**AlphaSigns — Initial Implementation Report**

**Group Members:**

* **Sibtain Wali** (21L-5098)
* **Abubakar Tahir** (21L-1837)
* **Ibrahim Malik** (21L-5445)

## **Project Title:**

**Sign Language Alphabet Detection using YOLO**

## **Project Overview:**

We are developing a **computer vision system** that detects and classifies **American Sign Language (ASL) alphabets**.  
The project uses **YOLO (You Only Look Once)**, a real-time object detection model, for fast and accurate recognition of static hand gestures.  
This system aims to assist in **communication** and **accessibility** for the deaf and hard-of-hearing community.

# **Code Overview**

This notebook contains a complete pipeline for training a **YOLOv8** object detection model to recognize **American Sign Language (ASL) letters**. The process involves:

1. Installing necessary libraries.
2. Downloading and preparing the ASL dataset.
3. Configuring and training the YOLOv8 model.
4. Evaluating performance using validation metrics.

The goal is to build a model that can accurately detect and classify hand signs representing different letters of the alphabet.

# **Libraries Used**

The code relies on several key Python libraries:

* **Ultralytics YOLOv8**: The main framework for training and running the YOLO (You Only Look Once) object detection model.
* **Roboflow**: Used to download and manage the ASL dataset in the correct format for YOLO training.
* **OpenCV (cv2)**: Handles image processing tasks.
* **NumPy & Matplotlib**: For numerical operations and visualization.
* **Torch & TorchVision**: PyTorch-based deep learning framework for model training.
* **Pillow (PIL)**: For image loading and manipulation.

These libraries ensure smooth data handling, model training, and evaluation.

# **Datasets Used**

The dataset consists of **labeled images of ASL hand signs** for each letter of the alphabet (A-Z). It was sourced from **Roboflow**, a platform that provides annotated datasets for machine learning.

### **Dataset Structure:**

* **Training Set (504 images)**: Used to train the model.
* **Validation Set (144 images)**: Used to evaluate model performance during training.
* **Test Set**: Reserved for final evaluation (though not explicitly shown in the logs).

The dataset is formatted in **YOLOv5/YOLOv8-compatible structure**, with images and corresponding label files.

# **Training Phase**

### **Model Selection**

The **YOLOv8n (nano)** variant was chosen—a lightweight but effective version of YOLOv8, suitable for real-time detection tasks.

### **Training Configuration**

* **Epochs:** 50 (full training cycles over the dataset).
* **Batch Size:** 16 (number of images processed at once).
* **Image Size:** 640x640 pixels (standard for YOLO models).
* **Optimizer:** **AdamW** (an improved version of Adam with weight decay).
* **Learning Rate:** Automatically adjusted for optimal training.

### **Key Observations During Training:**

* **Early Epochs (1-10):** The model quickly improves, with **mAP50 (mean Average Precision at 50% IoU) rising from 0 to ~0.75**.
* **Mid Training (10-30):** Performance stabilizes, with **mAP50 reaching ~0.80**.
* **Final Epochs (30-50):** Fine-tuning occurs, with **final mAP50 at ~0.897**, indicating high detection accuracy.

# **Training and Validation Results**

### **Final Validation Metrics:**

* **mAP50:** **0.897** (Excellent detection accuracy).
* **Precision:** **0.906** (High correctness in predictions).
* **Recall:** **0.893** (Good coverage of actual hand signs).

### **Class-wise Performance:**

* **Best Performing Letters:** B, C, E, F, G, H (near-perfect detection).
* **Challenging Letters:** I, J, L (lower recall, possibly due to complex hand shapes).

The model shows strong overall performance but may need slight improvements for certain letters.

### **Predicted Output**

The trained model successfully identifies ASL letters in test images. Sample predictions show accurate bounding boxes and labels. Real-time inference (via YOLO.predict()) confirms the model’s ability to generalize on unseen data.

# **Current Status of the Code**

**Training Completed Successfully** – The model has been trained and validated.  
 **Best Model Saved** – The weights (best.pt) are stored for future use.  
 **Performance Logs Available** – All training metrics are recorded for analysis.

**Validation Metrics Available**

**Conclusion**

So, overall, this project did a pretty great job training a YOLOv8 model to recognize and classify American Sign Language (ASL) letters with medium accuracy, hitting a mean Average Precision (mAP50) of about 89.7%. The model works really well on most of the letters, although there’s still some room for improvement, especially for trickier ones like I, J, and L. A little extra data or fine-tuning might do the trick there. These results show real promise for creating live ASL recognition tools, which could make things more accessible for the deaf and hard-of-hearing community, help with learning ASL, and even be part of communication tools. Moving forward, next steps include making the model more ready for real-world use, gathering more diverse data so it can handle a wider range of situations, and building an easy-to-use app for live sign detection. All in all, this lays a solid foundation for future AI-powered sign language projects.