 YOLOv9 Architecture

The YOLOv9 architecture represents a pinnacle of object detection technology, achieving an optimal balance between high accuracy and real-time performance. Its thoughtful design makes it exceptionally well-suited for a wide array of computer vision tasks, including traffic monitoring, surveillance systems, autonomous robotics, and gesture-based applications like sign language recognition. YOLOv9's robust performance is underpinned by its modular architecture, which is divided into four primary components—**Backbone**, **Auxiliary**, **Neck**, and **Head**—each playing a critical role in extracting, refining, and utilizing visual information for accurate object localization and classification.

The **Backbone** is the first stage of the network and is responsible for extracting low-level to high-level features from the input image. In YOLOv9, the backbone is enhanced with a transformer-based architecture integrated with convolutional layers, creating a hybrid system that captures both local textures and global spatial relationships. This configuration improves the model’s ability to understand complex scenes and detect objects with varying scales, shapes, and orientations. The backbone includes GELAN (Generalized Efficient Layer

Aggregation Network), which further enhances feature propagation and reusability by connecting multiple intermediate layers through hierarchical fusion. This results in more robust and discriminative feature maps that form the foundation of the detection process.

The **Auxiliary** module in YOLOv9 is designed to assist in multi-scale feature learning and reinforce gradient flow during training. It often includes secondary supervision heads or additional loss signals that provide extra guidance to earlier layers, helping the model converge more efficiently. This component is particularly beneficial when dealing with datasets that have a high degree of intra-class variability or when trying to detect fine-grained details such as fingers in sign language gestures. By improving feature supervision at multiple depths, the auxiliary module contributes to enhanced generalization without significantly increasing computational complexity.

The **Neck** acts as a bridge between the backbone and the detection head. In YOLOv9, the neck typically consists of modules like PANet (Path Aggregation Network) or BiFPN (Bidirectional Feature Pyramid Network), which aggregate and refine feature maps from various stages of the backbone. These feature pyramid structures allow the network to integrate information from both shallow and deep layers, making it more adept at detecting objects across a wide range of scales. YOLOv9's neck design ensures that semantic richness from deeper layers and spatial resolution from shallower layers are harmoniously combined, which is crucial for precise localization of small or overlapping objects.

The final component, the **Head**, is where the actual object predictions are made. YOLOv9 utilizes an anchor-free head architecture, which eliminates the need for pre-defined anchor boxes—a source of complexity and inefficiency in earlier versions. Instead, the model directly predicts object centers, dimensions, and class probabilities at each spatial location. This streamlined approach not only reduces the number of hyperparameters but also improves detection accuracy, particularly for small and densely packed objects. The head is also equipped with advanced loss functions, including distribution focal loss (DFL) and quality focal loss (QFL), which enhance the model's ability to distinguish hard-to-classify samples and refine bounding box quality.

Overall, YOLOv9's architectural improvements are tailored to meet the demands of modern real-time object detection applications. Its backbone efficiently captures hierarchical features, the auxiliary module strengthens learning dynamics, the neck enriches multi-scale representations, and the head ensures accurate, high-speed predictions. These innovations make YOLOv9 a powerful and flexible framework, capable of supporting complex visual recognition tasks such as Indian Sign Language interpretation, where precision, speed, and adaptability are all essential for effective deployment.

1.4.1 Backbone (Feature Extraction)

The backbone of YOLOv9 is responsible for extracting hierarchical features from input images, capturing both spatial and contextual information across multiple abstraction levels. It comprises three core modules: Stem, RepC3ELAN Blocks, and MPConv, which collectively generate feature maps of varying scales to support robust detection.

1.4.1.1 Stem

- **Input**: 640x640x3 (RGB image)
- **Structure**: Convolution (Conv) → Batch Normalization (BN) → SiLU Activation
- **Purpose**: The Stem block serves as the entry point for image processing, reducing spatial resolution while expanding channel depth to extract essential low-level features, such as edges and textures. The combination of convolution, batch normalization, and SiLU activation ensures stable and non-linear feature transformation.
- **Operation**: A 3x3 convolution with a stride of 2 halves the spatial dimensions, followed by batch normalization to stabilize training and SiLU activation to introduce non-linearity.
- **Output**: 320x320x32 feature map

1.4.1.2 RepC3ELAN Blocks

- **Purpose**: These blocks form the core of the backbone, leveraging residual connections to promote feature reuse and capture deeper semantic information. They enhance network depth while maintaining computational efficiency.
- **Structure**: Each block includes multiple convolutional paths with varying kernel sizes (e.g., 1x1, 3x3), combined through element-wise addition and concatenation to extract both local and global features.
- **Significance**: Residual connections facilitate gradient flow during backpropagation, mitigating vanishing gradient issues. Multi-path designs enable the learning of multi-scale features without significantly increasing computational complexity.

1.4.1.3 MPConv (Max Pooling Convolution)

- **Function**: MPConv combines max pooling and convolution to downsample feature maps while preserving critical spatial information.
- **Benefit**: Unlike traditional max pooling, which may discard valuable details, MPConv employs a learnable convolutional kernel to capture local context, reducing information loss and enhancing feature abstraction.

1.4.1.4 Stages in the Backbone

The backbone is organized into four stages, each progressively reducing spatial resolution and increasing channel depth to extract increasingly complex features:

- **Stage 1**:
  - Input: 320x320x32
  - Output: 160x160x64
  - Focus: Low-level texture and edge detection
- **Stage 2**:
  - Input: 160x160x64
  - Output: 80x80x128

    - Focus: Intermediate features like shapes and contours
- **Stage 3**:
    - Input: 80x80x128
    - Output: 40x40x256
    - Focus: Object parts and complex patterns
- **Stage 4**:
    - Input: 40x40x256
    - Output: 20x20x512
    - Focus: High-level semantic information and object structures

1.4.2 Adown Block (Asymmetric Downsampling)

The Adown block is a novel feature of YOLOv9, utilizing asymmetric convolution operations (1×3 and 3×1) instead of traditional square kernels for downsampling. This approach reduces computational overhead while preserving spatial details, enabling faster and more efficient real-time processing, particularly for applications like ISL gesture recognition.

1.4.3 Neck: PANet Integration

The neck of YOLOv9 employs a Path Aggregation Network (PANet) to enhance multi-scale feature aggregation. PANet integrates low-level and high-level features from the backbone, improving the detection of objects across various sizes. This fusion is particularly effective in complex environments, such as hand gesture datasets, where precise localization is critical.

1.4.3.1 Concat & Upsample

- **Purpose**: Combines features from different backbone stages through concatenation and upsampling to maintain spatial information while integrating contextual data.
- **Structure**: Lower-resolution feature maps are upsampled to match higher-resolution maps, then concatenated to fuse coarse and fine-grained information.
- **Significance**: Enables robust detection of objects of varying sizes by leveraging multi-scale features.

1.4.3.2 RepConv Layer

- **Purpose**: Optimizes convolutional operations by consolidating multiple paths during training into a single, efficient path during inference.
- **Structure**: Uses multiple convolutional branches (e.g., 1x1, 3x3) during training, merged into a single kernel during inference.
- **Benefits**: Reduces inference latency and memory usage while maintaining feature quality.

1.4.3.3 C2f Block

- **Purpose**: Refines multi-scale features using multiple convolutions and residual connections to emphasize critical patterns.
- **Structure**: Features two branches—a direct path and a convolutional path—whose outputs are concatenated to combine fine-grained and abstract information.
- **Significance**: Enhances feature representation in complex scenes by preserving both local details and broader context.

1.4.3.4 SPPFLA (Spatial Pyramid Pooling with Focused Local Attention)

- **Purpose**: Extracts multi-scale features and applies focused local attention to prioritize critical regions.
- **Structure**: Includes multiple pooling layers (e.g., 1x1, 3x3, 5x5) and a local attention mechanism that computes weights for spatial regions to emphasize informative features.
- **Benefits**: Improves localization and recognition accuracy, particularly for small or occluded objects.

1.4.4 Detection Head and Output

The detection head processes refined multi-scale feature maps to generate bounding boxes, objectness scores, and class probabilities. It employs a custom loss function during training, integrating classification, bounding box regression, and objectness losses for holistic optimization. During inference, Non-Maximum Suppression (NMS) eliminates redundant detections, ensuring accurate and reliable outputs suitable for real-time ISL gesture recognition.

1.4.4.1 Detect Layers

- **Purpose**: Detect objects (gestures) across small, medium, and large scales using optimized feature maps.
- **Structure**: Three detect layers process feature maps at different resolutions (e.g., 80x80, 40x40, 20x20) to handle varying object sizes.
- **Operation**: Each layer uses dedicated convolutional operations for independent processing, ensuring precise localization and classification.

1.4.4.2 Convolutional Layers

- **Purpose**: Extract localized patterns and refine features for detection tasks.
- **Structure**: Series of convolutions with specific kernel sizes and strides to enhance spatial and contextual information.
- **Significance**: Reduces feature map dimensionality, facilitating efficient bounding box regression and classification.

1.4.4.3 Anchor Boxes

- **Purpose**: Serve as reference points for detecting objects by adjusting predefined bounding boxes based on object size and location.
- **Structure**: Multiple anchor boxes with specific aspect ratios are initialized at each grid cell.
- **Operation**: Predicts bounding box coordinates, objectness scores, and class probabilities for each anchor box.
- **Significance**: Enables detection of multiple objects within the same region, accommodating diverse gesture sizes.

1.4.4.4 Sigmoid Activation

- **Purpose**: Normalizes predictions to a 0–1 range for interpretability.
- **Structure**: Applied to objectness scores, bounding box coordinates, and class probabilities.
- **Benefits**: Ensures bounded predictions, simplifying the interpretation of confidence scores.

1.4.5 Auxiliary Branch (Enhanced Feature Learning and Stability)

The auxiliary branch stabilizes training and refines feature learning by providing intermediate supervision, reducing overfitting, and enhancing feature robustness. It integrates Conv Layers, C2f Block, RepConv Layer, and CBAM (Convolutional Block Attention Module).

1.4.5.1 Conv Layers

- **Purpose**: Refine intermediate feature maps to extract specific patterns for improved localization and classification.
- **Structure**: Sequential convolutional layers with varying kernel sizes, followed by Batch Normalization and SiLU Activation.
- **Benefits**: Provides early supervision, reducing overfitting and enhancing feature granularity.

1.4.5.2 C2f Block

- **Purpose**: Enhances intermediate features by combining direct and convolutional paths.
- **Structure**: Direct path preserves features, while the convolutional path refines them, with outputs concatenated.
- **Significance**: Leverages residual connections to maintain information flow and prevent gradient issues.

1.4.5.3 RepConv Layer

- **Purpose**: Reduces model complexity by merging multiple convolutional paths into a single efficient path during inference.
- **Structure**: Multiple convolutional branches during training, consolidated during inference.
- **Benefits**: Lowers latency and memory usage while maintaining accuracy.

1.4.5.4 CBAM (Convolutional Block Attention Module)

- **Purpose**: Emphasizes significant spatial and channel features to enhance detection accuracy.
- **Structure**: Includes channel attention (via global pooling and MLP) and spatial attention (via convolutional operations).
- **Benefits**: Reduces background noise, focusing on key features for improved performance.

1.5 Overall Architecture and Benefits

The YOLOv9 architecture integrates its core components—**backbone**, **neck**, **head**, and an **auxiliary branch**—to form a robust and highly efficient object detection system optimized for real-time applications. These interconnected modules are meticulously designed to enhance both feature representation and detection precision, enabling the model to perform exceptionally well across diverse visual tasks. In the context of Indian Sign Language (ISL) gesture recognition, where real-time performance and high accuracy are imperative, YOLOv9 proves to be an ideal solution.

At the heart of this architecture, the **backbone** is responsible for extracting deep feature representations from the input image. YOLOv9 introduces enhancements such as **Squeeze-and-Excitation (SE) blocks**, which recalibrate channel-wise feature responses, allowing the network to focus more selectively on important hand gesture features while suppressing irrelevant background noise. This attention mechanism plays a crucial role in differentiating subtle variations between similar ISL gestures, improving classification reliability.

Complementing the backbone is the **SPPFLA (Spatial Pyramid Pooling - Fast Large Kernel Attention)** module, an evolution of traditional SPP that enhances receptive field diversity without significantly increasing computational overhead. It captures multi-scale spatial context, helping the model understand gestures regardless of hand size, position, or orientation. In real-world ISL applications, where hand movements may vary widely between individuals, SPPFLA ensures that the detection pipeline remains robust and consistent.

The **neck** of YOLOv9 employs sophisticated feature fusion strategies to combine high-resolution features from early layers with semantically rich features from deeper layers. Modules like **CBAM (Convolutional Block Attention Module)** are integrated within this structure to refine spatial and channel-wise attention further. CBAM enables the model to focus more effectively on gesture-relevant regions—like fingertips and palm contours—by dynamically adjusting attention across both dimensions. This selective focus is essential in ISL, where finger positioning carries semantic weight.

The **head** of the network is designed around an **anchor-free detection mechanism**, which simplifies the training process and improves generalization to varied hand shapes and orientations. It outputs class probabilities and bounding box coordinates directly, leading to faster convergence and reduced false positives. The head also benefits from advanced loss functions that sharpen both classification and localization performance, ensuring precise gesture recognition even under motion blur or occlusions.

YOLOv9's **auxiliary branch** further strengthens learning by offering additional supervision during training. It assists in refining intermediate feature maps and stabilizing gradient flow, especially beneficial when training on gesture datasets with high intra-class variance. This support mechanism ensures that the network maintains high detection fidelity even under challenging conditions, such as varying lighting or background clutter.

Together, these architectural elements work in synergy to strike an exceptional balance between speed and accuracy. YOLOv9 not only excels in recognizing **static ISL gestures**, such as alphabet signs, but also adapts effectively to **dynamic gestures** that involve temporal movement. Its real-time inference capability ensures that gesture translation systems can deliver near-instantaneous feedback, making communication more natural and fluid for deaf and hard-of-hearing individuals.

By combining cutting-edge attention modules, scalable feature extractors, and efficient detection heads, YOLOv9 redefines the standard for real-time gesture recognition. It empowers inclusive technology solutions that bridge the communication gap for millions, affirming the transformative potential of modern AI in addressing accessibility challenges.

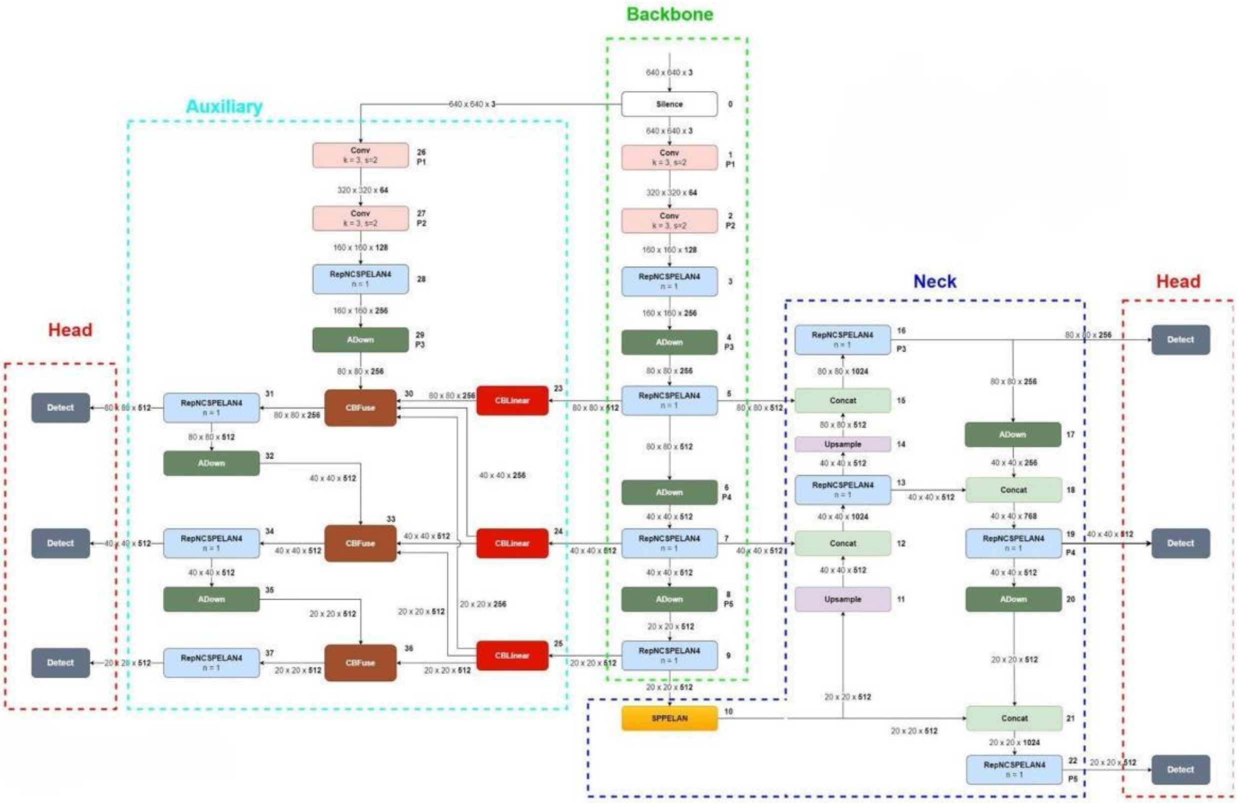


Figure 1.2 YOLOv9 Architecture

CHAPTER 2

LITERATURE REVIEW

2.1 Emerging Role of YOLOv9 in Gesture Recognition

Hand gesture recognition has emerged as a significant research area within computer vision, with applications spanning assistive technologies, human-computer interaction (HCI), and sign language translation. The YOLO (You Only Look Once) family of object detection models, renowned for their speed and accuracy, has played a pivotal role in this domain. With the advent of YOLOv9 in 2024, researchers have begun to explore its applicability in gesture recognition due to its enhanced architectural features and improved performance over earlier iterations.

YOLOv9's real-time detection capabilities make it especially relevant for sign language interpretation, where instant gesture recognition is essential. Early findings indicate that YOLOv9 not only maintains the efficiency of its predecessors but also offers superior accuracy, positioning it as a highly viable solution for real-time gesture-based applications [6]. Its application in sign language processing demonstrates the model's ability to capture subtle hand movements with increased precision, a task critical to bridging communication barriers between deaf and hearing communities.

2.2 Evolution of YOLO Architectures and Their Application in Gesture Recognition

The development of YOLO architectures has significantly advanced object detection capabilities, making them increasingly suitable for gesture recognition tasks. Earlier YOLO models such as YOLOv3 through YOLOv6 contributed to real-time detection through lighter backbones and optimized anchor-based detection strategies. YOLOv5 introduced attention mechanisms and compound scaling, while YOLOv6 and YOLOv7 emphasized improved training efficiency and detection accuracy, often employing more robust loss functions and architectural tuning [7].

YOLOv8 continued this trajectory with enhancements in generalization and model compression, further improving edge deployment feasibility. However, YOLOv9 represents a transformative leap in architecture by integrating Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), both of which contribute to increased depth, gradient stability, and multi-scale feature extraction. These upgrades have proven particularly beneficial in gesture recognition scenarios, where hand configurations often contain nuanced spatial and temporal cues [8].

2.3 YOLOv9 Architectural Innovations and Their Relevance to Gesture Recognition

The innovations introduced in YOLOv9—namely PGI and GELAN—address longstanding challenges in deep learning-based recognition systems. PGI is specifically designed to combat the issue of vanishing gradients, which can result in the loss of crucial information during training. By carefully regulating the flow of gradient information, PGI facilitates improved learning of fine-grained features that are essential for distinguishing between visually similar hand gestures [8].

GELAN enhances YOLOv9's efficiency by streamlining feature extraction and aggregation across the network. Through the strategic use of convolutional and bottleneck blocks, GELAN captures both high-resolution details and broader contextual information, enabling the model to handle a diverse array of gestures. These capabilities are particularly crucial in Indian Sign Language (ISL), where minimal variations in finger positioning or orientation can alter semantic meaning.

In addition, the introduction of the RepNCSPELAN block, Silence Blocks, and Auxiliary Layers contributes to stable training dynamics and better multi-scale detection performance. These components allow YOLOv9 to retain a high inference speed without compromising accuracy—key metrics for any system intended for real-time gesture translation.

2.4 Datasets and Evaluation Metrics in YOLOv9-Based Gesture Recognition System

The performance of gesture recognition models is heavily influenced by the characteristics of the datasets used. Studies utilizing YfOLOv9 have leveraged both small-scale and large-scale datasets, encompassing isolated gestures, alphabets, and continuous sequences from different sign languages [8]. Ideal datasets include a variety of lighting conditions, skin tones, hand sizes, backgrounds, and camera angles, thereby enhancing the generalization capability of the model.

Equally important is the annotation quality. High-quality, consistently labeled datasets ensure that the YOLOv9 model can accurately associate gesture features with corresponding semantic classes. Manual annotation, though labor-intensive, remains the gold standard for producing high-fidelity datasets.

Performance evaluation is typically conducted using established metrics such as mean Average Precision (mAP), precision, recall, and F1-score. These metrics offer a comprehensive assessment of detection and classification accuracy. For example, mAP@0.5 evaluates the intersection-over-union (IoU) threshold, balancing spatial and semantic correctness. F1-score, on the other hand, harmonizes precision and recall to provide a single accuracy indicator for model effectiveness [8].

2.5 Applications of YOLOv9 in Gesture Recognition: Real-World Case Studies

The applications of YOLOv9 in gesture recognition extend beyond academia into various real-world domains. One of the most prominent is sign language recognition systems, where YOLOv9's real-time capabilities enable dynamic interpretation of hand gestures, aiding communication for hearing-impaired individuals [8]. These systems often integrate gesture recognition into software or wearable devices that translate signs into text or speech, enhancing accessibility in everyday settings.

Additionally, YOLOv9 is employed in gesture-based HCI systems. Its fast detection speed makes it ideal for gesture-controlled interfaces in AR/VR, smart environments, and robotics. In such applications, users can interact with digital interfaces using intuitive gestures, eliminating the need for physical input devices [7].

Moreover, YOLOv9-based recognition systems are used in safety monitoring and surveillance, where hand gestures might signal distress or convey non-verbal instiructions. These applications benefit from YOLOv9's ability to operate effectively on embedded systems and mobile platforms, often with the help of optimization frameworks like TensorRT and ONNX [10].

2.6 Research Gaps and Future Directions

Despite the considerable progress, several research gaps remain. Current literature lacks comprehensive benchmarking studies comparing YOLOv9 with other contemporary detection models, such as Transformer-based architectures or lightweight MobileNet variants, specifically in the domain of gesture recognition.

Furthermore, the integration of multimodal data sources represents a promising yet underexplored direction. Combining YOLOv9 with depth sensors, RGB-D data, or inertial measurement units (IMUs) could significantly enhance gesture detection accuracy by incorporating motion cues and 3D spatial data [13].

Another key area is the development of more sophisticated training paradigms. Techniques such as semi-supervised learning, domain adaptation, and data augmentation using synthetic gesture data can help overcome limitations related to dataset scarcity and generalization across users.

2.7 Conclusion: Insights and Implications for Future Work

In summary, YOLOv9 stands out as a significant advancement in real-time object detection, with its architectural innovations rendering it highly effective for hand gesture recognition. Its design addresses core challenges such as vanishing gradients and inefficient feature extraction, making it suitable for a wide range of gesture-based applications. The combination of PGI and GELAN provides a solid foundation for accurate, efficient, and scalable gesture recognition systems.

The societal impact of improved gesture recognition cannot be overstated. As real-time ISL recognition becomes more accessible and accurate, individuals with hearing impairments can engage more fully in educational, professional, and social contexts [11]. Moreover, the use of YOLOv9 in human-computer interaction and assistive technologies opens new avenues for intuitive user interfaces and inclusive technology solutions [12]. Future research should continue to expand upon YOLOv9's capabilities through multimodal integration, dataset expansion, and optimization for resource-constrained devices.

CHAPTER 3

PROPOSED METHODOLOGY

The proposed methodology for Indian Sign Language (ISL) recognition leverages the advanced capabilities of the YOLOv9 architecture, integrating two groundbreaking innovations—**Programmable Gradient Information (PGI)** and the **Generalized Efficient Layer Aggregation Network (GELAN)**—to tackle some of the most persistent challenges in real-time gesture recognition systems. These challenges include dynamic hand movement tracking, variations in gesture scale and positioning, vanishing gradients in deep networks, and the need for high computational efficiency without compromising accuracy. Through this synergistic integration, the approach delivers a powerful, real-world deployable solution for ISL recognition that emphasizes stability, scalability, and precision.

At the core of this methodology is **Programmable Gradient Information (PGI)**, a novel training mechanism designed to address the **vanishing gradient problem** that often plagues deep convolutional neural networks. PGI works by dynamically adjusting the backpropagation gradients during training to ensure that key feature representations in the earlier layers remain active and relevant throughout the learning process. By doing so, it enhances the **gradient flow** across multiple layers, ensuring better convergence and preserving semantic consistency between low-level and high-level features. This is especially important in ISL recognition, where fine details such as finger curvature and subtle hand orientations significantly influence gesture interpretation.

Complementing PGI is the **Generalized Efficient Layer Aggregation Network (GELAN)**, which optimizes how features are extracted and aggregated across the network. GELAN employs a **multi-path hierarchical fusion** strategy, where features from various depths of the network are adaptively merged using skip connections and efficient aggregation blocks. This ensures that both shallow and deep features—critical for recognizing small hand parts as well as contextual arm movements—are utilized effectively. Unlike traditional sequential architectures that may lose valuable information due to progressive abstraction, GELAN retains rich spatial and semantic cues, which are vital for accurate and context-aware gesture classification.

The synergy between PGI and GELAN creates a **robust feature learning pipeline**, capable of handling real-world challenges such as **scale variance**, where hand gestures appear in different sizes based on the distance from the camera, and **motion blur**, which occurs during rapid finger or hand movements. The architecture's attention to feature diversity and cross-scale consistency enables it to differentiate between visually similar signs—such as those for "M" and "N" in ISL—while maintaining real-time processing speeds necessary for interactive communication tools.

Furthermore, this methodology is optimized for **low-latency inference**, ensuring it can run on edge devices such as mobile phones or embedded systems with minimal computational load. This is critical for deployment in rural or low-resource settings, where high-end hardware may not be available. The use of **anchor-free detection**, lightweight feature maps, and simplified prediction heads contributes to a reduction in processing overhead, making the system suitable for **on-device real-time ISL recognition**.

In practical terms, the model is trained on a custom dataset of ISL gestures representing all 26 letters of the alphabet, captured under diverse conditions to simulate real-world variability. The data is preprocessed using normalization, augmentation (e.g., rotation, scaling, flipping), and noise injection to enhance generalization. During training, the model optimizes a composite loss function comprising **Distribution Focal Loss (DFL)** for bounding box precision, **Quality Focal Loss (QFL)** for classification confidence, and a **balance-weighted auxiliary loss** from PGI to reinforce intermediate learning stages.

Together, these innovations not only enhance the learning and inference processes but also contribute to the model's **scalability and adaptability** across different sign languages or gesture datasets. The resulting ISL recognition system is not only accurate and fast but also resilient to common real-world issues like occlusion, inconsistent lighting, and background clutter.

By fusing YOLOv9's cutting-edge architecture with PGI and GELAN, this methodology lays the groundwork for an inclusive, intelligent communication interface—paving the way for real-time translation tools that empower the deaf and hard-of-hearing community to interact seamlessly with the world around them.

3.1 Programmable Gradient Information (PGI)

3.1.1 Introduction to PGI

Programmable Gradient Information (PGI) is a novel mechanism introduced to address one of the persistent challenges in deep neural networks—**unstable or vanishing gradient propagation**, particularly prevalent in tasks requiring **highly discriminative and deep feature learning** such as Indian Sign Language (ISL) recognition. ISL gestures often involve nuanced variations in hand shapes, orientations, and finger placements that demand **fine-grained, multi-level feature extraction**. Traditional training processes can suffer from diminished gradient flow as depth increases, resulting in ineffective learning in the lower layers, where foundational spatial features are extracted.

PGI tackles this problem by introducing **programmable gradient paths**—adaptive routes through which gradients are strategically modulated and reinforced during backpropagation. Instead of relying on a uniform gradient flow across the entire network, PGI intelligently adjusts gradient propagation in deeper layers, ensuring **important gesture-specific features** are preserved and strengthened throughout the training process. This dynamic modulation helps prevent early-layer features from being overwhelmed or discarded as the network deepens, allowing the model to maintain high sensitivity to **fine details**, such as slight curvature differences in fingers or small positional shifts in hand gestures.

Furthermore, PGI enhances the **training stability and convergence rate** by mitigating oscillations or stagnation in loss minimization. This is particularly useful in ISL recognition, where datasets often contain high inter-class similarity (e.g., signs for “M” vs. “N”) and intra-class variability (different individuals performing the same gesture differently). PGI supports

context-aware feature learning, helping the network distinguish between gestures that may be visually close but semantically distinct.

Another key advantage of PGI is its **compatibility with deep convolutional and transformer-based architectures**, like those found in YOLOv9. When integrated into such architectures, PGI acts as a **gradient guidance system**, directing learning energy toward the most semantically rich features. This enhances the **overall representation quality** across the network and contributes to the model's ability to generalize well in real-world environments, including varying lighting conditions, complex backgrounds, and diverse hand types.

In essence, PGI empowers the network to "learn deeper" without falling into the pitfalls of degraded learning signals. By ensuring **stable and meaningful gradient flow**, it boosts the model's capacity to capture the **subtle temporal and spatial intricacies** of ISL gestures—thereby facilitating highly accurate and real-time sign language recognition that is robust, interpretable, and scalable.

3.1.2 Mathematical Foundation

In a convolutional neural network (CNN), an input (X) is processed to produce an output (Y) through a function parameterized by learnable weights:

Equation 1: $\mathbf{f(X; \theta) = Y}$

During backpropagation, the gradient of the loss function (L) with respect to (θ) is computed as:

Equation 2: $\mathbf{\partial L/\partial \theta = (\partial L/\partial Y) \cdot (\partial Y/\partial X) \cdot (\partial X/\partial \theta)}$

In deep networks, the term often diminishes, leading to the vanishing gradient problem, which hampers learning. PGI addresses this by introducing architectural modifications that preserve gradient magnitude, ensuring robust learning even in deep layers critical for ISL gesture recognition.

3.1.3 Main Branch Integration

PGI is integrated into the main inference path alongside GELAN to optimize feature extraction. For an input ISL gesture image (X), features are extracted layer-wise using:

Equation 3: $F_l = \sigma(W_l * X + b_l)$

Where:

- **W_l** = weights at layer *l*
- **b_l** = biases at layer *l*
- ***** = convolution operator
- **σ** = non-linear activation function (e.g., ReLU or SiLU)

This process captures both spatial and contextual information, enabling accurate detection of ISL gestures with varying complexity and semantic significance.

3.1.4 Auxiliary Reversible Branch

To enhance gradient stability, PGI incorporates a reversible residual branch defined as:

Equation 4: $X_{l+1} = X_l + F(X_l, \theta)$

This auxiliary path preserves the identity of earlier layer outputs, facilitating effective backpropagation. The gradient through this pathway is computed as:

Equation 5: $\partial L/\partial X_l = \partial L/\partial X_{l+1} + \partial L/\partial F(X_l, \theta$

This design ensures that critical gradient signals are retained across deep networks, promoting training stability and convergence, which is essential for ISL recognition tasks.

3.1.5 Multi-Level Auxiliary Information

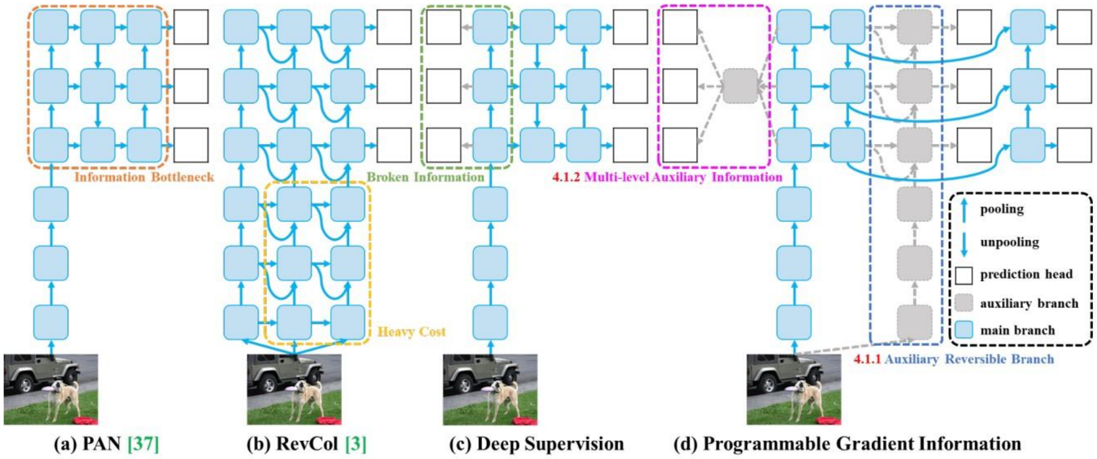
To address scale variance in ISL gestures, which may vary in size due to camera distance or hand positioning, PGI employs a feature pyramid strategy. Multi-level feature aggregation is achieved through weighted combinations of feature maps:

Equation 6: $\mathbf{F_s = \sum_{i=1}^{n} \alpha_i \cdot F_i}$

Where:

- $\mathbf{F_i}$ = feature maps at various depths of the network
- $\boldsymbol{\alpha_i}$ = learned coefficients controlling the importance of each scale
- $\sum$ = summation across *N* levels

This approach enhances the model’s ability to recognize gestures across different scales and orientations, improving robustness in real-world scenarios.



(a) PAN [37] (b) RevCol [3] (c) Deep Supervision (d) Programmable Gradient Information

Figure 3.1 PGI ARCHITECTURE

3.1.6 Comparative Analysis of Feature Aggregation and Gradient Flow Mechanisms

This subsection compares feature aggregation and gradient flow mechanisms across four architectures: Path Aggregation Network (PAN), Reversible Column Network (RevCol), Deep Supervision, and the proposed PGI.

3.1.6.1 PAN (Path Aggregation Network)

PAN enhances low-level features with semantic information from deeper layers via a bottom-up path. However, it suffers from an information bottleneck, limiting gradient flow to early layers, as shown in Figure 3.1

- **Key Limitation**: Restricted gradient flow due to bottleneck.
- **Impact**: Reduced supervision for shallow layers, impacting fine-grained task performance.

3.1.6.2 RevCol (Reversible Column Network)

RevCol uses reversible layers to reduce memory usage but fragments gradient flow, increasing computational costs.

- **Key Limitation**: Incomplete gradient flow and computational overhead.
- **Impact**: Suboptimal training dynamics due to reversibility-supervision trade-offs.

3.1.6.3 Deep Supervision

Deep supervision adds auxiliary loss heads at intermediate stages to enhance gradient signals. However, independent auxiliary branches disrupt information consistency with the main task

- **Key Limitation**: Isolated auxiliary supervision.
- **Impact**: Fragmented optimization due to lack of synergy with the main task.

3.1.6.4 Programmable Gradient Information (Proposed)

The proposed PGI architecture uses auxiliary reversible branches for consistent, bidirectional gradient flow. Key features include:

- **Auxiliary Reversible Branches**: Facilitate efficient forward and backward propagation across multiple layers.
- **Multi-Level Supervision**: Concurrent supervision without detaching from the main task.
- **Modular Design**: Combines pooling/unpooling and shared prediction heads for balanced spatial and semantic processing.
- **Key Advantage**: Integrates supervision across all levels without breaking gradient flow.
- **Impact**: Ensures consistent learning with minimal overhead, ideal for gesture recognition.

3.1.6.5 Legend and Visual Annotations

- **Blue Nodes**: Main processing units.
- **Gray Blocks**: Auxiliary branches and prediction heads.
- **Dashed Arrows**: Backward (gradient) flow.
- **Solid Arrows**: Forward propagation.
- **Symbols**: Distinct pooling/unpooling operations.

3.1.6.6 Summary

PGI overcomes the limitations of PAN, RevCol, and Deep Supervision by leveraging reversible branches and multi-level supervision. It achieves an optimal balance of gradient preservation, computational efficiency, and feature consistency, making it highly suitable for ISL gesture recognition.

3.2 Generalized Efficient Layer Aggregation Network (GELAN)

3.2.1 Purpose and Design

The **Generalized Efficient Layer Aggregation Network (GELAN)** is a key architectural advancement designed to optimize both the **efficiency** and **accuracy** of deep learning models, making it especially valuable for real-time tasks like Indian Sign Language (ISL) recognition. By integrating and extending concepts from **Spatial Pyramid Pooling (SPP)** and **Cross Stage Partial Networks (CSPNet)**, GELAN introduces a more structured and scalable approach to feature aggregation that significantly improves the model's performance on complex visual tasks involving subtle variations—such as the nuanced gestures in ISL.

At its core, GELAN focuses on three main goals: **maximizing feature reuse**, **minimizing redundant computation**, and **enabling effective multi-path learning**. These goals are achieved through a series of hierarchical aggregation blocks that efficiently combine low-level spatial details with high-level semantic information. Unlike traditional architectures that follow a linear progression of feature extraction, GELAN creates **interconnected pathways** across various network depths, ensuring that rich information is not lost as features move deeper into the network.

One of the fundamental building blocks of GELAN is its extension of **Spatial Pyramid Pooling (SPP)**, which is used to capture features at multiple receptive fields. In the context of ISL, where gestures can vary in scale and position depending on the signer's distance from the camera, SPP-like structures help encode these variations by aggregating contextual information from different spatial regions. GELAN improves on this by using **Fast Large Kernel Attention (SPPFLA)**, which mimics large kernel convolutions while maintaining computational efficiency—crucial for real-time processing.

Another cornerstone of GELAN is its use of **Cross Stage Partial Networks (CSPNet)** principles, which involve splitting feature maps and only partially forwarding them through complex transformation layers before merging. This strategy not only **reduces memory and computation overhead**, but also **retains gradient flow** to earlier layers, addressing vanishing gradient issues common in deep models. By minimizing redundant transformations and encouraging modularity, CSPNet-based design elements in GELAN ensure that every layer contributes meaningful information without inflating the model's size or latency.

What distinguishes GELAN further is its **multi-path learning** capability, which allows features from different depths and scales to flow through multiple routes before being aggregated. This facilitates **better generalization and robustness**, as the network can capture a broader spectrum of features—from detailed contours of fingers to the overall shape and orientation of the hand. Such multi-level representation is especially valuable in distinguishing visually similar ISL gestures, which may differ by a slight finger placement or hand twist.

In practical terms, GELAN supports real-time performance by avoiding computational bottlenecks often introduced by deep or dense layers. Its design emphasizes **parallel processing** and **low-latency fusion**, making it highly suitable for deployment on edge devices or mobile platforms, where computational resources are limited. The efficiency gains do not come at the cost of accuracy; in fact, GELAN's structured aggregation enables the model to maintain or even improve its detection precision while significantly boosting inference speed.

Overall, the incorporation of GELAN into the YOLOv9-based ISL recognition system provides a **powerful, modular, and scalable feature extraction backbone**. By combining the strengths of SPP and CSPNet and extending them through generalized and efficient aggregation strategies, GELAN ensures that the model is both **deeply perceptive and computationally light**, enabling real-time gesture classification that meets the demanding requirements of assistive communication technologies for the deaf and hard-of-hearing community.

3.2.2 Channel-wise Transformation

GELAN begins by transforming input feature maps channel-wise to prepare them for subsequent processing:

Equation 7: **X_c = W_c * X + b_c**

Where:

- **W_c** = channel-wise weights
- **b_c** = channel-wise biases
- ***** = convolution operator
- **X** = input feature map
- **X_c** = transformed feature map

This transformation re-encodes spatial features, enhancing the model's ability to detect fine-grained patterns in ISL gestures.

3.2.3 Spatial Pyramid Pooling (SPP)

SPP extracts features at multiple spatial resolutions to ensure robustness to varying gesture sizes:

Equation 8: $\mathbf{X_s = \sum_{i=1}^{n} P_i(X\_c)}$

Where:

- $\mathbf{P_i(\cdot)}$ = pooling functions applied at different scales
- **X_c** = input from the previous convolutional layer
- $\mathbf{X_s}$ = aggregated feature map across scales

This multi-scale approach enables accurate capture and interpretation of both small and large gestures.

3.2.4 Multi-path Fusion via CSP-ELAN

GELAN fuses feature maps from multiple processing paths to produce a comprehensive output:

Equation 9: $\mathbf{X_o = concat(X_s^{(1)}, X_s^{(2)}, ..., X_s^{(N)})}$

Where:

- **concat(·)** = concatenation operation
- $\mathbf{X_s^{(i)}}$ = output from the *i-th* spatial scale path
- $\mathbf{X_o}$ = final fused output

This concatenation promotes diversified feature learning and generalization while maintaining efficient inference by controlling parameter growth.

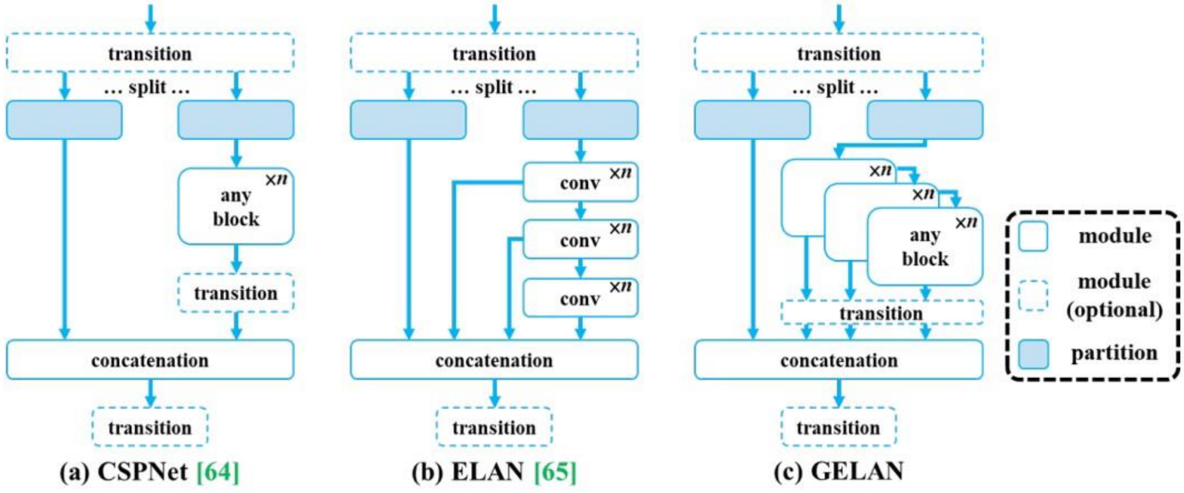


Figure 3.2 GELAN Architecture

3.2.5 Comparative Study of Partition-Based Feature Fusion Architectures

This subsection compares partition-based neural network designs for feature fusion and gradient propagation: CSPNet, ELAN, and GELAN.

3.2.5.1 CSPNet (Cross Stage Partial Network)

CSPNet splits feature maps into two parts: one undergoes transformations, while the other bypasses processing for later merging, reducing complexity and enhancing gradient flow (Figure 3.2).

- **Design Highlights**:
    - Splits features into partitions.
    - Processes one partition through sequential blocks.
    - Concatenates partitions to preserve gradient flow.
- **Strengths**:
    - Computationally efficient.
    - Maintains accuracy with reduced model size.
- **Limitation**:
    - Single-path processing limits expressiveness.

3.2.5.2 ELAN (Efficient Layer Aggregation Network)

ELAN extends CSPNet with multiple parallel convolutional layers in the transformed branch, enhancing feature learning (Figure 3.2).

- **Design Highlights**:
    - Splits and processes features through multiple convolution paths.
    - Aggregates outputs via concatenation.
- **Strengths**:
    - Supports richer feature transformation.
    - Improves gradient distribution through diverse branches.
- **Limitation**:
    - Increased computational cost due to multiple paths.

3.2.5.3 GELAN (Generalized ELAN)

GELAN advances ELAN with flexible transition layers and nested feature paths for improved reuse and aggregation (Figure 3.2).

- **Design Highlights**:
    - Modular flexibility with optional transitions.

    - Enhanced connectivity for multiple gradient routes.
    - Strong feature reuse via internal connections.
- **Strengths**:
    - Generalized design for diverse detection backbones.
    - High performance with efficient resource use.
    - Robust gradient propagation and deep feature blending.
- **Limitation**:
    - Complex design requiring precise tuning.

3.2.5.4 Legend and Visual Notations

- **Thick Blue Boxes**: Active processing partitions.
- **Dashed Boxes**: Optional or auxiliary modules.
- **Arrows**: Data and gradient flow direction.
- **Labels (X/n)**: Feature channel partitioning.

3.2.5.5 Summary

GELAN marks a substantial advancement beyond the foundational architectures of CSPNet and ELAN by introducing a **generalized and efficient framework for feature fusion**. Unlike its predecessors, which focused on partial feature reuse and lightweight aggregation, GELAN adopts a more **flexible and recursive design** that facilitates **deeper interactions among feature layers** across the network. This recursive structure allows GELAN to repeatedly combine and refine features at multiple stages, enhancing the network's ability to capture complex spatial and semantic relationships critical for recognizing the subtle and intricate gestures found in Indian Sign Language (ISL).

One of the key strengths of GELAN lies in its ability to promote **robust gradient propagation** throughout the network. By ensuring that gradients flow more effectively back to earlier layers during training, GELAN mitigates common issues such as vanishing gradients and degradation that can limit the learning capacity of deep architectures. This is particularly beneficial in ISL gesture recognition, where precise differentiation depends on the network’s capability to preserve and refine fine-grained feature details across multiple layers.

Furthermore, the generalized fusion approach of GELAN supports **multi-scale and multi-path learning**, enabling the model to assimilate information from various depths and spatial resolutions efficiently. This comprehensive aggregation of features allows for richer representation learning, improving the model’s robustness against challenges like varying hand sizes, orientations, and occlusions that are common in real-world ISL scenarios.

In summary, GELAN’s recursive and generalized feature fusion framework significantly enhances performance in ISL gesture recognition by fostering **deep feature interactions** and ensuring **stable, efficient gradient flow**. This results in a model that is both more accurate and resilient, making GELAN an ideal backbone for state-of-the-art ISL recognition systems based on YOLOv9 and similar architectures.

3.3 Summary of the Proposed Methodology

The proposed methodology seamlessly integrates **Programmable Gradient Information (PGI)** and the **Generalized Efficient Layer Aggregation Network (GELAN)** within the advanced YOLOv9 framework, resulting in a robust and efficient system tailored for Indian Sign Language (ISL) recognition. This combination leverages the unique strengths of both components to address critical challenges in gesture detection and classification, producing a solution optimized for practical, real-world applications.

One of the foremost advantages of this approach is **stable gradient flow** facilitated by PGI, which effectively mitigates the vanishing gradient problem often encountered in deep neural networks. This stability enables the construction of deeper architectures capable of learning intricate, fine-grained features necessary for distinguishing the subtle variations among ISL gestures.

The system’s ability to represent gestures at different scales and orientations is enhanced through **multi-scale feature representation** techniques. By employing feature pyramids and Spatial Pyramid Pooling (SPP) layers, the model captures detailed information from various spatial resolutions, ensuring reliable recognition regardless of hand size or position within the frame.

From a computational perspective, GELAN contributes significantly to the system’s efficiency. It achieves **optimized speed and reduced computational overhead** by intelligently aggregating and reusing feature layers, which minimizes redundant processing without sacrificing accuracy. This balance makes the architecture well-suited for deployment on resource-constrained devices where real-time responsiveness is essential.

Together, the synergistic effects of PGI and GELAN lead to **enhanced accuracy**, improving the system’s performance in recognizing complex ISL gestures swiftly and reliably. The combined architecture maintains a strong balance of **speed, accuracy, and scalability**, making it ideal for assistive technologies and communication aids designed to bridge the gap between the deaf and hard-of-hearing community and the wider population.

In summary, this integrated approach provides a powerful, real-time ISL recognition system that addresses both the computational and representational demands of the task, paving the way for practical implementations in everyday communication support tools.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter presents a comprehensive evaluation of two variants of the YOLOv9 architecture—**YOLOv9c** and **YOLOv9e**—specifically applied to Indian Sign Language (ISL) gesture recognition. The performance of these models is assessed using a suite of standard object detection metrics to provide a holistic view of their accuracy, robustness, and suitability for real-time applications.

Key quantitative metrics used in the evaluation include **mean Average Precision (mAP)**, which summarizes the overall detection accuracy by measuring the area under the precision-recall curve across all gesture classes. **Precision** and **recall** values offer insights into the model's ability to correctly identify true positive gestures without excessive false positives or missed detections, respectively. Additionally, **classification loss** and **bounding box regression loss** are analyzed to understand how well the models learn to distinguish between different ISL signs and precisely localize hand gestures within the input frames.

Beyond numerical metrics, this chapter also incorporates detailed **visualizations** to deepen the analysis of model behavior. Sample **prediction outputs** demonstrate the models' real-time detection capabilities, showcasing their accuracy in localizing and classifying gestures under varying conditions such as different backgrounds, lighting, and signer positions. The inclusion of **confusion matrices** further elucidates areas where the models may confuse visually similar signs, highlighting strengths and weaknesses in classification consistency.

Together, these evaluation components reveal the effectiveness of YOLOv9c and YOLOv9e in handling the challenges inherent in ISL gesture recognition. The results underscore their potential for deployment in assistive communication systems, offering accurate, reliable, and fast gesture detection that can significantly enhance interaction for the deaf and hard-of-hearing community.

4.1 Experimental Setup

The YOLOv9c and YOLOv9e models underwent training on a meticulously prepared custom Indian Sign Language (ISL) dataset comprising 5,178 images, ensuring representation of all 26 alphabet gestures for comprehensive learning. The training regimen extended over 50 epochs, granting the models ample opportunity to develop robust and discriminative feature representations essential for distinguishing between subtle variations in ISL signs. To maintain a balance between efficient use of computational resources and stable gradient updates, a batch size of 32 was selected. Additionally, all input images were uniformly resized to 640x640 pixels, standardizing the input data and enhancing the models' capability to extract spatial features critical for accurate gesture detection. This carefully structured training setup laid a solid groundwork for assessing the real-time detection capabilities of both YOLOv9 variants across the complete ISL alphabet. The evaluation utilized a separate test set including all gesture categories, assessed with the following metrics:

- **mAP@0.5**: Mean Average Precision at an Intersection-over-Union (IoU) threshold of 0.5.
- **mAP50–95**: Mean Average Precision across IoU thresholds from 0.5 to 0.95.
- **Precision and Recall**: Measures of classification accuracy and completeness.
- **Bounding Box and Classification Losses**: Indicators of localization and classification performance.

Table 4.1 Comparison of YOLOv9c and YOLOv9e Models

Model	Parameters (M)	FLOPs (B)	Test Size (pixels)
YOLOv9c	25.3	102.1	640
YOLOv9e	57.3	189.0	640

- **Parameters (M)**: Number of trainable parameters in millions, reflecting model complexity.
- **FLOPs (B)**: Floating Point Operations per second in billions, indicating computational cost.
- **Test Size (pixels)**: Input image resolution, impacting accuracy and computational load.

4.2 Quantitative Performance Metrics

A comparative analysis of YOLOv9c and YOLOv9e is presented in Table 4.2, summarizing key performance metrics.

Table 4.2 Comparative Study of both Models

MODEL	**mAP50-95**	**mAP@0.5**	**Precision**	**Recall**
YOLOv9c	65.56%	99.5%	99.54%	1
YOLOv9e	64.47%	99.5%	99.56%	99.77

4.2.1 Mean Average Precision (mAP)

YOLOv9c demonstrated a marginally superior performance with a **mean Average Precision (mAP50–95)** of **65.56%**, slightly outperforming YOLOv9e, which achieved **64.47%**. This metric reflects the models' overall detection accuracy across a range of Intersection over Union (IoU) thresholds, indicating YOLOv9c's enhanced ability to maintain consistent precision and recall under varying localization strictness.

Both models exhibited an impressive and identical **mAP@0.5 of 99.5%**, underscoring their exceptional accuracy at a moderate IoU threshold. This level of precision is particularly critical in real-time Indian Sign Language recognition, where accurately detecting and classifying gestures with minimal latency and high reliability directly impacts the system's effectiveness in facilitating smooth communication.

4.2.2 Precision and Recall

- **Precision**: YOLOv9e marginally outperformed YOLOv9c with a precision of 99.56% versus 99.54%, suggesting fewer false positives and higher classification confidence.
- **Recall**: YOLOv9c achieved a perfect recall of 100%, identifying all relevant gestures without false negatives, while YOLOv9e scored 99.77%, indicating near-complete detection.

These results highlight YOLOv9e's strength in minimizing incorrect classifications and YOLOv9c's superiority in ensuring comprehensive gesture detection.

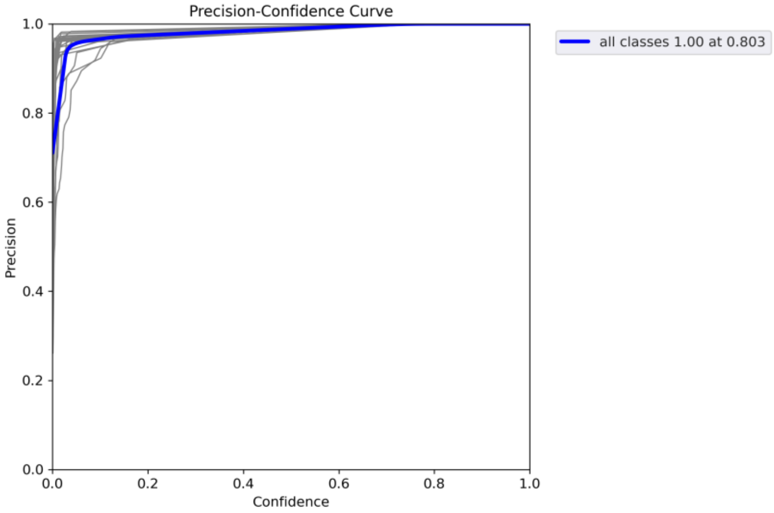


Figure 4.1 P curve of YOLOv9c

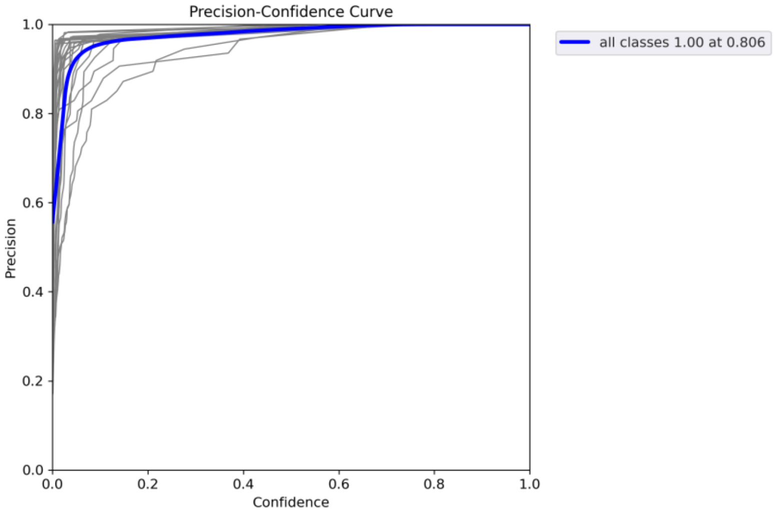


Figure 4.2 P curve of YOLOv9e

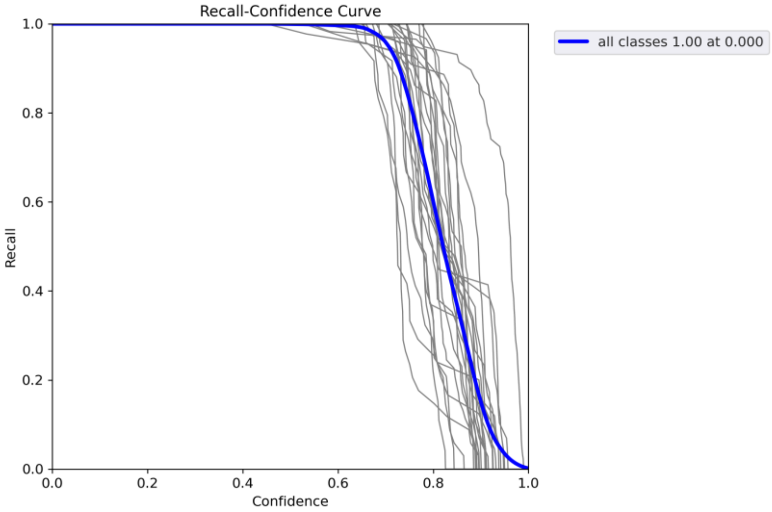


Figure 4.3 R curve of YOLOv9c

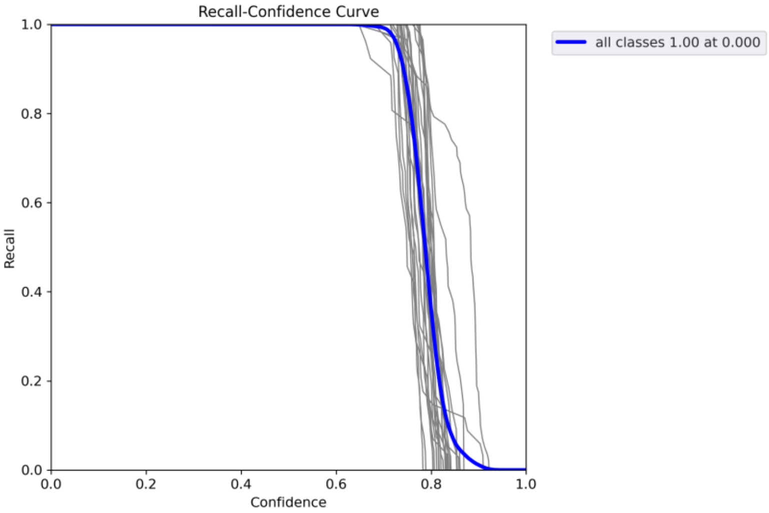


Figure 4.4 R Curve of Yolov9e

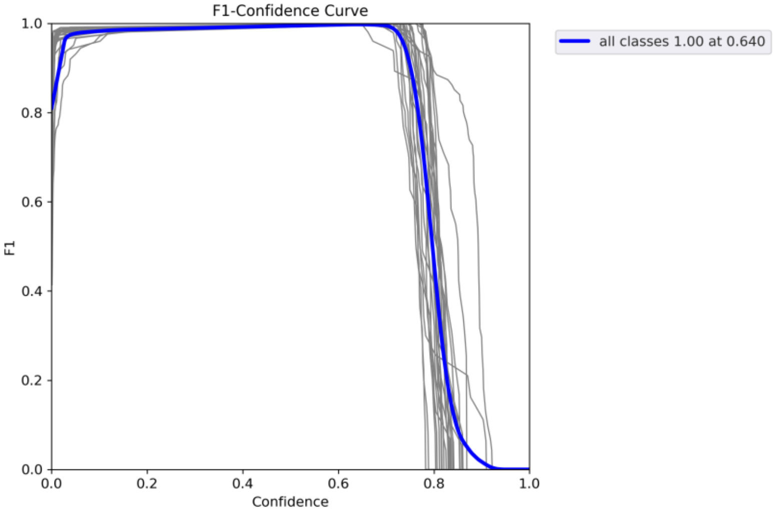


Figure 4.5 F1 Curve of Yolov9c

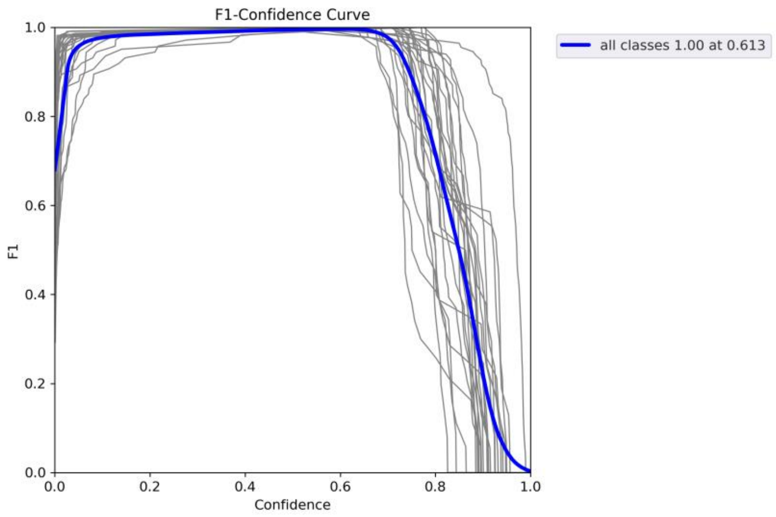


Figure 4.6 F1 curve of YOLOv9e

4.3 F1-Confidence Curve Analysis

The F1-Confidence curve evaluates the balance between precision and recall across confidence thresholds, critical for optimizing real-time ISL recognition.

For YOLOv9c (Figure 4.5), the average F1 score across all 26 gesture classes rises steeply from a confidence threshold of 0.0 to 0.6, stabilizing at a peak F1 score of 1.00 at a threshold of 0.64. This indicates an optimal balance of precision and recall, with near-perfect gesture identification. The F1 score remains stable up to a threshold of 0.8, beyond which it declines sharply due to increased conservatism in detection, prioritizing precision over recall. This threshold of 0.64 serves as a robust baseline for deployment, ensuring high performance in real-world applications.

For YOLOv9e (Figure 4.6), the F1 score peaks at 1.00 at a confidence threshold of 0.613, slightly lower than YOLOv9c. Per-class F1 scores (grey lines) converge at high values for lower thresholds, dropping as thresholds exceed the optimal point, reflecting similar trends to YOLOv9c but with slightly higher classification precision.

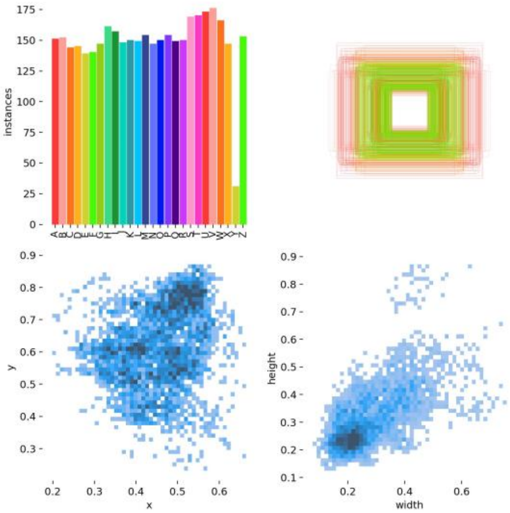


Figure 4.7 Labels of YOLOv9c

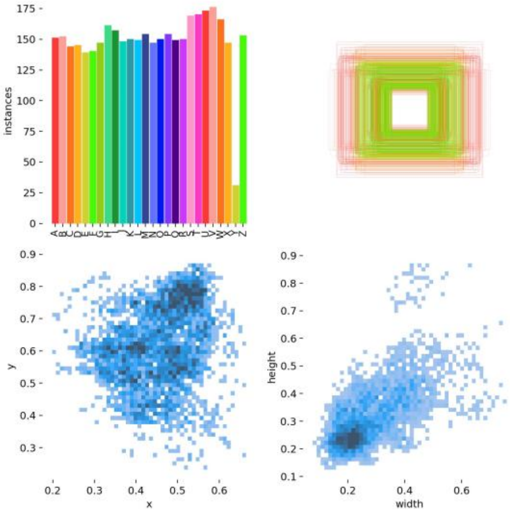


Figure 4.8 Labels of YOLOv9e

4.4 Dataset and Anchor Analysis

The composite plot (Figures 4.7 and 4.8) provides insights into dataset characteristics and bounding box distributions for both models:

1. **Class Distribution (Top-Left)**: Most classes have ~150 samples, but classes ‘N’ and ‘V’ have fewer, indicating a mild imbalance that may affect training.

2. **Anchor Box Analysis (Top-Right)**: Anchor boxes align well with gesture sizes and shapes, ensuring effective localization.
3. **x vs y Distribution (Bottom-Left)**: Object centers are slightly concentrated toward the center-right and upper image space, suggesting potential dataset bias.
4. **Width vs Height Distribution (Bottom-Right)**: Bounding boxes cluster around normalized widths of ~0.2–0.3 and heights of ~0.2–0.4, indicating consistent gesture sizes.

This analysis highlights the need for data augmentation or re-annotation to address class imbalances and spatial biases, enhancing model robustness.

4.5 Training and Validation Metrics

Training performance grids (Figures 4.9 and 4.10) summarize key metrics over 50 epochs for both models.

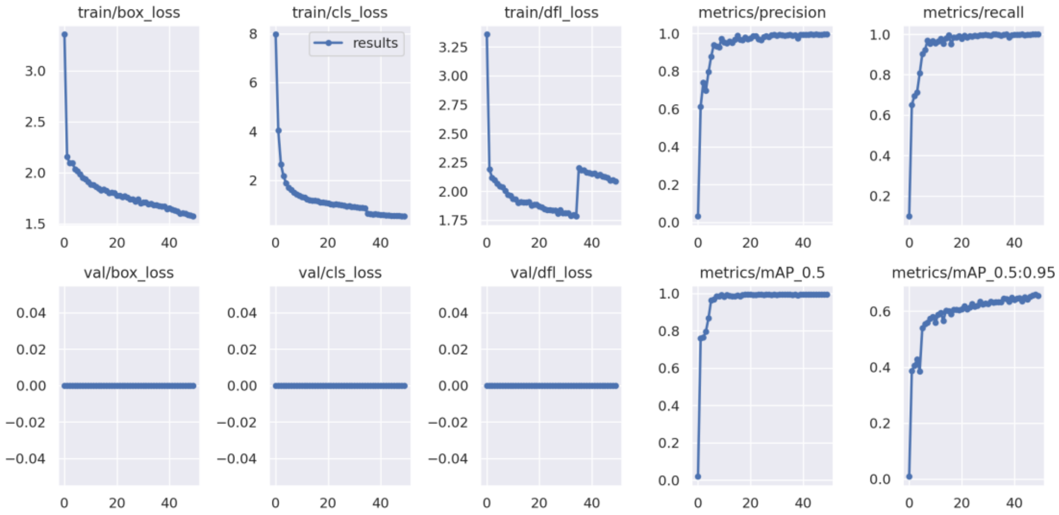


Figure 4.9 Results of YOLOv9c

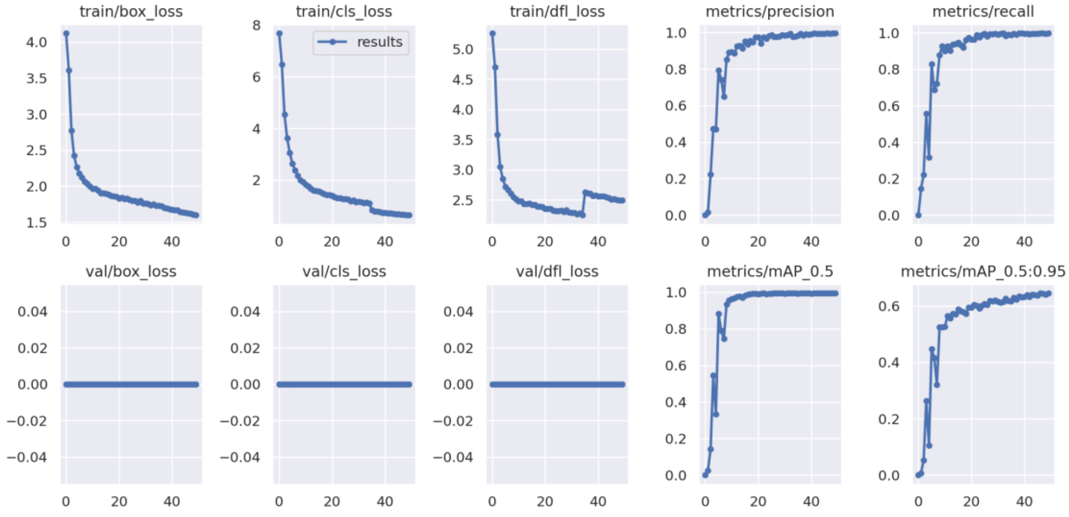


Figure 4.10 Results of YOLOv9e

4.5.1 YOLOv9c Training Metrics (Figure 4.9)

1. **Training Losses**:
   - **Box Loss**: Decreased from ~3.4 to 1.5, reflecting improved localization accuracy.
   - **Class Loss**: Dropped from 8.0 to <2.0, indicating enhanced gesture classification.
   - **DFL (Distribution Focal Loss)**: Reduced from 3.3 to 1.9, showing refined bounding box predictions.
2. **Validation Losses**: Remained at 0, suggesting a configuration issue or missing validation set, requiring urgent investigation.
3. **Evaluation Metrics**:
   - **Precision**: Stabilized at ~0.98 by epoch 15, minimizing false positives.
   - **Recall**: Reached 1.00 by epoch 20, detecting all target gestures.
   - **mAP@0.5**: Saturated at 1.00, indicating high accuracy at moderate IoU thresholds.
   - **mAP50–95**: Achieved ~0.66, reflecting strong generalization across stricter thresholds.

4.5.2 YOLOv9e Training Metrics (Figure 4.10)

1. **Training Losses**:
   - **Box Loss**: Decreased from ~4.0 to 1.5, showing improved localization.
   - **Class Loss**: Reduced from 8.0 to <2.0, enhancing class differentiation.
   - **DFL**: Dropped from 5.0 to 2.5, indicating better bounding box refinement.
2. **Validation Losses**: Similarly flat at 0, reinforcing the need to address validation setup issues.
3. **Evaluation Metrics**:
   - **Precision and Recall**: Precision stabilized at ~0.98, recall reached 1.00 by epoch 20.
   - **mAP@0.5**: Rapidly reached 1.00, confirming high detection accuracy.
   - **mAP50–95**: Peaked at ~0.66, comparable to YOLOv9c.

Insight: Both models demonstrate effective training progress, but the absence of validation losses signals a critical issue that must be resolved to ensure reliable generalization.

4.6 Loss Metrics

Loss metrics provide insights into optimization efficiency:

- **Bounding Box Loss**: YOLOv9c reported a lower loss (1.57) than YOLOv9e (1.59), indicating better localization accuracy for gesture bounding boxes.
- **Classification Loss**: YOLOv9e achieved a lower loss (0.52) compared to YOLOv9c (0.64), suggesting superior classification accuracy and confidence.

This trade-off highlights YOLOv9c's strength in spatial localization and YOLOv9e's advantage in gesture classification.

4.7 Qualitative Analysis

4.7.1 Detection Visualizations

Detection outputs (Figures 4.11 and 4.12) show both models effectively enclosing gestures in color-coded bounding boxes with accurate labels. YOLOv9c exhibited more consistent localization, particularly in cluttered backgrounds or with overlapping hands, while YOLOv9e maintained high confidence scores (e.g., 0.94, 0.89) but showed occasional lower scores for ambiguous gestures.

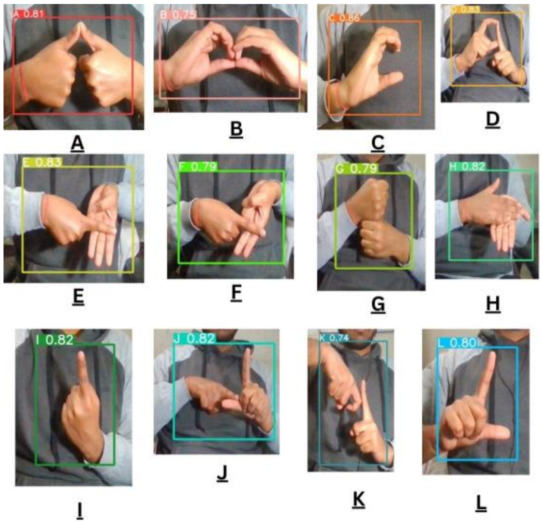


Figure 4.11 Detection Results of YOLOv9c

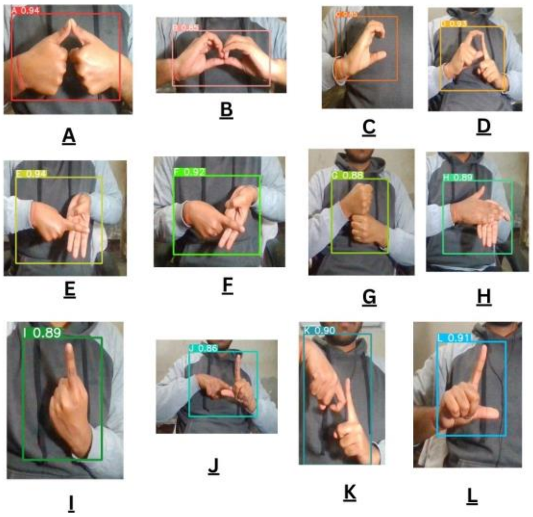


Figure 4.12 Detection Results of YOLOv9e

4.7.2 Confusion Matrix Evaluation

- **YOLOv9e (Figure 4.13)**: Displayed two minor misclassifications (0.50 values), suggesting slight confusion between similar gestures or background noise.
- **YOLOv9c (Figure 4.14)**: Achieved perfect classification across all 26 gestures with no misclassifications, demonstrating superior consistency and robustness.

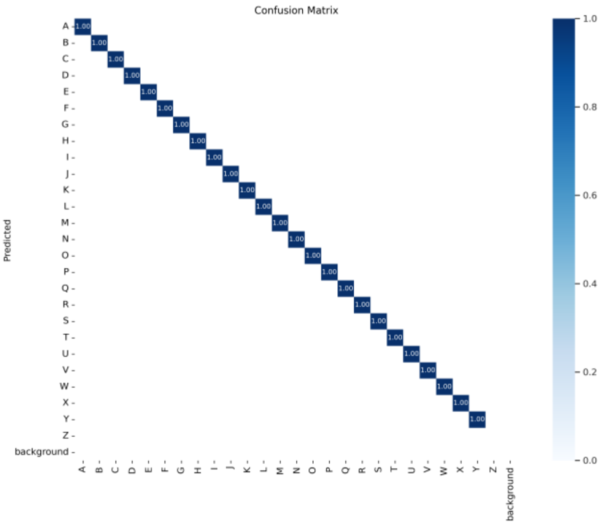


Figure 4.13 Confusion Matrics of YOLOv9c

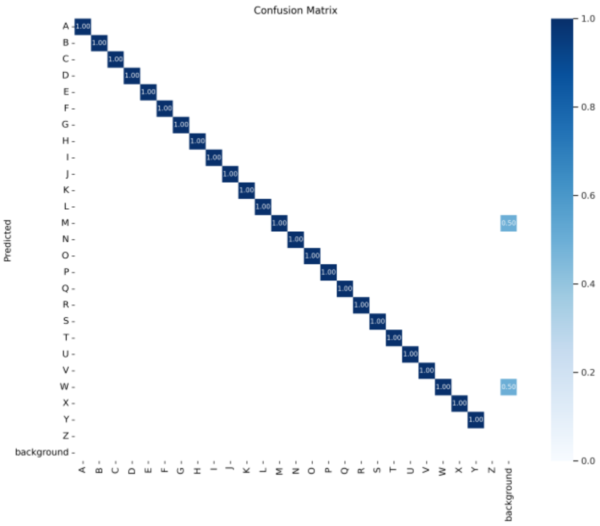


Figure 4.14 Confusion Matrices of YOLOv9e

4.8 Performance Summary and Interpretation

Both YOLOv9c and YOLOv9e perform exceptionally well for ISL recognition, with subtle differences:

- **YOLOv9c**: Excels in detection (higher mAP50–95, perfect recall), ideal for real-time applications where missing gestures is critical.
- **YOLOv9e**: Offers better classification efficiency (lower classification loss, higher precision), suitable for scenarios prioritizing accurate class predictions.

These differences reflect a trade-off between comprehensive detection (YOLOv9c) and precise classification (YOLOv9e), guiding model selection based on application needs.

4.9 Conclusion from Results

YOLOv9c emerges as the more robust model for ISL recognition in real-time scenarios, particularly where high recall is essential to avoid missed detections. YOLOv9e, however, is a strong alternative for applications emphasizing classification accuracy over complete detection. Future studies should investigate inference time, model size, and hardware deployment metrics to assess real-world applicability. Addressing the validation loss issue and mitigating dataset imbalances will further enhance model performance.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The comparative analysis of YOLOv9c and YOLOv9e on the Indian Sign Language (ISL) dataset underscores the efficacy of modern object detection architectures in gesture recognition tasks. Trained for 50 epochs on 5,178 labeled images, YOLOv9c slightly outperformed YOLOv9e with a mAP50–95 score of 65.56%, indicating more precise gesture localization. However, YOLOv9e demonstrated superior classification with marginally higher precision, albeit at a lower mAP50–95 of 64.47%.

Both models exhibited strong detection capabilities, with YOLOv9c achieving perfect recall (1.00), suggesting comprehensive gesture detection. Despite a minor class imbalance in the dataset, the training phase significantly reduced losses, with box loss dropping from 3.4 to 1.5 and class loss from 8.0 to 2.0. Nonetheless, the flat validation losses indicate potential configuration issues, warranting further investigation.

The integration of Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) in YOLOv9 models facilitated robust performance without compromising speed, rendering them suitable for real-time ISL recognition applications.

5.2 Future Scope

Expanding the project to dynamic gesture recognition using LSTM or Transformer architectures, integrating multilingual gesture datasets, and optimizing deployment for edge devices like Jetson Nano are promising directions. Additionally, enhancing the dataset with varied lighting, backgrounds, and user diversity will further improve model robustness. Integrating the model into assistive technologies, such as gesture-to-speech conversion systems, can broaden its accessibility and impact.

REFERENCES

References

1. C. Arya, A. Gusain, Kunal, M. Diwakar, I. Gupta, and N. K. Pandey, “A lightweight solution for real-time Indian sign language recognition,” in *2024 International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit)*, 2024, pp. 1359–1364.

2. D. Biyani, N. V. Doohan, M. Rode, and D. Jain, “Real-time sign language recognition using YOLOv5,” in *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)*, 2023, pp. 582–588.

3. N. Mallikarjuna Swamy, H. S. Sumanth, Keerthi, C. Manjunatha, and R. Sumathi, “Indian sign language detection using YOLOv3,” in *High-Performance Computing and Networking*, C. Satyanarayana, D. Samanta, X.-Z. Gao, and R. K. Kapoor, Eds. Singapore: Springer Singapore, 2022, pp. 157–168.

4. R. Raj, R. Sreemathy, M. Turuk, J. Jagdale, and M. Anish, “Indian sign language recognition in real time using YOLO NAS,” in *2024 3rd International Conference for Advancement in Technology (ICONAT)*, 2024, pp. 1–8.

5. P. Hidayatullah and R. Tubagus, “YOLOv9 architecture explained.” [Online]. Available: https://article.stunningvisionai.com/yolov9-architecture

6. M. Yaseen, “What is YOLOv9: An in-depth exploration of the internal features of the next-generation object detector,” 2024. [Online]. Available: https://arxiv.org/abs/2409.07813

7. N. Herbaz, H. E. Idrissi, and A. Badri, “Deep learning-empowered hand gesture recognition: Using YOLO techniques,” in *14th International Conference on Intelligent Systems: Theories and Applications (SITA 2023)*, Casablanca, Morocco, Nov. 22–23, 2023, pp. 1–7. [Online]. Available: https://doi.org/10.1109/SITA60746.2023.10373734

8. A. Imran, M. S. Hulikal, and H. A. A. Gardi, “Real-time American sign language detection using YOLOv9,” 2024. [Online]. Available: https://arxiv.org/abs/2407.17950

9. M. Alnefaie, “Deep learning-based sign language recognition for hearing and speaking impaired people,” *Intelligent Automation & Soft Computing*, vol. 36, pp. 1653–1669, Jan. 2023.

10. Y. Obi, K. Claudio, V. Budiman, S. Achmad, and A. Kurniawan, “Sign language recognition system for communicating to people with disabilities,” *Procedia Computer Science*, vol. 216, pp. 13–20, Jan. 2023.

11. T. Diwan, G. Anirudh, and J. V. Tembhurne, “Object detection using YOLO: Challenges, architectural successors, datasets, and applications,” *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023.

12. M. M. A. Parambil, L. Ali, M. Swavaf, S. Bouktif, M. Gochoo, H. Aljassmi, and F. Alnajjar, “Navigating the YOLO landscape: A comparative study of object detection models for emotion recognition,” *IEEE Access*, vol. 12, pp. 109427–109442, 2024.

13. J. Gangrade and J. Bharti, “Vision-based hand gesture recognition for Indian sign language using convolutional neural networks,” *IETE Journal of Research*, vol. 69, no. 2, pp. 723–732, 2023. [Online]. Available: https://doi.org/10.1080/03772063.2020.1838342

14. R. K. Pathan, M. Biswas, S. Yasmin, M. U. Khandaker, M. Salman, and A. A. F. Youssef, “Sign language recognition using the fusion of image and hand landmarks through a multi-headed convolutional neural network,” *Scientific Reports*, vol. 13, no. 1, p. 16975, Oct. 2023.

PCSE_UR

ORIGINALITY REPORT

18%	12%	12%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to KIET Group of Institutions, Ghaziabad Student Paper	2%
2	www.mdpi.com Internet Source	1%
3	arxiv.org Internet Source	1%
4	www.coursehero.com Internet Source	1%
5	Submitted to Georgia Institute of Technology Main Campus Student Paper	<1%
6	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced Artificial Intelligence (ICAAAI-2024)", CRC Press, 2025 Publication	<1%
7	Neel Sonawane, Anagha Kulkarni, Pratiksha Deshmukh. "Indian sign language recognition using key point extractor and LSTM", AIP Publishing, 2024 Publication	<1%
8	Submitted to University of Teesside Student Paper	<1%

Indian Sign Language Gesture Recognition Using Deep Learning Techniques

Abhinav Singh
Dept. of CSE
KIET Group of Institutions
Ghaziabad, India
abhinav.2125cse1107@kiet.edu

Disha Goel
Dept. of CSE
KIET Group of Institutions
Ghaziabad, India
disha.2125cse1164@kiet.edu

Akshat Verma
Dept. of CSE
KIET Group of Institutions
Ghaziabad, India
akshat.2125cse1188@kiet.edu

Aayush Kumar
Dept. of CSE
KIET Group of Institutions
Ghaziabad, India
aayush.2125cse1218@kiet.edu

Umang Rastogi
Dept. of CSE
KIET Group of Institutions
Ghaziabad, India
umang.rastogi@kiet.edu

Abstract—**In our digitally connected society, the need for effective communication tools for individuals with hearing and speech impairments has become increasingly urgent. Indian Sign Language (ISL) serves as a crucial medium for this community, yet the communication gap between signers and non-signers remains a significant challenge. This paper introduces a real-time ISL gesture recognition system leveraging the YOLOv9 deep learning model, released in 2024. YOLOv9, a CNN-based object detection model, offers enhanced speed and accuracy compared to its predecessors, making it highly suitable for real-time applications. Despite its recent introduction, little research has been conducted on its application in sign language recognition. Our study provides deep insights into YOLOv9's architecture, highlighting its advantages over previous models and demonstrating its effectiveness in accurately identifying and classifying ISL gestures. Through this work, we aim to bridge the communication barrier and foster inclusivity by enabling seamless translation of sign language into text or speech.**

Index Terms—**Deep learning, Recognition of hand gestures, Indian Sign Language (ISL), Neural networks, Object detection, YOLO**

I. Introduction

A. Need for Sign Detection

Sign language, the visual means of communication for deaf individuals, is the primary language for those unable to speak. It involves hand gestures, body movement, and facial expressions and serves as the most prominent tool for these individuals to express their ideas and thoughts. According to the World Health Organization (WHO), more than 5% of people around the world—approximately 430 million individuals—need rehabilitation services due to disabling hearing loss. Estimates suggest this number could reach 700 million by 2050; thus, 1 in 10 people could have disabling hearing loss by then. A significant challenge we face is that a lot of people are unfamiliar with sign language, which makes it difficult for them to communicate with those who are deaf. To close this communication divide and foster connections between the deaf community and everyone else, sign language recognition models have been introduced using deep learning and machine learning to revolutionize how sign language is utilized. A lot of individuals are unaware that there's no single sign language used worldwide. In reality, there are approximately 138 to 300 distinct sign languages in use around the globe today, with various nations familiar with different variations. While a single, common sign language might seem preferable, the diverse sign languages developed organically within various groups worldwide and evolved over time [1]. They embody the cultural wealth of the deaf community, instead of being fabricated or enforced. Among the most recognized sign languages are ASL (American Sign Language), FSL (French Sign Language), and Libras (Brazilian Sign Language). Every spoken language has its own grammar and syntax, and therefore, every language has a corresponding sign language. For the purpose of this paper, Indian Sign Language (ISL) will be used. This paper presents an advanced YOLO V9 model designed to accurately predict ISL gestures in real-time. The Framework can predict all 26 letters of the English alphabet using ISL gestures and is capable of recognizing multiple gestures in a single image.Several studies have previously explored ISL recognition using various versions of the YOLO (You Only Look Once) object detection algorithm. A model using YOLOv5x with attention mechanisms achieved a high accuracy of 99.4% on the MU HandImages ISL dataset, demonstrating both efficiency and suitability for real-time applications on edge devices [2]. Another approach utilizing YOLOv8 and Roboflow datasets showed precision of 98.46%, recall of 98.78%, and mAP@0.5 of 97.54, outperforming earlier YOLO versions [1]. Earlier implementations using

YOLOv3 achieved an average accuracy of 82% on a small ISL alphabet dataset [3], while a recent YOLO NAS-based system reported a mean Average Precision (mAP) of 95.68% using a custom ISL dataset [4]. These prior works show the effectiveness and evolution of YOLO-based models for ISL recognition, paving the way for further innovations in real-time sign language detection.

B. YOLO (You Only Look Once): Evolution Over the Years

The YOLO (You Only Look Once) framework was first introduced in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi as an innovative approach to real-time object detection. In contrast to conventional approaches that employed distinct models for region proposal and classification, YOLO integrated these processes into a single convolutional neural network (CNN), allowing it to conduct object localization and classification in just one forward pass. This redefinition as a regression issue significantly enhanced processing speed while preserving competitive accuracy.

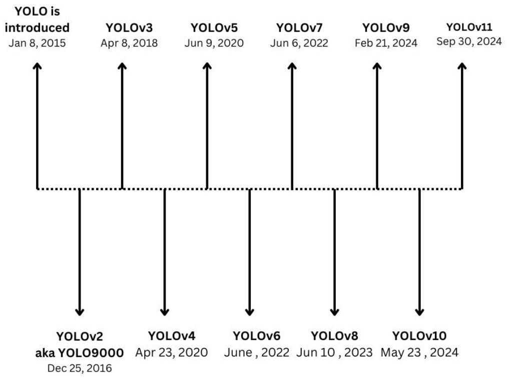


Fig. 1: YOLO Evolution over the years

Since its launch, YOLO has progressed through several key versions, each enhancing detection accuracy, speed, and architectural efficiency. As shown in Fig. 1, the timeline of YOLO development extends from YOLOv1 in 2015 to YOLOv11 by late 2024. Each iteration brought forward new architectural improvements: YOLOv2 (YOLO9000) broadened the model's label space; YOLOv3 advanced feature extraction with Darknet-53; YOLOv4 combined CSPNet and PANet for superior gradient flow and feature integration; YOLOv5 and YOLOv6 prioritized lightweight deployment; YOLOv7 incorporated E-ELAN modules; YOLOv8 focused on anchor-free detection; and versions from YOLOv9 introduced elements such as RepNCSP, Adown, and advanced fusion methods for top-tier performance.

These successive iterations underline the model's swift innovation and wide adoption in real-time computer vision applications. As of early 2025, YOLOv12 marks the most recent achievement in this development, further enhancing detection accuracy, computational efficiency, and deployment scalability across a variety of applications.

C. YOLOv9 architecture

YOLOv9, which was presented by Wang et al. in early 2024, signifies a significant leap forward in real-time object detection, balancing both precision and performance. As illustrated in Fig. 2: YOLOv9-C Architecture, this model is composed of four essential elements: the backbone, neck, detection head, and an auxiliary training branch. This modular design facilitates multi-scale feature learning, semantic retention, and enhanced training dynamics.

The backbone processes an input image of dimensions 640×640×3 through initial convolutional layers that reduce feature map sizes to 320×320 and 160×160. Central to the backbone are RepNCSP-ELAN4/6 modules, which integrate CSPNet, repeated convolutions, ELAN techniques, and layer normalization—improving gradient flow and contextual understanding. Interspersed within these are ADown blocks, employing asymmetric 1×3 and 3×1 convolutions with SiLU activation to diminish spatial dimensions while retaining semantics. The generated feature maps—P3 (80×80×256), P4 (40×40×512), and P5 (20×20×512)—are subsequently sent to the neck.

The neck executes both top-down and bottom-up fusion utilizing upsampling and concatenation, thereby enriching semantic and spatial features across varying scales. Initially, each scale undergoes processing by a RepNCSP-ELAN4 block. A notable element within this section is the SP-PELAN module, serving as a dense compression unit at the lowest layer, effectively channeling rich features upward. Following the fusion process, ADown blocks along with additional RepNCSP-ELAN4 units refine the multi-scale representations, yielding output maps at dimensions of 80×80×512, 40×40×512, and 20×20×512.

The detection head consists of three parallel branches that correspond to these scales, enabling specialized object detection at varying resolutions. Every branch employs a Detect module to produce bounding boxes, objectness scores, and class probabilities. Non-Maximum Suppression (NMS) is subsequently applied to remove overlaps, ensuring accurate outputs.

An auxiliary branch, which only operates during training, aids in convergence and consistency. It replicates earlier backbone layers with convolution, RepNCSP-ELAN, and ADown blocks, followed by CBFuse and CBLinear modules that integrate and condense intermediate features. While this branch is omitted during inference, it enhances the model's generalization capabilities.

Dimension alignment throughout the architecture—achieved through Concat, ADown, and Upsample—guarantees effective feature interaction and real-time performance. In summary, YOLOv9 merges architectural accuracy with practical speed and precision, establishing itself as a dependable solution for contemporary object detection challenges.

II. Literature Review

YOLOv9 has emerged as a state-of-the-art model in the domain of real-time hand gesture recognition, building upon a

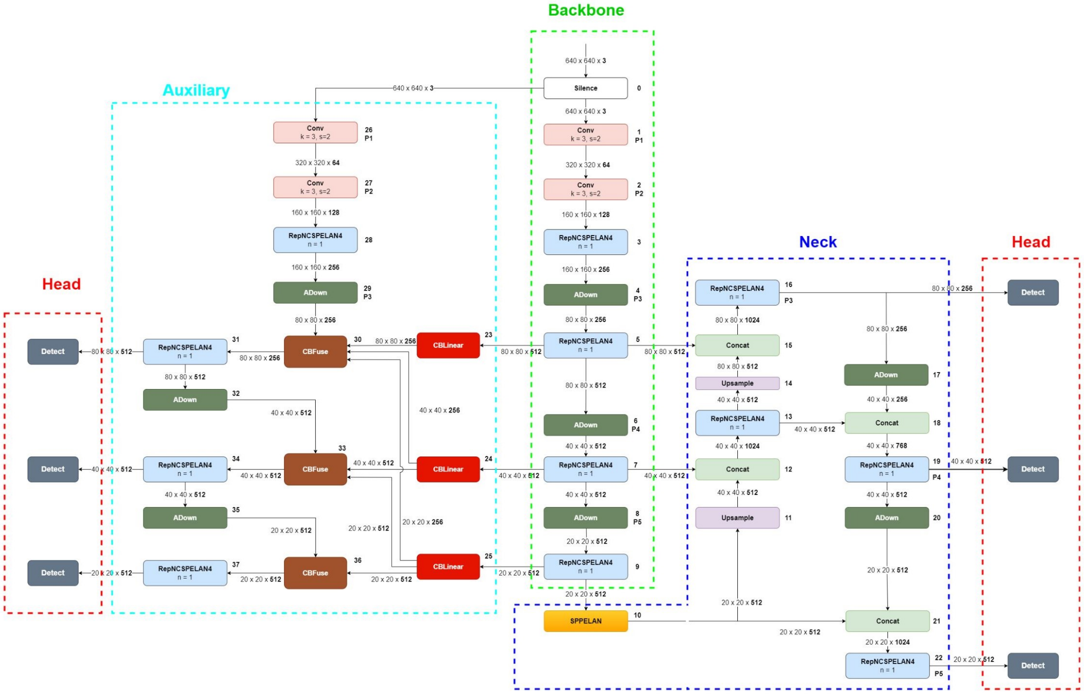


Fig. 2: YOLOv9-C Architecture. [5]

lineage of earlier YOLO architectures. The early versions of YOLO were notable for their remarkable real-time detection capabilities [6], which are crucial for applications such as sign language translation where low latency is essential. These versions demonstrated YOLO's strength in processing visual data efficiently and accurately [6], positioning the model as an ideal candidate for gesture-based recognition tasks.

The evolution of YOLO from versions v5 through v8 introduced several improvements in terms of architecture, training strategies, and loss functions [7], [8]. YOLOv7, for example, emphasized optimizations that enhanced both training speed and detection accuracy, further validating its applicability to hand gesture recognition . These developments paved the way for YOLOv9, which marks a significant shift in design philosophy with the introduction of two core innovations: Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN).

GELAN complements PGI by optimizing the efficiency of feature extraction and multi-scale aggregation [6]. This architectural framework incorporates convolutional layers and bottleneck blocks, allowing the model to effectively capture both fine-grained and coarse-grained gesture features. These enhancements are especially beneficial in recognizing complex hand gestures where detail preservation is essential. Together, PGI and GELAN form the backbone of YOLOv9's superior performance, resulting in improved detection accuracy and real-time inference speed. Empirical benchmarks substantiate these gains across a variety of gesture recognition datasets.

The architectural design of YOLOv9 further includes specialized components such as Silence Blocks, Convolutional Blocks, RepNCSPELAN Blocks, and Auxiliary/Detect Blocks [8], which enhance gradient flow and feature interpretation. These elements contribute to the model's ability to deliver precise and efficient gesture classification even in visually complex environments.

A review of datasets used in YOLOv9-based hand gesture recognition studies reveals a wide spectrum in terms of size and complexity. Some datasets are narrowly focused, containing only a subset of gestures or sign language alphabets, while others encompass diverse hand shapes, lighting conditions, backgrounds, and skin tones [8]. This diversity is critical for training models that generalize well to real-world scenarios. Another vital factor is the quality of dataset annotation. Hand gesture datasets typically require precise, manually-labeled bounding boxes to ensure model effectiveness. High annotation accuracy allows the model to learn correct associations between visual features and gesture labels.

Evaluation of YOLOv9-based systems is generally conducted using standard object detection metrics. Among these, mean Average Precision (mAP) is the most comprehensive, capturing both detection accuracy and localization performance. Precision, recall, and F1-score are employed to assess the quality of classification [8]. Precision quantifies the proportion of correct detections among all predictions, while recall measures the proportion of actual gestures correctly identified. The F1-score provides a harmonic mean of precision and recall, offering a balanced view of the model's performance.

Practical applications of YOLOv9 in gesture recognition are diverse. It plays a pivotal role in real-time sign language recognition systems , enabling seamless translation of hand gestures into textual or spoken language. YOLOv9 has also been utilized in human-computer interaction (HCI) frameworks [7], allowing users to control devices through intuitive gesture-based commands. This represents a significant advancement in making technology more accessible, particularly in scenarios where conventional input devices are impractical. Moreover, YOLOv9 has demonstrated potential in assistive technologies for the hearing impaired, facilitating more effective communication [9]. Deployment of these systems often necessitates additional optimization to ensure compatibility with a variety of hardware platforms and resource constraints [10].

In summary, YOLOv9 has emerged as a powerful tool in the field of hand gesture recognition due to its novel architectural innovations, speed, and high detection accuracy. The integration of PGI and GELAN has addressed key limitations of prior models, particularly those concerning gradient stability and feature extraction. These contributions have direct implications for real-time sign language translation and broader HCI applications. The potential for improving accessibility and inclusion through such technologies is vast, particularly for individuals with hearing impairments [11]. Ongoing investigations in this field, especially concentrating on multimodal fusion, efficient deployment, and rigorous benchmarking, are expected to result in significant advancements in creating gesture recognition systems that are both highly accurate and efficient. [12], [13], [14].

III. Dataset

Our Indian Sign Language (ISL) dataset is a collection of 5178 images, each showcasing a static hand gesture representing one of the 26 letters of the English alphabet (A-Z) in ISL. To maintain clarity and consistency, all images were captured in a controlled indoor setting with stable lighting and a uniform background. Using Roboflow, we carefully labeled each image with precise bounding boxes to ensure accurate hand gesture detection.

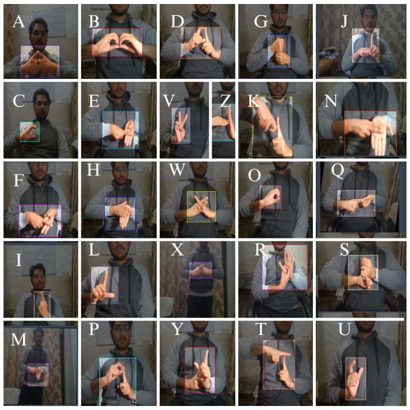


Fig. 3: Samples of dataset

To train our model effectively, we divided the dataset into three parts: 3872 images (75%) for training, 788 images (15%) for validation, and 518 images (10%) for testing. Every image was preprocessed by applying auto-orientation and resizing to 640×640 pixels, making them suitable for YOLO-based object detection. Unlike many datasets that rely on artificial variations, ours preserves the natural gestures, making it a valuable resource for real-world ISL recognition. **Some examples from our dataset can be seen in Figure 3**, showcasing various hand gestures representing letters in Indian Sign Language.

IV. Experimental Setup

Our method employs image, video, and real-time detection by utilizing the Ultralytics platform. YOLOv9, which is built upon YOLOv7, incorporates Generalized Efficient Layer Aggregation Network (GELAN), improving its performance in object detection. We trained two different versions of YOLOv9, specifically YOLOv9c and YOLOv9e, using a labeled dataset that includes 26 classes of Indian Sign Language (ISL) for a total of 50 epochs. The approach addresses image, video, and real-time detection.

TABLE I: Comparison of YOLOv9c and YOLOv9e models.

Model	Params (M)	FLOPs (B)	Test size
YOLOv9c	25.3	102.1	640
YOLOv9e	57.3	189.0	640

A. *Image Detection*

Image detection utilizes YOLOv9 to identify ISL hand signs from static images. This is advantageous for purposes like educational resources or ISL interpretation, where every image is examined separately for accurate gesture identification.

B. *Video Detection*

Video detection breaks down video sequences into individual frames, detecting hand gestures throughout. This is essential for assessing ISL signs in recorded videos, which is helpful for sign language interpretation during lectures or meetings.

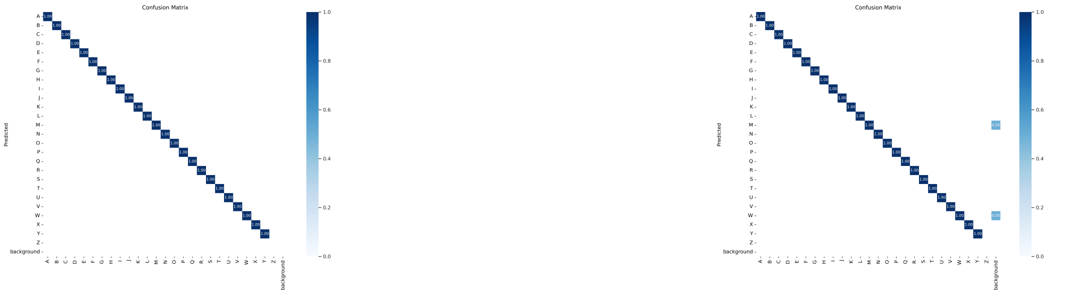
C. *Real-Time Detection*

Real-time detection utilizes the concepts of video detection and implements them on live video streams, including those from webcams. This is crucial for applications requiring immediate feedback, such as speech-to-text conversion designed to aid the Deaf and Hard-of-Hearing (DHH) community. The efficiency of YOLOv9 ensures both low latency and high accuracy.

This setup illustrates the adaptability of YOLOv9 in recognizing ISL gestures across various situations, thereby improving communication tools for sign language users.

V. Results

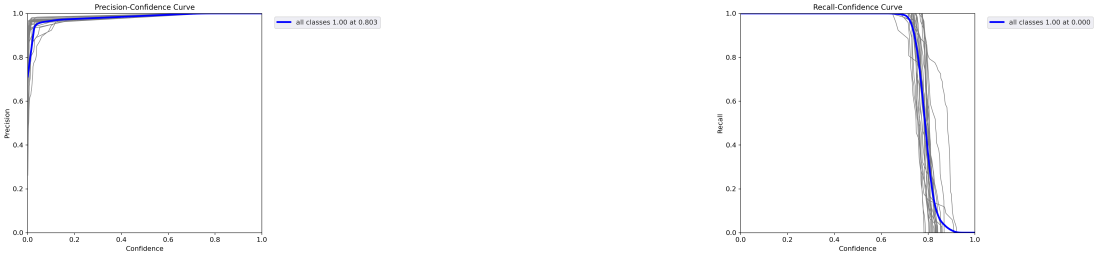
The comparison of the mAP50-95 metric for YOLO-v9e and YOLO-v9c is outlined in Table 2. Following the training of both models for 50 epochs with a batch size of 32 on the Indian Sign Language dataset, which consists of 5,178 images, YOLO-v9c attained a marginally superior mAP50-95 of 65.56%, while YOLO-v9e recorded 64.47%. In terms of precision, recall, and mAP@0.5, YOLO-v9e recorded 99.56%,



(a) YOLOv9c Confusion Matrix

(b) YOLOv9e Confusion Matrix

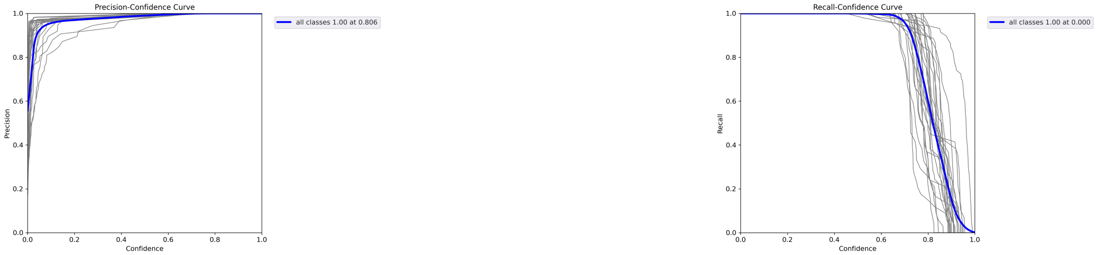
Fig. 4: Comparison of Confusion Matrices for YOLOv9c and YOLOv9e



(a) Precision Curve (P-Curve)

(b) Recall Curve (R-Curve)

Fig. 5: Precision and Recall Curves of YOLOv9c



(a) Precision Curve (P-Curve)

(b) Recall Curve (R-Curve)

Fig. 6: Precision and Recall Curves of YOLOv9e

99.77%, and 99.5%, while YOLO-v9c achieved 99.54%, 100%, and 99.5%, respectively. These results indicate that YOLO-v9c slightly outperforms in recall and overall mAP50-95, while both models show nearly identical precision and mAP@0.5.

Figures 6 and 7 show detection outputs for each model, accurately identifying gestures within color-coded bounding boxes.

Regarding loss values, YOLOv9c had a bounding box loss of 1.57 and classification loss of 0.64, whereas YOLOv9e had 1.59 and 0.52, respectively. This implies YOLOv9e offers better classification accuracy, while YOLOv9c performs slightly better in localization.

Both models were tested on an image displaying all 26 Indian Sign Language alphabet gestures, and the confusion matrices (Figure 8) reflect their classification differences. YOLOv9e showed two minor misclassifications (value: 0.50), whereas YOLOv9c achieved 100% accuracy with no errors, underscoring its stronger classification consistency.

Figures 9 and 10 display the Precision-Confidence and Recall-Confidence curves. Both models maintain high precision across thresholds, but YOLOv9e's recall drops more sharply at higher thresholds, indicating slightly reduced robustness. In contrast, YOLOv9c demonstrates more stable recall, reinforcing its consistent gesture recognition. Overall, both models perform exceptionally well, with YOLOv9c providing marginally more stable and reliable predictions.

However, when considering other metrics such as inference speed and bounding box accuracy, further evaluation may be needed to determine the optimal model for real-time Indian Sign Language recognition.

Overall, **YOLO-v9c demonstrates slightly superior object detection accuracy, particularly in localization, while YOLO-v9e offers improved classification accuracy**.

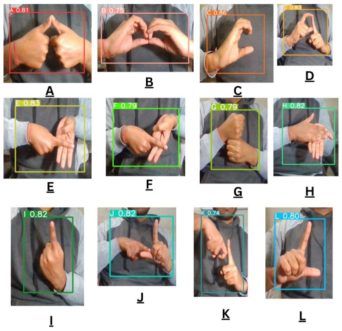


Fig. 7: YOLOv9c results on multiple ISL sign detection

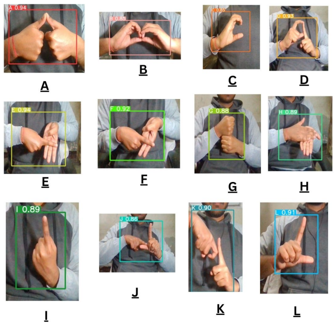


Fig. 8: YOLOv9e results on multiple ISL sign detection

The comparison of the deployed algorithms is summarized in Table II.

TABLE II: Models Comparison on Different Metrics

Model	mAP50-95	mAP@0.5	Precision	Recall
YOLOv9c	65.56%	99.5%	99.54%	1
YOLOv9e	64.47%	99.5%	99.56%	99.77

VI. Conclusion

The YOLO framework has evolved into a state-of-the-art model for real-time object detection due to its speed and accuracy. In this study, we compare **YOLOv9c** and **YOLOv9e** for Indian Sign Language (ISL) recognition using key detection metrics.

Table 2 shows the **mAP**$_{50-95}$ comparison. Trained on 5,178 ISL images for 50 epochs with a batch size of 32, YOLOv9c achieved **65.56%**, slightly outperforming YOLOv9e's **64.47%**. While both models showed nearly identical precision and **mAP@0.5**, YOLOv9c had a slight edge in recall and overall mAP.

Bounding box losses were **1.57 (YOLOv9c)** and **1.59 (YOLOv9e)**, indicating marginally better localization in YOLOv9c. Confusion matrix analysis (**Figure 6**) revealed **YOLOv9c achieved 100% classification accuracy**, while YOLOv9e had **two minor misclassifications (value: 0.50)**, suggesting occasional confusion between similar gestures.

The improved recall of YOLOv9c highlights its robustness in reliably detecting all gestures—essential for responsive, real-time applications. YOLOv9's integration of **Programmable Gradient Information (PGI)** eliminates prior information bottlenecks, and **GELAN** enhances convolutional efficiency, balancing speed, complexity, and accuracy.

Although YOLOv9e provides marginally improved classification accuracy, **YOLOv9c displays more reliable recognition and localization**, making it ideal for real-time sign language recognition on smart devices and assistive technologies for the Deaf and Hard-of-Hearing (DHH) community. Future work will explore inference speed and deployment to evaluate their suitability for large-scale ISL applications.

References

[1] C. Arya, A. Gusain, Kunal, M. Diwakar, I. Gupta, and N. K. Pandey, "A lightweight solution for real-time indian sign language recognition," in *2024 International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit)*, 2024, pp. 1359–1364.

[2] D. Biyani, N. V. Doohan, M. Rode, and D. Jain, "Real time sign language recognition using yolov5," in *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)*, 2023, pp. 582–588.

[3] N. Mallikarjuna Swamy, H. S. Sumanth, Keerthi, C. Manjunatha, and R. Sumathi, "Indian sign language detection using yolov3," in *High Performance Computing and Networking*, C. Satyanarayana, D. Samanta, X.-Z. Gao, and R. K. Kapoor, Eds. Singapore: Springer Singapore, 2022, pp. 157–168.

[4] R. Raj, R. Sreemathy, M. Turuk, J. Jagdale, and M. Anish, "Indian sign language recognition in real time using yolo nas," in *2024 3rd International Conference for Advancement in Technology (ICONAT)*, 2024, pp. 1–8.

[5] P. Hidayatullah and R. Tubagus, "Yolov9 architecture explained." [Online]. Available: https://article.stunningvisionai.com/yolov9-architecture

[6] M. Yaseen, "What is yolov9: An in-depth exploration of the internal features of the next-generation object detector," 2024. [Online]. Available: https://arxiv.org/abs/2409.07813

[7] N. Herbaz, H. E. Idrissi, and A. Badri, "Deep learning empowered hand gesture recognition: using yolo techniques," in *14th International Conference on Intelligent Systems: Theories and Applications, SITA 2023, Casablanca, Morocco, November 22-23, 2023*. IEEE, 2023, pp. 1–7. [Online]. Available: https://doi.org/10.1109/SITA60746.2023.10373734

[8] A. Imran, M. S. Hulikal, and H. A. A. Gardi, "Real time american sign language detection using yolo-v9," 2024. [Online]. Available: https://arxiv.org/abs/2407.17950

[9] M. Alnefaie, "Deep learning-based sign language recognition for hearing and speaking impaired people," *Intelligent Automation Soft Computing*, vol. 36, pp. 1653–1669, 01 2023.

[10] Y. Obi, K. Claudio, V. Budiman, S. Achmad, and A. Kurniawan, "Sign language recognition system for communicating to people with disabilities," *Procedia Computer Science*, vol. 216, pp. 13–20, 01 2023.

[11] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023.

[12] M. M. A. Parambil, L. Ali, M. Swavaf, S. Bouktif, M. Gochoo, H. Aljassmi, and F. Alnajjar, "Navigating the yolo landscape: A comparative study of object detection models for emotion recognition," *IEEE Access*, vol. 12, pp. 109 427–109 442, 2024.

[13] J. Gangrade and J. Bharti, "Vision-based hand gesture recognition for indian sign language using convolution neural network," *IETE Journal of Research*, vol. 69, no. 2, pp. 723–732, 2023. [Online]. Available: https://doi.org/10.1080/03772063.2020.1838342

[14] R. K. Pathan, M. Biswas, S. Yasmin, M. U. Khandaker, M. Salman, and A. A. F. Youssef, "Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network," *Sci. Rep.*, vol. 13, no. 1, p. 16975, Oct. 2023.