CONTINUOUS AMERICAN SIGN LANGUAGE TRANSLATION USING CONFORMER

Yazeed Mshayekh[1], Khaled Rabee[1], Ahmed Allaham[1], Hasan AL-Halabi[1] and Basel Darwazeh[1]

[1] *University of Jordan, King Abdullah II School of Information Technology, Department of AI, Amman, Jordan*

Delivery Date: 4, January 2023

Keywords—Continuous Sign Language Translation, Hand gestures, Machine Learning(ML), Deep Learning(DL)

I. Introduction

We are living in a world where communication is the key to achieve anything wither it was gaining information, getting a job or building relationships with others. So, being able to communicate freely without any barriers or restrictions is a blessing that we should be thankful for; because, in this communication-dependent world there is a group of people who encounters a lot of challenges in their daily lives that is caused mainly by the communication barriers.

The problem is that the sign language is not popular, so there is not a lot of people who did learn it, therefor, this will create a communication gap between mute and deaf people and others, also, dealing with such a gap will make them suffer from loss of opportunities.

Our project is going to address this problem by providing the mute and deaf people with an automatic translation for *Sign Language* (a) and that's mean the ability of converting the hand gestures into a spoken language. This is going to be done by applying many different fields and techniques that are related to the artificial intelligence such *Machine Learning* [1] (b), *Deep Learning* [2] (c), *Computer Vision* [3] (d), *Object Detection* [4] (e), *Object Recognition* [5] (f) and *Natural Language Processing* [6] (g)).

Sign language processing is a hot topic in the communication field where there are many studies that tried to develop methods and ways in order to improve the connection and the communication processes and the impact of other sign language applications, *Sign Language and Web 2.0 Applications* [7], this paper describes Dicta-Sign, a project aimed at developing the technologies required for making sign language-based Web contributions possible, by providing an integrated framework for sign language recognition(SLR), see Example in Figure 1, a Survey of *Advancements in Real-Time Sign Language Translators: Integration with IoT Technology*[8], the research aimed to analyze the advancements in real-time sign language translators developed over the past five years and their integration with IoT technology. By closely examining these technologies, it aimed to attain a deeper comprehension of their practical applications and evolution in the domain of SLT, *Sign Language Literature* [9], the goal of this paper is to emphasize the importance of sign language recognition and translation and provide a comprehensive review of relevant research conducted in this field. And of course, there are many more, so, having all of these researches and papers that are addressing the sign language emphasis on how important this field is.

Contact data: Yazeed Mshayekh, mshayekhy17@gmail.com

a. Problem Definition

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. while 1.1 billion young people are at risk of hearing loss due to exposure to noise and other related problems. Unaddressed hearing loss results in a global cost of 750 billion US dollars [10]. Today, there are more than 300 different sign languages in the world, spoken by more than 72 million deaf or hard-of-hearing people worldwide [11], and some of them with their acronyms are presented in Table 1.

Sign language is a special type of language that is used for deaf individuals as their mode of communication. Unlike other natural languages, it makes use of meaningful body movements to convey the messages, and these body movements are called gestures or signs. Hands and finger movements, head nodding, shoulder's movements, and facial expressions are used to convey meaning. It is used by the deaf people for communication between deaf–deaf or deaf–normal individuals.

Every particular sign means a distinct letter, word, or expression. Combination of signs makes a sentence just like words in spoken languages make sentences. Therefore, a sign language is a complete natural language with its own syntax and grammar[12]. Spoken languages vary from one region to another region, and about 6909 spoken languages

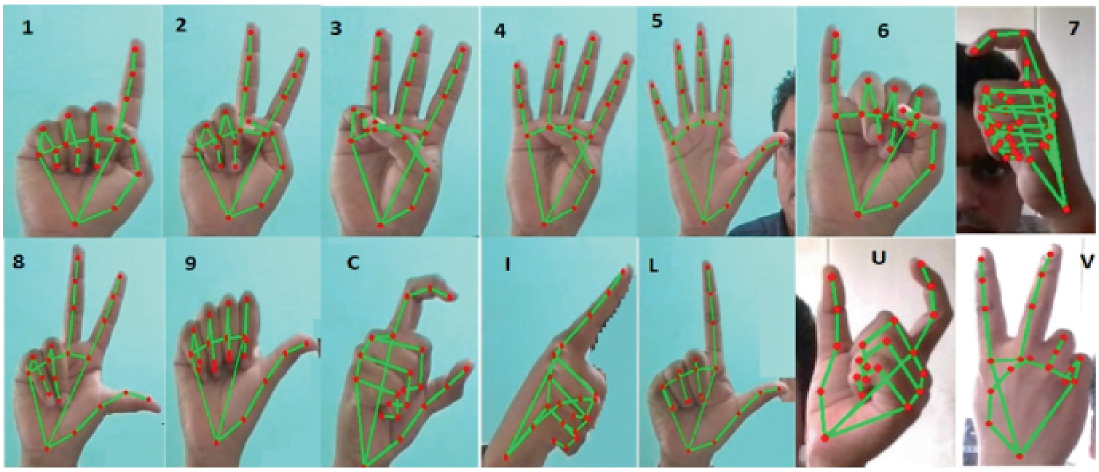

Fig. 1: Detect the Hand and then **Recognize** the Hand Gesture Meaning.

TABLE 1: SOME OF SIGN LANGUAGES AND THEIR ACRONYMS).

Sign language	Acronym
American Sign Language	ASL
Arabic Sign Language	ArSL
Argentinian Sign Language	ArgSL
Australian Sign Language	AusLan
British Sign Language	BSL
Brazilian Sign Language	LSB
Chinese Sign Language	CSL
Greek Sign Language	GSL
German Sign Language	DGS
Indian Sign Language	ISL
Irish Sign Language	IrSL
Japanese Sign Language	JSL
Malaysian Sign Language	MSL
Mexican Sign Language	MxSL
New Zealand Sign Language	NzSL
Pakistan Sign Language	PSL
Portuguese Sign Language	PorSL
Russian Sign Language	RSL
Spanish Sign Language	LSE
Turkish Sign Language	TSL

exist in the world[13].

A Sign Language Gesture involves two types of features, namely `manual features` and `non-manual features`.

The Manual Features (MF) depend upon the *shape*, *movement*, *location*, and *orientation of the hand*. There are gestures which are performed by one hand, while the others are performed by involving both hands. Here are some examples showing manual features Figure 2.

Non-Manual Features (NMF): Non-manual features include different *facial expressions*, *head tilting/nodding*, *shoulder raising*, *mouthing*, and related actions which adds meaning to our performed gesture/sign, see Figure 3. Mostly, non-manual markers are used along with manual markers. While Figure 4 gives examples showing non-manual features

.

The gestures that involve hand movements are referred to as dynamic signs, while the gestures that do not involve any hand movement are termed as static sign gestures. Similarly, the gestures that involve both hands are called **doubled-handed gestures**, and the ones which are performed by a **single hand** are called *single-handed gestures*. As shown in Figure 5.

b. Challenges

As been mentioned before, many researches and papers has been published to deal with the sign language and the gap that has been acquired in the communication field between the mute/deaf people and the world. Each one of those papers has mentioned a unique approach, techniques and methods to address this gap. None of them was perfect since they all got some limitations and challenges.

Dicta-Sign [8], this project aimed for developing the technologies required for making *sign language-based Web* contributions possible. The following is some of the challenges that encountered this project.

Incompatibility of Sign Language with Web 2.0 Applications, this acquired due to the lack of anonymization and easy editing of online sign language contributions. Challenges in Sign Language Recognition, the project faced issues with the robustness especially when low-resolution webcams are used and difficulties in incorporating linguistic research results into recognition systems.

Camgoz_Neural_Sign_Language_CVPR [14], this paper used *the RWTH-PHOENIX-Weather 2014T* [15] dataset, this dataset provides spoken language translations and gloss level annotations for German Sign Language videos of weather broadcasts. Using such data will limit the usability of this project since the data is weather-related and it's in German, so it will be hard to have a generalized solution than can be used in other fields and languages.

Transfer Learning for British Sign Language Modelling [16], the salient idea of this paper is whether transfer learning is a legitimate method for modelling one language with the knowledge of another, assuming the languages are different, but share some common properties, such as vocabulary. The transfer learning approach is tested by applying models trained on one language (English) to another lan-

`https : / / github . com / yazeedmshayekh2 / American-Sign-language-Recognition-Project/tree/main`

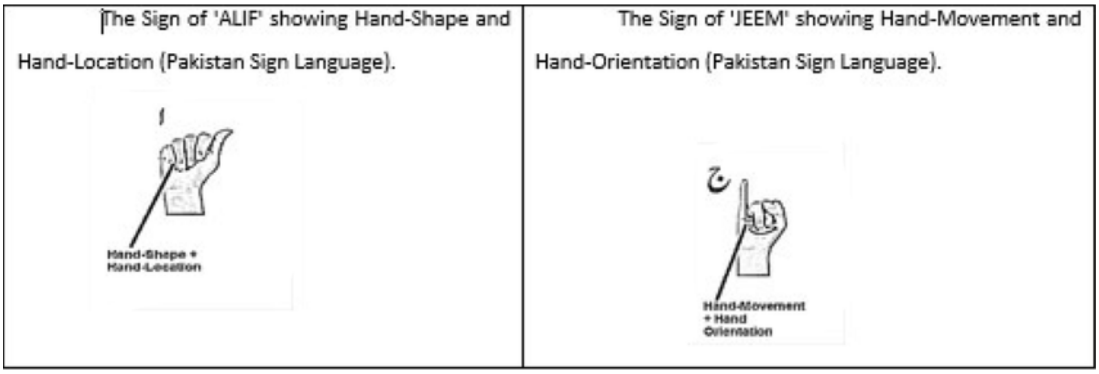

Fig. 2: Highlights the *Manual Features* with the help of some gestures.

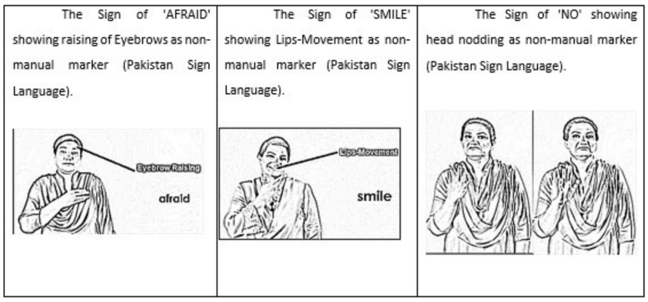

Fig. 3: Highlights some *Non-Manual Features* with the help of suitable gestures.

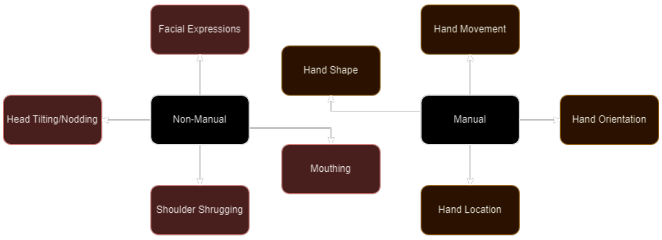

Fig. 4: Showing structural differences between *Manual* and *Non-Manual* gestures.

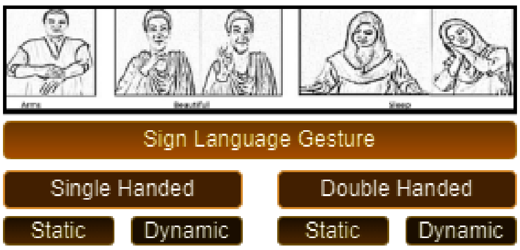

Fig. 5: Some suitable gestures that help understanding the concepts of *single-handed* and *double-handed static*(like one on the left) and *dynamic gestures*, where the dynamic gestures have been shown by presenting multiple frame (like two on the right).

guage (British Sign Language - BSL) and vice versa. The perplexity scores are high in both cases, indicating that the probability distribution over the next word in one language is far from the true distribution of words in the other language. This highlights the challenge of transferring language models across different languages.

c. Proposed Solution

In order to advance sign language recognition and make AI more accessible for the Deaf and Hard of Hearing community, it is crucial to address challenges faced by existing projects. Our objective is to create a reliable, efficient, and resilient model utilizing AI solutions such as automated speech recognition (ASR) and machine translation. Despite the global significance of sign language, over 70 million Deaf individuals and 1.5+ billion people with hearing loss lack access to these technologies [**?**].

Recognizing the need for improvement, we aim to leverage ASR and machine translation to enhance text entry through sign language. While many Deaf smartphone users can fingerspell words faster than they can type, existing sign language recognition AI for text entry falls behind voice-to-text and gesture-based typing due to the lack of robust datasets.

To achieve our goal, we prioritize obtaining a diverse and extensive dataset comprising a vast number of videos with varied backgrounds and lighting conditions. Training our model on such a dataset is essential to enhance its robustness and enable the development of a versatile model capable of handling diverse situations. We have chosen a dataset collected by Google [17], encompassing over three million fingerspelled characters from more than 100 Deaf signers. This dataset was captured via the selfie camera of smartphones, offering a wide range of backgrounds and lighting conditions for a more comprehensive and effective training process.

We will discuss in details the methodology of our project in the methodology section, we will showcase the whole process, models and dataset.

d. Structure of the article

The topics of the project will be as follows, Section 2 will showcase a background about the problem and some related works, Section 3 will discuss the followed methodology, work flow, dataset and model. Section 4 will include the challenges and limitations that encountered us throw the process. Section 5 will showcase the acquired results. Section 6 will talk some methods and techniques that we wanted to try but we didn't because of the time factor.

e. Glossary

`Sign Language` refers to a visual-gestural language used by individuals who are deaf or hard of hearing to communicate. Unlike spoken languages, sign languages rely on manual and facial expressions to convey meaning.

`Machine Learning` refers to a subset of artificial intelligence (AI) that empowers computer systems to learn patterns and make decisions without explicit programming.

`Deep Learning` stands as a pivotal branch of ML, specializing in the training and utilization of neural networks with multiple layers—commonly referred to as deep neural networks.

`computer vision` refers to a field within artificial intelligence that focuses on enabling machines to interpret and understand visual information from the world.

`Object Detection` refers to a computer vision technique employed to identify and locate specific objects or patterns within an image or video frame.

`Object Recognition` refers to the computer vision process of identifying and classifying specific objects or patterns within images or video frames.

`Natural Language Processing` refers to a branch of artificial intelligence that focuses on enabling machines to understand, interpret, and generate human language.

II. BACKGROUND

Background

Many approaches and techniques has been developed to bridge the communication gap for the mute and deaf community. This section will cover those solutions starting from the very beginning.

a. Background of the problem

For the traditional methods, one of the approaches for addressing this gap was the presence of sign language interpreters to facilitate communication between deaf/mute individuals and those who do not understand sign language[18]. This method might be effective but it costs a lot of money[19], also, it's not a dependent solution since the whole communication process depends on the existence of the sign language interpreters. Another way was the written communication, in this approach the mute/deaf individuals will write down but this method will consume a lot of time.

After many advancements in technologies and studies and getting access for more resources, all of this has paved the way for solutions to address these challenges and enhance communication for the deaf and mute community.

Gesture Recognition Technology(GRT)(a), with the rise of computer vision (b) and machine learning (c), GRT has been employed to interpret sign language gestures(SLG). This involves using cameras to capture and analyze hand movements, enabling the translation of sign language into text or speech [20]. Another way was the use of wearable devices equipped with motion sensors and ML algorithms have been developed to recognize and translate SLG. These devices aim to provide a portable and personalized solution for individuals with hearing and speech impairments [21]. Moreover, there is some mobile applications that leverage image and video processing algorithms to interpret sign language through smartphone cameras. These applications provide on-the-go translation (d), fostering independence and improving accessibility [22].

For the current state of the art, Machine translation from signed to spoken languages: state of the art and challenges [23]. This paper has adopted the video-based approach to translate sign languages, it will focus on translating videos containing sign language utterances to text, i.e., the written form of spoken language. This paper will discuss SLT (e) models that support video data as input, it also shows that this approach has benefits compared to wearable-based approaches, which require wearable bracelets or gloves or 3D cameras [21], they can be trained with existing data, and they could for example be integrated into conference calling software or used for automatic captioning in videos of signing vloggers.

b. Related Works

1. *Sign Language Recognitionign Language Recognition*

Early approaches for SLR rely on hand-crafted features `Tharwat et al., 2014`[24]; `Yang, 2010`[25] and use Hidden Markov Models `Forster et al., 2013`[26] or Dynamic Time Warping `Lichtenauer et al., 2008`[27] to model sequential dependencies. More recently, 2D convolutional neural networks (2D-CNN) and 3D convolutional neural networks (3D-CNN) effectively model spatio-temporal representations from sign language videos `Cui et al., 2017`[28]; `Molchanov et al., 2016`[29].

Most existing work on CSLR divides the task into three sub-tasks: alignment learning, single-gloss SLR, and sequence construction (`Koller et al., 2017`[30]; `Zhang et al., 2014`[31]) while others perform the task in an end-to-end fashion using deep learning (`Huang et al., 2015`[32]; `Camgoz et al., 2017`[30]).

2. *Sign Language Translation*

SLT was formalized in `Camgoz et al. 2018`[33] where they introduce the PHOENIX-Weather 2014T dataset[15] and jointly use a 2D-CNN model to extract gloss-level features from video frames, and a seq2seq model(Seq2Seq pipeline as shown in Figure 6) 6 to perform German SLT. Subsequent works on this dataset `Orbay and Akarun, 2020`[34]; `Zhou et al., 2020`[35] all focus on improving the CSLR component in SLT. A contemporaneous paper `Camgoz et al., 2020`[36] also obtains encouraging results with multi-task Transformers for both tokenization and translation, however their CSLR performance is sub-optimal, with a higher Word Error Rate than baseline models. Similar work has been done on Korean sign language by `Ko et al. 2019`[37] where they estimate human keypoints to extract glosses, then use seq2seq models for translation. `Arvanitis et al. 2019`[38] use seq2seq models to translate ASL glosses of the ASLG-PC12 dataset `Othman and Jemni, 2012`[39].

3. *Neural Machine Translation*

Neural Machine Translation (NMT) employs neural networks to carry out automated text translation. Recent methods typically use an encoder-decoder architecture, also known as seq2seq models. Earlier approaches use recurrent `Kalchbrenner and Blunsom, 2013`[40]; `Sutskever et al., 2014`[41] and convolutional networks `Kalchbrenner et al., 2016`[42];

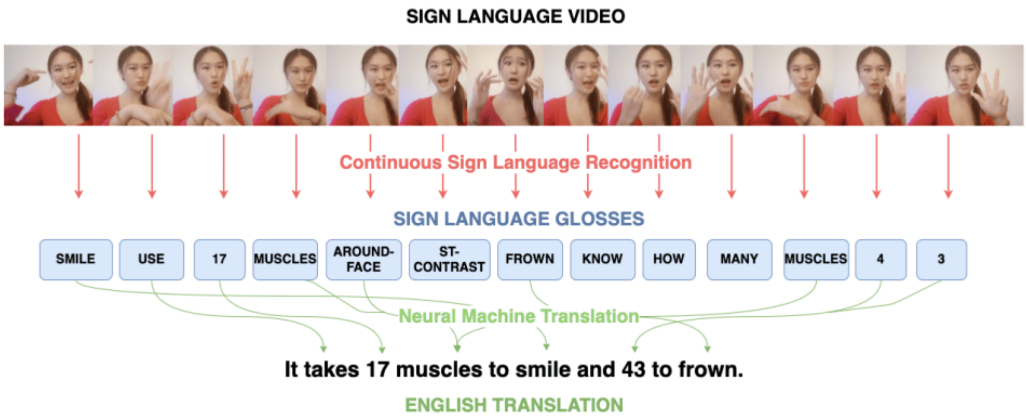

Fig. 6: Sign language translation sequence to sequence pipeline.

`Gehring et al., 2017`[43] for the encoder and the decoder. However, standard seq2seq networks are unable to model long-term dependencies in large input sentences without causing an information bottleneck. To address this issue, recent works use attention mechanisms `Bahdanau et al., 2015`[44]; `Luong et al., 2015`[45] that calculates context-dependent alignment scores between encoder and decoder hidden states. `Vaswani et al. (2017)`[**?**] introduces the Transformer, a seq2seq model relying on self-attention that obtains state-of-the-art results in NMT.

c. Glossary

`Gesture Recognition Technology (a)`: it involves using cameras to capture and analyze hand movements, enabling the translation of sign language into text or speech
`computer vision (b)`: refers to a field within artificial intelligence that focuses on enabling machines to interpret and understand visual information from the world.
`machine learning (c)`: refers to a subset of artificial intelligence (AI) that empowers computer systems to learn patterns and make decisions without explicit programming.
`on-the-go translation` (d): On-the-go translation for sign language represents a revolutionary advancement in assistive technology, specifically designed to empower deaf and mute individuals by providing instant and portable translation of sign language gestures into written or spoken language.
`SLT (e)`: sign language translation.
`Transformer (f)`:
`Conformer (g)`:
`Data Augmentation (h)`: is a technique used in machine learning and deep learning to artificially increase the diversity of the training dataset by applying various transformations to the existing data.
`Preprocessing (i)`: refers to the steps taken to prepare data for analysis or machine learning. It involves cleaning, transforming, and organizing raw data into a format that is suitable for further processing.

III. Methodology

The way that we are going to use to address this problem is by utilizing the facial/hand/pose gestures. There are two approaches, the first one is using a transformer (f) [46] while the second one is using a conformer (g) [47] with data augmentation (h) [48].

For the first approach, we did a preprocessing (i) for the data, we have emitted the missing hand gestures, then we resized the videos to 256 frames. After that, we added for the data the phrase type (phone number/url/address). We sat the maximum phrase length to (31 char+ 1 EOS token), the mean for the character length was 17.8 character, the median was 17 character and the mod was 12 character. Since we only want to use the (x, y) coordinates we transformed the parquet files into numpy arrays (X_train, y_train, X_val, y_val).

The reason behind using the transformer is that we wanted to develop a sequence-to-sequence model (seq2seq problems as shown in 6) to deal with the input, which is videos consisted of sequence of frames, so the flow of the process will be as follow, transforming the input video frames into embeddings using an activation function to capture the feature for each frame that we have. Then, those embeddings will go throw positional encoding to know the position of each frame in the video for the rest of all other frames in the video, after that, the output of this process will go to the encoder that belongs for the transformer.

The layers for the encoder: the normalization layer, multi-head attention layer (dimension 384), another normalization layer. Feet forward network (DENSE layer without using bias and The Gaussian error linear unit activation function, dropout layer 0.3 dropout ratio, another DENSE layer, softmax activation function).

a. Preprocessing and Data Augmentation

1. PreProcessing without Data Augmentation in the First Solution

Proprocessing

The Dataset preprocessing steps as follows, first select dominant hand based on most number of non empty hand frames, then we filter out all frames with missing dominant hand coordinates, resize video to 256 frames, and excluding samples with low frames per character ratio. one of the preprocessing steps was added phrase type(Phone Number, URL, Address).

2. PreProcessing And Data Augmentation in the Second Solution

In this solution, we used multiple different data augmentation techniques, in order to reduce the overfitting problem. The preprocessing steps and data augmentation techniques that we used, applied as follows:

Preprocessing:

Padding(short sequences), resizing(longer sequences), mean calculation with Ignoring Handling, standard deviation calculation with Ignoring NaN, normalization(Standardization), global normalization(Standardization of the pose keypoints).

Splitting, Rearranging , Resizing (lips, hands, nose, eyes, and pose), interpolation(resizes the sequence to a target length, random interpolation).

Data Augmentation:

Random Spatial Rotation (finger keypoints, degree(-10,10)) and Random Scaling(scales finger keypoints, scale(0.9, 1.1)).

Rotation, Shear, Scaling(degree=(-15,15), shear=(-0.10,0.10), scale=(0.75,1.5), inner flipping(around mean of coordinates) and left-right flipping(rigth, Left body like Left, Right hand and so on for each left, right data aspect).

Random Rotation and Scaling(each finger individually), temporal resampling (resampling the temporal length of the data sequence at a new rate), and subsequence resampling(resamples a subsection of the data sequence).

Masking(learns the mode to handle incomplete data), random rotation and scaling(for each finger individually), spatial masking(spatial mask to a random part of the data), and temporal masking(Masks a random temporal segment of the data).

Random Shifting(shift_range=0.1), partial rotation(Applies rotations to parts of the data or individual fingers, whole sequence or a subsection), partial shifting, combined masking(Combines temporal and feature masking in one step), and composite augmentation(applying a random combination of augmentation techniques).

b. Models

Here in this section we're to dive deeply in the models that we used and the architectures of these models.

1. Transformer

After the data is preprocessed and ready, it goes to the input embedding to extract the features from the input data(shape, hands, body posture, and facial expression), then positional encodings are added in order to add information to know the position of each frame in the video (to keep the sequence), then the output from this layer goes to the attention heads, to capture the relationships between each frame with all other frames, to add information about the relationships between frames by adding a value called Attention.

Decoder part from the transformer architecture takes the output from the encoder, which is actually the same as encoder structure but there is one difference which is the masked attention to prevent the model when predicting the current token from seeing the future tokens. Then Decoder-Encoder Attention which is very important to see the whole input when predicting the next token. Final step, the output from the decoder goes into linear layer and then softmax function to add probabilities for each possible output gloss in order to predict the one with the highest probability.

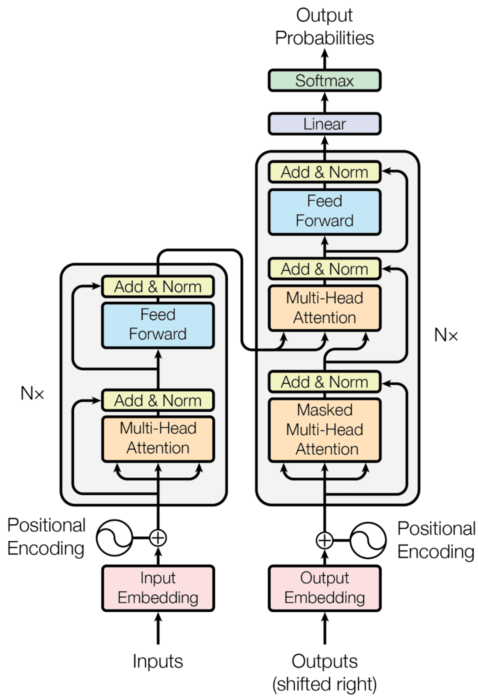

Fig. 7: Transformer Architecture

2. Conformer

In this approach, First we extract the features, then Convolutional Layers are added in order to capture features in each frame such as hand movement, the pose, body movements, and any other type of information in the frame, then the output goes into attention layers to understand the input as a sequence not as individual frames, then goes into Feet-Forward network, and the most important part ,which is used to capture the features within the same frame and capture the features about the sequence as a one block(relationships between frames at the same video), is Conformer-attention block, and then the decoder part comes to predict the word with the information provided by self-attention layers inside the decoder and the information comes from the Conformer Encoder block. Here we can take advantage from using CTC which is a mechanism to align the signs with the predicted glosses, which is a very useful when dealing with Seq2Seq problems like SLT(our problem), **Note** I didn't use CTC mechanism.

IV. Experiments

In this section, we're going to highlight what the values that we set for hyperparameters, the global configuration for model, preprocessing and learning rate(maximum learning rate and learning scheduler), for both first and second solu-

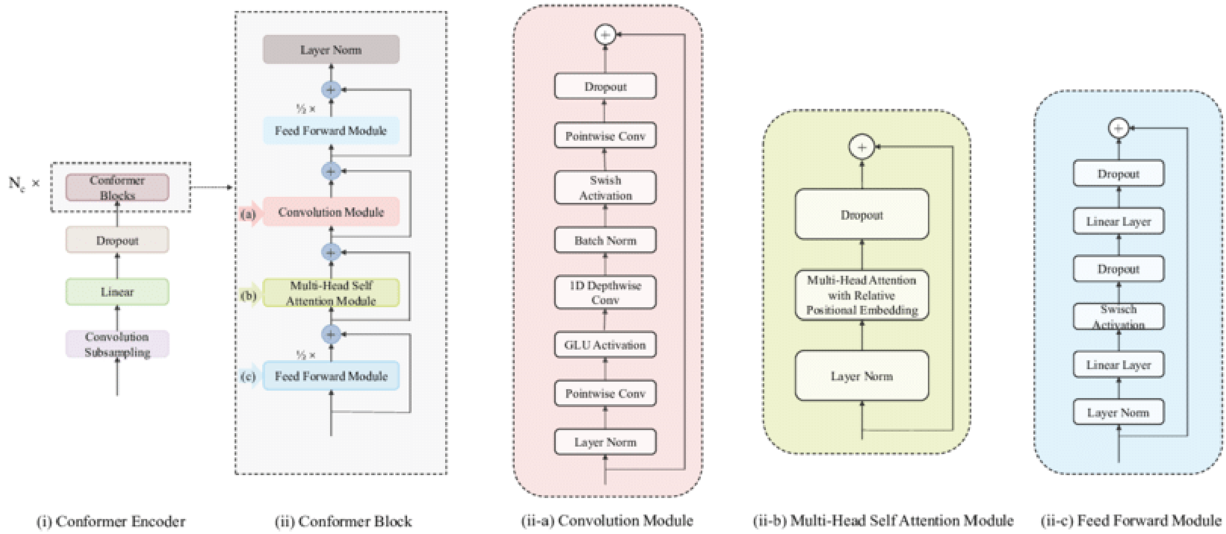

Fig. 8: Conformer Architecture.

tion, what are the hardware resoruces that we used, and what are the dataset that we found, what is the one that we choose and why.

a. Datasets

Here we're going to talk about all the datasets that we found (Here in this table 2[49], you can find more details about first three dataset), and what it is the one that we choose.

1. HOW2SIGN

Within the scope of our project, we looked at more than one datasets. The first dataset is the HOW2SIGN datasets[50], this dataset was an option that we gave up on it because of its huge size that requires high computational power and resources.

It consists of around 80,000 instructional videos (about 2,000 hours) with associated English sub-titles and summaries. It has About 300 hours have also been translated into Portuguese using crowd-sourcing. Its size is around the 300 gigabytes.

2. PHOENIX

The second dataset is PHOENIX14T[15], over a period of three years (2009 - 2011) the daily news and weather forecast airings of the German public TV-station PHOENIX[15] featuring sign language interpretation have been recorded and the weather forecasts of a subset of 386 editions have been transcribed using gloss notation. The problem with this dataset that it is using the German language so this will introduce a linguistic component to the translation endeavor.

3. ASLG

For the last dataset, it was the ASLG-PC12[51], this dataset started collecting English data from Gutenberg Project toward transform it to ASL gloss. Gutenberg Project offers over 38K free ebooks and more than 100K through their partners.

The collecting task was done throw these steps, extract only English texts, because there exist ebook in others languages than English like German, Spanish. Break the text into sentences (sentence splitting task). Prepare the corpora (normalization, tokenization).

4. Google Dataset

While we don't have that much powered resources in order to deal with the last datasets, so we found a solution that is to find a dataset that we can load it on Virtual Machine environment, without even download it locally, since the previous datasets are large to store and deal with them locally. The dataset that we choose, It was a dataset of competition for google on Kaggle[51], the data includes more than three million fingerspelled characters produced by over 100 Deaf signers captured via the selfie camera of a smartphone with a variety of backgrounds and lighting conditions. The size of the dataset is 190GB.

Landmarks extracted from videos using mediapipe holistic detection model, to know more about it Follow this link. Not all of the frames necessarily had visible hands or hands that could be detected by the model. The landmarks files were .parquet files, Google has given us the parquet files instead of us to extract the landmarks from videos then save them in parquet files. This allows you to take advantage of the Parquet format to entirely skip loading landmarks that you aren't using. Parquet file consisted of sequence_id, frame, [x/y/z][type][landmark_index].

Lankmarks.parquet files

Sequence_id => A unique identifier for the landmark sequence. landmark files contain approximately 1,000 sequences. The sequence ID is used as the dataframe index.

Frame=> The frame number within a landmark sequence.

[x/y/z][type][landmark_index] => There are now 1,629 spatial coordinate columns for the x, y and z coordinates for

TABLE 2: STATISTICS OF THE RWTH-PHOENIX-WEATHER 2014T AND ASLG-PC12 DATASETS. OUT-OFVOCABULARY (OOV) WORDS ARE THOSE THAT APPEAR IN THE DEVELOPMENT AND TESTING SETS, BUT NOT IN THE TRAINING SET. SINGLETONS ARE WORDS THAT APPEAR ONLY ONCE DURING TRAINING.

	German Sign Gloss			German			American Sign Gloss			English		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
Phrases	7,096	519	642	7,096	519	642	82,709	4,000	1,000	82,709	4,000	1,000
Vocab.	1,066	393	411	2,887	951	1,021	15,782	4,323	2,150	21,600	5,634	2,609
tot. words	67,781	3,745	4,257	99,081	6,820	7,816	862,046	41,030	10,503	975,942	46,637	11,953
tot. OOVs	19	22	-	57	60	-	255	83	-	369	99	-
singletons	337	-	-	1,077	-	-	6,133	-	-	8,542	-	-

each of the 543 landmarks. The type of landmark is one of ['face', 'left_hand', 'pose', 'right_hand'].

The spatial coordinates have already been normalized by MediaPipe. Note that the MediaPipe model is not fully trained to predict depth so you may wish to ignore the z values. The landmarks have been converted to float32.

character_to_prediction_index.json

It is a json file that contains the classes to predict, there is 59 classes.

[train/supplemental_metadata].csv file

path - The path to the landmark file.
file_id - A unique identifier for the data file.
participant_id - A unique identifier for the data contributor.
sequence_id - A unique identifier for the landmark sequence. Each data file may contain many sequences.
phrase - The labels for the landmark sequence. The train and test datasets contain randomly generated addresses, phone numbers, and urls derived from components of real addresses/phone numbers/urls. Any overlap with real addresses, phone numbers, or urls is purely accidental. The supplemental dataset consists of fingerspelled sentences. Note that some of the urls include adult content. The intent of this competition is to support the Deaf and Hard of Hearing community in engaging with technology on an equal footing with other adults.

b. Hardware Resources

In order to train the model on google's dataset[51], we use Kaggle's 1 Tesla P100 GPU, Its 3584 CUDA cores and 16GB of HBM2 vRAM linked via a 4096-bit interface provide performance to the order of 9.3 TFLOPS at single precision, 18.7 TFLOPS at half precision, and 4.7 TFLOPS at double precision. 29 GB RAM, 73 Disk Space + 189 GB Disk Space for Storage the data.

c. Models

1. Transformer

Transformer Model [46](Transformer Architecture as shown in Figure 7) with 4.887.936 Million Parameters(Embedding+ Landmark Embedding+ Encoder(2 Encoder Blocks)+ Decoder(2 Decoder Blocks)+ 4 Attention Heads in Encoder and Decoder+ Causal Attention Masking), without Data Augmentation.

Lips/Right_HAND/Left_HAND landmarkes that we used, X/Y dimensions used only without z dimension. Preprocessing steps, first we fill Nan with zeroes, then Filtering out Empty Hand Frames, set PAD token to zeros, and we Downsampled by resizing images to 128.

The number of epochs were 100 epochs, POD/SOS/EOS Tokens Used, Batch Size set to 64, learning rate set to 0.001, Weight Decay Ratio set to 0.05, Maximum Phrase length set 31+1 for EOS Token, splitting 10% of the data(7878 samples) into validation set(val_dataset) and the other is for training, It took like 3 hours to train were the epochs was 100. You can found the code Here in This link, and you can find the MLops that we make in Neptune.ai Here in This link, we took a reads for every run that we make and we save just the best, Note: ASL-32 was the best read and we used Transformer Model Architecture in this run.

2. Conformer

The model consists of 2 layer MLP landmark encoder + 6 layer 384-dim Conformer[47] + 1 layer GRU (You can see the conformer architecture in Figure 8). Total number of parameters was 15,892,142, there are 15,868,334 Trainable parameters and 23,808 Non-trainable parameters. It took like 7 hours to train were the epochs was 100, I stop it because the loss didn't improve that much, I tried using Kaggle TPUs but it didn't so if you how to use them, **Note:** If the kaggle TPUs used the number of epochs will increase to 500 and the batch size will increase as well.

Number of Epochs was 100, BATCH_SIZE was 64. Number of Unique Characters To Predict + Pad Token + SOS Token + EOS Token was 62. Maximum Learning Rate was 1e-3, weight decay ration for learning rate was 0.05. Maximum phrase length was 31 and 1 for Eos Token. Number of frames to resize recording to is 384. Drop out ration was 0.1. Causal Masking is applied. the number of landmarks that we choose to use (4 landmarks for Nose , 41 landmarks Lips , 17 landmarks Pose , 32 landmarks Eyes 16(Right) and 16(left), 42 landmarks Hands) In Total 42+76+33 = 151 (42 landmarks HAND_NUMS , 76 landmarks FACE_NUMS , and 33 landmarks POSE_NUMS). X/Y/Z dimensions used that means we add the depth in this approach and finally we splitting the data into 66208 sample for training, and 1000 samples for evaluation on validation dataset.

You can found the code Here in This link, and you can find the MLops that we make in Neptune.ai Here in This link, we took a reads for every run that we make and we save just the

TABLE 3: COMPARISON BETWEEN OUR SOLUTION AND OTHER SOLUTIONS, NOTE YOU CAN TAKE A LOOK AT THE SOLUTION BY CLICK ON THE WORD [HERE] AT THE END OF EACH SOLUTION

Solution Architecture	Levenshtein Distance	
	Private LB	Public LB
1DConv + Transformer + Augmentation using train + supplemental data Here	0.665	0.713
16 CNN-Transformer Blocks + 8 Transformer Blocks + CTC loss Here	0.671	0.714
Transformer and 1D-CNN Here	0.699	0.706
Improved Squeezeformer + TransformerDecoder + Clever augmentations Here	**0.803**	**0.836**
joint CTC + AttentionHere	0.82	0.81
17 layers Squeezeformer with timerduce and ROPE Here	-	0.809
Conformer Encoder-Decoder Ensemble with beam search and edit_dist optimization Here	-	0.807
Vanilla Transformer + Data2vec Pretraining + CutMix + and KD Here	-	0.792
Our Solution	0.686	-

best, Note: we used Conformer Model Architecture in this run ASL-27. It was not that good, because I didn't give that much time to modify the code to work in the best way and to modify the model architecture (change the number of heads, conformer blocks, decoder blocks, change the landmark indices, remove z, add or remove augmentation techniques), actually there are a lot of reasons behind this or maybe the whole idea of conformer is wrong.

V. RESULTS

In this Section, we're going to talk about the model results, using loss and evaluation metrics.

a. Model Results

Using transformer we got **1.854708** SCRELM training loss, **0.861631** accuracy using Top1Accuracy, **0.959309** accuracy using Top5Accuracy, SCRELM loss **2.05572** was validation loss. Training Levenstein distance was **0.814**, Validation Levenstein distance was **0.686**. Validation Top1Accuracy **0.77** and for Top5Accuracy **0.92** Accuracy. For training set, BLEU-1: **23.19**, BLEU-2: **25.76**, BLEU-3: **27.04**, BLEU-4: **27.66** METEOR: **12.15** For validation set, BLEU-1: **12.41**, BLEU-2: **13.79**, BLEU-3: **14.47**, BLEU-4: **14.80** METEOR: **6.840**. Note: In Table 3, you can find more details about other related solutions compared with our solution.

Using Conformer we got **2.284645** SCRELM training loss, **0.657783** accuracy using Top1Accuracy, **0.895256** accuracy using Top5Accuracy, SCRELM loss **2.798254** was validation loss. Sorry because I didn't know how to implement Levenstein distance, BLEU, and METEOR in this solution approach.

b. Evaluation Metrics and Loss

In this section, we're going to talk about loss function that we choose, and the evaluation metrics. Dive deeply in the equations for Loss, and Evaluation metrics.

1. Sparse categorical cross-entropy with label smoothing(SCRELM):

SCRELM is a variation of the standard cross-entropy loss function that is used for training neural networks for multi-class classification problems. It is similar to the standard cross-entropy loss function, but with two main differences. First Instead of using a single target probability distribution, we use a set of target probability distributions, Second we add a label smoothing term to the loss function, which encourages the model to produce probabilities that are close to the true labels[52]. The equation of SCRELM:

$$L(y,\hat{y}) = -\sum (y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})) + \alpha H(\hat{y}) \tag{1}$$

`L` is the loss function. `y` is the true label vector (a binary vector where each element is either 0 or 1). y_hat is the predicted probability distribution (a vector of probabilities, where each element is between 0 and 1). `a` is the hyperparameter controlling the strength of the label smoothing term (usually set to a small value such as 0.1 or 0.01). H(y_hat) is the entropic regularizer term, which encourages the model to produce probabilities that are well-calibrated and not too concentrated on a single class.

2. Levenshtein Distance:

$$d(x,y) = min(x,y) \tag{2}$$

Where x (let's consider it as actual label) and y (let's consider it as a prediction) are the two strings being compared, d(x, y) is the Levenshtein distance[53] between them, and delta(x, y) represents the edit distance between the two strings. Like levenshtein distance between (3creekhouse, 3creek house) is 1.

3. TopKAccuracy Score:

Mathematically, the Top-k accuracy[54] score can be calculated using the following equation:

$$\text{Top-k Accuracy} = \frac{\text{No\_samples where T label is in top k predictions}}{\text{Total No\_samples}} \tag{3}$$

Where:

TP (True Positives) is the number of times that the true class label appears within the top-k predictions made by the model. **FN** (False Negatives) is the number of times that the true class label does not appear within the top-k predictions made by the model.

c. Bilingual Evaluation Understudy

BLEU[55] is computed using a couple of n-gram modified precisions. Specifically,

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4}$$

where p_n is the modified precision for n-gram, the base of log is the natural base e, w_n is weight between 0 and 1 for log p_n and $\sum_{n=1}^{N} w_n = 1$, and BP is the brevity penalty to penalize short machine translations.

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases} \tag{5}$$

where c is the number of unigrams (length) in all the candidate sentences, and r is the best match lengths for each candidate sentence in the corpus. Here the best match length is the closest reference sentence length to the candidate sentences. For example, if there are three references with lengths 12, 14, and 17 words and the candidate translation is a terse 13 words, ideally the best match length could be either 12 or 14, but we arbitrary choose the shorter one which is 12.

Usually, the BLEU is evaluated on corpus where there are many candidate sentences translated from different source texts and each of them has several reference sentences. Then c is the total number of unigrams (length) in all the candidate sentences, and r is the sum of the best match lengths for each candidate sentence in the corpus.

It is not hard to find that BLEU is always a value between 0 and 1. It is because BP, w_n, and p_n are always between 0 and 1, and

$$\exp\left(\sum_{n=1}^{N} w_n \log p_n\right) = prod_{n=1}^{N} p_n^{w\_n} \in [0,1] \tag{6}$$

Usually, BLEU uses $N = 4$ and $w_n = \frac{1}{N}$.

VI. Limitations

The use of low-resolution cameras presents a significant challenge to the project. Because of this limitation, understanding gestures may be inaccurate, making it difficult to capture small differences in hand movements and facial expressions. The performance of the system is permanently linked to the quality of visual input, and the use of low-resolution cameras may compromise the overall effectiveness of gesture translation.

Furthermore, the project's only focus on translation into English introduces a significant constraint. This decision limits the system's generalization to other languages, limiting its reach to a broader segment of potential users.

When dealing with scenarios involving multiple people, a unique challenge arises If more than one person make gestures in front of the camera at the same time. To address this challenge, robust algorithms capable of distinguishing and interpreting overlapping gestures in crowded visual environments are required.

An additional consideration is the possibility of unintentional signing movements. The system must effectively differentiate between deliberate signing and non-signing gestures to avoid misinterpretations.

On the technical front, the lack of devices with high specifications devices limits the possibilities. This limitation limits the project's ability to handle large datasets and necessitates the use of cloud computing for efficient data processing. Cloud-based solutions, on the other hand, have financial implications too.

Furthermore, due to the specialized nature of sign language, specific training and experience are required. While the system is intended to meet the needs of young users, the language acquisition patterns of children present a significant challenge. Children, unlike adults, may not acquire language skills quickly, making it a challenge to ensure their proficiency in using the system effectively.

VII. Conclusion

Because I didn't give that much time to modify the code to work in the best way and to modify the model architecture (change the number of heads, conformer blocks, decoder blocks, change the landmark indices, remove z, add or remove augmentation techniques), actually there are a lot of reasons behind this or maybe the whole idea of conformer is wrong.

Since there is no time to try these concepts to solve this problem, I think using clever augmentation from the first place solution with Flash-Attention, Squeezeformer, CTC with Conformer, or STMC transformer will give a better Performance. Try to solve this problem by using the simple solutions and then go deeper with more complex solutions. Thanks a lot if you reach this part. And try to use Supplementary dataset, since the top solutions in this competition used Supplementary dataset and they said it was a useful.

With doing a lot of experiments, changing the hyperparameters, implement different preprocessing techniques, different augmentation techniques, I think it will give you a very good accuracy.

References

[1] I. Sarker. Machine learning: Algorithms, real-world applications and research directions. sn comput. sci. 2, 160. https://doi.org/10.1007/s42979-021-00592-x.

[2] ——. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. sn comput. sci. 2, 420. https://doi.org/10.1007/s42979-021-00815-1.

[3] V. Wiley and T. Lucas, "Computer vision and image processing: a paper review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 29–36, 2018.

[4] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[5] P. Suetens, P. Fua, and A. J. Hanson, "Computational strategies for object recognition," *ACM Computing Surveys (CSUR)*, vol. 24, no. 1, pp. 5–62, 1992.

[6] S. M. Mohammad, "NLP scholar: A dataset for examining the state of NLP research," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari *et al.*, Eds. Marseille, France: European Language Resources Association, May 2020, pp. 868–877. [Online]. Available: https://aclanthology.org/2020.lrec-1.109

[7] E. Efthimiou, S.-E. Fotinea, C. Vogler, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, and J. Segouat, "Sign language recognition, generation, and modelling: A research effort with applications in deaf communication," in *Universal Access in Human-Computer Interaction. Addressing Diversity: 5th International Conference, UAHCI 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009. Proceedings, Part I 5*. Springer, 2009, pp. 21–30.

[8] M. Papatsimouli, P. Sarigiannidis, and G. F. Fragulis, "A survey of advancements in real-time sign language translators: Integration with iot technology," *Technologies*, vol. 11, no. 4, p. 83, 2023.

[9] M. Alaghband, H. R. Maghroor, and I. Garibay, "A survey on sign language literature," *Machine Learning with Applications*, vol. 14, p. 100504, 2023.

[10] W. H. Organization. (27 February 2023) "deafness and hearing loss". "https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss".

[11] N. G. Society". (October 19, 2023) "title sign language". "https://education.nationalgeographic.org/resource/sign-language/".

[12] A. G. "Bell, ""the question of sign-language and the utility of signs in the instruction of the deaf: Two papers by"," *"Journal of deaf studies and deaf education"*, vol. '10', no. '2', "2005".

[13] E. F. J. "182. Woll B, Sutton-Spence R, ""multilingualism: the global approach to sign languages."," *Sociolinguist Sign Lang*, vol. 8, no. 32, 2001.

[14] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7784–7793.

[15] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, "Extensions of the sign language recognition and translation corpus rwth-phoenix-weather." in *LREC*, 2014, pp. 1911–1916.

[16] B. Mocialov, G. Turner, and H. Hastie, "Transfer learning for british sign language modelling," *arXiv preprint arXiv:2006.02144*, 2020.

[17] A. Othman and M. Jemni, "English-asl gloss parallel corpus 2012: Aslg-pc12," in *sign-lang@ LREC 2012*. European Language Resources Association (ELRA), 2012, pp. 151–154.

[18] R. M. Kagalkar and S. V. Gumaste, "Mobile application based translation of sign language to text description in kannada language." *Int. J. Interact. Mob. Technol.*, vol. 12, no. 2, pp. 92–112, 2018.

[19] M. De Coster, D. Shterionov, M. Van Herreweghe, and J. Dambre, "Machine translation from signed to spoken languages: State of the art and challenges. arxiv 2022," *arXiv preprint arXiv:2202.03086*.

[20] R. Ruben, "Sign language: Its history and contribution to the understanding of the biological nature of language." *Acta oto-laryngologica*, vol. 125, no. 5, 2005.

[21] U. T. Services". (2023) "universal translation services () how much does a sign language interpreter cost?". "https://www.universal-translation-services.com/how-much-does-a-sign-language-interpreter-cost/".

[22] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," *journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[23] R. Ambar, C. K. Fai, M. H. Abd Wahab, M. M. Abdul Jamil, and A. A. Ma'radzi, "Development of a wearable device for sign language recognition," in *Journal of physics: conference series*, vol. 1019. IOP Publishing, 2018, p. 012017.

[24] A. Tharwat, T. Gaber, A. E. Hassanien, M. K. Shahin, and B. Refaat, "Sift-based arabic sign language recognition system," in *Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014*. Springer, 2015, pp. 359–370.

[25] Q. Yang, "Chinese sign language recognition based on video sequence appearance modeling," in *2010 5th IEEE Conference on Industrial Electronics and Applications*. IEEE, 2010, pp. 1537–1542.

[26] J. Forster, O. Koller, C. Oberdörfer, Y. Gweth, and H. Ney, "Improving continuous sign language recognition: Speech recognition techniques and system design," in *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies*, 2013, pp. 41–46.

[27] J. F. Lichtenauer, E. A. Hendriks, and M. J. Reinders, "Sign language recognition by combining statistical dtw and independent classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 2040–2046, 2008.

[28] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7361–7369.

[29] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4207–4215.

[30] N. Cihan Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3056–3065.

[31] J. Zhang, W. Zhou, and H. Li, "A threshold-based hmm-dtw approach for continuous sign language recognition," in *Proceedings of international conference on internet multimedia computing and service*, 2014, pp. 237–240.

[32] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3d convolutional neural networks," in *2015 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2015, pp. 1–6.

[33] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7784–7793.

[34] A. Orbay and L. Akarun, "Neural sign language translation by learning tokenization," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*. IEEE, 2020, pp. 222–228.

[35] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for continuous sign language recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 009–13 016.

[36] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 023–10 033.

[37] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human keypoint estimation," *Applied sciences*, vol. 9, no. 13, p. 2683, 2019.

[38] N. Arvanitis, C. Constantinopoulos, and D. Kosmopoulos, "Translation of sign language glosses to text using sequence-to-sequence attention models," in *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2019, pp. 296–302.

[39] A. Othman and M. Jemni, "English-asl gloss parallel corpus 2012: Aslg-pc12," in *sign-lang@ LREC 2012*. European Language Resources Association (ELRA), 2012, pp. 151–154.

[40] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1700–1709.

[41] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[42] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *arXiv preprint arXiv:1610.10099*, 2016.

[43] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International conference on machine learning*. PMLR, 2017, pp. 1243–1252.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[45] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[47] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[48] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[49] K. Yin and J. Read, "Better sign language translation with stmc-transformer," *arXiv preprint arXiv:2004.00588*, 2020.

[50] A. Duarte, S. Palaskar, L. Ventura, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. Giro-i Nieto, "How2sign: a large-scale multimodal dataset for continuous american sign language," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2735–2744.

[51] "Google". (2023) "google - american sign language fingerspelling recognition". "https://www.kaggle.com/competitions/asl-fingerspelling/data".

[52] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, and E. A. Chavez-Urbiola, "Loss functions and metrics in deep learning. a review," *arXiv preprint arXiv:2307.02694*, 2023.

[53] R. Haldar and D. Mukhopadhyay, "Levenshtein distance technique in dictionary lookup methods: An improved approach," *arXiv preprint arXiv:1101.1232*, 2011.

[54] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/fcf55a303b71b84d326fb1d06e332a26-Paper.pdf

[55] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.