American Sign Language Recognition and Translation using Deep Learning frameworks

Abstract: *This paper explores the realm of American Sign Language (ASL) recognition and translation through extensive research and implementation. A bespoke dataset consisting of 80 labels was meticulously curated for the food industry, training and evaluation was done utilizing a range of deep learning models such as LSTM, LRCN, Bi-LRCN, Bi-LSTM, and object detection models YOLO v5 and YOLO v8. The results obtained from these models were rigorously analyzed and compared, providing valuable insights into their efficacy and limitations in ASL recognition and translation tasks. Beyond technical advancements, this research carries significant social impact. Effective ASL recognition and translation technology can break down barriers for the deaf and hard of hearing community, facilitating communication and creating a more inclusive society. Improved accessibility in education, employment, and daily interactions can enhance opportunities and foster equal participation. By contributing to the advancement of ASL technology, this study helps bridge the communication gap between individuals who use ASL and those who rely on spoken language. The findings and recommendations presented in this paper serve as a valuable resource for researchers, developers, and practitioners working towards improving ASL recognition and translation systems. Ultimately, this research contributes to the larger goal of promoting inclusivity, equal access, and empowerment for individuals in the deaf and hard of hearing community.*

Keywords: *American Sign Language recognition, Long Short Term Memory, Deep Learning, YOLO*

1. Introduction

The fundamental building block of human contact, communication enables the sharing of thoughts, feelings, and information. Languages in their spoken and written forms have long been regarded as essential channels for this process. A lively and colorful form of communication known as sign language has, however, been marginalized and unappreciated in this linguistic tapestry. More than 300 different sign languages are used by signers to communicate with one another and have been recognized by the United Nations (UN). The UN asserts that sign languages are completely natural languages that differ structurally from spoken languages. Individuals with hearing impairments are the main beneficiaries of sign language, a visual-spatial language that uses complex hand gestures, emotive facial expressions, and intentional body motions. With all of its syntactic patterns, grammatical norms, and lexical resources, it makes up a complete and intrinsic language system. Despite its intrinsic linguistic diversity, sign language has long been marginalized in favor of its spoken counterparts, leaving the deaf and hard-of-hearing communities with little access to resources. The genesis of sign language can be traced back to antiquity, wherein visual communication emerged as a means to surmount the barriers imposed by auditory impediments. Throughout the annals of history, distinct sign languages have arisen within different regions and communities, exemplified by American Sign Language (ASL), British Sign Language (BSL), Australian Sign Language (Auslan), and Indian Sign Language (ISL). These unique linguistic modalities have organically evolved within their respective cultural milieus, serving as integral components of the deaf communities' collective identity and cultural heritage. Despite the great diversity of sign languages that have developed throughout numerous cultures and geographical areas, there still exists a pitiful gap in the lack of a comprehensive tool that can accurately categorize, identify, and translate these sign languages into written text. Between native signers and non-native signers, there has been a separation. The urgent importance of AI is made clear when it comes to sign languages, which exhibit a wide range of regional and community-specific variations. AI has the potential to close the current gap by offering cutting-edge methods for the identification, classification, and transcription of sign languages into written text.

According to the World Health Organization (WHO) there are over 1.5 billion people globally who live with hearing loss and this number could rise to over 2.5 billion by 2050. Over 5% of the world's population – or 430 million people – require rehabilitation to address their disabling hearing loss (432 million adults and 34 million children). It is estimated that by 2050 over 700 million people – or 1 in every 10 people – will have disabling hearing loss [1]. There are over 300 different sign languages across the globe and sign languages such as Indian Sign Language (ISL), American Sign Language (ASL), and numerous other diverse sign languages are utilized by millions of individuals worldwide, constituting a significant portion of the global population with hearing impairments. The distribution can be understood with the help of Figure 1, statistics of which are sourced from [2].

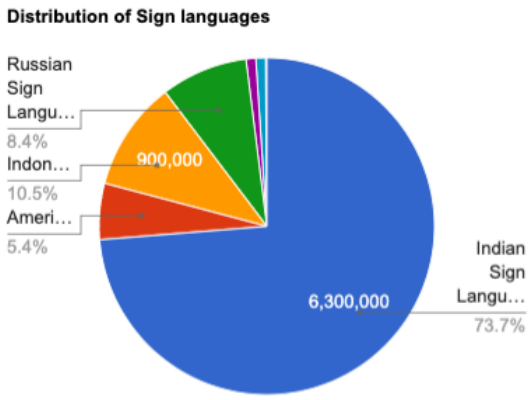

Figure 1 Distribution of Sign Languages

The salience of sign language resides in its transformative potential to bridge the communication chasm between the deaf and the hearing realms, thereby facilitating meaningful interactions and enabling equitable participation across multifarious spheres of life. By due recognition and embracing sign language as a bonafide linguistic system, society can dismantle impediments and embolden individuals with hearing impairments to express themselves holistically. The acknowledgment of sign language bolsters the development of inclusive educational frameworks, endowing deaf students with the tools to access quality education and pursue their academic aspirations unfettered. Furthermore, sign language assumes momentous implications within the healthcare and professional domains. Communication barriers often engender disparities in healthcare outcomes for deaf individuals, as they grapple with impediments in accessing medical services and acquiring accurate information. By integrating sign language interpreters and fostering cognizance regarding the linguistic entitlements of the deaf community, healthcare providers can ensure efficacious communication and deliver parity in healthcare provision. Moreover, sign language has made notable strides within the realm of technology. Advancements in sign language recognition and translation systems, fortified by machine learning and computer vision methodologies, have emerged as propitious tools to facilitate seamless communication between sign language users and non-sign language users. These technological breakthroughs hold immense promise in augmenting accessibility, engendering social inclusivity, and nurturing egalitarian opportunities for individuals with hearing impairments.

This paper aims to delve into the historical backdrop and underscore the paramount significance of sign language as an invaluable tool for fostering inclusivity and empowering marginalized communities and thereafter using Deep Learning techniques to predict the signed actions, thereby helping us to bridge the gap between non-signers and native signers.

The contribution of the study includes:

- ✔ Develop a comprehensive dataset: Collect a large and diverse dataset of sign language gestures and expressions to train and evaluate the models. Include various hand shapes, movements, and facial expressions to capture the complexity of sign language.
- ✔ Pre-process the sign language data: Explore pre-processing techniques such as image resizing, normalization, augmentation, and noise reduction to enhance the quality and variability of the sign language images. Evaluate the impact of different pre-processing methods on the model's performance.
- ✔ Comparative analysis of deep learning algorithms: Implement and compare multiple deep learning algorithms, such as LSTM [3], LRCN [4], BiLRCN, BiLSTM [5] for sign language detection and classification. Evaluate their accuracy, computational efficiency, and suitability for real-time applications.
- ✔ Implement YOLO v5 [6] and v8 [7] architectures: Adapt and modify the YOLO (You Only Look Once) object detection architectures, namely YOLO v5 and v8, for sign language detection and classification. Investigate the effectiveness of both versions and compare their performance.
- ✔ Evaluate and compare performance: Measure the detection and classification performance of LSTM[3], LRCN [4], BiLSTM, BiLRCN, YOLO v5 and v8 models using standard evaluation metrics such as mean Average Precision (mAP), Precision, Recall, Accuracy and Loss. Compare their accuracy, speed, and robustness to determine which architecture performs better for sign language detection and classification tasks.

- ✔ Use Natural Language Processing for real world applications: In order to convert the classified labels into text we will use natural language processing to convert it into sentences or paragraphs and then further translate into our desired language using suitable methods

2. Related Work

Sign Language Recognition is a fundamental interpretation task. To gain a deeper understanding of the related works in this domain, we perused a variety of research papers and gained insights to identify the existing research gaps. Sign Language Recognition [8] explores CNN architectures along with Image Processing for hand gesture recognition and production of corresponding text. The system introduced automatic SLR and translation of static hand gestures, 26 English alphabets (A-Z) and 10 digits (0-9). They trained an horizontal voting ensemble of 3 architectures , namely LeNet5, MobileNetV2 and a self-designed network consisting of 3 Conv plus 2 Dense layers. While [9] , explored end-to-end Sign Language Translation using CNN based spatial embedding, RNN-HMM hybrids [10], and attention-based encoder-decoder networks to align, recognize and translate sign videos to text. Along similar lines, towards achieving SLR, [11] proposed Gaussian HMM trained on appearance-based features from original data and from a multilayer perceptron (MLP), plus incorporating PCA. Another approach depicted in [12] , presented the use of CNNs with a number of custom reformation (RF) layers proven to enhance the accuracy for detecting ASL fingerspelling data. Maruyama proposed a multi-stream neural network WSLR framework, introducing a stream with local region images and another with skeletal information by extending the I3D network, enabling extraction of global appearance features along with fine-grained hand gestures in quick motion and the positional relationships [13]. The framework consists of three streams: Base stream for global and optical flow information, Local Image stream for the hand shapes and facial expression, and Skeleton stream for the positional relationship among the body and both hands. Model was trained on dynamic WLASL [14] and MS-ASL [15] datasets, using YOLOv3 [16] for object detection. Computer Vision-based hand gesture recognition using CNN is proposed to convert the recognized input into speech, which enables one-way communication [17]. To enable two-way communication, text to sign language or fingerspelling conversion is proposed. They use the output text of the model which becomes the string input for the text-sign language conversion system. The string is broken down into elements, and the sign of each element is fetched from a local directory containing images for all ASL alphabets. Although this paper proposes a novel approach, the image dataset does not enable dynamic gesture recognition which is crucial for sign language recognition tasks. Moreover, the text to sign language alphabet conversion isn't a very efficient task since reading the finger spellings word by word doesn't make sense as sign languages are continuously represented.

They propose a framework to convert signed ASL words to grammatically sound English sentences, starting from recognition of signed input as words using a CNN model with 89.2% accuracy, followed by tokenization and POS tagging [18]. The sentences obey the self-defined rules in the form of CFG represented as trees called Lexicalized Tree Adjoining Grammar (LTAG). This is then used in the LALR parser along with some essential word insertions when needed. The parsed output varies every time and is somewhat meaningful, which is checked for errors using LanguageTool. [19] present an architecture to recognise ASL signs by first identifying hand movements using MediaPipe [20], a tool which examines hands in a video stream. Further, LSTM is used to predict signs. The research focused on minimalistic background, thus disabling real-time unrestricted viability of the proposed model. They suggest gradient masking to combat contrast issues and detect clear objects. [21] provides a clear contrast between CNN and LSTM for British Sign Language translation into text, proving CNN to be the winner with 97.4% accuracy as opposed to LSTM's 48.7%. LSTM and GRU model is proposed to recognize signs from isolated Indian Sign Language (ISL) video frames [22]. They experimented with six different sequential combinations of LSTM and GRU with their custom dataset, IISL2020, recorded in natural yet isolated conditions. They finally proposed a model consisting of a single layer of LSTM followed by GRU which achieved around 97% accuracy over 11 different signs. They observed that increasing the number of layers in the LSTM and GRU may guarantee higher accuracy. Their current model only works well with isolated signs and only for ISL, where the dataset itself is insufficient. A comprehensive evaluation of deep models and optimizers for Indian Sign Language recognition [23] performs a systematic evaluation and statistical analysis of pre-trained deep models, gradient-based optimizers and optimization hyperparameters for static ISL recognition. They also proposed a three-layered CNN model which is built and trained from scratch. Among the pre-trained models, ResNet152V2 was the best performer with a promising accuracy of 96.2% on numerals and 90.8% on alphabets of the same dataset. They highlight Adam to be the most significant optimiser even though it took the highest time to train. They incorporate dataset augmentation techniques like rotation, blurring, horizontal flipping, and adding random noise to the original images

of ISL dataset. The metrics used were Accuracy, Precision, Recall, F1-score, Coefficient of Variation (CV) and a loss function of categorical cross entropy. The mentioned features took into account the impact of batch size, tuning of hyperparameters, and evaluation of tuned optimizers and CNN models. Their proposed techniques were aimed for static ISL dataset only , which proves to be less useful in real-world scenarios which require dynamic gestures to be recognised.

Golda introduced gesture recognition for ISL by using wearable sensor based gloves, consisting of five flex sensors, one pressure sensor, one Bluetooth module connected to microcontrollers of both sides of each hand [24]. The sensor data is fed to the CNN with 18 input neurons for each sensor value, which then performs word classification. The design is tested on 50 signs, achieving 97.8% accuracy. To tackle word ordering and alignment corresponding to visual signs, [25] an encoder-decoder hierarchical LSTM model to translate sequence of CSL video frames to sequence of text words. They incorporate 3D CNN to explore spatio-temporal cues, and grouping visemes by a technique called online key clip mining, along with a temporal attention weighting mechanism. After 3D CNN and HLSTM, 2 LSTM layers are introduced to translate semantics soundly while preserving visual content. The last two layers shorten encoding time, ensuring less computation with high nonlinearity. The model achieves stellar performance for seen sentences, but meager adaptability for unseen sentences due to unbalanced data distribution and word obscurity. Sign Language Recognition [26] translates the signer's conversation into tokens, whereas Sign Language Identification [26] targets to identify the signer language. Static and dynamic sign language datasets from various corpora were considered, with contents including numerical, alphabets, words and sentences from various sign languages. The fundamental preprocessing steps incorporated include skin-detection, ROI, Image resizing, image segmentation, feature extraction and tracking. Devices like Glove-based with built-in sensors, Vision-based, and Virtual button approach are accounted for, and a number of ML techniques are compared with CNN, proving the latter to be a clear winner. Getting rid of gloves and sensors proves to be more beneficial and does not restrict user interaction with the system. Bahia reviews data acquisition techniques i.e. Glove-based, Kinect-based, Vision-based, etc. where Vision-based acquisition proves to be the better performer and easy to use [26]. A taxonomy to represent the modern research is divided into three levels: Elementary - Recognition of sign characters, Advanced - Recognition of sign words, Professional - Sentence interpretation. Issues with SL research are accounted for which mainly include the availability of a benchmarked dataset for ISL, high computational costs, difference in background conditions, grammar detection, signer variation, etc. Table 1 encompasses all the referred papers and provides a detailed comparison between their approaches, models, recognition levels & accuracies.

Table 1 Comparison of works on Sign Language Recognition

Reference	Dataset	Recognition Level	Model	Accuracy
[8]	Public ASL	Alphabets and Numerals	Ensemble of LeNet5, MobileNetV2, 3 Conv + 2 Dense layers	99.8%
[13]	WLASL, MSASL	Words	Tri-stream (Base, Local, Skeletal) Inflated 3D ConvNet framework + YOLOv3	WLASL- 81.38% MSASL - 83.86%
[17]	Public ASL	Alphabets	Modified VGG16	99.62%
[22]	IISL2020	Words	LSTM + GRU	97%
[23]	Public ISL	Alphabets and Numerals	3 layered CNN	Alphabets - 97.6% Digits - 99%
[25]	CSL	Sentences	3D CNN + HLSTM + LSTM	92.4% (Precision)
[24]	ISL	Words	5 layered CNN	97.8%
[19]	ASL	Words	LSTM	99.35%
[28]	ASL	Alphabets	Single Shot Multi Box Detection (SSD) + InceptionV3 + SVM (Cross Validation technique)	99.9%
[29]	ASL	Words	HOG + SVM	NA

[30]	ASL	Alphabets + Space + Nothing + Delete	CNN	98.69%

3. Proposed work

Sign language is majorly used by deaf or hard of hearing people. There are more than 300 sign languages around the globe, spoken by more than 72 million deaf or hard of hearing people worldwide [31]. As the main objective is to bridge the gap of communication that is prevalent between sign language speakers and people who are not acquainted with sign language, translation has to happen at the word level, rather than spelling each letter of a word. Sign language at the word level promises instant and real time understanding of what the person is trying to communicate. This significantly enhances the speed and efficiency of communication, especially in situations where quick translation is required. Whereas fingerspelling each letter would consume a lot of time and would not help in communicating efficiently. Word level sign language also handles complex linguistic structures and grammar present in sign languages, whereas fingerspelling primarily focuses on representing individual letters which do not convey the grammatical structure of the sentence. Hence, the research was inclined towards word level sign language interpretation and translation. During the course of the research, the authors focussed on developing a model built to translate and recognize words relevant to the food service industry. Upon thorough research and analysis, the authors found that a large percentage of deaf people are not employed in the labor force and employment rates for the same community have not risen from 2008 to 2017 [32]. This strongly points at the obstacles they face at the workplace. Problems due to communication also exist in the food service industry. Deaf or hard of speaking communities find it difficult to place, customize or communicate their orders at a regular drive through or sit down restaurants. If a person belonging to such communities suffers from any allergies - like lactose, peanuts, poultry etc. it becomes difficult for them to mention the same to the food-server. Taking note of this ever increasing gap of communication and obstacles that are common in lives of deaf or hard of speaking communities the authors decided to curate a bespoke dataset, containing words pertaining to the food service industry like; “burger”, “chicken”, “cheese”, “manager”, “lactose”, “allergy”, “sauce”, “water”, “soda” etc. from various well known data sources like WLASL dataset, YouTube. A total of 80 labels were collected spanning the food service industry for the research.

Initially we worked with Long Short-Term Memory [3] which is a specialized form of Recurrent Neural Network(RNN) with deep learning. This choice was made to surmount limitations which we found in traditional RNNs, as it allowed us to effectively capture lengthy temporal connections in a sequential data. Our dataset consists of videos which can be of varying lengths, an added advantage of LSTM lies in its adaptability to process sequences of varying lengths. Moreover the LSTM network has the ability to capture and comprehend protracted temporal dependencies, a crucial trait for tasks involving sequences with temporal lags or dependencies. LSTMs afford the luxury of selective focus on pertinent information, disregard superfluous details, and sustain the retention of crucial insights across protracted sequences.

If we consider a LSTM cell, there are three different gates as shown in Equation 1 - 6 : a forget gate (f_t), an input gate (i_t) and an output gate (o_t). The forget gate decides which information needs to be forwarded or needs attention which makes the model retain relevant information and avoid accumulation of noise. The input gate controls the flow of new information into the cell states which allows the model to update its memory in a controlled manner. The output gate regulates the flow of information from the cell state to output, influencing the relevance and amount of information passed onto the next time step, ensuring that the model focuses on important information and produces accurate outputs. Figure 2 depicts the architecture of an LSTM cell state.

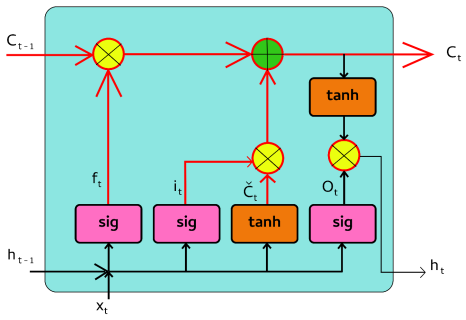

Figure 2 LSTM Cell

$$c'_{<t>} = \tanh (W_c [a_{<t-1>}, x_{<t>}] + b_c) \quad (1)$$

$$T_u = g(W_u [a_{<t-1>}, x_{<t>}] + b_u) \quad (2)$$

$$T_f = g(W_f [a_{<t-1>}, x_{<t>}] + b_f) \quad (3)$$

$$T_o = g(W_o [a_{<t-1>}, x_{<t>}] + b_o) \quad (4)$$

$$c_{<t>} = T_u * c'_{<t>} + T_f * c_{<t-1>} \quad (5)$$

$$a_{<t>} = T_o * \tanh c_{<t>} \quad (6)$$

While working on LSTM we realized that it was not capable of capturing spatial and temporal features from the input video, so to counter those drawbacks we worked on Long-term Recurrent Convolutional Network [4], which is an innovative deep learning architecture tailored for video captioning, where it generates descriptive captions for videos. LRCN uniquely combines convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to tackle the challenge. The LRCN model addresses the challenge of video captioning by integrating CNNs for visual feature extraction and RNNs for modeling temporal dependencies. Figure 3 depicts the architecture of an LRCN unit. As our dataset consists of videos we needed a model which could capture both visual context and time-based dependencies, LRCN tackles these challenges as it have CNN component that process individual frames, and an RNN component, often based on LSTM network, that generates captions based on sequential dynamics of frames.

The advantages of LRCN over traditional LSTM networks in video captioning stems from its dual integration of CNNs and RNNs. This integration allows LRCN to effectively capture both visual nuances and temporal patterns. The model aims to minimize the difference between generated captions and ground truth captions through techniques like maximum likelihood estimation. This would result in generation of contextually accurate descriptions for videos and therefore will improve the performance in video captioning tasks.

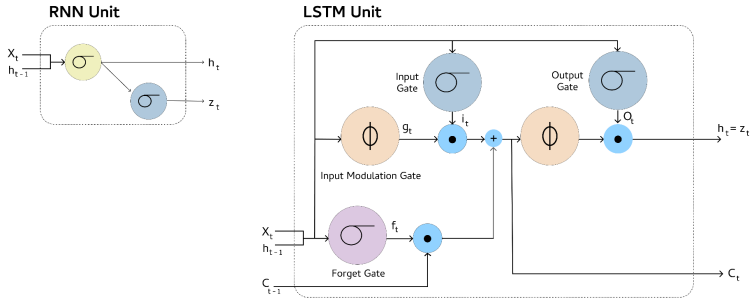

Figure 3 LRCN unit

Further during the course of the research, there was a need to retain relevant information over a longer period of time to capture the English language's contextual understanding faster and better. BiLSTM (Bidirectional Long Short-Term Memory) is an extension of the LSTM architecture that processes sequential data in both forward and backward directions simultaneously. It employs two separate LSTM layers to capture past and future context, enhancing the model's understanding of temporal dynamics. By combining outputs from both directions. The bidirectional approach helps BiLSTMs to retain relevant information over longer sequences and has demonstrated superior performance in various applications, leading to more accurate predictions and classifications in tasks involving sequential data. Figure 4 depicts the architecture of the BiLSTM network. In this architecture we can notice that the network has access to both past and future input features.

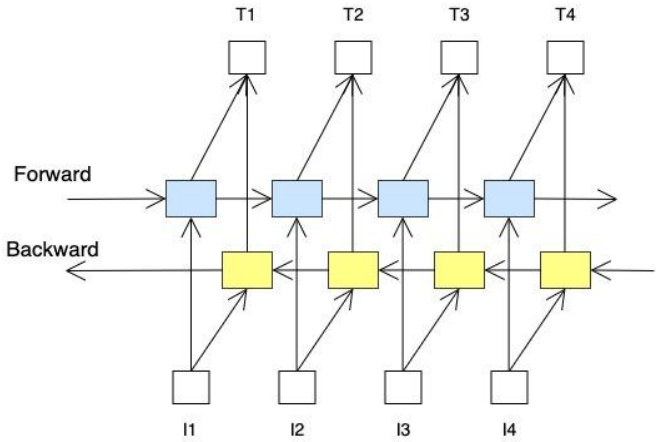

Figure 4 BiLSTM Network

Where I1,I2,I3,I4 indicate the inputs in an sequential order and T1,T2,T3,T4 indicate the predicted or tagged sequences. 'Forward' represents the forward LSTM network and 'Backward' represents the backward LSTM network. Equations 7 - 9 shows the mathematical calculations that take place during the Forward pass in BiLSTM, where i_t is the input gate, f_t is the forget gate and o_t is the output gate. Equation 7 describes g_t which represents the cell state candidate and Equation 8 describes c_t , the cell state. Throughout the set of equations, $\odot$ represents element-wise multiplication, σ is the sigmoid activation function, and tanh is the hyperbolic tangent activation function. W and U are weight matrices, and b is the bias vector associated with each gate and candidate value.

$$g_t = \tanh(W_g * x_t + U_g * h_{t-1} + b_g) \quad (7)$$

$$c_t = \sigma(W_f * x_t + U_f * h_{t-1} + b_f) \odot c_{t-1} + \sigma(W_i * x_t + U_i * h_{t-1} + b_i) \odot g_t \quad (8)$$

$$h_t = \sigma(W_o * x_t + U_o * h_{t-1} + b_o) \odot \tanh(c_t) \quad (9)$$

Equations 10 - 15 shows the mathematical calculations that take place during the Backward pass in BiLSTM, where i_t is the input gate, f_t is the forget gate and o_t is the output gate. The superscript "b" indicates the backward direction.

$$i_t^b = \sigma(W_i^b * x_t^b + U_i^b * h_{t-1}^b + b_i^b) \quad (10)$$

$$f_t^b = \sigma(W_f^b * x_t^b + U_f^b * h_{t-1}^b + b_f^b) \quad (11)$$

$$o_t^b = \sigma(W_o^b * x_t^b + U_o^b * h_{t-1}^b + b_o^b) \quad (12)$$

$$g_t^b = \tanh(W_g^b * x_t^b + U_g^b * h_{t-1}^b + b_g^b) \quad (13)$$

$$c_t^b = f_t^b \odot c_{t-1}^b + i_t^b \odot g_t^b \quad (14)$$

$$h_t^b = o_t^b \odot \tanh(c_t^b) \quad (15)$$

The Bidirectional Long-term Recurrent Convolutional Network (Bidirectional LRCN) was later incorporated as part of our research, which is an extension of the LRCN model, designed to further enhance video understanding tasks. While the traditional LRCN processes video frames sequentially in one direction, the Bidirectional LRCN processes them in both forward and backward directions simultaneously. By incorporating bidirectional processing, the model can capture not only past contextual information but also future context, providing a more comprehensive understanding of the temporal dynamics within videos. This dual-processing capability is achieved by adding an additional set of LSTM units that read the video frames in reverse order. Bidirectional LRCN has shown improved performance in various video-related tasks, such as action recognition, video captioning, and video summarization, by leveraging bidirectional context and effectively capturing temporal dependencies in videos.

Through the progression of our study, the authors found the need to perform object detection in order to counter noisy input. YOLO (You Only Look Once) version 5 [6] is an advanced object detection algorithm that has gained significant popularity in the fields of computer vision and deep learning very recently. It aims to improve both speed and accuracy in object detection tasks, building upon the success of its predecessors.The key advancement in YOLO version 5 lies in its streamlined architecture, which allows for faster and more efficient object detection. Unlike other algorithms that rely on region proposal methods, YOLO v5 utilizes a single neural network to simultaneously predict bounding boxes and class probabilities for multiple objects in an image, enabling real-time detection.

A notable improvement in YOLO v5 is the introduction of "PANet" (Path Aggregation Network), a novel detection head that facilitates feature fusion at different scales. This multi-scale approach enhances the algorithm's ability to detect objects of various sizes, especially smaller objects that were challenging to detect in previous versions. YOLO v5 is known for its real-time object detection capabilities and it can achieve FPS ranging from 15-60 on modern GPUs.

An advanced algorithm for object detection YOLO (You Only Look Once) v8 [7] was also incorporated in the study that has gained considerable attention in computer vision. It builds on the success of its previous versions, introducing improvements for more accurate and faster object detection.The primary objective of YOLO v8 is real-time object detection and classification. It divides an input image into a grid, with each grid cell predicting fixed bounding boxes, class probabilities, and confidence scores.

Noteworthy is its multi-scale approach for detecting objects of varying sizes, achieved through feature pyramids that merge feature maps of different resolutions. YOLO v8 employs a potent Darkenet architecture-based convolutional neural network (CNN) to extract pertinent features, enabling accurate predictions of bounding boxes and class probabilities, If we compare YOLO v5 and YOLO v8, YOLO v8 excels due to its refined multi-scale detection, advanced architectural choices, and optimization techniques, contributing to superior object localization and classification.

IV. Dataset

The dataset was carefully curated and manually created specifically for the purpose of research. The process of data acquisition involved engaging ASL signers proficient in the Food Domain to perform the actions associated with each label. Multiple signers were recorded to introduce variations in signing style, hand shapes, and gestures, ensuring a diverse and representative dataset.

To enhance the dataset's versatility and improve the model's robustness, augmentation techniques were applied to the original videos. These techniques included random rotation, scaling, translation, and other transformations.

Augmentation introduces variability in the appearance and pose of the signers, allowing the model to generalize better and recognize ASL gestures under different conditions.

Before training the ASL detection and recognition model, pre-processing steps were applied to the dataset. This included standardizing the video resolution, frame rate, and format to ensure consistency across the dataset. Additionally, noise reduction techniques may have been applied to minimize background noise or artifacts present in the videos, further enhancing the dataset's quality. Thereafter came the task of annotations.

To annotate frames, we used YOLO V8, an annotator would typically draw bounding boxes around the region of interest corresponding to the ASL gesture in each frame. The annotator would then label the bounding box with the appropriate class or category, representing the specific ASL gesture being performed. This process is repeated for all frames in the dataset, ensuring that each instance of an ASL gesture is accurately annotated.

YOLO v8 is a state-of-the-art deep learning architecture specifically designed for object detection tasks. It employs a single neural network to simultaneously predict bounding boxes and classify objects within an image. The use of YOLO v8 for labeling frames allows for accurate and efficient annotation of ASL gestures present in the dataset. The labeling process involves annotating frames or images by identifying and delineating objects of interest within the scene. The labeling process was performed using Roboflow [33], a widely used online annotation tool which is described later in this section.

While conducting this research, Roboflow was used as the platform to facilitate the annotation process. Roboflow is a popular platform that provides a comprehensive suite of tools for managing and preprocessing computer vision datasets. It offers an intuitive interface and various functionalities that streamline the labeling and annotation process. The platform likely provided features such as image uploading, annotation tools, and collaboration capabilities, allowing multiple annotators to work on the dataset simultaneously. Roboflow may have also offered functionalities for data augmentation, preprocessing, and data management, making it a convenient choice for the research activity’s annotation needs.

In the course of our research, there were two instances of annotation procedures used, with a difference in the inclusion of the "Person" label. In the first instance, the annotation procedure included a "Person" label to indicate the presence of a person performing the ASL gesture. This label helped provide contextual information and differentiate between instances where a person is present versus situations where only the ASL gesture is depicted.

As an example, let's consider an image from the training set present on Roboflow that depicts the ASL gesture for "bag". In this image, a person is performing the ASL gesture associated with the word "bag". The annotation process using YOLO V8 and Roboflow would involve drawing a bounding box around the hand region where the "bag" gesture is being performed. The annotator would label this bounding box with the class or category "bag". This annotation serves to indicate to the ASL detection and recognition model that the specific gesture in the image represents the sign for "bag".

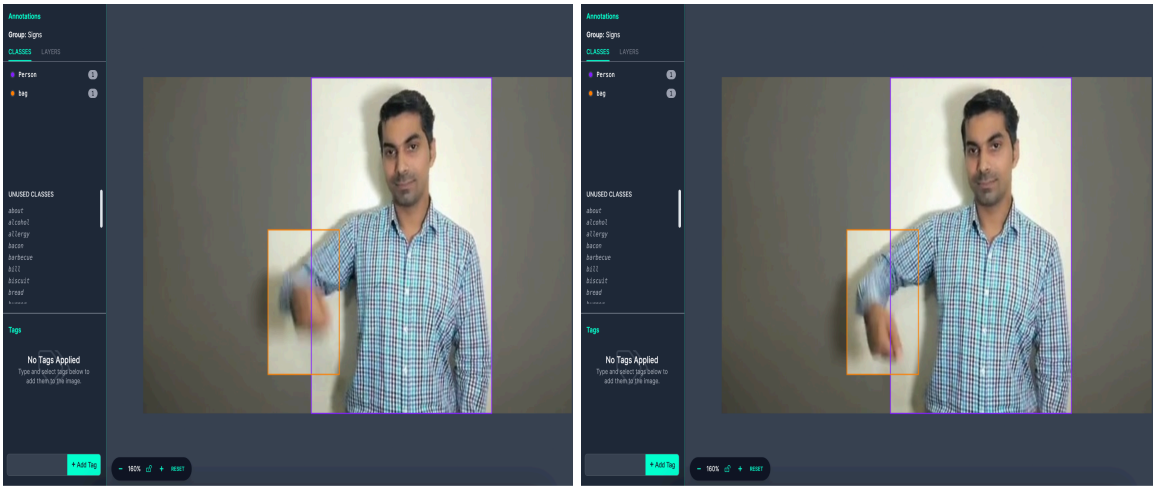

Figure 5 & 6: Annotations of Bag (Orange) and Person (Violet)

But we noted that our model,in the case when “Person” label was included, was extremely susceptible to background noise and prone to wrong detection of the Signer even in the slightest of movements in the background. So, we decided to remove the “Person” label and proceed with the annotation of the sign only. This modification in the annotation procedure allowed the model to concentrate on the ASL gestures independently, potentially improving the accuracy and efficiency of the recognition system.

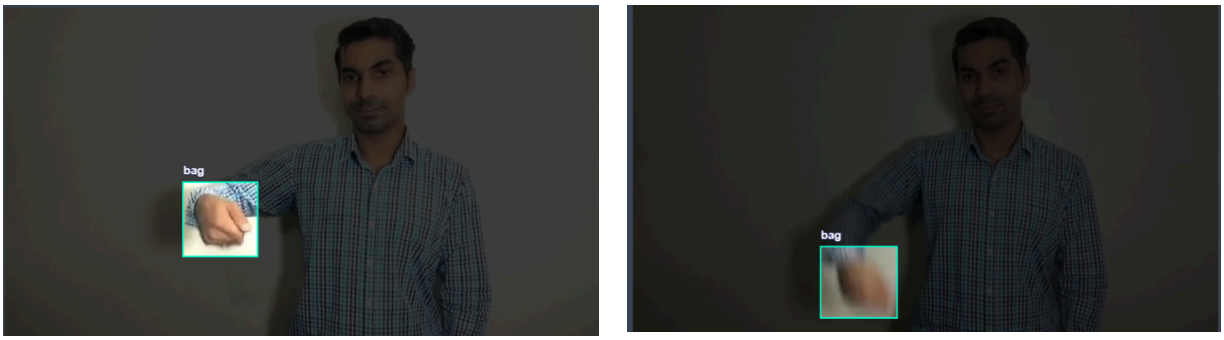

Figure 7 & 8: Annotations of Bag (Green)

The inclusion or exclusion of the "Person" label in the annotation procedure can be considered a design choice based on research specific requirements and objectives. By excluding the "Person" label, the model can focus solely on ASL gesture recognition, potentially leading to more precise and dedicated performance in that aspect and in our case, improved the overall accuracy, precision and other analytics and made the model more resistant to the background noise.

V. Result and Analysis

5.1 Performance Measures

Measuring the performance of models is of utmost importance as it defines the next step to take, when it comes to tuning the hyperparameters of the model for improvements. Performance measures act as unbiased benchmarks to evaluate the efficacy and efficiency of their models. Through quantitative assessment of model performance, weaknesses and potential limitations can be identified, allowing for focused optimizations. This iterative process of measurement, analysis, and fine-tuning of model hyperparameters ensures the attainment of desired outcomes. Precise performance measures further facilitate model comparisons, which assists the authors in selecting the optimal model for their specific objectives. Ultimately, robust performance measurement plays a pivotal role in refining and enhancing models, resulting in improved performance and practical usability in real-world scenarios, which aligns with our research objectives.

Several commonly used performance measures throughout the research include loss, accuracy, precision, recall and mean Average Precision (mAP). These measures provide valuable insights into different aspects of model performance.

$$Loss\ Function\ =\ (1/N)\ \Sigma\ (y_i\ -\ \hat{y_i})^2 \tag{19}$$

Loss is a metric that quantifies the dissimilarity between the predicted and actual values. In Equation 19, L is the objective function that is to be minimized, N is the total number of samples, y_i is the actual value of target function and $\hat{y_i}$ is the predicted value of the target function. It reflects how well the model is learning during training, with the goal of minimizing the loss function.

$$Accuracy\ =\ Number\ of\ correctly\ classified\ samples\ /\ Total\ number\ of\ samples \tag{20}$$

Accuracy [34] is a widely used measure that calculates the proportion of correctly predicted instances out of all instances. It provides an overall assessment of the model's performance but may not be suitable for imbalanced datasets.

$$Precision\ =\ TP\ /\ (TP\ +\ FP) \tag{21}$$

Precision [34], [35] is a measure that quantifies the proportion of correctly predicted positive instances out of all instances predicted as positive. It indicates the model's ability to avoid false positives.

$$Recall\ =\ TP\ /\ (TP\ +\ FN) \tag{22}$$

Recall [34], [35], on the other hand, measures the proportion of correctly predicted positive instances out of all actual positive instances. It highlights the model's ability to avoid false negatives.

$$mAP\ =\ (1/N)\ \Sigma\ (AP_i) \tag{23}$$

mAP, or mean Average Precision [36], is particularly common in object detection and image classification tasks. In Equation 5.1.5, AP_i stands for the average precision of class 'i'. In our model, all labels can be thought of as classes. It considers both precision and recall across different confidence thresholds and provides a comprehensive evaluation of model performance. When interpreting these performance measures, it is essential to consider the specific requirements and objectives of the task at hand.

In our evaluation, we tested the model's performance on a set of images using the YOLO v8, YOLO v5 algorithms. When applying YOLO v8 to our images, the model generated a set of predictions or outputs. Each output consists of bounding boxes that localize and classify objects within the image, along with their corresponding confidence

scores. These confidence scores indicate the model's level of certainty for each prediction. Figure 9 depicts a frame out of a video where the label “biscuit” is being signed. As you can see, it predicts the probability of the sign being “biscuit” as 98%.

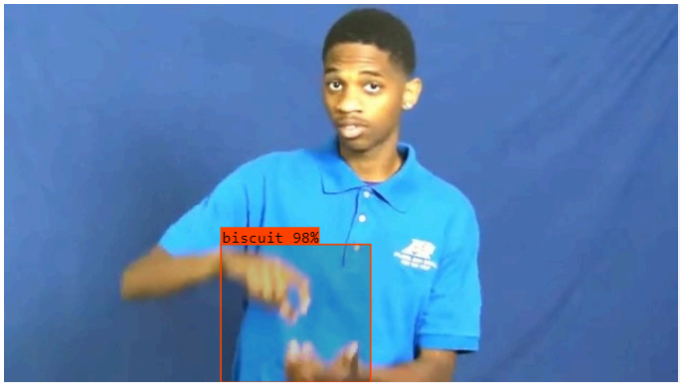

Figure 9 Biscuit

Similarly, Figure 10 corresponds to a frame from a video of label "alcohol". Notably, the model predicts with a high probability of 89% that the sign in the frame represents the "alcohol" label.

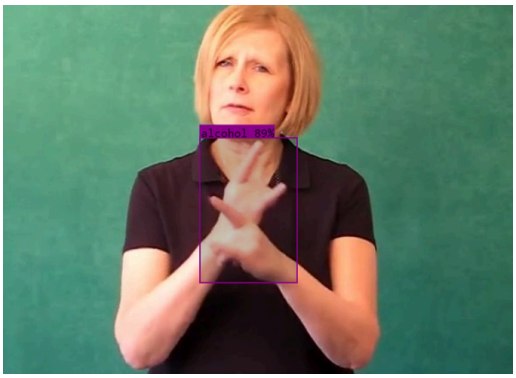

Figure 10 Alcohol

When utilizing LRCN and LSTM models for our analysis, the process differs from object detection with YOLO v8. Instead of generating bounding boxes and localizing objects within the images, these models focus on sequence-based data analysis.

Figure 11 depicts the output of our LRCN model. It is important to note that while YOLO v8 focuses on object detection and classification in individual images, LRCN and LSTM models excel at capturing temporal patterns and dependencies in sequential data, making them particularly suitable for video analysis, action recognition, and other tasks that involve analyzing data over time.

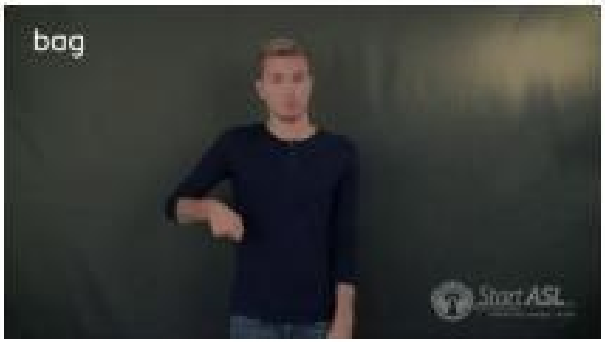

Figure 11 Bag

Building upon the previous discussion, Table 2 compares the performance measures obtained from our object detection models whereas Table 3 provides an insightful comparison between LRCN, LSTM, BiLRCN and BiLSTM. Table 2, Table 3 offers a detailed analysis of the model's performance, enabling a thorough examination of its strengths and limitations.

Table 2 Comparison of YOLO v5 and YOLO v8

Model	**mAP**	**Precision**	**Recall**
YOLO v5	96.6%	**94.6%**	92.2%
YOLO v8	**96.7%**	92.4%	**93.3%**

Table 3 Comparison of LSTM, LRCN and BiLRCN models

Model	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**	**Number of Epochs**
LSTM	96.48%	86.56%	**90.00**%	65
LRCN	94.68%	**86.87**%	88.59%	50
BiLRCN	**97.34**%	82.83%	81.38%	65

The LSTM model demonstrates strong performance across all three accuracy measures. It achieves a high training accuracy of 96.48%, indicating its ability to effectively learn and predict the training data. The validation accuracy of 86.56% suggests that the model generalizes well to unseen data, though it might slightly overfit the training data. The test accuracy of 90.00% indicates the model's overall performance on an independent test set, showcasing its capability to make accurate predictions on new and unseen data. The model was trained for 65 epochs, suggesting a significant amount of training time.

The LRCN model also demonstrates promising results, although slightly lower compared to the LSTM model. It achieves a training accuracy of 94.68%, indicating its ability to learn patterns within the training data. The validation accuracy of 86.87% suggests reasonable generalization to unseen data, while the test accuracy of 88.59% represents its performance on independent test data. The model was trained for 50 epochs, which is relatively fewer epochs compared to the LSTM model.

The BiLRCN model exhibits the highest training accuracy of 97.34%, indicating its effectiveness in learning the training data. However, the model's performance seems to degrade when it comes to validation and test accuracies. The validation accuracy of 82.83% and test accuracy of 81.38% suggest potential overfitting or difficulties in generalization to unseen data. The model was also trained for 65 epochs, similar to the LSTM model, indicating a considerable training time. In summary, the LSTM model demonstrates the highest test accuracy among the three models, followed by LRCN and BiLRCN. The LSTM and LRCN models exhibit relatively closer performance, with the BiLRCN model trailing behind in terms of accuracy.

When it comes to YOLO, analysis can be done at a granular level, that is the mAP50-95 value for each label can be analyzed post training. Table 4 depicts the resulting top five mAP values after being trained on a v8 architecture.

Table 4 Top five mAP values post training on YOLO v8

Sr No.	**Label Name**	**mAP value**
1.	bacon	**0.981**
2.	vegetables	**0.981**
3.	pepper	**0.981**
4.	mustard	**0.981**
5.	fresh	**0.980**

VI. Conclusion

This paper focuses majorly on recognizing and translating ASL gestures to text, using models like LSTM, LRCN, Bi-LSTM, Bi-LRCN, and object detection models like YOLO v5, YOLO v8. The object detection model YOLO v5 achieves a staggering recall value of 92.3% successfully recalling and translating labels like “bacon”, “fresh”, “mustard”, “pepper” etc. In order to improve the recall value, we used YOLO v8, which had a recall of 93.3%.

While the research provided valuable insights into American Sign Language Recognition and Translation, it is important to note the limitations that have influenced the results and conclusions. The scarcity of data posed a limitation to the robustness of the models generated during the course of the research. Insufficient data availability restricted the ability to develop comprehensive and reliable models, potentially impacting the accuracy and generalizability of the results obtained. Future research should consider acquiring a larger and more diverse dataset to mitigate this limitation and improve the overall effectiveness of the models. The model’s accuracy can also be improved by developing a dataset under ideal conditions, changing orientation of the camera, including more number of signers to increase diversity.

For future research in this domain, frameworks of natural language processing can be incorporated to ensure a more comprehensible output, for instance, displaying sentences instead of just the signed words. Further, the proposed model can be trained for other sign languages like the Indian Sign Language, ensuring higher regional inclusivity and usage. The only issue with the incorporation of other sign languages is the availability of a robust, universal dataset, which can be combated by working towards creation of the same with the cooperation of local deaf and dumb communities.

VII. References

[1] World Health Organization, “Deafness and hearing loss,” Feb. 27, 2023.

[2] “List of sign languages by number of native signers,” *Wikipedia*, Sep. 13, 2022.

[3] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[4] J. Donahue *et al.*, “Long-term Recurrent Convolutional Networks for Visual Recognition and Description,” Nov. 2014, [Online]. Available: http://arxiv.org/abs/1411.4389

[5] Raghav Aggarwal, “Bi-LSTM,” *The Medium*, Jul. 04, 2019.

[6] M. Horvat, L. Jelečević, and G. Gledec, “A comparative study of YOLOv5 models performance for image localization and classification Hascheck-Croatian Academic Spelling Checker View project A comparative study of YOLOv5 models performance for image localization and classification.” [Online]. Available: https://www.researchgate.net/publication/363824867

[7] Jacob Solawetz and Francesco, “What is YOLOv8? The Ultimate Guide.,” Jan. 11, 2023.

[8] N. Pavan Kumar, S. Gl, S. Ram Kodandaram, and S. G. L, “Sign Language Recognition”, doi: 10.13140/RG.2.2.29061.47845.

[9] N. Cihan Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, “Neural Sign Language Translation.” [Online]. Available: https://www-i6.informatik.rwth-aachen.de/

[10] O. Koller, S. Zargaran, and H. Ney, “Re-Sign: Re-Aligned End-to-End Sequence Modelling with Deep Recurrent CNN-HMMs.” [Online]. Available: http://www-i6.informatik.rwth-aachen.de/

[11] Y. L. Gweth, C. Plahl, and H. Ney, “Enhanced continuous sign language recognition using PCA and neural network features,” in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, Jun. 2012, pp. 55–60. doi: 10.1109/CVPRW.2012.6239187.

[12] S. Sharma, K. Kumar, and N. Singh, “Deep Eigen Space Based ASL Recognition System,” *IETE J Res*, vol. 68, no. 5, pp. 3798–3808, Sep. 2022, doi: 10.1080/03772063.2020.1780164.

[13] M. Maruyama, S. Ghose, K. Inoue, P. P. Roy, M. Iwamura, and M. Yoshioka, “Word-level Sign Language Recognition with Multi-stream Neural Networks Focusing on Local Regions,” Jun. 2021, [Online]. Available: http://arxiv.org/abs/2106.15989

[14] D. Li, C. R. Opazo, X. Yu, and H. Li, “Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison,” Oct. 2019.

[15] H. Reza, V. Joze, M. Redmond, and O. Koller, “MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language.” [Online]. Available: https://www.microsoft.com/en-us/research/project/ms-asl/

[16] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement.” [Online]. Available: https://pjreddie.com/yolo/.

[17] A. Thakur, P. Budhathoki, S. Upreti, S. Shrestha, and S. Shakya, “Real Time Sign Language Recognition and Speech Generation,” *Journal of Innovative Image Processing*, vol. 2, no. 2, pp. 65–76, Jun. 2020, doi: 10.36548/jiip.2020.2.001.

[18] K. Gajjar, A. Agrawal, A. Gonsalves, G. Singh, and B. Tech, “Sentence Formation Using NLP on the Basis of American Sign Language,” 2022. [Online]. Available: www.ijraset.com

[19] R. M. Abdulhamied, M. M. Nasr, and S. N. Abdulkader, “Real-time recognition of American sign language using long-short term memory neural network and hand detection,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, pp. 545–556, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp545-556.

[20] F. Zhang *et al.*, “MediaPipe Hands: On-device Real-time Hand Tracking,” Jun. 2020.

[21] S. Dhulipala, F. F. Adedoyin, and A. Bruno, “Sign and Human Action Detection Using Deep Learning,” *J Imaging*, vol. 8, no. 7, Jul. 2022, doi: 10.3390/jimaging8070192.

[22] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A. B. Gil-González, and J. M. Corchado, “Deepsign: Sign Language Detection and Recognition Using Deep Learning,” *Electronics (Switzerland)*, vol. 11, no. 11, Jun. 2022, doi: 10.3390/electronics11111780.

[23] P. Sharma and R. S. Anand, “A comprehensive evaluation of deep models and optimizers for Indian sign language recognition,” *Graphics and Visual Computing*, vol. 5, p. 200032, Dec. 2021, doi: 10.1016/j.gvc.2021.200032.

[24] P. Golda Jeyasheeli and N. Indumathi, “Deep Learning Based Indian Sign Language Words Identification System,” 2021. doi: 10.3233/apc210272.

[25] D. Guo, W. Zhou, H. Li, and M. Wang, “Hierarchical LSTM for Sign Language Translation.” [Online]. Available: www.aaai.org

[26] A. Sultan, W. Makram, M. Kayed, and A. A. Ali, “Sign language identification and recognition: A comparative study,” *Open Computer Science*, vol. 12, no. 1. Walter de Gruyter GmbH, pp. 191–210, Jan. 01, 2022. doi: 10.1515/comp-2022-0240.

[27] N. K. Bahia and R. Rani, “Multi-level Taxonomy Review for Sign Language Recognition: Emphasis on Indian Sign Language,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 1, pp. 1–39, Jan. 2023, doi: 10.1145/3530259.

[28] R. H. Abiyev, M. Arslan, and J. B. Idoko, “Sign language translation using deep convolutional neural networks,” *KSII Transactions on Internet and Information Systems*, vol. 14, no. 2, pp. 631–653, 2020, doi: 10.3837/tiis.2020.02.009.

[29] A. A. Kulsoom, C. Y. P, F. Farheen, and N. Halima, “Real Time Sign Language Recognition and Translation to Text for Vocally and Hearing Impaired People.” [Online]. Available: www.ijert.org

[30] S. Khadar Sharif, V. Vjiet, C. Sri Varshini, G. Sreekanth, G. Hruday, and M. Chandu, “Sign Language Recognition.” [Online]. Available: www.ijert.org

[31] National Geographic Society, “Sign Language,” May 20, 2022.

[32] MyDisabilityJobs, “Deaf Employment Statistics,” Oct. 03, 2022.

[33] F. Ciaglia, F. S. Zuppichini, P. Guerrie, M. McQuade, and J. Solawetz, “Roboflow 100: A Rich, Multi-Domain Object Detection Benchmark,” Nov. 2022.

[34] H. M and S. M.N, “A Review on Evaluation Metrics for Data Classification Evaluations,” *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.

[35] D. M. W. Powers and Ailab, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION.”

[36] Jonathan Hui, “mAP (mean Average Precision) for Object Detection,” *The Medium*, Mar. 07, 2018.