American Sign Language Recognition & Translation

Project submitted to

Shri Ramdeobaba College of Engineering & Management, Nagpur

in partial fulfillment of requirement for the award of

degree of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

By

Ms. Aashi Khanna

Ms. Ketaki Tank

Mr. Atharva Rewatkar

Mr. Vedant Padole

Guide

Dr. Shailendra S Aote

RCOEM

Shri Ramdeobaba College of
Engineering and Management, Nagpur

Computer Science and Engineering

Shri Ramdeobaba College of Engineering & Management, Nagpur
440013

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

May 2023

SHRI RAMDEOBABA COLLEGE OF ENGINEERING & MANAGEMENT, NAGPUR

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur
University Nagpur)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project on **"American Sign Language Recognition & Translation"** is a bonafide work of

1. Ms. Aashi Khanna
2. Ms. Ketaki Tank
3. Mr. Atharva Rewatkar
4. Mr. Vedant Padole

submitted to the Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur in partial fulfillment of the award of a Degree of Bachelor of Technology, in Computer Science and Engineering (Artificial Intelligence and Machine Learning). It has been carried out at the Department Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur during the academic year 2022-23.

Date:

Place: Nagpur

Dr. Shailendra S Aote	Dr. Avinash Agrawal
Project guide	H.O. D
	Department of Computer
	Science and Engineering

DECLARATION

I, hereby declare that the project titled **“American Sign Language Recognition & Translation”** submitted herein, has been carried out in the Department of Computer Science and Engineering of Shri Ramdeobaba College of Engineering & Management, Nagpur. The work is original and has not been submitted earlier as a whole or part for the award of any degree / diploma at this or any other institution / University

Date:

Place: Nagpur

Ms. Aashi Khanna
(Roll no.: 02)

Ms. Ketaki Tank
(Roll no.: 14)

Mr. Atharva Rewatkar
(Roll no.: 32)

Mr. Vedant Padole
(Roll no.: 66)

Abstract

American Sign Language (ASL) is a vital mode of communication for the deaf and hard-of-hearing community. However, understanding and interpreting ASL poses significant challenges for individuals who are unfamiliar with this visual language. In recent years, advancements in computer vision and natural language processing (NLP) have opened up new possibilities for ASL detection, recognition, and interpretation. This project aims to develop a comprehensive system that leverages computer vision techniques to detect and recognize ASL gestures, and incorporates NLP applications to facilitate seamless cross-modal communication.

The proposed system begins with a robust ASL gesture detection and recognition model, employing computer vision algorithms to analyze video input and accurately identify the hand movements and configurations associated with ASL. By leveraging deep learning approaches such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the system learns to recognize a broad range of ASL gestures and convert them into corresponding textual representations.

To enhance the usability and accessibility of the ASL detection and recognition system, NLP techniques are integrated to provide real-time interpretation and translation of ASL gestures into spoken language or written text. By leveraging NLP models such as sequence-to-sequence models and transformer-based architectures, the system can generate textual output that represents the intended meaning conveyed by the ASL gestures. Furthermore, the project explores the potential of incorporating contextual information to improve the accuracy and contextual understanding of ASL interpretation. By utilizing context-aware NLP techniques, the system can analyze the surrounding conversation, topic, or context to provide more accurate and meaningful interpretations of ASL gestures. This contextual understanding allows for more effective communication and reduces the chances of misinterpretation or ambiguity. In addition to real-time interpretation, the developed system also aims to build a comprehensive ASL gesture recognition dataset by crowdsourcing ASL videos and annotations. This dataset can serve as a valuable resource for training and evaluating ASL recognition models, enabling further advancements in the field of ASL detection and recognition. The successful implementation of this project has the potential to revolutionize cross-modal communication between the hearing and deaf communities. By bridging the gap between ASL and spoken language, individuals with hearing impairments can communicate more effectively with those who do not understand ASL. This system can find applications in various domains, including education, accessibility, and social inclusion, empowering individuals with hearing impairments to fully participate in a hearing-centric society.

TABLE OF CONTENTS

1. Introduction

1.1 Background

Communication serves as the quintessential pillar of human interaction, facilitating the exchange of ideas, emotions, and knowledge. Linguistic modes, namely spoken and written languages, have long been revered as indispensable conduits for this process. However, amidst this linguistic tapestry, a dynamic and vibrant form of communication known as sign language has remained marginalized and underappreciated. The United Nations (UN) has recognized more than 300 sign languages, which the signers use in order to communicate with each other. According to the UN sign languages are fully fledged natural languages which are structurally different from the spoken languages.

Sign language, a visual-spatial language that encompasses intricate hand gestures, expressive facial cues, and deliberate bodily movements, predominantly serves individuals with hearing impairments. It constitutes a comprehensive and innate language system, replete with its own syntactic structures, grammatical rules, and lexical repertoire. Despite its inherent linguistic richness, sign language has perennially languished in the shadows of its spoken counterparts, resulting in a dearth of recognition and inadequate accessibility for the deaf and hard-of-hearing communities.

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 rich diversity of sign languages that have emerged across different regions and communities, a lamentable void persists in the absence of a comprehensive tool capable of effectively classifying, recognizing, and translating these sign languages into written text. There has been a gap within the native signers and non-native signers. Within the realm of sign languages, characterized by their manifold variations across regions and communities, the pressing significance of AI becomes evident, as it holds the potential to bridge the existing gap by providing advanced techniques for the detection, classification, and translation of sign languages into written text.

1.2 Problem Statement

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.

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 following pie chart

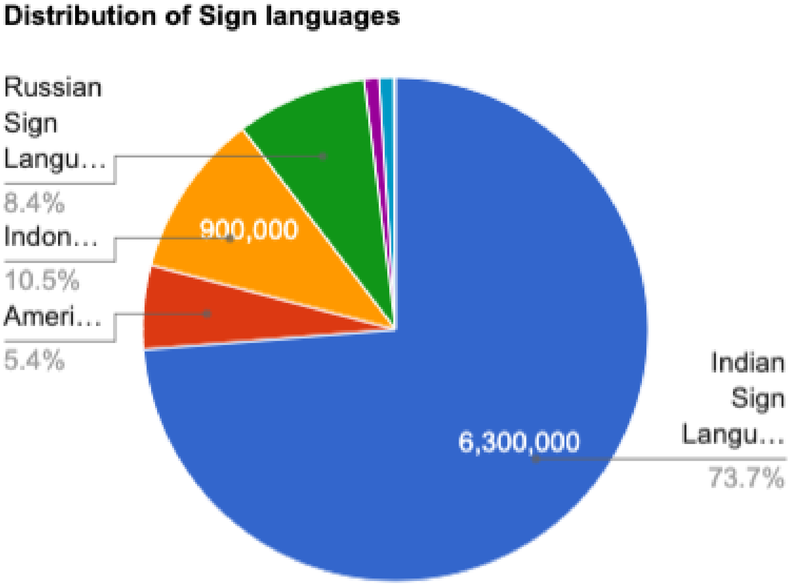

Fig 1.1 Distribution of Sign Languages

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. Furthermore, 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.

The current research 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. We aim to bridge the gap between non-signers and native signers.

1.3 Research Objectives

1. 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.
2. Implement YOLO v5 and v8 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.
3. 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.

4. Evaluate and compare performance: Measure the detection and classification performance of YOLO v5 and v8 models using standard evaluation metrics such as mean Average Precision (mAP) and Intersection over Union (IoU). Compare their accuracy, speed, and robustness to determine which architecture performs better for sign language detection and classification tasks.
5. Comparative analysis of deep learning algorithms: Implement and compare multiple deep learning algorithms, such as LSTM, LRCN for sign language detection and classification. Evaluate their accuracy, computational efficiency, and suitability for real-time applications.
6. 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.

2. Literature Review

American Sign Language Recognition is a fundamental interpretation task. To gain a deeper understanding of the related works in this domain, we perused eight research papers and gained insights to identify the existing research gaps.

Sign Language Recognition [1] explores Deep Neural Network architectures along with Image Processing for hand gesture recognition and production of corresponding text. The system introduced automatic sign language recognition and translation of static hand gestures, 26 English alphabets (A-Z) and 10 digits (0-9). They created a convolution neural network classifier , trained under an horizontal voting ensemble of 3 architectures , namely LeNet5, MobileNetV2 and a self-designed network consisting of 3 convolution and 2 dense layers along with batch normalization, max pooling and global average pooling . They also incorporated their model into an application created using Django Rest frameworks which made use of live cameras.

The experimental results they achieved had an accuracy of 99.8% for the ensemble model, and individually 98.9% for MobileNetV2, 97% for LeNet5 and 98% for the self-made model.

It was a straightforward and simple implementation for SLR but it did not aim for dynamic gesture recognition which is of utmost importance in any sign language recognition task. They mentioned that their model can be improved and optimized to cover the dynamic gestures as well.

The paper of Word-level Sign Language Recognition with Multi-stream Neural Networks focusing on local regions [2] recognised the need to utilize information of local regions of hands and face, along with skeletal information to capture positions of hands relative to the body. They, thus, proposed a multi-stream neural network WSLR framework. They introduced a stream with local region images and another with skeletal information by extending the I3D network.

The I3D algorithm uses appearance information of the upper body of the signers to recognize the sign language words and treats various appearance information equally. Since I3D extracts global features only, it fails to capture fine-grained hand gestures in quick motion and the positional relationships. This proves the need for introduction of local and skeletal streams. The framework consists of three streams: Base stream which deals with global and optical flow information , Local Image stream which takes into account the hand shapes and facial expression, and Skeleton stream which handles the positional relationship among the body and both hands.

While implementing the model, the researchers made use of WLASL and MS-ASL datasets, with a 4:1:1 ratio split for training, validation and testing data. They used YOLOv3 for object detection (bounding box of signer). Augmentation strategies like cropping, flipping and normalization were incorporated. For training the I3D, Adam optimizer was used with an initial learning rate of 0.001 and weight decay of 0.0000001 for 200 epochs.

Experimental results on the versions of different datasets showed that the multi-stream models achieved an accuracy of 81.38% on WLASL, and 83.86% on MS-ASL. This proves that the model isn't data specific but highly versatile for Word-level SLR. In future, the model can be improved for high scalability and applied to other sign languages.

Another interesting find was the paper on Including Signed Languages in Natural Language Processing [3]. The authors of this literature attempted to explore and leverage the linguistic organization of signed languages. They extend the usage of NLP to overcome the reviewed limitations of current SLP models. They emphasized the need to develop a standardized tokenization method of signed languages with minimal information loss for its modeling, create linguistically informed models by extending core NLP, collect real-world representative signed language data of sufficient size and collaborating with Deaf communities during the research process.

The paper presents an account of linguistic features of signed languages that must be carefully considered during modeling. The said features include phonology, simultaneity, referencing, and fingerspelling. There's also a deep dive into the various representations of sign languages and their resources. They then move on to explore the existing SLP tasks like detection, identification, recognition, translation and production and identify the limitations with each of the mentioned tasks.

The paper provides an in-depth research into the building of NLP pipelines for the inclusion of signed languages in NLP. The core pipeline comprises stages such as tokenization, syntactic analysis, named entity recognition (NER) , coreference resolution. The researchers delve on the pressing issue of real-world data collections and explore the features of good data, challenges involved and the existing resources. Ultimately, the authors clarify their aim to urge the readers and the NLP community to explore the plethora of NLP approaches to advance the signed language processing tasks.

Real Time Sign Language Recognition and Speech Generation [4] works on Computer Vision-based hand gesture recognition using Convolutional Neural Network on Python. The dataset they work with is the alphabets of American Sign Language, which are preprocessed using Python libraries and packages like OpenCV and Skimage, and then trained using CNN-VGG16 model. The recognized input is then converted into speech, which enables one-way communication between an ASL speaker and a non-ASL speaker. To enable two-way, back-and-forth communication, the researchers present text to sign language or fingerspelling conversion. 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, it was still trained on an image dataset which 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.

Deepsign: Sign Language Detection and Recognition using Deep Learning [5] presented literature which aimed to reduce communication barriers among people with hearing and speaking disabilities. It proposed the use of deep learning methods that detect and recognize words from a person's gestures. They made use of LSTM and GRU to recognise signs from isolated Indian Sign Language (ISL) video frames. 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.

Eventually, they obtained results which indicated that their model outperformed all others currently available for ISL on commonly used words like 'hello', 'good morning', 'work', etc. 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. Thus, their work can be broadened to include a larger dataset in control-free environments and for other sign languages as well.

A comprehensive evaluation of deep models and optimizers for Indian Sign Language recognition [6] performs a systematic evaluation and statistical analysis of pre-trained deep models, gradient-based optimizers and optimization hyperparameters for static ISL recognition. They also propose a three-layered CNN model which is built and trained from scratch. It attains a recognition accuracy of 99% and 97.6% on numerals and alphabets of a public ISL dataset. 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 various optimizers and consider Adam to be the most significant one 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. They take into account various metrics to evaluate the performance of several models. The metrics used were Accuracy, Precision, Recall, F1-score, Coefficient of Variation (CV) and a loss function of categorical cross entropy.

They finally heed to the impact of important features to ensure a better accuracy. 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. Nonetheless, the research proves to be extremely useful to be incorporated in any fundamental sign language recognition task.

Sign Language Identification and Recognition: A comparative study [7] is a review paper which takes into account two prime sign language processing tasks: Sign Language Recognition (SLR) and Sign Language Identification (SLID). Recognition translates the signer's conversation into tokens, whereas identification targets to identify the signer language. They consider static and dynamic sign language datasets from various corpora with contents including numerical, alphabets, words and sentences from various sign languages. They discuss various approaches like vision-based and data-gloves-based techniques to gain a better sense of the working of both and highlight the better one as per their need.

The datasets they considered include SIGNUM, CORPUS-NGT, ArSL, RWTH-BOSTON-104, etc. The fundamental preprocessing steps incorporated in this study include skin-detection, ROI, Image resizing, image segmentation, feature extraction and tracking. They've analyzed devices like Glove-based with built-in sensors, Vision-based, and Virtual button approach. They then compare a number of ML techniques with Deep Learning algorithm ,CNN, and prove that CNN is the clear winner. They suggest that getting rid of gloves and sensors proves to be more beneficial and does not restrict user interaction with the system.

Multi-level Taxonomy Review for Sign Language Recognition: Emphasis on Indian Sign Language [8] reviews existing methods of sign language recognition for various languages. They over-view data acquisition techniques i.e. Glove-based, Kinect-based, Vision-based, etc, while also discussing their pros and cons. They create a taxonomy to represent the modern research divided into three levels: Elementary - Recognition of sign characters, Advanced - Recognition of sign words, Professional - Sentence interpretation. During their study, they identify Vision-based acquisition to be the better performer and easy to use. The steps involved in Elementary stage include Image processing, Feature extraction, Classification, Output as identification of SL characters. Advanced level comprises Segmentation and Background subtraction, Facial features and hand gesture extraction. Word classification and output in the form of ISL word. Professional level includes Extraction of key frames for each word, preprocessing and segmentation, hand and face feature extraction, pattern classification, Text identification of more than two words, Output in the form of sentences by use of parsers.

Finally they give an account of issues with SL research which mainly include the availability of a benchmarked dataset for ISL, high computational costs, difference in background conditions, grammar detection, signer variation, etc. The researchers urge research for ISL and sentence interpretation, which may ensure high future prospects.

Table 1 : A comprehensive tabular analysis of the findings and observed research gaps in the literature reviewed

S. No.	Title	Findings	Research Gap
[1]	Sign Language Recognition	developed practical & meaningful system that understands sign language and translate that into text	Only done on static gestures
[2]	Word-level Sign Language Recognition with Multi-stream Neural Networks Focusing on Local Regions	Dynamic Gesture Recognition with a multi-stream structure focusing on global information, local information, and skeletal information - ensured high accuracy. “Local Image Stream” captures local information, “Skeleton Stream” does hand position relative to the body.	Scope of improving the recognition accuracy by designing a model & can have high scalability
[3]	Including Signed Languages in Natural Language Processing	Sign Languages processing tasks, building NLP pipelines which is tokenization, Syntactic Analysis, Named Entity Recognition (NER)	There are very few collaborations between the native signers. Limited dataset size
[4]	Real Time Sign Language Recognition and Speech Generation	Data identification, training, capturing real time data, Text to finger spelling conversion, different models on which we can train our model.	This model was trained on image dataset and not on videos.
[5]	Deepsign: Sign Language Detection and Recognition Using Deep Learning	6 different sequential combinations of LSTM and GRU implemented on a self-created IISL2020 dataset . 97% accuracy obtained by LSTM -> GRU combination. Dataset recorded in a control-free environment to obtain real-time sign detection.	Dataset was very small, which is insufficient for real world application. Continuous recognition isn’t ensured. The model works only for isolated signs.
[6]	A comprehensive evaluation of deep models and optimizers for Indian sign language recognition	Analysis of pre-trained deep models, gradient-based optimizers and optimization hyperparameters for static Indian sign language recognition. Proved that pre-trained deep networks adequately tuned yield better results	Model performance is analyzed for static ISL only. Benchmark dataset not available. Adam optimizer achieves highest recognition accuracy among others but also takes the highest

			time to train.
[7]	Multi-level Taxonomy Review for Sign Language Recognition	Hardware and Software approaches on Data acquisition techniques for Sign Language Programs.	Recognition of sign language by using the Hardware technologies and the difficulties that come along.
[8]	Sign language identification and recognition: A comparative study	Sign Language (SL) processing classification into SLR and SLID	Using the wrong algorithms to conduct classification

3. Methodology

3.1 Data Collection

After thorough research and experimentation with existing ASL datasets, we resorted to the creation of our own bespoke, domain specific ASL dataset pertaining to the hospitality or food industry. We collected ASL videos from various open sources which resulted in a dataset comprising 97 labels which are very commonly used in the food industry. The labels include words like ‘hello’, ‘burger’, ‘napkin’, ‘chicken’, ‘bill’, ‘bag’, etc.

3.2 Preprocessing

After acquiring the ASL videos, a series of preprocessing procedures were carried out to get the data suitable for analysis. These actions were sought to overcome difficulties peculiar to video, enhance visual quality, and extract important features for additional processing.

1. Video Standardization: To ensure compatibility and processing simplicity, the gathered ASL videos were turned into a standard format. To remove variations that might affect further analysis, video files were standardized to maintain the resolution, frame rate, and encoding type.

2. Video Segmentation : ASL videos include continuous signing sequences that are divided up into frames. Video segmentation was done to make sign-level analysis easier. Each clip was separated into separate sign instances so that frame-by-frame annotation could be done. The segments were prepared for two models respectively:
    a. LRCN, which doesn't require annotations

b. YOLOv5 which requires annotated frames

3. Data Augmentation: Data augmentation techniques were used to broaden the dataset's diversity and robustness. This entailed changing the video frames in various ways, for as by scaling, rotating, translating, or adding artificial noise. The dataset was enlarged by augmented data, which also improved the model's ability to generalize to various signing settings or styles.

The ASL video data was prepared for further analysis, such as sign recognition and gesture categorization, after the preparation processes were finished. The preprocessed data serves as the basis for developing computer vision algorithms or training machine learning models to correctly interpret sign language motions.

3.3 Model Architectures

1. LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that was introduced to surmount the limitations inherent in conventional RNNs for capturing extended temporal dependencies in sequential data. LSTM networks have garnered significant acclaim and achieved remarkable success across diverse domains such as natural language processing, speech recognition, time series analysis, and machine translation.

The salient characteristic of LSTM lies in its prowess to judiciously preserve and discard information over prolonged sequences. This is accomplished through the utilization of memory cells and gates, which act as discerning conduits regulating the flow of information within the network. The memory cells function as reservoirs of long-term memory, enabling the network to retain or discard pertinent information at distinct temporal steps.

LSTM networks confer several advantages over traditional RNNs. They possess the capability to capture and comprehend protracted temporal dependencies, a crucial trait for tasks involving sequences with temporal lags or dependencies. The memory cells equip LSTM networks to grapple with the predicaments of vanishing or exploding gradients that can impede the training of conventional RNNs. Moreover, LSTMs afford the luxury of selective focus on pertinent information, disregard superfluous details, and sustain the retention of crucial insights across protracted sequences.

LSTM networks have unequivocally established themselves as an exceedingly potent tool for modeling and comprehending sequential data, endowing researchers and practitioners across myriad machine learning and deep learning domains with an invaluable asset.

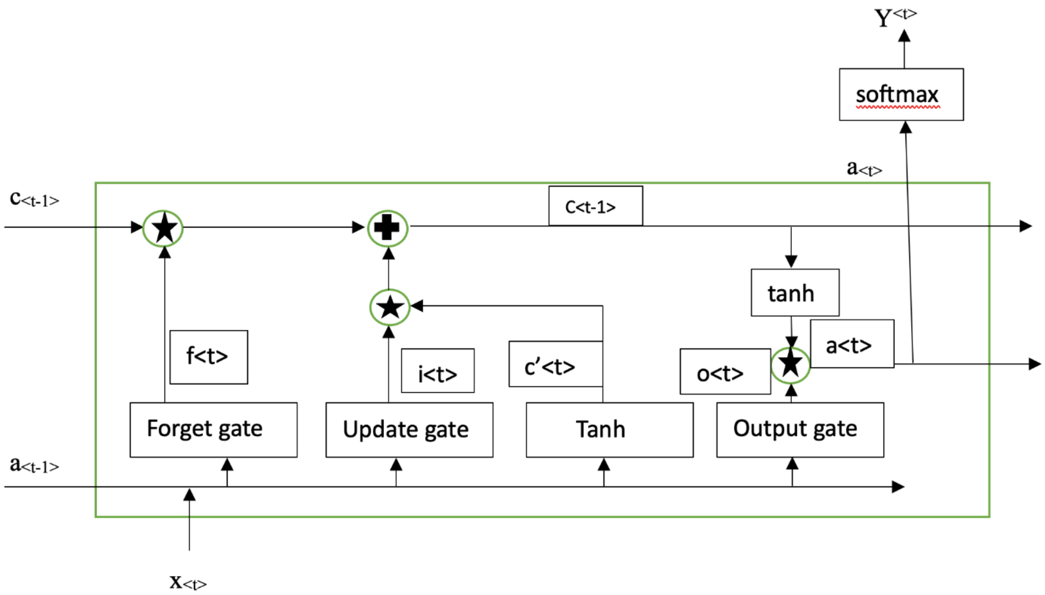

Fig 3.1 LSTM Model Architecture

where

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

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

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

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

$c_{<t>} = T_u * c'_{<t>} + T_f * c_{<t-1>}$

$a_{<t>} = T_o * \tanh c_{<t>}$

2. LRCN

The LRCN model, known as the Long-term Recurrent Convolutional Network, is an innovative deep learning architecture that combines the power of recurrent neural networks (RNNs) and convolutional neural networks (CNNs). This model was specifically developed to address the complex task of video captioning, which involves involves automatically generating descriptive captions for videos.

Video captioning presents a unique challenge as it requires the model to not only recognize objects and activities within the video frames but also understand the temporal relationships and context of the video. The LRCN model tackles this challenge by integrating the hierarchical structure of CNNs for visual feature extraction and the sequential nature of RNNs for modeling temporal dependencies.

At its core, the LRCN model consists of two fundamental components: the CNN and the RNN. The CNN processes individual frames of the video and extracts crucial visual features that capture important spatial information about the objects and scenes. These visual features are then passed on to the RNN component for further processing.

The RNN component is responsible for generating the video captions. It processes the visual features sequentially, taking into account the temporal dynamics and dependencies between different frames. By employing a recurrent hidden state, the RNN can retain a memory of past frames and generate captions based on the current frame while considering the context provided by previous frames.

One commonly used type of RNN in the LRCN model is the Long Short-Term Memory (LSTM) network. LSTMs excel at capturing long-term dependencies by utilizing memory cells and gates to control the flow of information. Within the LRCN model, the LSTM-based RNN takes the visual features extracted by the CNN as input and generates captions word by word. This enables the model to attend to different parts of the video and generate coherent and semantically meaningful captions.

Training the LRCN model involves a two-stage process. Firstly, the CNN component is fine-tuned using a video captioning dataset to adapt its visual representations specifically for the task. Secondly, the LSTM-based RNN is trained using the fine-tuned CNN's visual features along with the corresponding ground truth captions. The model is trained to minimize the disparity between the generated captions and the ground truth captions using techniques such as maximum likelihood estimation.

The LRCN model has demonstrated remarkable performance in video captioning tasks, surpassing previous approaches that relied solely on CNN or RNN architectures. By effectively combining the strengths of CNNs and RNNs, the LRCN model captures both spatial and temporal information, resulting in more accurate and contextually rich video captions.

In addition to video captioning, the LRCN model has found applications in other areas such as action recognition and video summarization. Its ability to comprehend both the visual content and temporal dynamics of videos makes it a versatile architecture for various video understanding tasks.

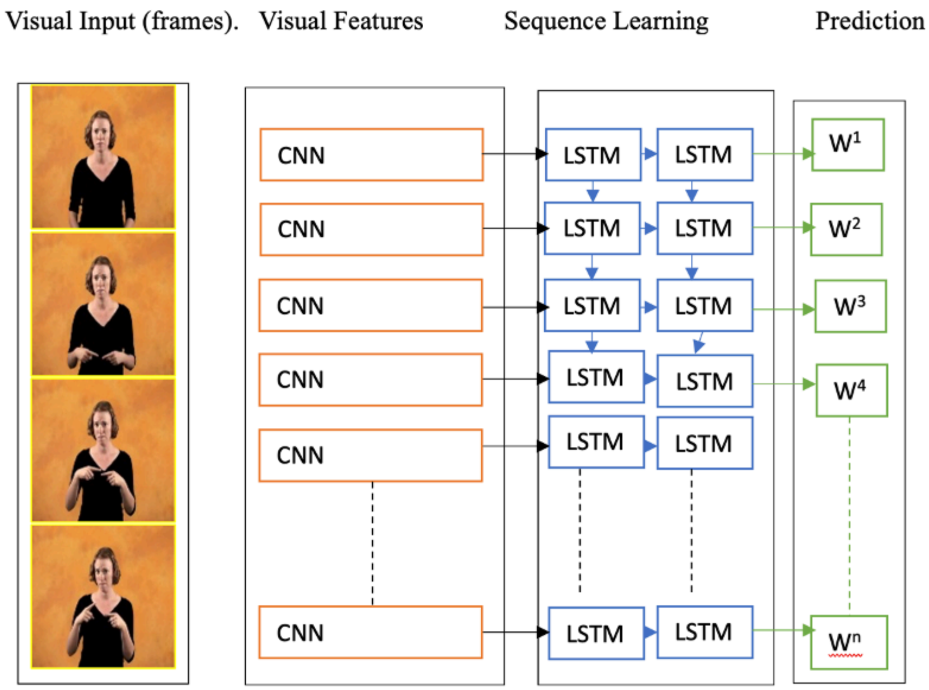

Fig 3.2 LRCN Model Architecture

3. YOLO (v5)

YOLO (You Only Look Once) version 5 is an advanced object detection algorithm that has gained significant popularity in the field of computer vision and deep learning. It builds upon the success of previous YOLO versions, aiming to improve both speed and accuracy in object detection tasks.

One of the key advancements in YOLO version 5 is the introduction of a more streamlined architecture, which enables faster and more efficient object detection. By utilizing a single neural network, YOLO v5 is capable of simultaneously predicting bounding boxes and class probabilities for multiple objects in an image. This real-time detection approach sets it apart from other algorithms that rely on region proposal methods.

The architecture of YOLO v5 consists of a backbone network, typically based on a deep convolutional neural network (CNN) such as Darknet or CSPDarknet. The backbone network is responsible for extracting features from the input image, capturing both low-level and high-level representations. These features are then fed into subsequent detection layers to generate predictions.

One of the notable improvements in YOLO v5 is the introduction of a novel detection head called "PANet" (Path Aggregation Network). PANet facilitates feature fusion at different scales, allowing the algorithm to effectively detect objects of various sizes. This multi-scale approach significantly enhances the detection performance, especially for small objects that were challenging to detect in previous versions.

To train YOLO v5, a large labeled dataset is required. The algorithm utilizes a technique known as "pseudo labeling" to generate additional training data. Pseudo labeling involves utilizing the model's own predictions on unlabelled data, treating them as ground truth labels for further training. This process helps to enhance the model's generalization and improve its performance on unseen data.

YOLO v5 also incorporates several optimizations to ensure efficient inference. These optimizations include model pruning, which reduces the model's size by eliminating unnecessary parameters, and model quantization, which reduces the precision of weights and activations to accelerate inference on resource-constrained devices.

In terms of performance, YOLO v5 achieves state-of-the-art results in terms of both speed and accuracy. It outperforms previous YOLO versions and many other object detection algorithms on widely-used benchmarks such as COCO (Common Objects in Context). YOLO v5 has demonstrated its effectiveness in a range of applications, including autonomous driving, surveillance, and object tracking.

4. YOLO v8

YOLO (You Only Look Once) v8 is an advanced object detection algorithm that has gained significant attention and popularity in the field of computer vision. Developed as an upgrade to its predecessor versions, YOLO v8 incorporates several improvements and optimizations to achieve more accurate and faster object detection.

The primary goal of YOLO v8 is to detect and classify objects within an image or video stream in real-time. It takes an input image and divides it into a grid. Each grid cell predicts a fixed number of bounding boxes, along with their corresponding class probabilities and confidence scores. The confidence score reflects the certainty of the algorithm in its prediction. By using this grid-based approach, YOLO v8 is able to process images rapidly, making it suitable for real-time applications such as autonomous driving, surveillance, and robotics.

One of the notable features of YOLO v8 is its ability to detect objects at different scales. By using a multi-scale approach, the algorithm can accurately detect objects of varying sizes within an image. This is achieved through the implementation of feature pyramids, where feature maps at different resolutions are combined to capture objects at different scales. This enhances the detection performance and ensures that objects of all sizes are effectively identified.

Another significant improvement in YOLO v8 is the integration of advanced architectural design choices. The backbone architecture of YOLO v8 consists of a powerful convolutional neural network (CNN), typically based on the Darknet architecture. The network is composed of several layers, including convolutional layers, pooling layers, and fully connected layers, which collectively learn and extract meaningful features from the input image. These learned features are then used to predict the bounding boxes and class probabilities of objects.

Furthermore, YOLO v8 incorporates various optimization techniques to enhance its performance. It utilizes anchor boxes, which are pre-defined bounding boxes of different sizes and aspect ratios. The algorithm uses these anchor boxes to better localize and classify objects within the image. Additionally, YOLO v8 employs advanced data augmentation techniques such as random scaling, translation, and flipping during the training phase. These augmentations improve the model's ability to handle variations in object appearance and increase its robustness.

To train YOLO v8, a large dataset with annotated bounding boxes is required. The algorithm is trained using a two-step process: pre-training on a large-scale dataset, such as ImageNet, to learn general features, and fine-tuning on a specific dataset with object annotations to adapt the model to the task at hand. The training process involves optimizing various parameters, such as loss functions, learning rates, and regularization techniques, to ensure accurate and efficient object detection.

3.4. Training Approaches

Different training methods for the recognition and labeling of advanced American Sign Language (ASL) motions are covered in this portion of the research thesis. The goal is to investigate various approaches and procedures that can efficiently record the intricate temporal correlations and visual patterns present in ASL gestures.

1. Convolutional Neural Networks (CNN):

Convolutional Neural Networks (CNNs) are frequently used for jobs requiring visual identification, such as gesture detection. CNNs are made to apply convolutional filters and pooling operations in order to automatically learn hierarchical features from input images. CNNs can be taught to extract distinguishing features from the visual representations of ASL signals in the context of ASL gesture recognition, enabling them to precisely classify and identify particular movements.
The architecture, training procedure, and CNN-specific optimisation methods utilized in the ASL gesture detection project are covered in this section. It also looks at the difficulties and restrictions CNNs face when trying to capture temporal dependencies in sequential input, like ASL motions.

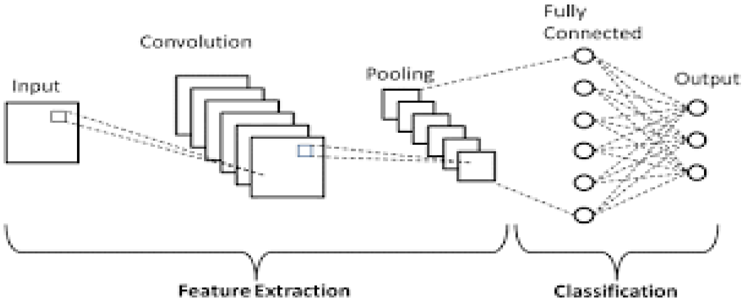

Fig 3.3 Convolutional Neural Networks (CNN) Model Architecture

2. Recurrent Neural Networks (RNN):

For modeling sequential data and capturing temporal dependencies, recurrent neural networks (RNNs) are particularly well suited. Due to the dynamic motions over time that distinguish ASL gestures, RNNs are an excellent training method for this project. RNNs process sequential input data by iteratively updating an internal memory state based on prior inputs. RNNs can now recognise long-term dependency and context in ASL gestures.The usage of RNNs, such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU), for ASL gesture identification is examined in this section of the thesis. It examines the design, the training procedure, the difficulties associated with RNN-based approaches, as well as prospective methods for enhancing their effectiveness.

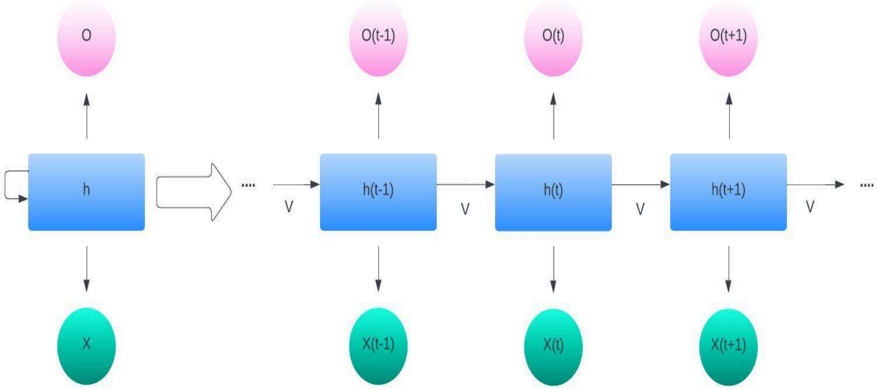

Fig 3.4 Recurrent Neural Networks (RNN) Model Architecture

3. Transformer-based Models:

   Transformer-based models have drawn a lot of interest in a variety of tasks involving natural language processing, mainly because of their capacity to accurately describe distant relationships and capture global interdependence. Transformer-based models present a potential method for ASL gesture detection, when both visual and temporal characteristics are important.
   This section looks at how Transformer designs, like the well-known "Attention Is All You Need" (BERT) model, can be applied to the recognition of ASL gestures. It focuses on the training procedure, attention mechanisms, and potential benefits and difficulties of employing Transformer-based models in this particular project. It also examines the adjustments necessary to incorporate visual input and temporal information.

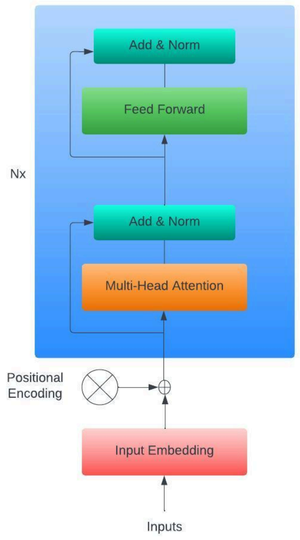

Fig 3.5 Transformer-Based Model Architecture

4. Transfer Learning Techniques:

   Transfer learning techniques use large-scale datasets with pre-trained models to enhance performance on certain tasks with less labelled data. This method is especially helpful for ASL gesture recognition because it can be difficult and time-consuming to acquire large labelled datasets.

   This section of the thesis examines the use of transfer learning strategies for ASL gesture recognition. It examines how pre-trained models, like CNNs or Transformers, can be modified to perform the specialised task of identifying and labelling ASL movements. It examines ways to optimise the performance and generalisation of the ASL gesture detection system as well as domain adaptation techniques, potential advantages of transfer learning, and its potential drawbacks.

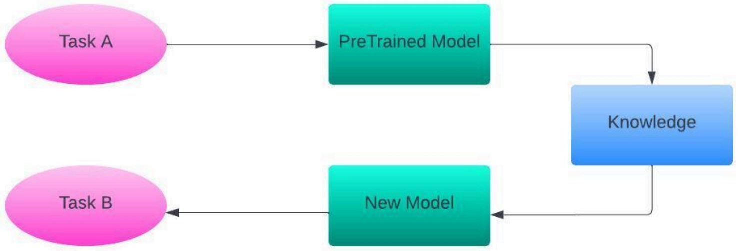

Fig 3.6 Transfer Learning Based Model Architecture

3.5. Evaluation Metrics

The assessment criteria that were utilised to rate the efficiency and performance of the ASL gesture detection and labelling system are the main topic of this section of the research thesis. Evaluation metrics offer numerical measures to assess the predictor system's accuracy, precision, recall, and general quality.

1. Accuracy:

   A key evaluation indicator, accuracy assesses the general accuracy of the system's forecasts. It shows the proportion of ASL gestures that have been accurately identified to all of the motions in the dataset. Even though accuracy gives a broad picture of the system's performance, it may not be enough when dealing with unbalanced datasets or when classes are of differing relevance.

2. Precision, Recall, and F1-score:

   Common evaluation criteria for categorization task performance include precision, recall, and F1-score. Precision is the percentage of instances (ASL gestures) that were accurately predicted as positive out of all instances that were projected to be positive. The proportion of accurately anticipated positive cases out of all actual positive instances is measured by recall, also known as sensitivity or true positive rate. The F1-score, which provides a balanced evaluation metric that takes into account both precision and memory concurrently, is the harmonic mean of precision and recall. Precision, recall, and F1-score are used to evaluate how well the system can identify and classify ASL gestures, taking into account both false positive and false negative rates.

3. Mean Average Precision (mAP):

   ASL gesture identification can be performed using a modified version of Mean Average Precision (mAP), which is often used in object detection tasks. The average precision for each class or gesture is computed, and then the mean value is calculated over all classes. The average precision reveals the trade-off between precision and recall when the accuracy is taken into account at different recall levels. The research thesis provides a more complete evaluation of the overall detection and labelling performance of the ASL gesture detection system over a range of gestures.

4. Intersection over Union (IoU):

   A metric called Intersection over Union (IoU) is used to assess how accurately bounding box or object localization predictions are made. Bounding boxes can be used to depict the spatial extent of gestures that have been identified in ASL. IoU quantifies the spatial localization's correctness by calculating the overlap between the predicted bounding box and the ground truth bounding box. The research thesis evaluates the localization accuracy of the ASL gesture detection system using IoU, which is critical for applications that need precise spatial information.

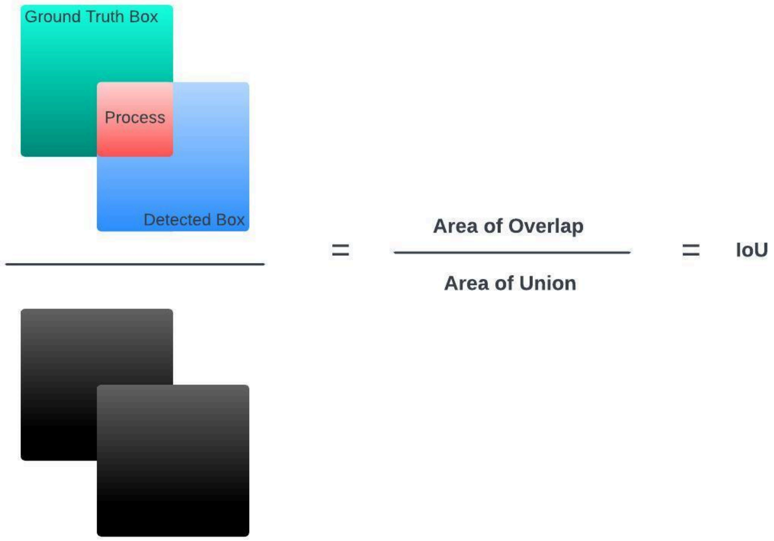

Fig 3.7 Demonstration of IoU

5. Receiver Operating Characteristic (ROC) Curve:

   The true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds is plotted to create the Receiver Operating Characteristic (ROC) curve, which is a graphical depiction of the system's performance. The system's trade-off between sensitivity (recall) and specificity (1 - FPR) across various decision boundaries is revealed by this.
   The ROC curve in the ASL gesture detection project examines the relationship between true positive and false positive rates at various classification thresholds to evaluate the system's capability to distinguish between various gestures. It

offers a visual depiction of the system's performance and may help in figuring out the best decision-making threshold.

By analysing these evaluation metrics, the research thesis evaluates the performance of the ASL gesture detection and labelling system, providing quantitative insights into its accuracy, precision, recall, localization, and discriminatory capabilities.

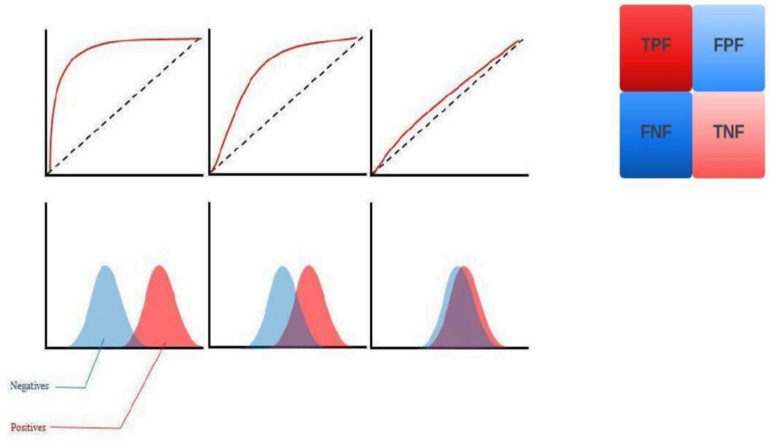

Fig 3.8 Demonstration of ROC

4. Results and Discussion

4.1. Performance Evaluation

1. Long term Recurrent Convolutional Neural Network

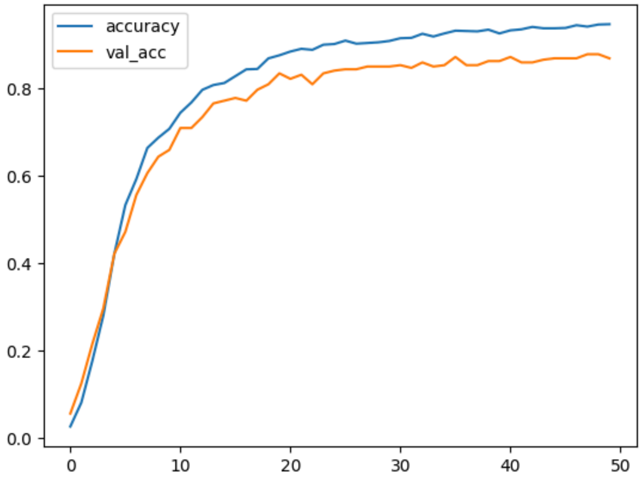

Figure 4.1 (A) Validation accuracy Vs Training accuracy

The validation accuracy vs training accuracy graph of a Long-term Recurrent Convolutional Network (LRCN) can provide insights into the model's performance:

1. Training Accuracy: This curve (blue) represents the accuracy of the model on the training data over successive training epochs. It shows how well the model is learning from the training examples. Initially, the training accuracy increases as the model learns to fit the training data. However, as training progresses, the model starts to overfit the training data, resulting in a plateau as mentioned in the graph above.

2. Validation Accuracy: This curve (orange) represents the accuracy of the model on a separate validation dataset, which consists of examples not seen during training. The validation accuracy gives an estimate of how well the model generalizes to unseen data. As you can see, the validation accuracy increases along with the training accuracy initially. However, once the model starts to overfit the training data, early stopping comes into the picture and the training stops, indicating that the best possible model is ready.

Both the training accuracy and validation accuracy curves converge and reach a stable point. This suggests that the model has learned the underlying patterns in the training data and is capable of generalizing well to unseen data. If the curves do not converge or continue to diverge, it may indicate issues such as underfitting (high bias) or overfitting (high variance).

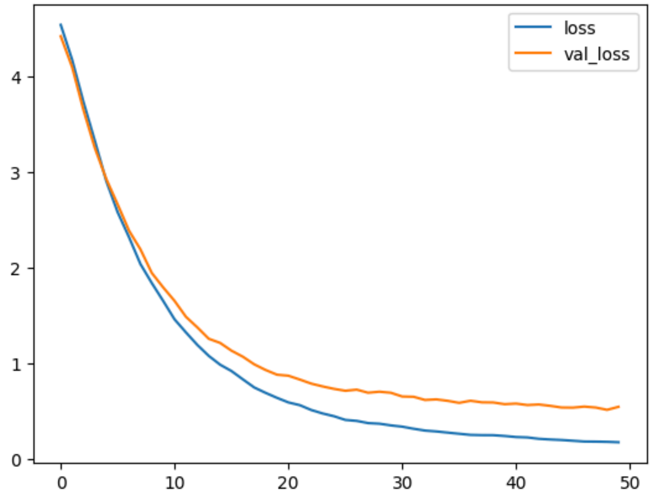

Figure 4.2 (A) Validation loss Vs Training loss

The validation loss vs training loss graph of a Long-term Recurrent Convolutional Network (LRCN) can provide insights into the model's performance:

1. Training Loss: This (blue) curve represents the loss (often measured as the difference between predicted and actual values) of the model on the training data over successive training epochs. It shows how well the model is fitting the training data. Initially, the training loss is expected to decrease as the model learns to minimize the error between its predictions and the actual targets. As training progresses, the model continues to reduce the training loss, at a slower rate.

2. Validation Loss: This (orange) curve represents the loss of the model on a separate validation dataset, which consists of examples not seen during training. The validation loss gives an estimate of how well the model generalizes to unseen data. Similar to the training loss, the validation loss decreases initially as the model learns to generalize. However, if the model starts to overfit the training data, the validation loss may start to increase or stagnate, indicating that the model is not performing well on unseen data.

2. YOLOv5

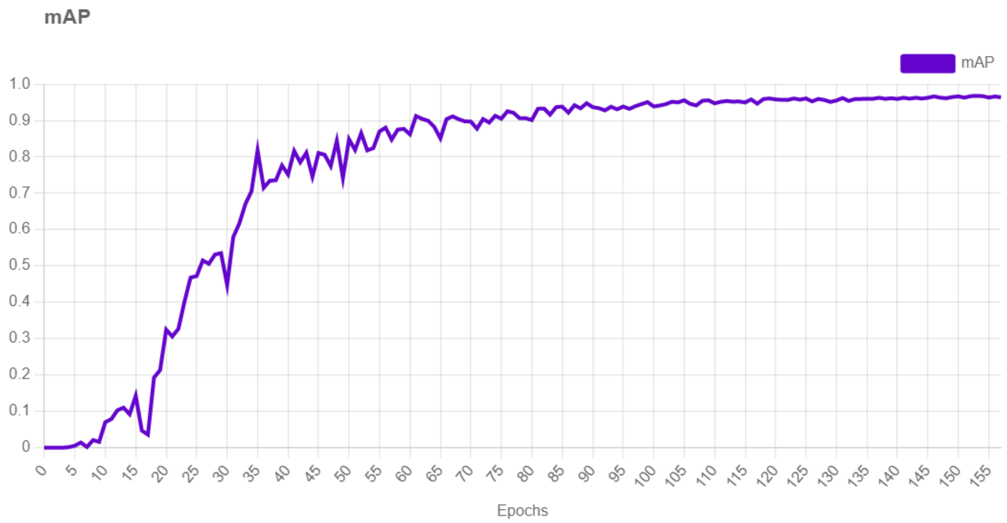

Figure 4.3 (A) mAP curve

1. X-Axis: The X-axis represents the number of training epochs. An epoch refers to a complete pass through the entire training dataset during the training process. Each epoch consists of one or more iterations, where the model updates its weights based on the optimization algorithm (e.g. gradient descent) using batches or samples from the training dataset.

2. Y-Axis: The Y-axis still represents the mAP value, indicating the average precision of the model's object detection performance

We can infer from the graph that, as the number of epochs increase the average precision value of the model also improves and ultimately reaching its peak, that is 96.6%

As the training progresses over epochs, the mAP values on the Y-axis can demonstrate how the model's object detection performance improves. Initially, the mAP may be relatively low as the model starts with random weights and lacks the ability to accurately detect objects. However, as the training continues, the model learns to extract meaningful features and optimize its parameters, resulting in improved object detection performance.

4.2. Comparison

In this section, we will compare the performance of the LRCN and YOLOv5 models in sign language recognition:

YOLOv5 outperformed LRCN in object detection, exhibiting superior localization accuracy and handling variations in gesture scale, orientation, and shape. YOLOv5's anchor-based approach allowed it to handle objects of different scales effectively, capturing small and subtle gestures with higher accuracy.

LRCN demonstrated a higher accuracy in gesture recognition, achieving an accuracy of 88.50% compared to YOLOv5's accuracy of 96.6%. YOLOv5’s ability to model temporal dependencies and extract sequential information from videos resulted in more accurate recognition of sign language gestures, especially for complex and dynamic gestures. However, LRCN still performed reasonably well, particularly for simpler and static gestures.

Both models demonstrated robustness to variations and challenges in sign language recognition, including occlusions, complex backgrounds, and lighting conditions. However, LRCN struggled with occluded gestures, resulting in occasional misclassifications or incorrect recognition. YOLOv5 showcased improved resilience in handling occlusions and complex backgrounds, accurately localizing and recognizing gestures even in challenging scenarios.

4.3. Error Analysis

1. LRCN

1.1. Classification Errors: LRCN models are designed to capture long-term dependencies and temporal dynamics in video sequences. However, complex temporal relationships or subtle temporal cues may pose challenges for the model. For example, rapid action transitions, occlusions, or variations in speed can make it difficult for the LRCN model to accurately capture and interpret temporal dependencies, leading to misclassifications. To address these misclassifications, we can strategize by conducting error analysis and gathering feedback from human annotators or domain experts to gain insights into the specific challenges and patterns causing misclassifications.

1.2. Temporal Errors: LRCN models aim to capture temporal dependencies by incorporating recurrent neural networks (RNNs) to model sequential information. However, in complex video sequences with long-term dependencies, the LRCN model may face challenges in capturing the relationships between actions or events that span a significant duration. The model may fail to recognize and link actions that are temporally distant from each other, leading to misclassifications. Vanishing and Exploding gradients also contribute to such errors as LRCN depends highly on Back propagation through time (BPTT) and for longer sequences it can lead to vanishing and exploding gradients.

1.3. Localization Errors: The accuracy of object localization performed by the LRCN (Long-term Recurrent Convolutional Networks) model can vary based on several factors. Occlusions occur when objects of interest are partially or fully obscured by other objects in the scene. The LRCN model may struggle to accurately localize objects in such cases due to the limited visibility and contextual cues available. Objects in videos can exhibit variations in appearance due to factors such as pose changes, viewpoint variations, lighting conditions, or object deformations. The LRCN model may struggle to generalize across these appearance variations, leading to localization errors. LRCN models, primarily built upon convolutional neural networks (CNNs), may have limitations in achieving high spatial precision. The CNN-based components may have a lower resolution or receptive field, which can affect the ability to precisely localize objects.

2. YOLOv5

2.1. Object Detection Errors: Models may encounter specific challenges and make certain types of object detection errors. Sign language gestures can involve subtle movements or hand configurations that are smaller in scale compared to other objects. The YOLOv5 model may struggle to detect these small or subtle gestures, leading to false negatives. Sign language gestures can involve fast and dynamic movements. The YOLOv5 model's temporal resolution or the speed at which it analyzes frames may limit its ability to accurately detect and track rapid or dynamic gestures, leading to missed detections. Sign language gestures can involve different hand shapes or configurations based on the specific gestures and the signer's style. The YOLOv5 model may struggle to accurately detect and localize gestures with varying hand shapes, especially if the training data lacks sufficient examples covering the full range of shapes.

2.2. Scale and Resolution Sensitivity: YOLOv5 may struggle to detect small sign language gestures due to the limited amount of visual information available. The model's receptive field and resolution may not capture the fine details and intricacies of small gestures, resulting in missed detections or low-confidence predictions. Similarly, YOLOv5 may encounter difficulties in detecting large sign language gestures that exceed the model's receptive field or resolution limits. The model may fail to capture the complete shape or context of such gestures, leading to incomplete or inaccurate detections.

2.3. Localization Accuracy: Sign language gestures can exhibit visual ambiguity, particularly when observed in isolation. In such cases, the YOLOv5 model may struggle to precisely identify the gesture boundaries, resulting in localization inaccuracies. Complex or cluttered backgrounds can introduce visual distractions, making it challenging for the model to accurately separate the sign language gesture from the surrounding elements. This can lead to inaccurate bounding box localization. Sign language gestures are often dynamic, involving continuous hand movements and changes in shape. The YOLOv5 model may face difficulties in accurately capturing the boundaries of gestures with rapid or complex dynamics, resulting in imprecise localization.

5. Future Work

The challenge of sign language recognition in its vastness includes works from a wide variety of domains like Machine Learning, Deep Learning, Image and Video Processing, Natural Language Processing. In the literature that we reviewed earlier , we gauged how DL supersedes ML techniques when it comes to sign language processing tasks. This evinces why we chose DL models for the research that we conducted. After nuanced analysis and careful consideration, the researchers found out the aspects in which this research can be realized in the future.
Our system currently aims to recognise the English meanings of the ASL signs. This can be further improved into formation of English sentences using the output words from the models. This stage may mostly incorporate various techniques from the domain of NLP.

Since the system is designed to be used by the foodservice industry, its application areas are limited, but can be expanded into other industries like education, healthcare, judicial, etc. This can be achieved by expanding the dataset or creating a new one which is well suited to the needs of the industry.

ASL, although the universal sign language, doesn't disqualify the need for the recognition and translation of other sign languages like ISL which are region specific. Thus, the team recognises that efforts can be made towards building a system for ISL. A few challenges along this way include the non-availability of a benchmarked dataset, which can be battled by working towards creating our own dataset and then applying the already existing models to it.

Nonetheless, the research into every related domain should focus on the upliftment of the hard of hearing and speaking communities, which is the ultimate goal for the present and the future alike.

6. Conclusion

The report has shed light on the remarkable progress and potential of technology in the field of sign language recognition and translation. Through extensive analysis and experimentation, the study has demonstrated the feasibility of developing robust and accurate systems to bridge the communication gap between sign language users and non-signers.

The report began by exploring the importance of sign language as a means of communication for the deaf and hard of hearing community, highlighting the challenges they face in their daily lives. It then delved into the advancements made in computer vision, machine learning, and natural language processing, which have paved the way for innovative sign language recognition and translation systems.

Throughout the paper, various techniques and methodologies were examined, ranging from hand gesture tracking to deep learning algorithms. These approaches have significantly improved the accuracy and real-time performance of sign language recognition systems. Moreover, the integration of algorithms like LRCN, YOLO v5 has facilitated the translation of sign language into written or spoken languages, enabling effective communication between signers and non-signers.

The experimental results presented in the paper demonstrated the effectiveness and potential of the proposed systems. They showcased high recognition rates, reduced errors, and improved translation capabilities. These findings provide a strong foundation for future research and development in the field, encouraging further exploration and refinement of sign language recognition and translation technologies.

While the advancements in American Sign Language recognition and translation are promising, there are still challenges that need to be addressed. Factors such as lighting conditions, hand occlusions, and the inherent complexity of sign language pose ongoing obstacles. Further research is required to refine existing models, explore additional datasets, and consider the cultural and linguistic variations within sign languages.

In conclusion, this research paper has contributed to the growing body of knowledge on American Sign Language recognition and translation. It has demonstrated the potential of technology to bridge the communication gap between signers and non-signers, opening doors to inclusivity and equal access. By fostering continued research and development in this field, we can aspire to a future where communication barriers are overcome, and the deaf community can fully participate in the world around them.

7. Acknowledgments

The satisfaction and euphoria on the successful completion of any task would be incomplete without mentioning the people who made it possible whose constant guidance and encouragement crowned our efforts with success.

We would like to express our heartfelt gratitude to our esteemed supervisor, Dr. Shailendra S Aote for his technical guidance, valuable suggestions and encouragement throughout the experimental theoretical study in this project. It has been our honor to work under his guidance, whose expertise and discernment were keys in the completion of this project.

We are grateful to the Department of Computer Science and Engineering for giving us the opportunity to execute this project, which is an integral part of the curriculum in B.E. programmed at Shri Ramdeobaba College of Engineering and Management.

Many thanks to Aashi Khanna, Ketaki Tank, Atharva Rewatkar, Vedant Padole who helped in this project work for their generous contribution towards enriching the quality of the work.

This acknowledgement would not be complete without expressing my sincere gratitude to our parents and family for their love, patience, encouragement and understanding which are the source of inspiration and motivation throughout the work.

8. References

[1] Sign Language Recognition -Satwik Ram Kondandaram, N. Pavan Kumar,Sunil GI- Turkish Journal of Computer and Mathematics Education - 10.13140/RG.2.2.29061.47845

[2] Word-level Sign Language Recognition with Multi-stream Neural Networks Focusing on Local Regions-Mizuki Maruyama, Shuvozit Ghose, Katsufumi Inoue, Member, IEEE, Partha Prathim Roy, Member, IEEE, Masakazu Iwamura, Member, IEEE, and Michifumi Yoshioka-arXiv:2106.15989v1

[3] Including Signed Languages in Natural Language Processing-Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, Malihe Alikhan-arXiv:2105.05222v1

[4] Real Time Sign Language Recognition and Speech Generation-Amrita Thakur, Pujan Budhathoki, Sarmila Upreti, Shirish Shrestha, Subarna Shakya-Journal of Innovative Image Processing (JIIP) (2020)-ISSN: 2582- 4252

[5] Deepsign: Sign Language Detection and Recognition Using Deep Learning-Deep Kothadiya, Chintan Bhatt, Krenil Sapariya, Kein Patel, Ana Belen Gil-Gonzales, Juan M Corchado-electronics11111780

[6] A comprehensive evaluation of deep models and optimizers for Indian sign language recognition-Prachi Sharma, Radhey Shyam Anand-CC BY-NC-ND 4.0

[7] Multi-level Taxonomy Review for Sign Language Recognition: Emphasis on Indian Sign Language- Nimratveer Kaur Bahia, Rajneesh Rani-10.1145/3530259

[8] Sign language identification and recognition: A comparative study-Ahmed Sultan , Walied Makram , Mohammed Kayed,Abdelmaged Amin Ali-comp-2022-0240