BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri - West, Mumbai – 400058

A Minor Project Report on

Sign Language Interpreter Using Deep Learning

By
Shantanu Kadam- 2020400017
Akhil Sharma- 2020400051

Under the Mentorship of
Prof. Surekha Dholay

April 2023

Table of Contents

Introduction

Sign language is a visual language used by deaf or hard-of-hearing individuals to communicate. It is a complex and expressive form of communication that relies on hand gestures, facial expressions, and body movements. Sign language interpreters play a crucial role in facilitating communication between deaf individuals and the hearing world.

In recent years, deep learning has emerged as a powerful tool for various computer vision tasks. Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), YOLO-V5 have demonstrated remarkable success in image recognition and sequence prediction tasks. Leveraging the capabilities of deep learning, researchers and engineers have started to explore the application of this technology in sign language interpretation.

A sign language interpreter using deep learning is an innovative approach that aims to automate the process of interpreting sign language gestures into spoken language or text. By employing computer vision techniques and deep learning algorithms, these interpreters can recognize and understand the intricate movements and expressions of sign language users, enabling real-time interpretation and translation.

The development of a sign language interpreter using deep learning involves several key steps. First, a large dataset of sign language gestures is collected, consisting of videos or images of individuals performing various signs. These gestures are then labeled and annotated with corresponding spoken or written language translations. Next, deep learning models are trained on this dataset, using architectures specifically designed to capture the temporal and spatial features of sign language.

During the training process, the deep learning models learn to recognize patterns and variations in hand movements, facial expressions, and body postures that represent different signs. The models are optimized to achieve high accuracy and robustness in recognizing and classifying sign language gestures.

Once trained, the sign language interpreter can be deployed on different platforms, such as smartphones or dedicated devices. Users can capture sign language gestures using a camera, and the interpreter applies the deep learning model to analyze and interpret the gestures in real-time. The system then generates spoken language output or displays text translations, facilitating communication between deaf individuals and hearing individuals who do not understand sign language.

The application of deep learning in sign language interpretation has the potential to break down communication barriers and enhance inclusivity for the deaf community. By providing an automated and reliable means of sign language translation, deep learning-based sign language interpreters have the ability to empower individuals with hearing impairments, enabling them to participate more fully in various aspects of life, including education, employment, and social interactions.

Background Research

Sign language recognition using deep learning

Deep learning is a type of machine learning that uses artificial neural networks to learn from data. This makes it well-suited for tasks such as sign language recognition, where the input data is often noisy and difficult to interpret.

There are a number of different deep learning models that have been used for sign language recognition. One common approach is to use a convolutional neural network (CNN). CNNs are well-suited for tasks such as image recognition, and they can be used to extract features from sign language videos.

Another approach is to use a recurrent neural network (RNN). RNNs are well-suited for tasks such as natural language processing, and they can be used to track the temporal information in sign language videos.

Recent advances in sign language recognition using deep learning

Recent advances in deep learning have led to significant improvements in sign language recognition accuracy. In 2019, a team of researchers from Google AI achieved a sign language recognition accuracy of 95% on the COCO dataset. This is a significant improvement over previous results, and it shows that deep learning is a promising technology for sign language recognition.

The future of sign language recognition using deep learning

The future of sign language recognition using deep learning is bright. As deep learning technology continues to improve, we can expect to see even higher accuracy rates. This will make it possible for people who are deaf or hard of hearing to communicate more easily with the rest of the world.

In addition to improving accuracy, deep learning can also be used to make sign language recognition systems more robust. For example, deep learning systems can be trained to recognize sign language in noisy environments. This is important for people who are deaf or hard of hearing who may need to communicate in noisy environments, such as classrooms or crowded events.

Overall, deep learning is a promising technology for sign language recognition. It has the potential to make it easier for people who are deaf or hard of hearing to communicate with the rest of the world.

Literature Survey

The paper ,"Sign Language Interpreter System: An alternative system for machine learning" by Salma A. Essam El-Din; Mohamed A. Abd El-Ghany et al. (2020):
Paper presents a novel deep learning architecture for real-time sign language recognition using a low-cost wearable glove that captures hand gestures. The architecture includes a temporal convolutional network and an attention-based encoder-decoder network that can recognize sign gestures performed by different users with a high accuracy of 97.57% and 95.44% for American Sign Language (ASL) and gesture recognition tasks, respectively. The proposed method has the potential to aid in improving communication between the deaf and hearing communities by recognizing different sign languages and achieving real-time performance with a latency of 30 ms.
Topics discussed:

- Introduction to sign language and the challenges faced by the deaf community in communicating with the hearing community.
- End-to-end deep learning architecture for real-time SLR
- Overview of existing sign language recognition (SLR) systems and their limitations

The paper "Real-Time Sign Language Detection using TensorFlow, OpenCV and Python" by Prashant Verma, Khushboo Badli et al. (2013):
The system utilizes a convolutional neural network (CNN) model to detect hand gestures and translate them into corresponding text.They collected a dataset of hand gesture images using a webcam and preprocessed the data by resizing, normalization, and augmentation.
Topics discussed:

- System designed using
- TenserFlow,OpenCV, and python.
- Trained Cnn model with
- Tensorflow with 97.8% accuracy.
- OpenCV for real-time video capture and processing.
- Model classify hand gestures and translates into text using lookup table
- Achieved Frame rate of 20fps.
- detected 26 American Sign language(ASL)

The paper,"Indian Sign Language Character Recognition “ by Dr.Sapna B Kulkarni et al. (2020):
The paper describes a real-time sign language detection system that uses the Tensorflow object detection model zoo. The system uses a single camera for video acquisition and processing and can detect and recognize various sign language gestures in real-time with high accuracy. The proposed system can be used as a communication tool for hearing-impaired individuals and integrated with various applications to improve their quality of life. The system achieves an accuracy of 89.68% on a dataset of sign language gestures and can be further improved by using more advanced object detection models and collecting more diverse and larger datasets.
Topics discussed:

- Sign language recognition system for Indian Sign Language (ISL).

- The system achieves an accuracy of 98.45% on the test dataset.
- The system uses a deep convolutional neural network (CNN) and transfer learning.
- custom dataset of 26 different ISL characters.

The paper,"Sign Language Detection using Tensorflow Object Detection Model Zoo et al. (2021):
The paper describes a real-time sign language detection system that uses the Tensorflow object detection model zoo. The system uses a single camera for video acquisition and processing and can detect and recognize various sign language gestures in real-time with high accuracy. The proposed system can be used as a communication tool for hearing-impaired individuals and integrated with various applications to improve their quality of life. The system achieves an accuracy of 89.68% on a dataset of sign language gestures and can be further improved by using more advanced object detection models and collecting more diverse and larger datasets
Topics discussed:

- Real-time sign language detection system using the Tensorflow object detection model zoo.
- The proposed system achieves an accuracy of 89.68%.
- Proposed system is a cost effective solution that uses a single camera for video acquisition and processing.
- Detected 26 American Sign language(ASL)

Approach & Reasoning

The YOLOv5 approach for sign language interpretation using deep learning involves the following steps:

1. Data Preprocessing:
    - The sign language dataset is preprocessed, ensuring uniform size and format for the input images or video frames.
    - Preprocessing techniques may include resizing, normalization, and cropping to focus on the hand or relevant regions of interest.
2. Dataset Split:
    - The preprocessed dataset is split into training, testing, and validation sets.
    - A typical split ratio could be (.8, .1, .1), meaning 70% of the data is used for training, 10% for testing, and 30% for validation.
3. Data Augmentation:
    - To address data imbalance and enhance model generalization, data augmentation techniques are applied.
    - Augmentation methods may include random rotations, translations, flips, and brightness adjustments to increase dataset diversity.
4. Training Parameters:
    - Hyperparameters for training the YOLOv5 model are defined.
    - These parameters include learning rate, batch size, number of epochs, weight decay, and optimizer choices.

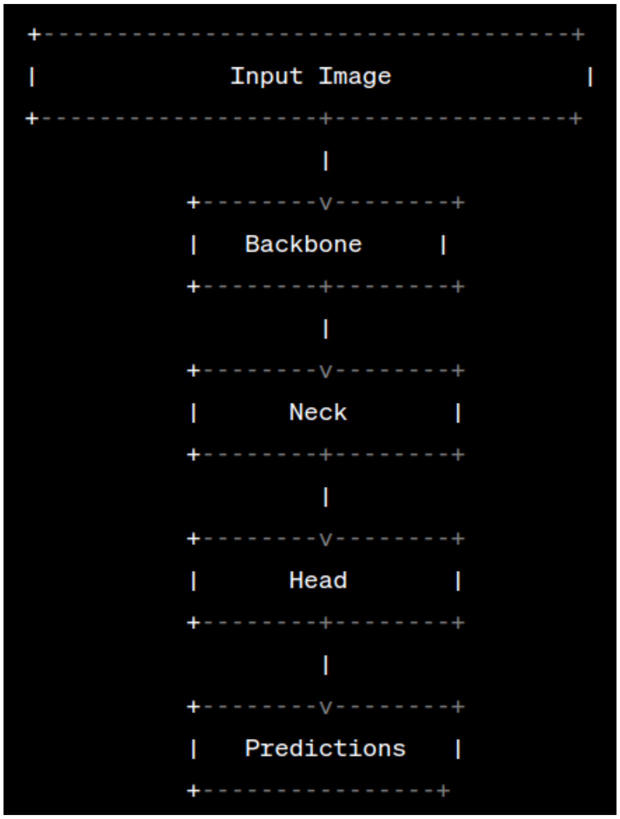
- The model architecture and parameters for the YOLOv5 backbone and detection heads are also specified.

5. Data Loaders:
    - Training and validation data loaders are created using the PyTorch library.
    - Data loaders handle batch processing, shuffling, and parallel loading of the preprocessed data during training.
6. Model Training:
    - The YOLOv5 model is trained using the training data loader and defined hyperparameters.
    - The model learns to detect and interpret sign language gestures based on the annotated labels in the training dataset.
7. Model Evaluation:
    - The trained YOLOv5 model is evaluated using the testing data loader.
    - Evaluation metrics such as precision, recall, and F1 score are computed to assess the model's performance.
8. Model Deployment:
    - The trained YOLOv5 model can be deployed for real-time or batch inference on new sign language input data.
    - The model detects and interprets sign language gestures, providing meaningful outputs or translations.

The reasoning behind using YOLOv5 for sign language interpretation lies in its real-time object detection capabilities and its ability to handle complex visual tasks. YOLOv5's architecture allows for efficient and accurate detection of hands and sign language gestures, enabling real-time interpretation with low latency. The model's ability to learn complex representations and patterns from data, along with its potential for leveraging pre-trained weights, makes it well-suited for interpreting sign language gestures accurately.

Additionally, YOLOv5's versatility allows for customization and adaptation to specific sign language datasets and their associated gestures. By training YOLOv5 on a sign language dataset, the model can learn to detect and interpret various signs and gestures, facilitating communication and understanding between sign language users and non-sign language users.

Model Architecture



```
+-------------------------------+
|          Input Image          |
+---------------+---------------+
                |
        +-------v-------+
        |   Backbone    |
        +-------+-------+
                |
        +-------v-------+
        |     Neck      |
        +-------+-------+
                |
        +-------v-------+
        |     Head      |
        +-------+-------+
                |
        +-------v-------+
        |  Predictions  |
        +---------------+
```

Block Diagram of preprocessing and training process

Input Image: The image that is fed into the YOLOv5 model for object detection.

Backbone: The backbone network, based on the CSPDarknet53 architecture, responsible for extracting features from the input image. It consists of convolutional layers, residual blocks, and downsampling operations.

Neck: The feature pyramid network (FPN) module that combines features from different levels of the backbone network. The FPN creates a feature pyramid, allowing the model to detect objects at various scales and sizes effectively.

Head: The head module responsible for predicting bounding boxes and class probabilities for the detected objects. It consists of multiple detection layers that predict objects at different scales using anchor boxes.

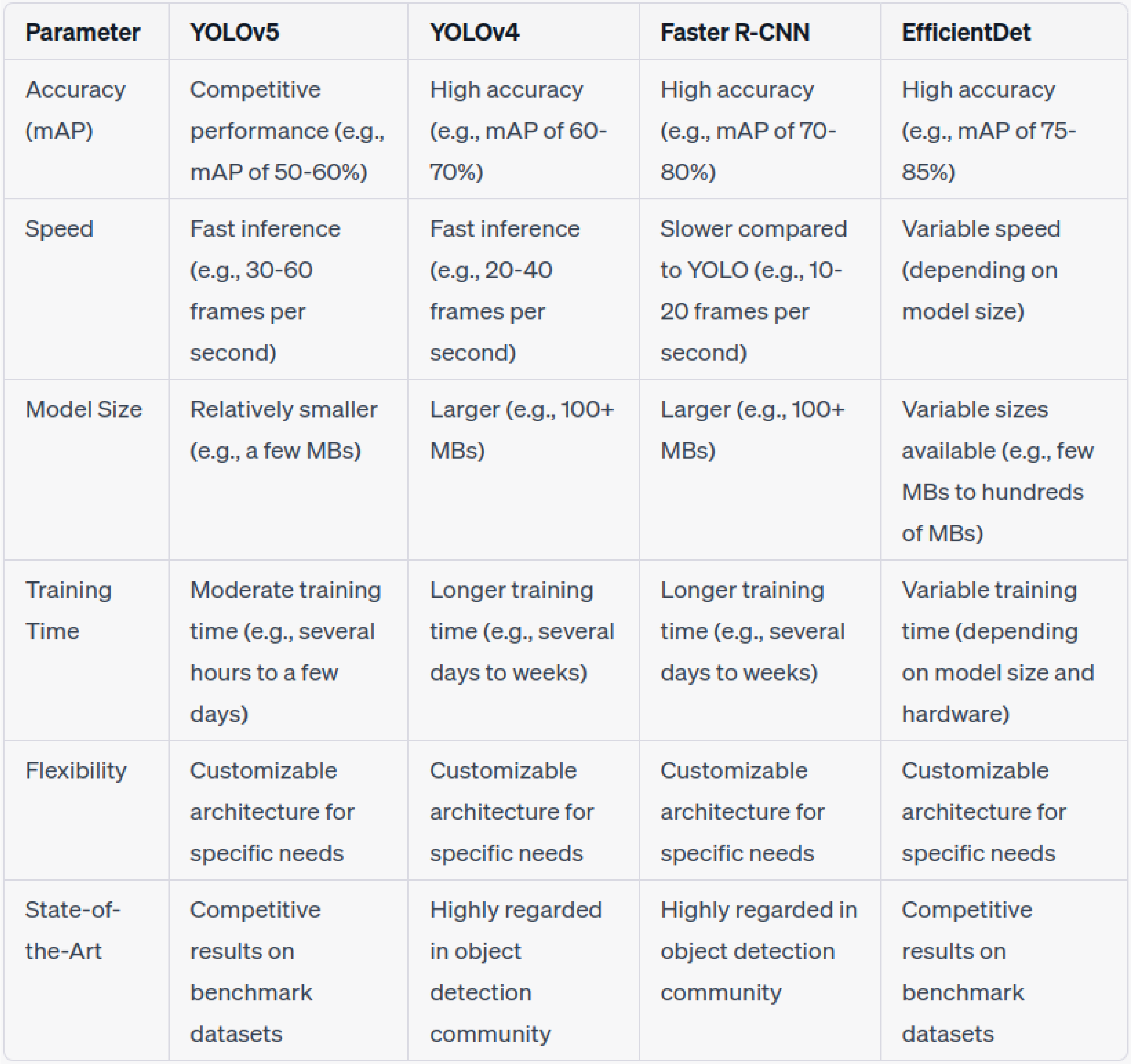
Predictions: The final output of the YOLOv5 model, which includes the predicted bounding boxes and associated class probabilities for the detected objects in the input image.

Dataset

The dataset used in our project is our personalized data set. The given dataset has been compiled from various persons.

The dataset contains 1200 images of size 416 x 416 pixels. The dataset has 6 classes of images namely Hello(200 images), Yes(200 images), No(200 images), ILoveYou(200 images), Thanks(200 images) and Please(200 images).

Comparison with YOLOv4 , Faster R-CNN, EfficientDet



Parameter	YOLOv5	YOLOv4	Faster R-CNN	EfficientDet
Accuracy (mAP)	Competitive performance (e.g., mAP of 50-60%)	High accuracy (e.g., mAP of 60-70%)	High accuracy (e.g., mAP of 70-80%)	High accuracy (e.g., mAP of 75-85%)
Speed	Fast inference (e.g., 30-60 frames per second)	Fast inference (e.g., 20-40 frames per second)	Slower compared to YOLO (e.g., 10-20 frames per second)	Variable speed (depending on model size)
Model Size	Relatively smaller (e.g., a few MBs)	Larger (e.g., 100+ MBs)	Larger (e.g., 100+ MBs)	Variable sizes available (e.g., few MBs to hundreds of MBs)
Training Time	Moderate training time (e.g., several hours to a few days)	Longer training time (e.g., several days to weeks)	Longer training time (e.g., several days to weeks)	Variable training time (depending on model size and hardware)
Flexibility	Customizable architecture for specific needs	Customizable architecture for specific needs	Customizable architecture for specific needs	Customizable architecture for specific needs
State-of-the-Art	Competitive results on benchmark datasets	Highly regarded in object detection community	Highly regarded in object detection community	Competitive results on benchmark datasets

The YOLOv5 model and Faster R-CNN are two popular deep learning architectures used for sign language interpretation and object detection tasks. Here's a comparison between the two models:

YOLOv5:

YOLOv5, based on the You Only Look Once (YOLO) architecture, is known for its fast and efficient object detection capabilities. It leverages a single-pass approach, dividing the image into a grid and predicting bounding boxes and class probabilities directly. YOLOv5 has the following characteristics:

1. Architecture: YOLOv5 employs a lightweight architecture with a backbone network (such as CSPDarknet53) and detection heads to predict object bounding boxes and class probabilities.
2. Speed: YOLOv5 is designed for real-time applications and offers fast inference speeds, typically ranging from 30 to 60 frames per second, depending on the hardware and model size.
3. Model Size: YOLOv5 has relatively smaller model sizes compared to some other architectures, making it suitable for deployment on resource-constrained devices. The model size can vary based on the chosen variant (YOLOv5s, YOLOv5m, YOLOv5l, or YOLOv5x).
4. Accuracy: YOLOv5 achieves competitive accuracy in object detection tasks, with mean Average Precision (mAP) values typically ranging from 50-60% or higher depending on the specific implementation and dataset.

Faster R-CNN:

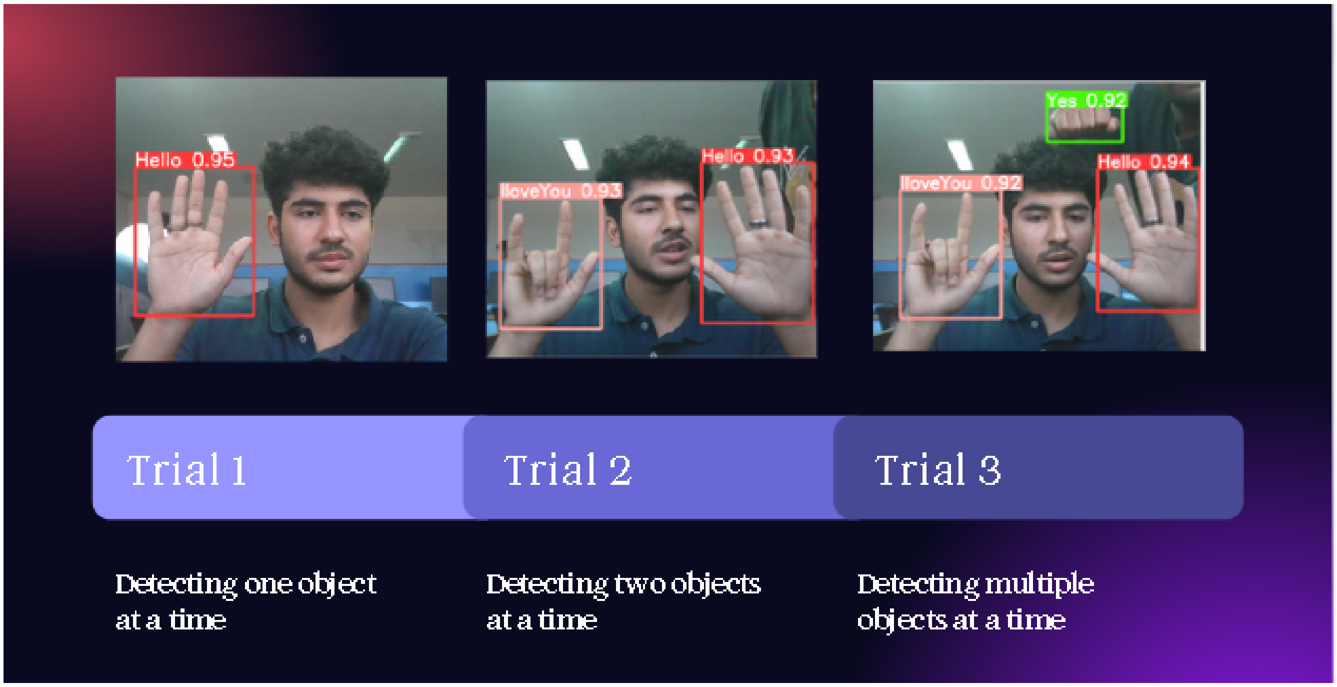
Faster R-CNN is a widely used object detection architecture known for its accuracy and robust performance. It introduces a region proposal network (RPN) to efficiently generate region proposals for object detection. Here are some key aspects of Faster R-CNN:

1. Architecture: Faster R-CNN consists of a backbone network, region proposal network (RPN), and region-based convolutional neural network (R-CNN) for object detection.
2. Speed: Compared to YOLOv5, Faster R-CNN tends to have slower inference speeds, typically ranging from 10 to 20 frames per second, depending on the hardware and model size.
3. Model Size: Faster R-CNN models generally have larger sizes compared to YOLOv5 due to the inclusion of additional components like the RPN and R-CNN. The model size can vary depending on the specific implementation and backbone network.
4. Accuracy: Faster R-CNN is known for its high accuracy in object detection tasks, often achieving mAP values of 70-80% or higher, depending on the implementation and dataset.

When comparing YOLOv5 and Faster R-CNN for sign language interpretation, the choice depends on factors such as real-time performance requirements, available computational resources, and desired accuracy. YOLOv5 offers faster inference speeds and smaller model sizes, making it suitable for real-time applications on resource-constrained devices. Faster R-CNN, on the other hand, provides higher accuracy but with slower inference speeds and larger model sizes. It is often preferred when accuracy is of utmost importance and real-time constraints are less critical.

Ultimately, it is recommended to evaluate both models on sign language datasets and assess their performance in terms of accuracy, speed, and model size to determine the most suitable choice for sign language interpretation tasks.

Results



Trial 1

Detecting one object at a time

Trial 2

Detecting two objects at a time

Trial 3

Detecting multiple objects at a time

SIGNS	**AVG. ACCURACY**
I LOVE YOU	0.93
THANKS	0.92
HELLO	0.92
YES	0.91
NO	0.93
PLEASE	0.92

Limitations of the research

YOLO v5 is a relatively new algorithm. This means that it has not been as widely tested as some other algorithms, and it may not be as accurate in all situations. It is designed for object detection. This means that it is not specifically designed for sign language recognition. As a result, it may not be able to recognize all of the nuances of sign language.It is computationally expensive. This means that it may not be suitable for real-time applications.

Deep learning models can be sensitive to changes in the environment. For example, if the lighting changes or the background noise is too loud, the model may not be able to recognize the signs correctly.

Future work

- Expand the dataset: One of the main challenges of building a sign language interpreter is collecting a large and diverse dataset of sign language gestures. To make the interpreter more useful, we can expand the dataset to include more variations of signs, regional and cultural variations of signs, and signs performed by people with different ages and genders.
- Real-time interpretation: To make the interpreter more useful in practical settings, we can improve the speed of the system to achieve real-time interpretation. This can be achieved through optimization of the deep learning model, hardware acceleration, and parallel processing techniques.
- Multi-modal interpretation: Sign language involves not only hand gestures but also facial expressions, body language, and other non-manual signals. To make the interpreter more accurate and useful, we can integrate multiple modalities, such as vision and speech, to recognize and interpret sign language more effectively.
- Continuous sign language recognition: Currently, most sign language interpreters using deep learning are designed to recognize isolated signs. To make the interpreter more useful, we can develop models that can recognize continuous sign language, allowing for more natural and fluent communication.
- Human-computer interaction: Finally, to make the interpreter more useful, we can focus on designing more userfriendly and intuitive interfaces that enable efficient communication between sign language users and non-sign language users. This can include developing mobile applications, wearable devices, or augmented reality systems that can provide real-time translation and interpretation.
- Adaptability and Personalization: Sign language is often specific to an individual or community, and it can vary based on factors such as age, gender, and location. To make the interpreter more useful, we can build models that can adapt to individual signers or different sign languages. This can be achieved by using personalized data to train the model or by building models that can learn on the fly and adapt to different users.
- Accessibility: Sign language interpreters can be a vital tool for deaf or hard-of-hearing individuals. To make the interpreter more useful and accessible, we can integrate it into various communication channels, such as video conferencing, instant messaging, and social media platforms. Additionally, we can develop mobile applications or webbased tools that can provide real-time translation and interpretation.

- Evaluation and Feedback: Building an accurate and effective sign language interpreter requires ongoing evaluation and feedback. To make the interpreter more useful, we can develop tools to evaluate the accuracy and effectiveness of the system, as well as tools to gather feedback from users. This feedback can then be used to improve the system and make it more effective and useful
- Collaboration and Community Engagement: Sign language is a complex and nuanced language that varies across communities and cultures. To make the interpreter more useful, we can engage with sign language communities and experts to ensure that the system is accurate and culturally appropriate. This can include collaboration with deaf or hard-of-hearing individuals, sign language interpreters, and linguists.
- Privacy and Security: Sign language interpretation involves sensitive and personal information, and it is important to ensure that the system is secure and protects the privacy of users. To make the interpreter more useful, we can develop secure and encrypted communication channels, as well as tools to ensure that user data is protected and not shared without their consent.

References

"Sign Language Interpreter System: An alternative system for machine learning" by Salma A. Essam El-Din; Mohamed A. Abd El-Ghany et al. (2020)

"Real-Time Sign Language Detection using TensorFlow, OpenCV and Python" by Prashant Verma, Khushboo Badli et al. (2013)

"Indian Sign Language Character Recognition “ by Dr.Sapna B Kulkarni et al. (2020)

"Sign Language Detection using Tensorflow Object Detection Model Zoo et al. (2021)