DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Sign Language Recognition from a webcam video stream

Rifa Khan

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Sign Language Recognition from a webcam video stream

Erkennung von Gebärdensprache aus einem Webcam-Videostream

Author:	Rifa Khan
Supervisor:	Prof. Dr. Christian Mendl
Advisor:	Dr. Felix Dietrich
Submission Date:	January 15th, 2022

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, January 15th, 2022 Rifa Khan

Acknowledgments

I would like to express my sincere appreciation to my advisor Dr. Felix Dietrich and supervisor Prof. Dr. Christian Mendl for accepting the thesis topic and supervising it. Their support, guidance, and valuable feedback were crucial for the successful completion of this thesis.

I would like to thank my parents for always believing in me and providing me with opportunities, by going out of their way at times. They always trusted me with my decisions and supported me in my dreams. I am very grateful to my siblings and my friends who constantly motivated me and kept me going through the course of this thesis.

Being an international student is not an easy task. I am indebted to the people of WG 06, who became my second family and always supported me emotionally.

Abstract

Sign language recognition has been an active research field for almost two decades. From early electric signal-based sign language recognition to modern-day recognition using deep learning techniques, researchers all over the world have tried to automate this task. While sign language recognition could be seen as a naive gesture recognition problem, sign language does not translate to spoken language word by word. Translation of sign languages simply aims to detect the individual words from the individual signs used while signing a sentence, recognition majorly refers to detecting the complete meaningful text sentence communicated with signs. In this thesis, this translation issue of Sign Languages is addressed and several solution approaches are demonstrated. We mainly aim to carry out key point detection based sign language recognition (SLR) to infer the meaning that the speaker wants to communicate by generating captions. We use MediaPipe to collect the hand key points from images and OpenPose to collect holistic pose keypoints from videos. We work with American Sign Language (ASL), specifically, ASL image data set[1] and How2sign [1] data set of ASL videos. We use a fully-connected neural network with ReLU activation function to detect alphabet gestures from images. We achieve an accuracy of 83% and a precision of 90% for recognition of single alphabets. Additionally, we test the recognition for images captured through a webcam. We also provide the architecture of a model using transformer cells for recognition of complete sentences from sign language videos.

[1]https://public.roboflow.com/object-detection/american-sign-language-letters

Contents

1 Introduction

Sign Languages are the primary means of communication for over 5% of the world's population or 430 million deaf or hard-of-hearing people [2].The advent of technology has made communication easier and simpler for people using spoken or written languages. While, there has been considerable progress in machine comprehension of spoken languages in recent years [3], automatic sign language recognition, detection and identification still remains challenging. Even today, human interpreters are used majorly to bridge the gap between spoken and sign languages. This dependency on interpreters poses a strong need for automation of sign language detection and recognition.

Earlier research in this area focused mainly on the use of external devices such RGB [4] or depth cameras [5], sensor [6] or colored gloves [7]. These requirements for external devices limit the applicability to only when these devices are available. The developments in computer vision and machine learning architectures have assured possibility of getting rid of these overheads for sign language users. Successful recognition of sign language would not only lead to a more inclusive society, it will also allow the Deaf community access features like voice activated services, text-based systems, spoken media based content, etc.

Although, by nature, sign language recognition looks similar to the domain of action recognition, characteristic features of sign languages make this problem more challenging and one of its kind. Sign languages utilize multiple complementary channels to convey information [8]. This includes manual features such as hand shape, movement and pose as well as non-manual features, such as facial expression, mouth and movement of the head, shoulders and torso [9]. Thus, it is essential to observe the motion, expression and posture changes of the upper torso and not just the hand movements and posture for sign language recognition. In addition to these challenges, the variations in signs when performed by different signers, i.e., body and pose variations, variations in background and illumination, make this problem even harder [10].

Development of various pose estimation frameworks have opened a new possibility for sign language recognition tasks. These frameworks make it possible to infer positions of the various joints of the human body. By utilizing these pose estimation techniques to the Sign Language data sets, one can simply work with these joint positions in an image and a sequence of frames, instead of working with images and videos, respectively. This is exactly how this thesis work explores the problem Sign Language Recognition. Figure 1.1 depicts the two pipelines used for estimating human pose.

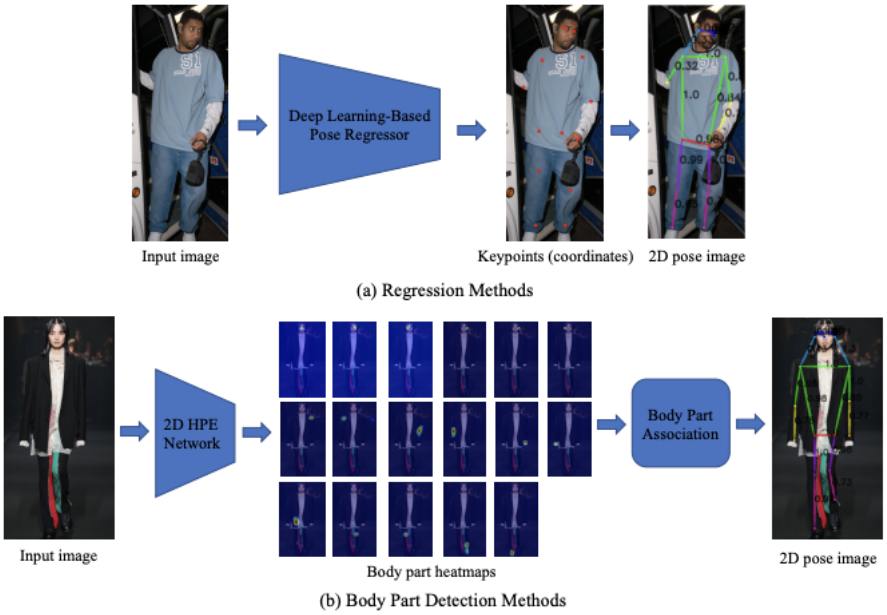


Figure 1.1: Two pipelines for single person 2D human pose estimation using deep learning methods as presented in [11]. In this thesis we mainly use MediaPipe [12] based on the upper pipeline.

This thesis aims at studying the complexity of sign language recognition, exploring the data sets available for research work in this domain and using human pose estimation frameworks to address the problem of sign language recognition. The organization of the thesis is outlined as follows. In chapter 2, we discuss the background of sign languages and the proposed methods, talk through the current research in the field of sign language recognition and highlight related works that use similar approaches (human keypoint estimation) and/or datasets for sign language recognition. Chapter 3 is dedicated to the data, model and approaches implemented in this work. In chapter 4, we discuss the results achieved and critically examine the same. This chapter also presents the main conclusions of this thesis and suggests improvements, possible directions and alternative approached for future work.

2 State of the art

This section briefs about the various important topics related to sign language recognition using human pose estimation and highlights the important research work in the related domains. We mainly discuss about researches related to sign language identification, detection and recognition, types of sign language recognition, American Sign Language recognition, pose estimation, gesture identification and language translation.

2.1 Sign Language

Sign languages have been a topic of research among the computer vision community for the past three decades. Formally speaking, Sign language [13] is a visual language performed with the dynamic movement of hand gestures, body posture, and facial expressions. As highlighted in chapter 1 the sign languages employ multiple channels. Understanding and using sign language requires considerable amount of time and effort. Additionally, sign languages are impacted with the change in language and culture (e.g., English and German sign languages are entirely different), thus making it more difficult for people to just learn them. The grammar of sign and spoken languages are very distinct. Among the various differences are different word ordering, use of direction and space in sign languages as opposed to spoken languages. The correlation between speech and sign is complex and there is no simple word to sign mapping [14]. In case of sign languages, the meaning of the same sign changes, depending on how many repetitions of the sign are performed [15]. Different signers perform sign language differently because of variations in individual's speed, localism, handedness, body shape, etc. [15], hence making it difficult to achieve the generality when trying to automate the task.

2.1.1 Sign Language Detection, Identification and Recognition

We begin by formally defining the three terms and then looking into past researches in the associated areas. Sign language detection [16], on one hand, is defined as the binary-classification task for any given frame of a video if a person is using sign-language or not, basically when something is being signed [17]. Sign language recognition, on the other hand, refers to identifying the meaning of the signs from the video or images [18, 19]. Sign language identification typically means identifying which sign language is being used to make the signs [17].

Most researches in the field of sign language processing focus mainly on sign language recognition and rarely speak about detection in its proper meaning. In [16], a multi layer

Recurrent Neural Network (RNN) is employed for sign language detection. [16] also emphasizes the lack of a proper publicly available data set for sign language detection, as most sign language data set are aimed at sign language recognition and are obtained in controlled environments. Hence, they worked on a data set created from videos from YouTube. They searched using a multitude of keywords to obtain the sign videos, thus creating a rich set comprising of different sign languages, single and multiple signers, natural signing, complex camera and signer motion. For the non signer videos, they included videos such as people speaking with hand gestures, miming, hand exercise videos, etc. Another prominent work in this field employs the use of human pose estimation [17]. This, rather recent work aims at distinguishing when exactly something is being signed in video conferences. They mainly work with Public DGS Corpus (German Sign Language) [20], using full body pose estimation and achieved a prediction accuracy of 87%-91%.

Quite similar to sign language detection, studies on sign language identification have been scarce. Early research such as [21] propose use of random forest for sign language identification. These systems used low level visual features and were able to differentiate between British Sign Language and Greek Sign Language with average F1 score of about 95%. In [22], authors use features learned by unsupervised techniques to identify six sign languages and achieve an average accuracy of 84%. A later work [23] extends on the work by [21] and distinguish British Sign Language (BSL) and French Sign Language (LSF) videos with static backgrounds with F1 score of 98%. An F1 score of 70% was obtained when identifying American Sign Language (ASL) and British Sign Language (BSL) videos found on video sharing sites.

Sign language recognition has been a hot research topic for almost past 30-40 years. One of the earliest work in this research area dates back to 1983 with the invention of gloves with sensors to detect flex of finger joints [24]. In this system discrete hand positions were translated into electrical signals representing alpha-numeric characters [24]. For the majority of the period from 1990 to 2000, researches were using statistical methods such as Hidden Markov model for sign language recognition [25, 26]. These systems were mostly signer dependent and aimed at isolated sign recognition. In [26] authors additionally discuss an approach for continuous sign language recognition. By 2000, researchers were already looking at local feature recognition and using clustering techniques for sign language recognition [27]. With the development of deep learning, researchers started exploring methods for learning general video and time-series representation(e.g., RNN, LSTM) and also frameworks for action recognition for SLR tasks [28, 29, 30]. Later attention modules were also used in combination to attain higher accuracy [31, 32] and also two stage pipelines with semantic detection and segmentation models were being used to maneuver recognition network [33].

2.1.2 Types of Sign Language Recognition

Past researches in the field of sign language recognition can be differentiated using a number of factors such as type of data (image or video) they deal with, vocabulary size, target sign

language, isolated, continuous or contextual recognition, etc. In this section, we highlight how researches have been affected by these factors, thus listing out the research trends in this field. In the subsequent subsections, we discuss more about Isolated and Continuous SLR and also researches targeting American Sign Language Recognition.

Researches targeting sign language recognition have been using different data modalities. While RGB is the most popular input data types for small and larger vocabulary ranges, colored gloves remain limited to smaller and medium vocabulary recognition tasks [34]. Depth information usage as input became popular after the release of Kinect sensor in 2010. Researches in this area also differ in terms of the features or parameters that were collected from the above discussed input modalities and used for sign language recognition. Hand shape has been the most covered parameter, followed by location and movement [34]. Global features such as body joints, full-frame, depth and motion became increasingly popular for language recognition because of the shift to deep learning techniques.

2.1.3 Isolated and Continuous SLR

The different Sign Language Recognition tasks can be roughly divided in two categories: in [35]:

- **Isolated Sign Recognition**: Sign Language Recognition methods belonging to this category aim to recognize a single letter or word (in the form of sign) at a time [35]. In [25] authors performed signer dependent recognition of 262 different signs in videos. They utilized hidden Markov modelling and considered a sign as a doubly stochastic process, represented by an inobservable state. Lim et al. [36] proposed a two phased isolated sign language recognition system. In the first phase called hand tracking, hand patches are extracted to pre-train Convolutional Neural Network (CNN) hand models and hand tracking is performed by particle filter. A square hand region centered around predicted hand position is served as input to second phase. In the second phase, called hand representation, a compact hand representation is computed by averaging the segmented hand regions. These hand representations were called "Hand Energy Image (HEI)" and outperformed other methods of Isolated SLR. Though over the years, the recognition rates have improved as much as 96% [10], proposed methods for isolated SLR still fail to correctly recognize very similar signs, specifically those that have similar hand trajectories.

- **Continuous Sign Recognition**: Continuous Sign Language Recognition (CSLR) aims identifying one or more complete sentences or finger spelled signs as continuous data [35]. These methods have the characteristics that can prove to be most suitable for real time SLR applications [37]. Continuous Sign Language Recognition deals with the problem of occlusion and has to identify sign gestures from the transition movements. Additionally there are no pauses between the signs, hence making it difficult to identify when the sign for a particular word is completed and the next sign starts. Bauer et al. developed a CSLR system consisting of one continuous density Hidden Markov Model

(HMM) for each sign [38]. During the recognition task, beam search was employed and they achieved an accuracy of 91.7% on a lexicon of 97 signs of German Sign Language (GSL). In [19], authors propose a weakly supervised framework with deep neural networks for vision based CSLR. The dataset used for training had ordered gloss labels but not the exact temporal locations for the videos of sign sentence. They utilized Recurrent Neural Networks (RNN) for spatio-temporal feature extraction and sequence learning of video segments to glosses. A spatial-temporal multi-cue (STMC) network was proposed in [39] to learn the implicit visual grammars by collaboration of different visual cues. The STMC network consist of two modules: spatial multi-cue (SMC) module that decomposes visual features from different cues, and temporal multi-cue (TMC) module that models temporal correlations. The proposed network achieved new state-of-the-art performance on three large scale CSLR benchmark datasets: PHOENIX-2014, CSL and PHOENIX-2014-T.

Until 2000, the growth in the number of studies targeting isolated sign language recognition has been exponential whereas, this growth is almost linear for continuous sign language recognition studies [34]. This is majorly because of the complexity of the continuous sign language recognition problem and the less number of available datasets for training purposes. On average, the number of studies for isolated sign language recognition are almost double of the studies focusing on CSLR. Also, most isolated SLR works model smaller vocabulary size comprising of below 50 signs [34]. For continuous SLR, studies targeting large vocabulary sizes (i.e. greater than 1000 signs) have been low until 2015, and these studies and also those focusing on a vocabulary size of 50 to 200 (small vocabulary) have experienced a gain in the number of published results since 2015 [34]. This was majorly because of the two benchmark datasets, namely RWTH-PHOENIX-Weather 2014 [40] (vocabulary size of 1080 signs) and CSL corpus [32] (vocabulary size of 178 signs), being the focus of the research community.

2.1.4 American Sign Language Recognition

Most researches aiming at sign language recognition or translation are conducted on sign languages corresponding to the researchers native language. American Sign Language (ASL) has the most published results with Chinese Sign Language (CSL) being the second most frequently researched sign language [34]. This is majorly because of the popularity of the English language. This thesis work also focuses on American Sign Language Recognition. In this section, we discuss the past researches targeting American Sign Language, highlighting the dataset used, basic model used for recognition and the achieved performance.

In [7], authors look into the potential of Kinect depth-mapping camera for sign language recognition and verification for educational games for deaf children and compared its performance against a system using colored gloves. Garcia et al. used transfer learning for sign language recognition and presented a real time finger spelling translator, utilizing GoogLeNet [41] architecture, for American Sign Language letters [42]. In [33], authors tackle the problem of figerspelling recognition of ASL alphabets "in the wild", i.e. from naturally occurring video

data collected from the websites (YouTube, aslized.org and deafvideo.tv) and not from videos specially collected for recognition tasks. Attention based recurrent encoder-decoders and CTC-based approaches were explored for sequence modeling and an accuracy of 42% was achieved using a CTC-based recognizer. One of the research working with body pose and hand shape features for ASL recognition is [43]. Authors use trajectories of estimated 2D skeletal data from videos and embeddings of hand images. Because of the use of skeletal data, proposed model is signer independent. The model was trained and evaluated on GMU-ASL51 dataset [44] of 12 users and 51 ASL gestures and showed superior performance compared to baseline models.

In [45], researchers studied recognition of ASL alphabets and numerals on four publicly available ASL image datasets. They propose use of a convolutional neural network (CNN) model and realize an improvement in accuracy by 9%. Another study using CNN to extract spatial features and a RNN to train on temporal features is [46]. The study focused on a dataset created by authors comprising of videos with ASL signs were made by a single signer. In [47], authors propose to use a YOLOv5 based solution for American Sign Language Recognition. MU_HandImages_ASL dataset [48] was used to train and evaluate the proposed model and a precision of 95% was achieved.

Recently in [49], authors look at dynamic ASL recognition using 3D CNNs. The CNN is trained on Boston ASL LVD (Lexicon Video Dataset) to classify 100 words. This work achieved a precision of 3.7% with a computing time of 0.19 seconds per frame, leading to a possibility of real time usage. Lee et al. [50] developed an application prototype with the use of leap motion controller. They used Long-Short Term Memory Recurrent Neural Network with k-Nearest Neighbor as the classification method. Proposed model was trained on 2600 samples with 100 samples of each alphabet of ASL and achieved an accuracy rate of 99.44%. Hosain et al. [51] present a method that uses both motion and hand shape and body pose information for word-level sign recognition from ASL video. They pooled spatio-temporal feature maps from different layers of 3D CNN and attained improved performance on WLASL benchmark dataset [29].

2.2 Pose Estimation

Human Pose estimation(HPE) involves estimating the configuration of human body parts from input data captured by sensors, specifically images and videos [11]. It is used for a variety of applications such as human computer interaction, activity recognition, augmented reality, etc. Human pose estimation is divided into two main categories, namely 2D HPE and 3D HPE. Most human pose estimation methods use an N-point rigid kinematic model to represent key points and features extracted from input data.

2.2.1 2D Human Pose Estimation

2D Human pose estimation estimates the 2D position of human body key points from images or videos [11]. Traditionally, researchers used hand crafted feature extraction techniques for human body parts and described human body as stick figure. A two layer random forest was employed as a joint regressor in [52], where the first layer act as a discriminative, independent body part classifier and the second layer predicts the joint locations by modeling the interdependence and co-occurrence of the parts. Similar to this, [53] utilizes a deformable part model with k poselet parts for keypoint prediction.

With the development of deep learning methods, high performance have been achieved for 2D human pose estimation from images and videos. Two kinds of pipelines are generally used for single person 2D pose estimation, namely, regression methods and body part detection methods. Regression methods directly learn a mapping from the original image to the human body model and produce joint coordinated [11]. Deep neural networks are generally used to learn this mapping.

Body part detection methods predict approximate body joint locations using the supervision of heatmaps [11]. The working of 2D single person human pose estimation methods is shown in fig. 1.1.

Lot of researchers have used the regression pipeline for human pose estimation [54, 55, 56, 57] among others. In [54] authors use deep convolutional neural networks (ConvNets) to estimate human pose from videos. Carreira et al. [55] proposed a self correcting model that progressively changed an initial solution by feeding back error predictions, this process was termed as Iterative Error Feedback (IEF). In another work by Luvizon et al. [56], authors suggested use of soft-argmax function to convert feature maps directly to body joint coordinates in a fully differentiable framework to estimate human pose from still images. In [57], Zhang et al. presented a Fast Pose Distillation (FPD) model learning strategy, which is a lightweight variant of Hourglass network [58] and training is done in small pose networks in knowledge distillation fashion [59].

Body part detection methods predict the positions of the body joint by training a body part detector. Recent methods view pose estimation as a heatmap prediction problem, where K heatmaps $H_1, H_2, ..., H_k$ are estimated for K keypoints. The pixel value of $H_i(x, y)$ in each heatmap is indicative of the probability that the keypoint lies in position (x,y). There has been a growing interest in detecting human poses by using heatmaps. Some of the works in this direction are [60, 61, 62, 63]. Authors proposed a hybrid architecture in [60] consisting of deep Convolutional Network and Markov Random Field, exploiting geometric relationships between body joint locations, for articulated human pose estimation in monocular images. Wei et al. [61] introduced a convolutional networks based sequential framework named Convolutional Pose Machines (CPM). This framework predicted locations of key joints using a multi stage process. Luo et al. [62] built a novel recurrent architecture with Long Short-Term Memory (LSTM) for pose estimation in images and videos. The LSTM cells captured temporal

geometric consistency and dependency between different frames, thus reducing the time for training HPE network for videos. UniPose, a unified framework with a ResNet backbone and Waterfall module for human pose estimation was proposed in [63]. This framework incorporates contextual segmentation and joint localization to predict human pose in a single stage for both single images and videos.

2.2.2 3D Human Pose Estimation

3D human pose estimation aims at finding approximate positions of the human body joints in 3D space [11]. While 2D human pose estimation have achieved significant performance, 3D human pose estimation is still a challenging task. One of the major limitation is the lack of large datasets as accurate 3D pose estimation is time consuming and manual annotation is not practical. Most 3D human pose estimation employ images and videos captured from monocular camera. Deep learning based 3D human pose estimation methods are broadly divided into single-view and multi-view human pose estimation.

Liang et al. [64] proposed a scalable neural network framework to reconstruct 3D mesh of human body from multi-view images. Using multi-view images reduced the projection ambiguity and helped in increasing reconstruction accuracy. In [65] authors build a system to predict 3d positions from the given 2d joint locations. This lifting of 2d joint locations to 3d positions was performed using a simple deep feed forward network with a low error rate. In [66] researchers discuss a solution for multi-human 3D pose estimation from multiple calibrated camera views. The authors exploit the temporal consistency in videos and retain the 3D pose for each person and update it iteratively using cross-view multi-human tracking.

2.2.3 Pose Estimation Frameworks

Presently, there are several popular models to perform human pose estimation such as OpenPose, PoseNet, BlazePose, DeepPose, DensePose and DeepCut. In this section we discuss about MediaPipe and OpenPose which have been used to collect keypoints from images and videos. MediaPipe is an open-source cross-platform framework for building multimodel machine learning pipelines. It is based on the BlazePose model and can be used to implement human face detection, multi-hand tracking, keypoint detection, object detection and tracking and so on.

2.3 Gesture Identification through Pose Estimation

Gesture identification refers to recognizing specific human gestures and using them to convey information or for command and control purposes [67]. Gesture identification is a very popular research field due to its applications in human-computer interaction, robotics, home automation, etc. This field still remain challenging because of the complexity of human motion. While gesture and pose might seem to be similar terms, gesture focuses more on hand movement rather than shape, as in case of pose. In this section we briefly look into how

gesture identification/recognition methods have developed over the years and later on dive in detail on the researches focusing on gesture recognition using human skeletal data, i.e. estimated pose information.

The very earliest methods for gesture identification involved use of sensor gloves. While these sensors could provide exact coordinates of palm and finger locations, they required the user to be connected to the computer physically. Also, these sensors were quite expensive, thus limiting their use for general people. To overcome this dependency on physical sensors, researchers started looking into computer vision methods to identify gestures. Use of computer vision led to the development of color based, motion based, appearance based, depth based and skeleton or pose based methods for gesture recognition.

Earlier, gesture identification was mainly done using conventional classification techniques and handcrafted features. Support Vector Machine (SVM) [68, 69], Hidden Markov Models (HMM) [70] were among the popular classical methods for gesture identification. With the advent of deep learning, researchers started working with methods based on Convolutional Neural Networks (CNN) [71], Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Networks [72]. These methods have achieved impressive performances compared to traditional methods but certain challenges still remain because of occlusion, multiple people in the background, poor lighting conditions, etc. To overcome these challenges, researches now focus on combining multiple modalities of input data such as skeleton joint information, RGB and depth frames, human body shapes, etc. One such recent work [73] proposes a multi modal gesture recognition method for RGB data input with a multi modal algorithm. The algorithm uses three sub models: two 3D convolutional neural networks based on ResNet architecture (3DCNN_ResNet) [74] to perform on RGB images and color body part segmentation, and a long short term memory network (LSTM) to work with 3D human skeleton joints.

Pose estimation based gesture recognition are the methods where representations of skeleton data are used for classification. The most common pose estimation based features used for gesture recognition include joint orientation, the space between joints, trajectories and curvature of the joints [67]. In [75], authors used a combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for automatic recognition of hand gesture and achieved an overall accuracy of 85.46% on the dynamic hand gesture-14/28 dataset. Authors discussed a two stage training strategy that first focused on CNN training and secondly, on CNN+LSTM network in [76], for human activity and hand gesture recognition using 3D data sequences obtained from full-body and hand skeletons. Nguyen et al. propose a neural network based on SPD manifold learning for skeleton-based hand gesture recognition [77]. The discussed pipeline work in three stages: first the convolutional layer is used to increase the discriminative power, second stage performs spatial and temporal aggregation of joint features and lastly, third stage learns an SPD matrix from skeletal data. In [78], a combination of 3D hand pose estimation, data fusion and deep neural network is used to improve recognition accuracy of dynamic hand gestures. A 3DCNN + ConvLSTM

framework is used to classify the combined dynamic hand gesture data (RGB, depth and 3D skeleton data) and an accuracy of 92.4% is achieved.

2.4 Language translation problems

Language is the most significant means of communication for humans. The need for language translation emerges when people speaking different languages interact. This need has increased enormously in current world of globalization. In this section, we discuss the major issues in automated language translation or neural machine translation, highlighting past research trends in thus area and then look into researches in area of neural sign language translation and recognition.

2.4.1 Neural Machine Translation

Machine translation refers to converting a natural source language into another natural target language by computer [79]. Though there is still no system that provides "fully automatic high-quality translation" (FAHQT), many programs such as Google Translate are available that provide useful output. Deep neural networks do not possess the property required for translating a sequence of words of one language into a sequence of words in another language. For this, encoder-decoder or sequence-to sequence models were developed. One such work [80], authors propose a RNN Encoder-Decoder consisting of two recurrent neural networks (RNN). One RNN encodes a sequence of symbols into a fixed length vector representation, and the other decodes the representation into another sequence of symbols. Multi-layer LSTM encoder-decoder architecture was used for English to French translation task on VMT'14 dataset and a BLEU score of 34.8 was achieved on the test set in [81]. These encoder decoder architectures are well suited for smaller phrases but fail to translate longer sequences. This limitation led to the development of attention mechanism. Bahdanau et al. proposed extension of encode-decoder architecture using attention mechanism to search for parts in source sentence that are relevant for predicting target words [82]. Later works evolved into a new simple network architecture, the Transformer, that were solely based on attention mechanism instead of recurrent and convolution units [83]. These transformer architecture showed better performance and lesser training time for translation tasks. In [84], an Action Transformer model is presented for human action recognition and localization in video clips. This model outperformed the existing state-of-the-art by a significant margin. Transformer-XL consisting of segment-level recurrence and novel positioning scheme were proposed in [85]. The presented architecture learns 85% longer dependency than RNNs and 45% longer dependency than vanilla Transformers [85]. Generative Pre-Trained Transformer 3 (GPT-3) [86] showing superior performance on translation tasks were proposed recently.

2.4.2 Neural Sign Language Translation and Recognition

Bragg et al. [87] define sign language recognition, generation and translation as an interdisciplinary field, which requires knowledge of computer vision, computer graphics, natural

language processing, human-computer interaction, linguistics, and Deaf culture. In [18], authors use CNN to to extract frame level spatial representations and sequence-to sequence model with attention mechanism to translate sign language to spoken language. They used an updated version of RWTH-PHOENIX2014 dataset and tested two approaches namely sign-to-text and sign-to-gloss-to-text, where sign-to-gloss-to-text approach showed more promising results. In a later work [14], they used a transformer architecture and sign-to-text approach achieving better results than the previous work. Authors proposed a sign language translation system based on human keypoint estimation in [88]. This work focuses on Korean Sign Language Translation using KETI (Korea Electronics Technology Institute) dataset, comprising of 14,672 videos of high resolution and quality. They utilized a sequence-to-sequence architecture for translation where human keypoints extracted from a face, hands, and body parts as input. Their translation model achieved an accuracy of 93.28% and 55.28% on validation and test set respectively. An interesting work by Saunders et al. [89] talks about Sign Language Production (SLP) to translate spoken language to a continuous stream of sign language video. They proposed use of Progressive Transformers to translate from discrete spoken language sentences to continuous 3D skeleton pose outputs that represent sign language.

3 Sign Language Recognition (SLR) from webcam video stream

This thesis aims to study American Sign Language (ASL) recognition by employing features, namely human body joint coordinates acquired from human pose estimation. In this chapter, we discuss the datasets used, the frameworks utilized and the models that were implemented and used for various experiments, and the obtained results. This thesis work was divided into two phases: the first phase focused on understanding the problem of sign language recognition and how key points detected from the human body can be used for sign language recognition task. Due to the limitations of the author in terms of language understanding, this work only focuses on American Sign Language, though some of the discussed experiments and techniques are inspired by earlier works dealing with other sign languages. We started with developing a basic understanding of the simple American Sign Language alphabet dataset. A brief overview of this dataset is presented in the later sections. We performed a basic visual analysis of the dataset to understand the general data distribution and presence of outliers. This was followed by experimenting with different classifiers for sign alphabet recognition on this dataset. In the second phase of the thesis, we moved to the more complex problem of detecting complete sign language sentences from videos, namely continuous sign language recognition. For this phase, How2Sign dataset was used for training and evaluation. This chapter paints a detailed picture of the entire thesis work while highlighting the various important details and intricacies.

3.1 American Sign Language

To better understand the problem of sign language recognition, we need to understand intricacies of sign languages. In this section, we provide a brief outline of how signs in different sign languages are signed, specifically focusing on American Sign Language (ASL).

There are over 300 sign languages in the world that are used by Deaf and Hard-of-Hearing (DHH) people to communicate. Some of the popular sign languages include American Sign Language (ASL), Chinese Sign Language (CSL), German Sign Language or Deutsche Gebärdensprache (DGS), and British Sign Language (BSL). Most sign languages are completely independent of their spoken counterpart and have their own grammar and lexicon, for example, even though British and American English are quite similar, British Sign Language (BSL) and American Sign Language (ASL) are a lot different. Signs used in sign languages are arbitrary and often do not necessarily relate to the referred meaning visually. Sign languages often use simultaneous expressions because of their visual nature.

(a) Sign for alphabet "K"

(b) Sign for alphabet "P"

Figure 3.1: Hand gestures for letter "P" and "K" are similar but differ in orientation

According to [43] approximately 6000 gestures are used to sign common words in American sign language. These gestures are characterized by fast, highly articulate motions of the upper body, including arm movements with complex hand shapes and facial expressions [51]. The words that are obscure and proper nouns are signed using fingerspelling, where the signer spells out the word by signing for the individual alphabets. Other than hand postures and facial expressions, important informative cues are provided by the motion of particular body parts such as hand-tip, neck, and arm. ASL uses 19 hand shapes to sign 26 alphabets of the English language (called as American manual alphabet). This is achieved, for instance, by changing the orientation of the same hand gesture to refer to different alphabets, e.g., signs for "P" and "K" use the same handshape but different orientation. This can be seen in fig 3.1 While all the signs for alphabets are done by using one hand, a sign for a word may also use both hands. All the signs can be described using the five parameters, namely hand-shape, movement, palm orientation, location, and non-manual markers (these include movement of the eyebrows, the cheeks, the nose, the head, the torso, and the eyes) [90]. While signing ASL, the signer usually signs the subject, verb, and object of the sentence. The order in which signs for the subject, object, and verb is done can differ for the same sentence and is affected by various factors. Fig. 3.2 provides an example of the two ways in which a simple sentence taken from [91] can be signed. In figure 3.2a a subject-verb-object (SOV) order is used for signing, while for the sentence in figure 3.2b, object-subject-verb (OSV) order is utilized. This is the case where an object is made the topic of the sentence and moved to the sentence-initial position. This is realized in sign by a forward head-tilt and a pause. The signs for these subject, verb and object parts of a sentence are mostly performed without pauses,

thus making it difficult to mark the boundary between different signs.

(a) subject-verb-object (SOV) order of signing

(b) object-subject-verb (OSV) order of signing

Figure 3.2: Example sentence to show how different order of signs for subject, object and verb can be interpreted as the sentence with same meaning. The bold italicized text are the words, corresponding to which signs are performed. The lower normal text is the conveyed meaning.

3.2 Datasets

For any Machine Learning problem, the choice of the dataset plays a very important role. In this section, we provide a concise overview of the various publicly available datasets for American Sign Language and other benchmark datasets for Sign Language Recognition research and later provide a detailed description of the datasets used in this thesis work. We specifically highlight the vocabulary size, available data modalities, and the target sign language for the datasets.

Following are the various popular publicly available datasets for continuous sign language recognition research:

- **Video-based CSL** [32]: It is Chinese sign language dataset for continuous sign language recognition. The dataset was recorded using 50 signers, where each signer performs each sentence 5 times. It contains a total of 100 hour of video duration that covers 178 sentences of Chinese sign language vocabulary. The dataset includes RGB videos (resolution of 1280×720 and frame rate of 30 fps), depth videos (resolution of 512×424 and frame rate of 30 fps) and 25 skeleton joint locations of each frame.
- **SIGNUM** [92]: It is a video dataset of German Sign Language (DGS) for signer-independent continuous sign language recognition. It covers a vocabulary of 450 basic signs and includes video of 55 hours of duration. The corpora comprises of 780 sentences performed by 25 native signers of different sexes and ages. The dataset includes RGB videos of resolution 776×578 and frame rate of 30fps and gloss level annotations.
- **RWTH-Phoenix-2014T** [18]: This dataset contains spoken language translations and gloss level annotations for German Sign Language videos of weather broadcasts. The dataset contains more than 67K signs (vocabulary of 1K signs) and 99K words (vocabulary of 2.8K). The dataset comprises of 11 hours of RGB videos, transcription and gloss level annotations. The signs have been performed by 9 signers.

- **Public DGS Corpus** [20]: It is an extension of DGS corpus that was built as a reference for German Sign Language. The dataset includes 50 hours of dialogue and natural signing on 20 different topics by 330 signers belonging to different sex, age and religious groups. The dataset after extension, also includes pose information collected through OpenPose.
- **BSL corpus** [93]: It is a machine-readable digital corpus of British Sign Language (BSL). The dataset covers a vocabulary of 5K signs performed by 249 signers and includes gloss level annotations.
- **Boston104** [94]: It is a video dataset of American Sign Language (ASL). The data was collected from 3 native signers using multiple synchronized digital cameras to capture different views of the signer. The dataset is rather small and cover a vocabulary of 104 signs.
- **NCSLGR** [95]: It is a dataset of ASL and includes multi-view videos data more than 5 hours covering a vocabulary of 1.8K signs for sign language recognition tasks. The signs are performed by 4 signers.
- **How2Sign** [1]: It is a multi-modal multi-view video dataset for American Sign Language Recognition. Section 3.2.2 discusses the dataset in detail.

3.2.1 ASL alphabet dataset

This section is dedicated to the image dataset used in the thesis work. Each image in this dataset corresponds to an alphabet sign from the American Sign Language (ASL). The dataset was downloaded from roboflow [96]. Figure 3.4 provides example images corresponding to different alphabets of ASL from this dataset. The dataset contains 1728 images in total with 1512, 144, and 72 images in the train, validation, and test split respectively. Figure 3.3 provides information about class balance and depicts the number of images per alphabet present in the test, train, and validation splits of the used dataset.

The dataset has been recorded by a single person signing various alphabets of ASL. The images are captured from different views and angles for the various alphabets. The dataset contains 720 unique images that have been augmented using a number of techniques namely, horizontal flip, crop (zooming in by 20%), rotation between -5° and +5°. Almost 10% of the images are grayscaled, some have undergone brightness changes and some are blurred. These augmentations have been applied randomly and led to a total of 1752 images. The background is variable for the different images, for some it is a plain white background while for other images, the background is rather cluttered with instruments and things lying around. These variations in background make this dataset suitable for training robust systems.

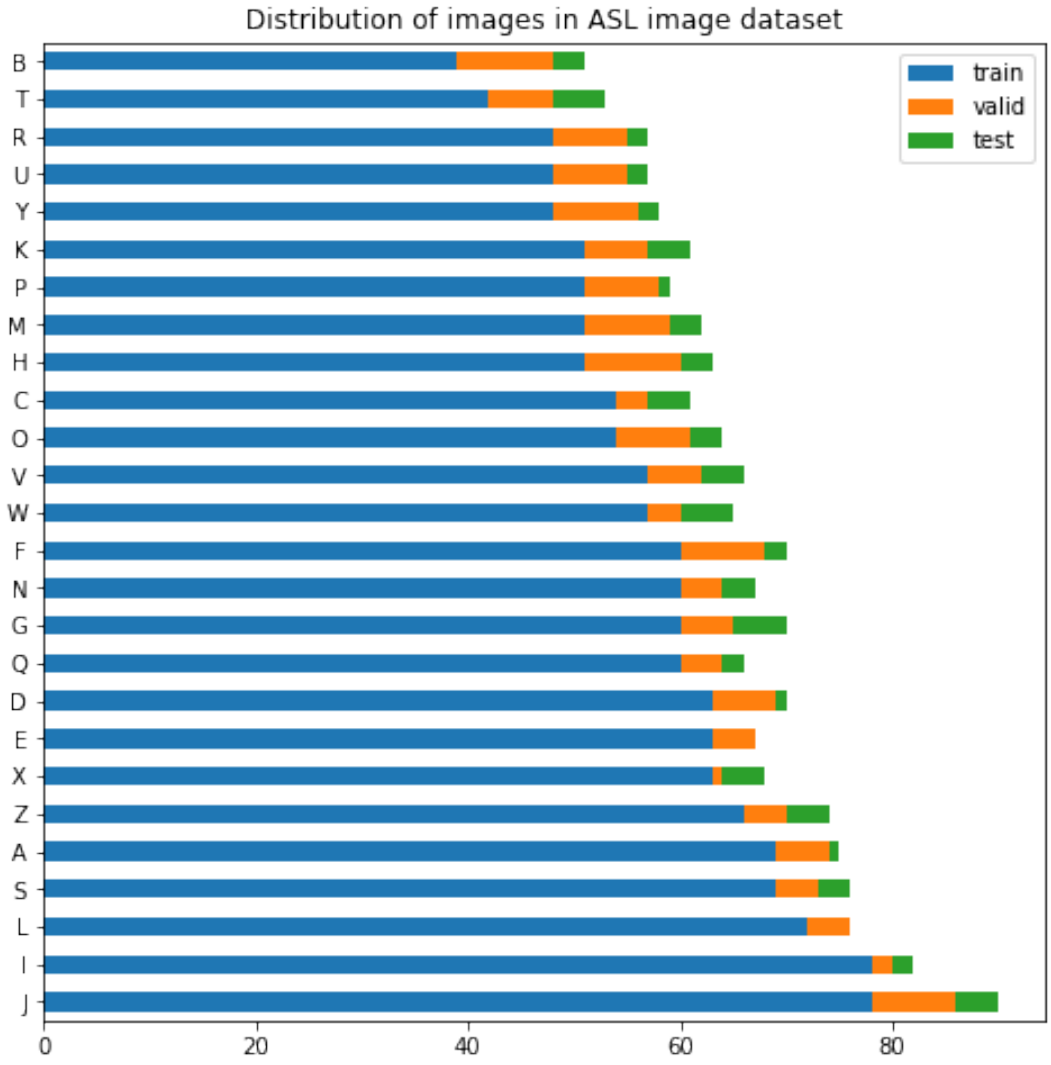


Figure 3.3: Bar graph showing distribution of images of various alphabets across train, validation and test splits in the image dataset used for this thesis work.

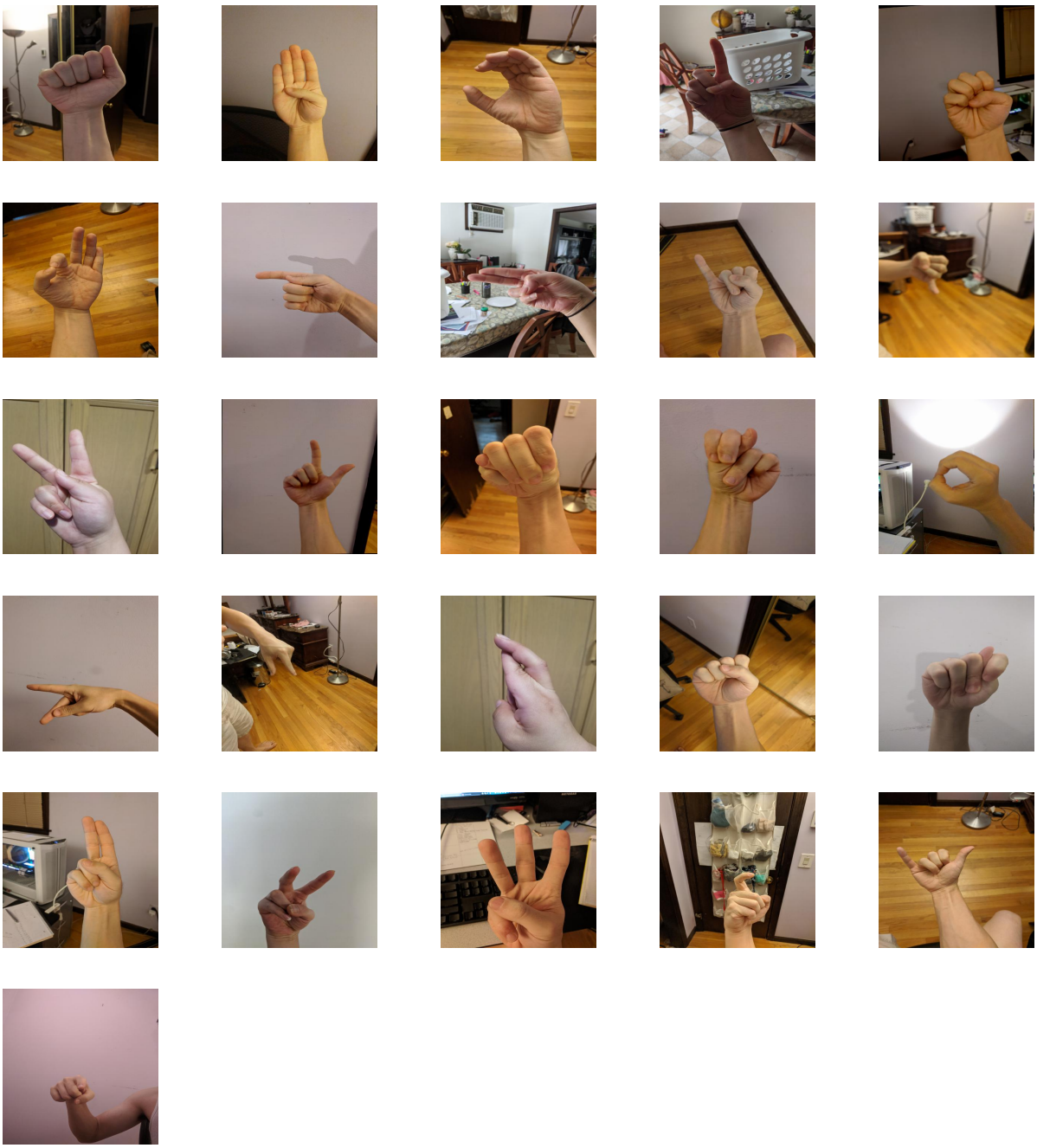


Figure 3.4: Some example images of the 26 alphabets from the image dataset[1] for American Sign Language. The images read as A to Z, starting from top left image and reading from left to right. Recognition of the signs from these images is a challenge for the machines, as they have different backgrounds.

3.2.2 How2sign

One of the limitations in sign language recognition research has been the absence of large annotated datasets. To overcome this limitation, a dataset was curated by collaboration of Universitat Politècnica de Catalunya, Barcelona Supercomputing Center, Carnegie Mellon University, Facebook research and Gallaudet University, namely How2Sign [1]. How2sign is a multi-modal and multi-view continuous American Sign Language (ASL) data set. Previous available datasets for Continuous Sign Language Recognition, such as RWTH-Phoenix-2014T, Boston104 (refer to section 3.2), etc. either had limited vocabulary size, short video or total duration and limited domain. How2Sign dataset provides comparatively larger vocabulary than previously available datasets and targets continuous sign language domain. This is also the first sign language dataset that contains speech because it has been created keeping in alignment with the existing How2 [97] dataset.

How2Sign consists of a parallel collection of almost 80 hours of instructional sign language videos and other corresponding modalities like speech, English transcripts and depth information. A three hour subset additionally has detailed 3D pose estimation. 11 people were used for collection of this dataset, these people are referred as signers. Out of these 11 signers, 5 people identified themselves as hearing, 4 as Deaf and 2 as hard-of-hearing. Out of the 5 hearing signers, 4 were professional ASL interpreters and one was ASL fluent.

For purpose of recording the dataset, the signers were first made familiar with the content of How2 videos by watching the video with the transcript as subtitles. After this, ASL translation videos were recorded while signers were watching the corresponding video from How2 dataset with subtitles and a slower speed of 0.75. The recordings were done under supervision in two different locations, namely the Green Screen Studio and the Panoptic studio. The complete 80 hours of dataset was recorded in the green screen studio and a smaller 3 hour subset chosen from test and validation split was re-recorded in the panoptic studio. Later on, these recorded videos were trimmed and cut into sentence level clips.

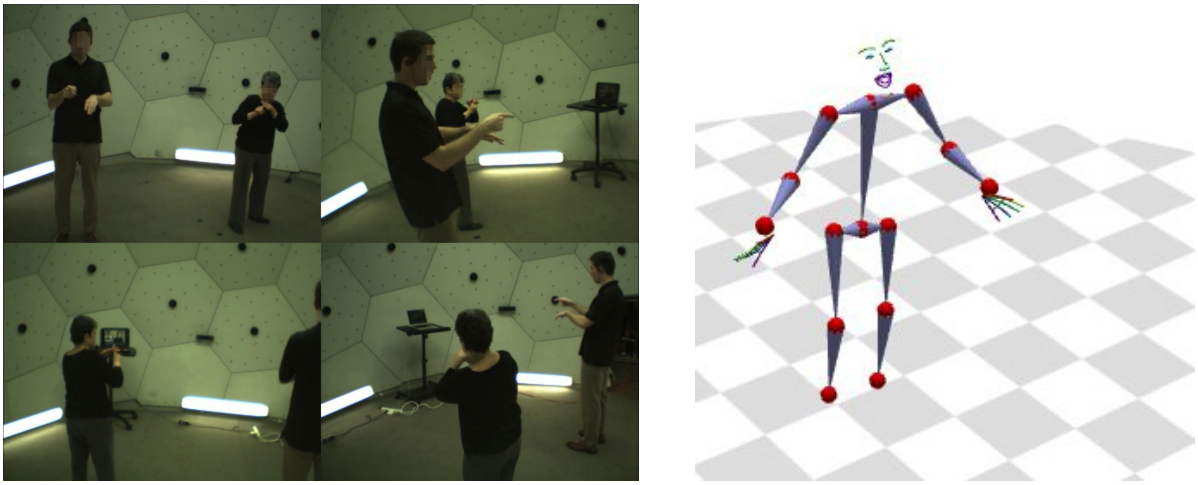
Videos recorded in the green screen studio were recorded from a frontal and lateral view. A depth and a high-definition (HD) camera was placed in the front and another HD camera in a lateral position to acquire these views. The recorded videos have a resolution of 1280×720 resolution and a frame rate of 30 fps. An example of the data recorded in green studio is presented in figure 3.5. The Panoptic studio [98] is a system with synchronized 480 VGA cameras, 30 HD cameras and 10 RGB-D sensors. This system also has the capability of estimating 3D keypoints of the signers. Figure 3.6 gives a snapshot of the data recorded in panoptic studio and the estimated 3D pose.

How2Sign contains multiple data modalities that include multi-view videos, English transcription, Glosses, Pose (2D and 3D keypoint information), Depth and also associated speech. These modalities are either automatically collected, extracted or taken from How2 dataset. The videos collected from multiple views help to reduce occlusion and vagueness in case of

(a) Sample of the video recorded from front view

(b) Sample of the video recorded from side view

Figure 3.5: Stills from the videos recorded from the two cameras placed in different positions in the green screen studio from How2Sign dataset.



(a) Multi-view VGA and HD videos

(b) Estimated 3D keypoints

Figure 3.6: Stills from the videos recorded in panoptic studio and the estimated 3D pose from How2Sign [1] dataset.

hands specifically. The English transcriptions are extracted from How2 dataset and aligned at the sentence level. Gloss, in terms of language, indicate what individual part of a sign mean. This is not the true translation but provides appropriate spoken language morpheme to express the meaning of signs in spoken language [99]. These glosses are not yet part of the publicly available version for download. Human pose information, specifically body, hand and facial keypoints are extracted for all the recorded videos. For the 80 hour portion recorded in green studio, the 2-dimensional (2D) pose information was extracted using OpenPose [100]. This pose information consists of 25 body keypoints, 70 facial keypoints and 21 keypoints extracted from each hand and is provided for both frontal and lateral view of the videos. The extracted keypoints for lateral view were not released publicly at the time of this thesis work. Figure 3.7 provides a sample of the extracted pose information from a video recorded from front view. For the 3 hour part recorded in panoptic studio, 3-dimensional (3D) pose information is extracted using Panoptic studio internal software [98]. The dataset also contains depth information collected from a frontal view using a Depth sensor. The speech modality comes from the videos of How2 dataset.

Gloss annotations were collected by ASL linguists using ELAN [101] software. The dataset also categorizes the videos into 10 categories. These categories are Cars and Other Vehicles, Games, Arts and Entertainment, Personal Care and Style, Food and Drinks, Education and Communication, Home and Garden, Pets and Animals, Hobbies and Crafts and Sports and Fitness. Total 2,456 videos from the How2 [97] were used to create How2Sign dataset. 21 videos from the training set, 17 videos from the validation set and 35 videos from the test set were recorded multiple times by a different signer, leading to a total of 2529 videos in the How2Sign dataset. All the recorded videos were split into videos corresponding to

single sentences, these sentence-level videos are approximately 5 seconds long and contain on average 162 frames and 17 words. How2Sign dataset in all cover above 35,000 sentences with around 16,000 different words. A small percentage of this vocabulary i.e. 20% includes fingerspelling.

The test set contains 26 videos that have been recorded by a signer not present in training set. This allows researchers to measure for generalization and signer dependency while evaluating their model. How2Sign was recorded with signers having different body proportions, thus providing variation in body sizes of the signers. The signers included both males (6) and females (5). This provides gender diversity to the dataset. The data collection was done during a period of 6 months, enabling changes in clothes and accessories of the signer. The dataset though does not contain large diversity in race, skin tone, background, illumination and camera quality.

3.3 Evaluation Metrics

In order to evaluate and compare the performance of machine learning models, evaluation metrics are used. In this section we provide an overview of the evaluation metrics that were utilized for measuring the performance of the various models implemented in this thesis work. To evaluate the performance of the ASL alphabet recognition techniques, we use accuracy and precision as an evaluation criterion. For the case of continuous sign language recognition of ASL on How2Sign dataset, we use Bilingual Evaluation Understudy Metric (BLEU) [102], Metric for Evaluation for Translation with Explicit Ordering (METEOR) [103] and Recall Oriented Understudy for Gisting Evaluation (ROUGE) [104].

Accuracy is the fraction of predictions that were correct in comparison to the total number of predictions. Accuracy can be between 0 and 1, with 0 being the worst value and 1 being the best value.

Figure 3.7: Extracted 2D keypoints from a sample frame of the front view video recorded in green screen studio [1]. This estimation of 2D keypoints is done using OpenPose [100].

Precision is the number of true positives (tp) divided by the number of positive predictions. True positive refer to the test cases that belong to a particular class and were classified as belonging to that class. Positive predictions refer to the test cases that were classified as belonging to the class in consideration. Qualitatively, precision tells about the quality of positive predictions. Precision can also be between 0 (worst value) and 1 (best value).

For the case of continuous sign language recognition, metrics to evaluate language translation or summarizing techniques are generally used. Sign language recognition can be evaluated using the textual subtitles as the ground truth sentence. The recognition method has to predict a sentence close or similar in meaning to this ground truth subtitle sentence, in order for the recognition to be good. So, in order to evaluate the quality of recognition, we need to compare these ground truth subtitled text in sign language videos with the predicted sentence from the recognition model. In the following paragraphs we discuss three methods that can be used for this comparison. During this discussion, the ground truth sentence (textual subtitle from sign language videos) is referred to as the "reference sentence". The sentence predicted by the recognition model is termed as the "candidate sentence".

3.3.1 Bilingual Evaluation Understudy Metric (BLEU)

BLEU [102] is a method of evaluating automatic machine translation. The method is quick, inexpensive and language-independent. It is one of the first metrics that relates closely with human evaluation. The prime idea behind BLEU is to compare n-grams of the predicted sentence with the n-grams of the reference sentence/s (ground truth) and count the number of matches. These matches are independent of the position and more is the number of matches, closer is the predicted sentence to ground truth. n-gram is a sequence of n words from the sentence in consideration. For example, for the sentence "The cat is outside the door", the 1,2,3 and 4-grams will be as follows:

- 1-gram: the, cat, is, outside, the, door
- 2-gram: the cat, cat is, is outside, outside the, the door
- 3-gram: the cat is, cat is outside, is outside the,...
- 4-grams: the cat is outside, cat is outside the,...

Rather than using simple precision (defined as in equation 3.1) as a metric for matches, BLEU uses modified-precision.

$$\text{precision} = \frac{\text{number of words/n-grams matches in predicted and reference sentence}}{\text{total number of words/n-grams in predicted sentence}} \tag{3.1}$$

For the case of modified precision for unigram or words, the count of a word in predicted sentence is clipped by the maximum count of the word in reference (ground-truth) sentence as shown in equation 3.2.

$$Count_{clip} = min(Count, Max\_Ref\_Count) \tag{3.2}$$

The modified precision (p) is then calculated by summing up the clipped counts for all the words and dividing the sum by the total (unclipped) number of predicted words, as given in equation 3.3.

$$p = \frac{Count_{clip}(word)}{Count(word)} \tag{3.3}$$

Modified precision for n-gram is similarly calculated for the case of n-grams where the clipped and unclipped count of words is simply replaced by clipped and unclipped counts of n-gram. Equation 3.4 shows the formula for modified precision calculation for n-grams, where PS refers to predicted sentence.

$$p_n = \frac{\sum_{\text{n-gram} \in PS} Count_{clip}(\text{n-gram})}{\sum_{\text{n-gram} \in PS} Count(\text{n-gram})} \tag{3.4}$$

The modified n-gram precision scoring accounts for two features of translation, namely adequacy and fluency. A predicted sentence (translation from sign to text) that uses the same words (1-grams) as the words in ground truth sentence satisfies adequacy. Fluency is captured by matching of longer n-grams. To account for the sentence length, that is, penalize sentences that are "too short", a Brevity penalty (BP) is calculated as follows:

$BP = \begin{cases} \exp^{(1-\frac{rl}{cl})}, & \text{if } cl \leq rl \\ 1, & \text{if } cl > rl \end{cases}$ In the above equation, rl refers to the length of the reference (ground truth) sentence and cl refers to the length of candidate sentence (predicted sentence in our case). Finally, weights are calculated as $w_n = \frac{1}{n}$, where the number n of n-grams, and a geometric averaging of modified precision scores using these weights is utilized to calculate the BLEU score as given below in equation 3.5.

$$BLUE = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{3.5}$$

The BLEU metric ranges from 0 to 1. The BLEU score is 1 if the predicted sentence is identical to the ground truth sentence. In this thesis work, we propose BLEU score to be calculated for n-grams with n=1,2,3 and 4, called as BLEU-1, BLEU-2, BLEU-3 and BLEU-4. The BLEU scores for lower n-grams, namely BLEU-1 and BLEU-2, help to recognize similarities between predicted and ground truth sentences, specifically sharing of same words and word-pairs. The BLEU scores for lower n-grams (n=3,4), are more indicative of whether the recognition is good as they look at higher level sentence structure.

3.3.2 Metric for Evaluation for Translation with Explicit Ordering (METEOR)

BLEU score, described above, has a lot of limitations. First, BLEU metric computations do not take recall into account. Recall is defined as the proportion of matched n-grams out of the total number of n-grams in the reference (ground-truth) sentence. Recall plays an important role in obtaining higher correlation to human judgement or ground truth data, as it reflects the degree to which the predicted sentence covers the entire content of the actual

referenced sentence. Second, BLEU does not have an explicit way of measuring the level of grammaticality (word order) but relies on higher-order n-gram precision for this. In order to account for these shortcomings, another metric for automated translation evaluation was evaluated, termed as Metric for Evaluation for Translation with Explicit Ordering (METEOR).

METEOR [103] is a metric for evaluating machine translations based on the generalized concept of unigram matching. Each possible matching between the reference sentence and the candidate sentence is scored using a combination of factors. These factors include unigram-precision, unigram-recall and a measure of fragmentation. The fragmentation measure captures how well-ordered the matched words are in the translated candidate sentence with respect to the reference sentence. In order to compare two sentences (the reference sentence and the candidate sentence), METEOR method creates an *alignment* between them. An alignment is a mapping where every unigram in each sentence is mapped to zero or one unigram in the other sentence. This alignment is produced using a series of stages, each involving two phases. The first phase lists all the possible mappings using some modules such as "exact" (maps unigrams that are exactly same), "porter stem" (maps unigrams that are same after Porter stemmer is applied to them), etc. In the second phase, the largest subset of these mappings that constitutes an alignment is selected. In case of conflicts, METEOR selects the alignment with least crosses. Cross between two unigram mappings (t_i, r_j) and (t_k, r_l) (where t_i and t_k are unigrams in translated/predicted candidate sentence and are mapped to unigrams r_j and r_l in the reference sentence respectively) is said to happen if the following formula evaluates to a negative value-

$$(pos(t_i) - pos(t_k)) \times (pos(r_j) - pos(r_l)) \tag{3.6}$$

where $pos(t_y)$ refers to numerical position of unigram t_y in the sentence in consideration. One can change the number of stages, the external mapping modules used in phase 2 of each stage and also the order of the stages for METEOR calculation. By default, METEOR uses three stages, which in turn use "exact", "porter stem" and "WN synonymy" as external modules respectively.

Once the system has a final alignment, METEOR score is computed by using combination of precision, recall and a penalty to account for longer matches. The formula for calculating unigram precision (P) and unigram recall(R) are given below.

$$\text{Precision (P)} = \frac{\text{number of mapped unigrams in candidate sentence}}{\text{total number of unigrams in candidate sentence}} \tag{3.7}$$

$$\text{Recall (R)} = \frac{\text{number of mapped unigrams in candidate sentence}}{\text{total number of unigrams in reference sentence}} \tag{3.8}$$

Then, a harmonic mean of P and 9R is calculated, termed as F_{mean} [105], as below:

$$F_{mean} = \frac{10PR}{R + 9P} \tag{3.9}$$

For penalty calculation, the mapped unigrams are grouped into *chunks* such that the unigrams in a chunk are in adjacent position in the candidate sentence and are also mapped to adjacently placed unigrams in reference sentence. The penalty is calculated using the formula:

$$Penalty = 0.5 \times \left(\frac{\text{number of chunks}}{\text{number of matched unigrams}} \right)^3 \tag{3.10}$$

The parameters of the above formula were determined by authors of METEOR metrics [103] through experiments. The METEOR score is finally calculated as:

$$\text{METEOR score} = \text{F}_{\text{mean}} \times Penalty \tag{3.11}$$

3.3.3 Recall Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE is a set of evaluation techniques that measure the quality of a summary or translation by comparing it to summaries created by humans or ground truth sentences. This is done by counting the number of overlapping units such as n-gram, word sequences, word pairs [104]. While ROUGE includes a number of metrics such as ROUGE-N, ROUGE-W, ROUGE-L, etc. we only discuss ROUGE-L here. Sentence level ROUGE-L can be applied to measure quality of sign language recognition by comparing the predicted sentence with the original transcription associated with the sign.

ROUGE-L computes similarity based on Longest Common Subsequence (LCS). The metric helps to identify sentence level structure similarity. To apply ROUGE-L, the reference sentence X and the candidate sentences Y are viewed as a sequence of words. The longest common subsequence (LCS) of X and Y is a common subsequence (sequence of words in this case) with maximum length (number of words), written as LCS(X,Y). Based on this longest common subsequence, precision (P_{lcs}) and recall (R_lcs) are calculated as follows:

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{3.12}$$

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{3.13}$$

where n and m are the number of words (length) in the candidate sentence Y and reference sentence X, respectively. ROUGE-L between sentences X and Y is then calculated as in equation 3.14

$$ROGUE - L = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{3.14}$$

where $\beta = \frac{P_{lcs}}{R_{lcs}}$. From equation 3.14, one can see that ROUGE-L is 1 when the candidate sentence and the reference sentence are exactly same, that is, X=Y. The measure is zero when there is nothing common between the two sentences X and Y being compared, i.e. LCS(X,Y)=0.

3.4 Methodology

In this section, we discuss the methods used to achieve sign language recognition in this thesis work. While doing so, we highlight the main steps and describe the reasons why a certain approach was preferred. We mainly talk about the image processing techniques utilized to understand the dataset in section 3.4.1. In section 3.4.2, we explain how pose estimation was carried out for the purpose of this work. Section 3.4.3 and 3.4.4 provide details about gesture detection and gesture to sentence translation, respectively.

3.4.1 Image processing

We begin by working with the image dataset described in section 3.2.1 to explore the recognition of ASL using human keypoint locations. To better understand the dataset, we perform Principal Component Analysis (PCA) of the training, validation and test split of the data.

Principal Component Analysis (PCA) is one of the most fundamental techniques for data representation. PCA refers to the process of computing principal components and then using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest. PCA is generally used to visualize higher dimensional data by reducing the data to lower dimensions. The dataset discussed in section 3.2.1 contains RGB images that are of variable sizes. To perform PCA, all the images were first converted to gray scale and then resized to a size of 256×256 using OpenCV methods. The conversion to grayscale helped to reduce the number of channels from three to one. The resulting data was flattened and reduced to two dimensions using pca method of *sklearn*. The result showing the distribution of images across the first two principal components is shown in figure 3.8. The colours in the figure depict the various alphabets of the ASL.

As one can see in figure 3.8, there are no major outliers in the dataset. The data is rather uniformly distributed across the two principal components for the different alphabet images. Some of the data points are overlapping, showing the similarity among the signs of some alphabets. We also performed PCA of the validation and test split of the dataset. The results are shown in figure 3.9 and 3.10 for validation and test split respectively. All these images show that the two dimensions fall short to capture the distribution of the data into various classes (26 alphabets). Thus, we require more than two dimensions to clearly distinguish the various images into the target 26 alphabet classes.

3.4.2 Pose estimation

This section focuses on the methods and frameworks used for pose estimation from the individual images and videos in this work. We also provide a brief overview of the working of the frameworks used for pose estimation.

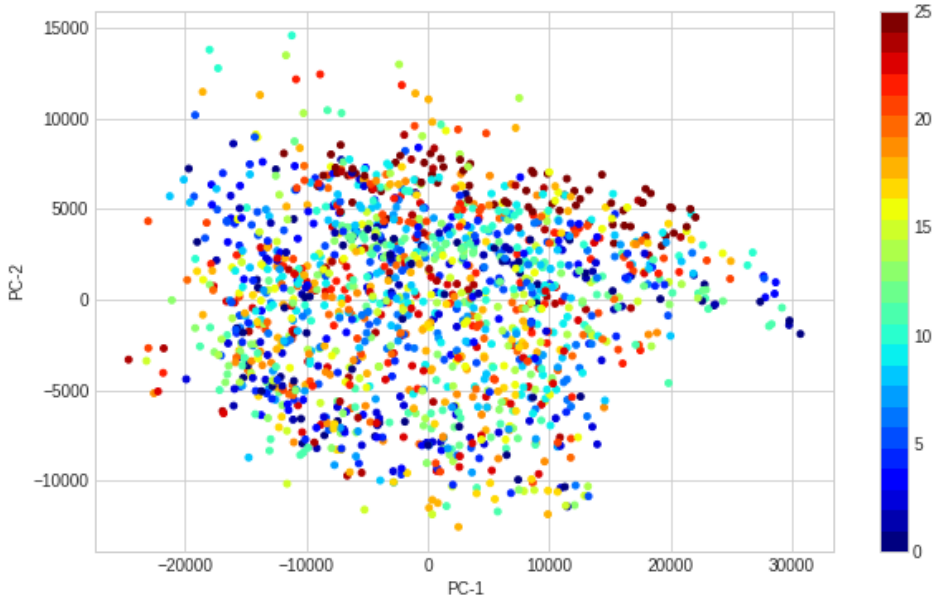


Figure 3.8: Figure shows the two dimensional visualization of training split of the image dataset discussed in section 3.2.1. The dataset has been projected using only the first two principal components. It can be seen from the image that it is hard to distinguish between the 26 hand poses using only the two dimensions. Additional dimensions are thus required to classify images into the 26 alphabet classes.

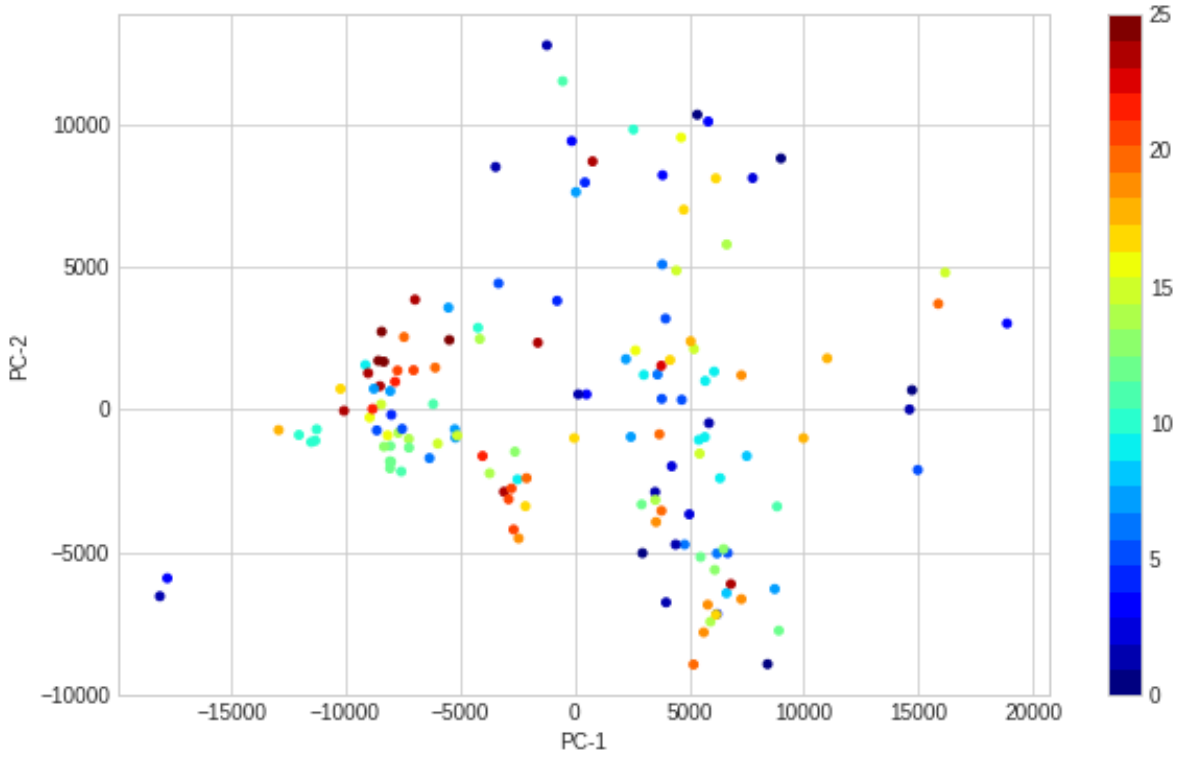


Figure 3.9: Figure shows the two dimensional visualization of validation split of the image dataset discussed in section 3.2.1. The dataset has been projected using only the first two principal components. Similar to training data, the projected 2 dimensions fall short for successful distinction of images in 26 alphabet classes.

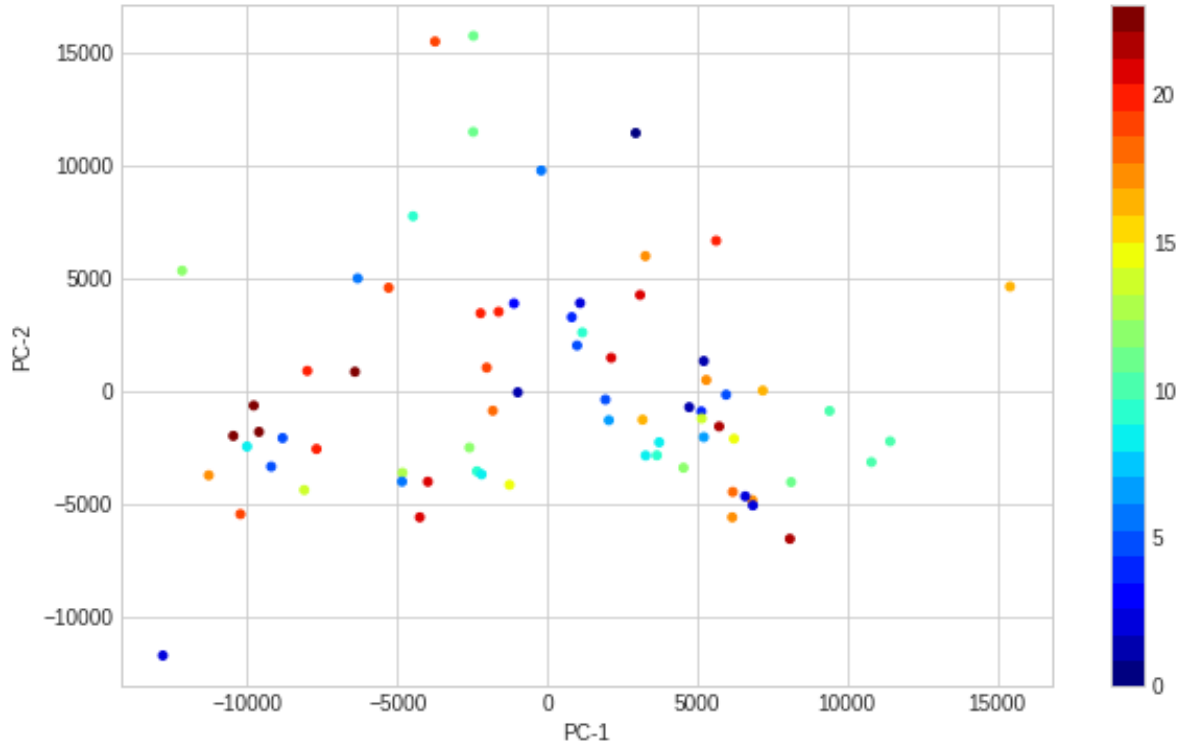


Figure 3.10: Figure shows the two dimensional visualization of test split of the image dataset discussed in section 3.2.1. The dataset has been projected using only the first two principal components. No clear segregation can be seen between the 26 classes, by using just two dimensions.

ASL image dataset

This thesis work focuses on the usability of skeletal data for sign language recognition tasks. We, first apply our approach to static image dataset and study recognition of ASL alphabets given these joint coordinates instead of the individual images. In this section, we discuss how pose estimation is carried out on the ASL image dataset 3.2.1 and how these evaluated joint coordinates are pre-processed to be used further for classification.

To collect the various joint coordinates from the images, MediaPipe[2] was used. MediaPipe is an open-source framework for building perception pipelines of machine learning software [106]. MediaPipe provides the functionalities to handle data processing pipeline, synchronize multiple information sources for the applications working on constant stream of input information such as, audio, video, etc. MediaPipe offers customizable Machine Learning solutions for a variety of applications that include:

- face detection
- object detection and tracking
- hair and selfie segmentation
- hand detection and tracking
- pose detection and tracking
- holistic tracking (combination of hand, face and pose tracking)

These solutions are available for a variety of platforms and are accessible via Python, C++ and JavaScript APIs, and also on Android and iOS. The above solutions utilize trained TensorFlow inference models.

Using MediaPipe, the perception pipelines are built as graphs with nodes corresponding to modular components such as algorithms to pre-process media, transform data, etc. These nodes are called calculators. Input data streams in the form of packets enter these nodes, are processed and are passed on to the next node in the graph. This graph based architecture makes MediaPipe easily customizable, user-friendly and flexible. Figure 3.11 provides an example of a simple graph with one calculator node. The input block refers to the input stream such as audio and video stream and the output block represents the output stream such as object-localization and face-landmark streams.

As the signs for the alphabets of American Sign Language involves hand gestures, we use MediaPipe Hands[3] solution [12] to infer hand poses. MediaPipe Hands uses a machine learning pipeline comprising of a palm detection model and hand landmark model. The palm detection model takes as input the complete image and outputs an oriented hand

[2]https://mediapipe.dev/
[3]https://google.github.io/mediapipe/solutions/hands.html

Figure 3.11: Basic MediaPipe graph with a placeholder for one calculator node.

bounding box. The hand landmark model operates on this cropped image output by the palm detection model and detects the 3D hand keypoints. The whole pipeline is implemented as a MediaPipe graph. As hands often occlude themselves and lack high contrast patterns, detection of hands is a challenging computer vision task. To overcome these challenges, MediaPipe Hands solution employs a palm detector instead of a hand detector, encode-decoder feature extractor and minimize focal loss during training. The hand landmark model uses regression to infer precise keypoint localization of 21 3D hand-knuckle coordinates inside the detected hand regions. Figure 3.12 shows these 21 landmarks, depicting their relative position and names. Figure 3.13 shows the detected 21 3D keypoints for the sample images

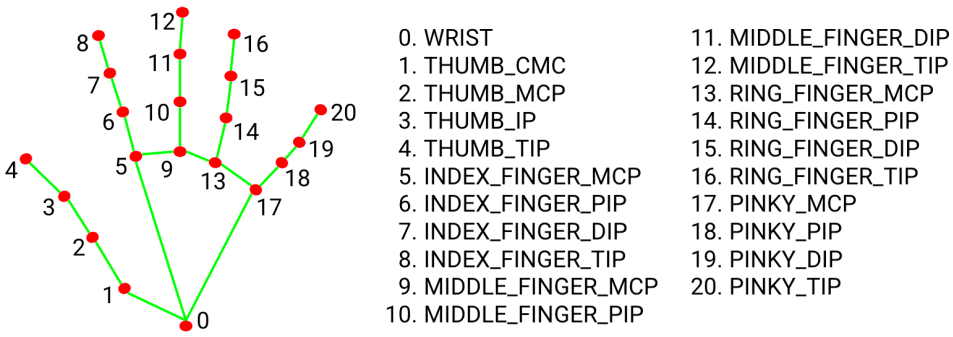


Figure 3.12: 21 hand landmarks and their relative position as detected by MediaPipe Hands [12] solution.

corresponding to different alphabets of ASL from this dataset shown in figure 3.4. These images are flipped horizontally by MediaPipe before keypoint detection is done. In addition to the 21 3D landmark positions, MediaPipe Hands solution also outputs the handedness, specifically, whether the detected hand in the image is left or right. To have a view of one to one mapping between the detected keypoints and hand poses and the original hand image containing the sign for the alphabet one can look at figure 3.14.

Figure 3.13: 21 3D landmarks detected for the sample 26 alphabets from the image dataset for American Sign Language shown in figure 3.4. The images read as A to Z, starting from top left image and reading from left to right.

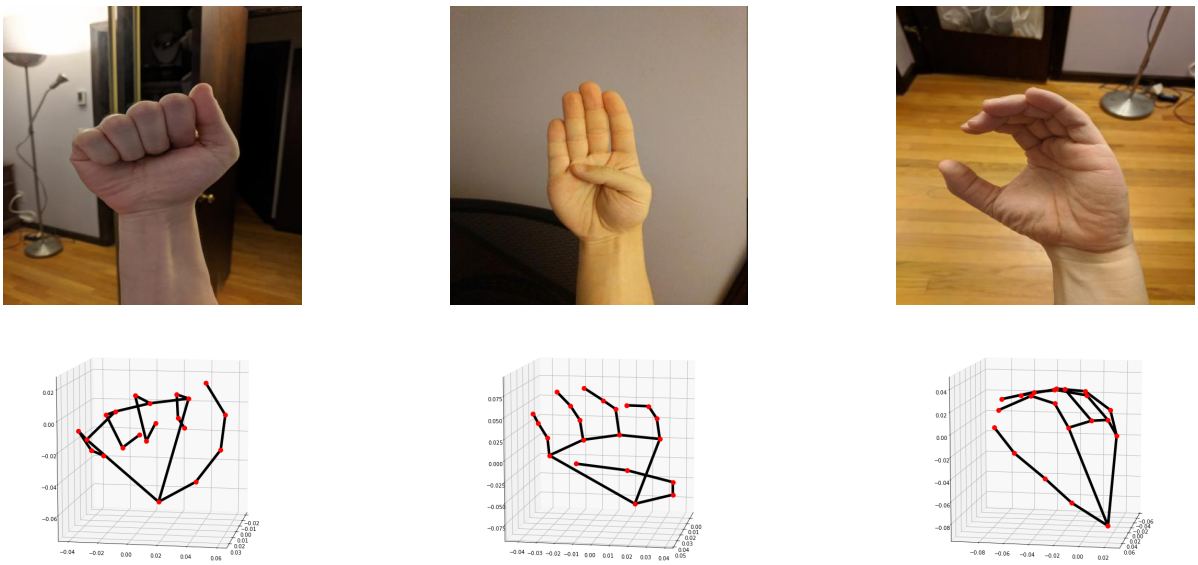


Figure 3.14: The sample images for the first three alphabets from the image dataset for American Sign Language (top) and the corresponding detected keypoints and hand poses using MediaPipe (bottom). One can see that the skeletal hand poses differ between the three alphabets and can be used for detection of alphabet instead of the original image.

3.4.3 Gesture detection

In this subsection, we describe how gestures are inferred from the estimated human body pose. We explain the methods employed for recognition of individual alphabets from the images and also the pre-processing steps necessary for ASL sentence recognition from the sign language videos.

ASL alphabet recognition

We use the MediaPipe Hands solution to collect the 21 3D landmarks (x and y coordinates and depth information) for all the images of the dataset for ASL alphabet recognition. We consider these collected landmark coordinates as $21 \times 3 = 63$ features and think of the recognition problem as a classification task. The landmark coordinates values are pre-processed by shifting each coordinate value to a new coordinate system where the wrist coordinates are chosen to be the origin (0,0,0). The input to this classification problem are these processed 63 features and the output is a label out of the 26 alphabets of the ASL language. It is worth noting that handedness feature is not utilized in classification because the signs of the alphabets can be performed by either hands and are invariant of the change of hands. There are certain images for which no hand is detected by MediaPipe. These images are discarded from the training, validation and test set.

We employ the classification methods discussed in tables 3.1 and 3.2 for detection of ASL alphabets as described above. The tables also highlights the *scikit-learn* [107] library functions used to implement these classification methods. For the case of nearest neighbors, the value of k was chosen to be 10. For decision tree, we chose a maximum depth equal to 20. For random forest classifier, maximum depth of 30, 10 estimators and 5 maximum features were chosen. We use fully connected neural networks. Additionally, grid search was performed to choose the hyperparameters with best accuracy over the validation test.

ASL sentence recognition

As discussed earlier, we work on How2Sign dataset for sentence level recognition of American Sign Language. This thesis work explores the suitability of using the human body keypoints for the detection, rather than working on spatio-temporal features of the videos in the dataset. The very first step of this approach is as gesture detection, that is to collect the keypoint coordinates of the signer from each frame of the videos in the dataset. How2Sign dataset already has this pose information modality and includes the detected 137 3D coordinates which include 25 body keypoint locations, 70 facial keypoints and 21 keypoints detected from each hand. This keypoint extraction has been done using OpenPose [100].

OpenPose is a real-time multi person system to jointly detect human body, hand, facial and foot keypoints [100]. OpenPose detects the keypoints based on a two-branch, multi-stage convolutional neural network (CNN) [100]. OpenPose works on a bottom-up method of human keypoint extraction and uses a non parametric representation, termed as Part Affinity Fields (PAFs), to learn the association of body parts of individuals in the images or videos. OpenPose allows a variety of inputs such as image, video and webcam and a choice of the output format (display, JSON file with the keypoint information, image+keypoints). OpenPose can be used on different platforms, such as Ubuntu, Windows, Mac OSX, and embedded systems. It also supports various hardware including CPUs and, CUDA and OpenCL GPUs. Initially OpenPose was released to detect 2D coordinates but has been later extended to detect 3D coordinates as well. OpenPose is very popular among the research community because of its availability and real-time performance.

The detected keypoints for the How2Sign dataset are included in the form of JSON file, where each file includes the 3D coordinates of 25 body keypoints, 70 facial keypoints and 21 keypoints of each hand of the signer detected in a particular frame of the video. Thus, the keypoints information corresponding to a single sentence (video with sign language being performed by a signer) is presented as multiple JSON files, where the number of JSON file is equal to the number of frames in the video under consideration. In addition to the 3D coordinates of the detected keypoints, the JSON files also include a signer ID.

Datasets used for machine learning tasks need some pre-processing to convert them into suitable format to be able to input them to a machine learning model. The detected keypoints acquired from the How2Sign dataset were also processed to be used for sentence recognition.

It is observed from the dataset that a lot of keypoint coordinates for the body of the signer are zero. These are majorly the coordinates corresponding to the lower body skeletal keypoints. This happens because the signers were seated at the time of recording of the dataset and the videos capture the signers body from above the thigh portion. The values for these keypoints have been set to 0 by OpenPose. We propose to discard these 10 keypoints corresponding to the lower body. This is done to remove features (here keypoints) that do not contribute any knowledge to the dataset. Also, 0 is an acceptable keypoint value and might lead to conveying some wrong pose information.

Another issue in the dataset is the case of missing keypoint coordinates, that can happen because of occlusion or bad lighting conditions. There are several ways to handle this issue. One possibility is to discard the frame that has these missing keypoints. While this solves the issue, it could also lead to deleting potentially valuable information of the other keypoints in the frame. Also, by skipping frames, one can lose the continuity between different gestures. Another possibility is to set this value to 0 or some other number. This solution is not plausible as any float value is an acceptable value for a keypoint and can lead to wrong inference of gestures. We propose to use mean value of this keypoint in the preceding and next frame. If the missing value is in the first or last frame, it is be filled using the value from the following and preceding frame, respectively. Though, this solves the issue, it still has some shortcomings. This method fail to handle the case of fast and sudden movements and can lead to inferring incorrect movement of body parts. We suggest that handling of missing keypoints should be further studied.

The dataset includes sentences of different lengths. This implies that the duration for performing the signs for these sentences is also variable. This leads to a difference in number of frames for each video in consideration. The difference in number of frames for different sentences is reflected as different number of JSON files (containing pose information). In order to standardize the input for sentence recognition, we padded the shorter (lesser frames) sentences with JSON files containing 0 value for the coordinates of the different keypoints. This leads to the number of frames (associated JSON files) per sentence being equal to the number of frames in the longest collected sample of sign language.

3.4.4 Gesture to sentence translation

In this section, we discuss the implemented model to perform sentence recognition given the series of keypoint coordinates as input. The problem of predicting sentences from these keypoint features for a series of frames can be viewed as similar to caption generating problem. Although, in case of caption generation, we go from images or videos to textual description. Here, we are targeting prediction of sentences (text or series of words) from the skeletal keypoint information collected from the various frames of the video. Transformers are the state-of-the-art for the problem of caption generation. In this section, we provide detailed description of the architecture of the model transformer with employed for sentence recognition from these gestures (series of keypoint coordinates) and also list out the various

pre-processing steps applied to the ground-truth sentences or text corresponding to each video (that is, sentence in American Sign Language).

For continuous sign language recognition tasks, ground truth labels to a sample data point is textual description of the sign. This textual description could either be subtitles, translations or gloss annotations. How2Sign dataset used for continuous sign language recognition in this thesis work has English subtitles in the form of sentences for every sentence-level video in the dataset. Table 3.3 provides few example sentences from the How2Sign dataset. Models designed for recognition of continuous sign languages predict a combination of individual words that are present in the vocabulary of the dataset. The creation of vocabulary for How2Sign dataset is done as follows:

- All words are converted to lower case. This is done in order to not include "This" and "this" as two different words in the dataset.
- Remove short form and replace them with longer form of the word. For instance, "don't" is replaced with do not, "he'll" with "he will", etc.
- The unique words after performing the above two steps are added to the vocabulary.

Figure 3.15 provides the overall architecture of the model that translates an ASL video into American English sentence. The model inspired from [88] uses sequence to sequence model based on transformer cells. The model uses SLR transformer architecture from Camgöoz et al. [14]. The model takes a series of feature vectors as input. In our case, these feature vectors correspond to the upper human body keypoints extracted from the various frames of the video using OpenPose. These feature vectors are normalized and input to the encoder unit. This is followed by multi head self attention layer. Self attention is a sequence to sequence operation that helps to identify the parts of the input sequence to be paid attention to, in order to obtain the desired outputs [83]. The other layers in the encoder include fully connected Feed Forward (FF) network and a layer for normalization.

Word embedding is done to allow words of similar meaning to be represented similar. Special tokens, i.e., <sos> (start-of-sentence) and <eos> (End-of-sentence) are added to each sentence. This is followed by positional encoding (PE), that helps to add positional information to the input sequence. In this model architecture sinusoidal positional encoding is used. The positionally encoded embeddings are passed to a masked self attention layer. Masking helps to ensure that each token only uses the tokens before it while extracting information on context. The output of masked self attention layer along with the output from the encoder side is passed to encoder-decoder attention module. This module learns the mapping between input and target sequences. The output of this model is passed to non-linear feed forward layer and softmax loss is applied. The network predicts one word at a time, until the <eos> token is reached.

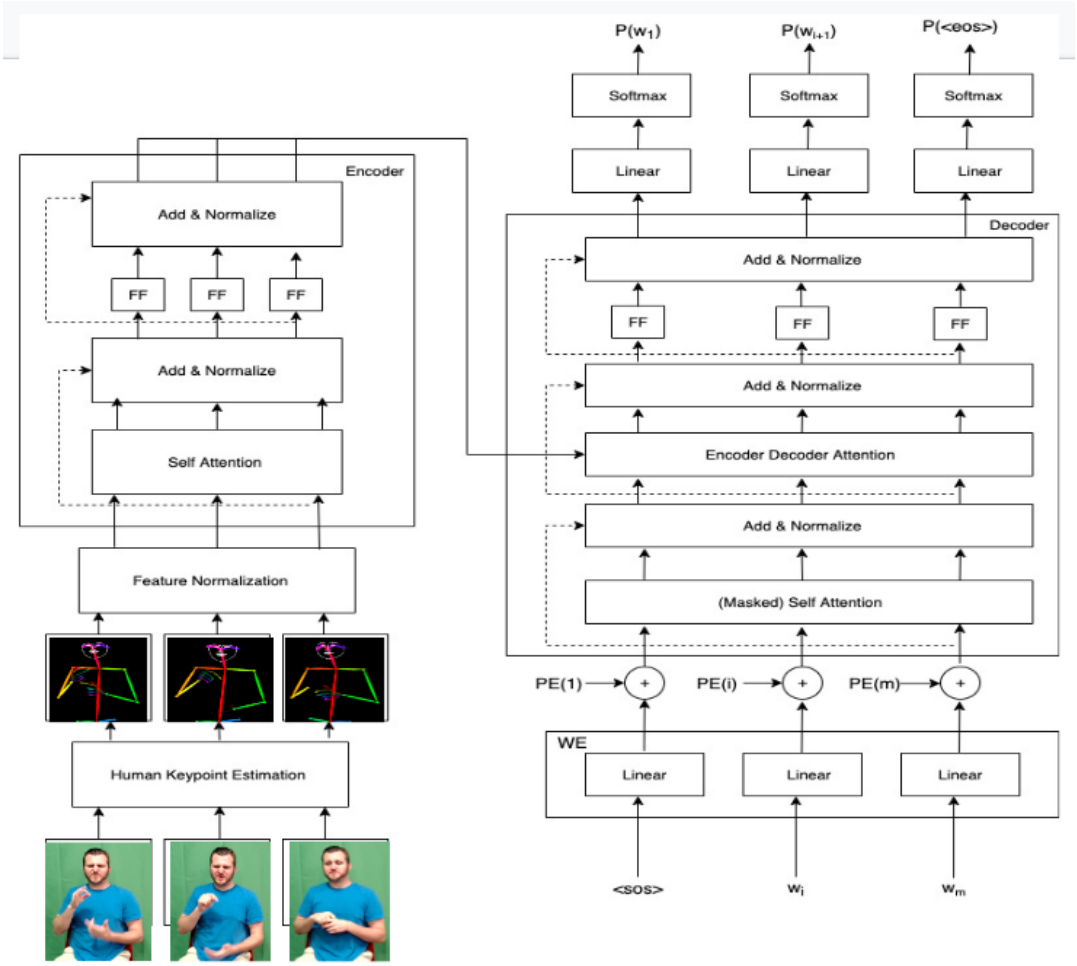


Figure 3.15: Detailed overview of the model architecture for continuous sign language recognition task.(FF: Feed Forward, PE: Positional Encoding, WE: Word Embeddings)

3.5 Results

In this section, we present the results obtained by employing the various classifiers discussed in section 3.4.3 for ASL language alphabet recognition, and continuous sign language recognition results obtained by using the gesture to sentence translation techniques discussed in section 3.4.4.

Table 3.4 summarizes the accuracy and precision (as defined in section 3.3) of the various classification methods for both the validation and test sets of the used ASL alphabet dataset. Neural Network classifier performs best in terms of accuracy and precision for both the validation and the test set. The quoted accuracy and precision is achieved using a fully-connected neural network with ReLU activation function and Adam solver. To obtain the best performing combination of hyperparameters, grid-search is utilized. The trained neural network classifier was additionally used to recognize the various alphabets signed in a webcam video stream.

Due to time constraints, the model architecture for continuous sign language recognition could not be successfully trained and hence, no results are provided here.

Table 3.1: The various classification methods used for ASL alphabet recognition, along with the *scikit-learn* library functions used (part I).

Method	**Description**	***scikit-learn* function**
Nearest Neighbors (KNN)	unknown data point is labelled as belonging to the data class which has the majority vote among the nearest neighbors of the unknown datapoint.	*neighbors.KNeighborsClassifier()*
Linear SVM	learns hyperplanes which separate the given data into target number of classes.	*svm.SVC(kernel='linear')*
RBF SVM	learns curve shaped boundaries (based on RBF kernel) which separate the given data into target number of classes	*svm.SVC(kernel='rbf')*
Decision Tree	predicts the target class of a test data point by learning simple decision rules inferred from data features	*tree.DecisionTreeClassifier()*
Random Forest	learns various decision tree classifiers on different subsamples of the data.	*ensemble.RandomForestClassifier()*
Neural Network	learns a non-linear function to convert input of certain dimensionality to required output dimensions.	*neural_network.MLPClassifier()*
AdaBoost	ensemble estimator that first fits a classifier on the original dataset and later fits additional classifiers on the same dataset focusing more on difficult cases.	*ensemble.AdaBoostClassifier()*

Table 3.2: The various classification methods used for ASL alphabet recognition, along with the *scikit-learn* library functions used (part II).

Method	**Description**	***scikit-learn* function**
Naive Bayes	supervised classification method based on Bayes' theorem and assumption of conditional independence between features (pairwise)	*naive_bayes.GaussianNB*
QDA	classifier that learns quadratic decision surface to classify data into classes.	*discriminant_analysis.QuadraticDiscriminantAnalysis*

Table 3.3: Some examples of the English sentences from the How2Sign [1] dataset.

Sentence Name	**Sentence**
–dANj_01AU_0-5-rgb_front	In the side position
–dANj_01AU_13-5-rgb_front	So we're going to start again on this one.
-06_nJnhORg_18-5-rgb_front	I have none at home
-0N0jbyBW6g_10-5-rgb_front	My name is Daniel King
5CGdJ5Cuv5M_11-8-rgb_front	You get two more tries
5CGdJ5Cuv5M_8-8-rgb_front	Well you don't want to be in jail while other people are buying up property and making all the money.

Table 3.4: Obtained accuracy and precision for various classification methods used for ASL alphabet recognition using the 21 hand landmarks coordinates as features.

	Validation set		**Test set**	
Classification Methods	**Accuracy**	**Precision**	**Accuracy**	**Precision**
Nearest Neighbors	0.52	0.62	0.59	0.75
Linear SVM	0.57	0.68	0.56	0.69
RBF SVM	0.45	0.44	0.45	0.68
Decision Tree	0.51	0.60	0.56	0.78
Random Forest	0.61	0.72	0.59	0.72
Neural Network	0.78	0.82	**0.83**	**0.90**
AdaBoost	0.08	0.84	0.11	0.82
Naive Bayes	0.42	0.52	0.38	0.48
QDA	0.125	0.87	0.11	0.85

4 Conclusions

As highlighted in this thesis work recognition of sign languages is a complex Computer Vision problem. It is a very promising field with numerous possibilities for exploration. This thesis work explored the recognition of signs by using human body key point information. In this section, we summarize the basic findings of this thesis work, provide a detail discussion of the results obtained and suggest possible alternatives of improvement and directions for future work.

4.1 Summary

Like any other problem involving natural languages, Sign Language Recognition is a challenging problem not only because of the complexity involved in sentence formation and detection, but also due to the complex nature of the signs that are used in the language. Signs for any sign language, in general, involve communicating information using three channels. These channels include the hands, upper body and the face. The information from these three different channels is communicated in a parallel fashion. While the hands are majorly used for making the signs for words, the posture of the upper body and the facial expressions convey the grammatical information. This thesis work looked closely at how signs in American Sign Language (ASL) are signed and detailed out the complexities involved.

The development and performance of any Machine Learning model is largely dependent on the quality and size of the dataset used for training, validation and testing of the model. This thesis provides a list of the popular datasets used for sign language recognition tasks and provides information about the target language, vocabulary size, various data modalities available, number of signers used for collection of data and also the suitability of the data for continuous or isolated sign language recognition tasks. This thesis provides a detailed overview of How2Sign dataset, which is a recent dataset for ASL recognition task. The dataset covers a large vocabulary of around 16,000 words. This dataset contains multi-view (front and lateral) videos and also pose and depth information. This dataset appears to be very promising for future sign language recognition tasks because of the availability of different modalities, larger vocabulary size and generalization capability of the dataset.

This work explores the problem of Sign Language Recognition through a different lens. We looked at the problem of recognizing signs as a two step problem. The first step is detecting the gesture that is being performed. The second step is translating this gesture to a sentence or word. The step of gesture recognition is done by collecting the coordinates of the

key points of the human body. The second step takes these detected keypoints and learns to infer sentences or words from it.

We explored the feasibility of working with these key points for recognition of sign language by first trying to detect single alphabets of American Sign Language (ASL) using a dataset consisting of images, and then working on continuous sign language detection on How2Sign dataset. For single alphabet detection, we used MediaPipe Hands solution to collect the 3D coordinates of the 21 landmarks of the hand involved in making the sign and use these $21 \times 3 = 63$ positional values as features and the alphabet for the sign, as label. Using these features, a number of classification techniques were experimented with and trained, to find the best classification method to classify this keypoint information to individual alphabets. The trained models were also tested for real time input signs captured from the webcam video stream.

Continuous sign language recognition (CSLR) is a more complex problem compared to single letter or word recognition (Isolated Sign Language Recognition). The thesis work highlighted the requirements for efficient CSLR and the challenges involved in achieving the same. We presented the pre-processing steps required to use How2Sign dataset for sign language recognition task. Additionally, we presented a model architecture that could be potentially used for detecting sentences given the keypoint information for the various frames. The thesis also provided a detailed understanding of the various metrics that can be used for evaluating the performance of sign language recognition models, both for single letter recognition and continuous sign language recognition (recognition of complete sentences).

4.2 Discussion

In this section, we discuss the results obtained by the various experiments that were carried out during the course of this thesis work. We analyze these results and propose the best course of performing the task using the results obtained from the various experiments.

Table 3.4 provides the precision and accuracy achieved on American Sign Language (ASL) alphabet recognition for the various classification techniques employed. As evident from the table, the best accuracy and precision is obtained for the case of neural networks. This is possible because of the capability of neural networks to learn and recognize patterns and learn correlated features. ReLU activation function showed better results compared to tanh function. Adam was used as the solver and was chosen over SGD and limited memory BFGS due to better performance. AdaBoost, Quadratic Discriminant Analysis (QDA) and Naive Bayes gave the worst recognition accuracy. QDA assumes that the features are drawn from a Gaussian distribution, which is not the case here. Naive Bayes classifier is based on the assumption of conditional independence between features. For the case of hand keypoints, the features are not independent but are rather correlated. The position of finger joints is correlated to position of the wrist. Also, not all combination of values of the 21 keypoints are

valid because of the underlying human skeleton structure and limited motion of the various joints. Because of this correlation, some of the features are voted twice while using Naive Bayes model. Their importance is over inflated and we achieve poor accuracy. Second best accuracy is achieved for Random Forests. This could be attributed to increased maximum depth of the individual decision trees and the ensemble based approach of random forests.

The achieved best accuracy is not very high. This could be seen as a limitation of the dataset as it only contains 1728 images, which are not sufficient to learn the various possible orientations of the signs being performed. The class imbalance in the dataset is another limitation to the achieved accuracy levels. MediaPipe framework fails to detect hand and the corresponding keypoints for certain images within the dataset. For this thesis work, those images were simply discarded making the dataset even smaller. Nevertheless, we can conclude that neural networks perform best among the tested classifiers for detection of sign language alphabets of American Sign Language (ASL).

For the case of real time recognition of alphabets using webcam, first MediaPipe Hands solution is used to get the 21 landmarks' positions, these positions are pre-processed and then the trained neural network is used for classifying this test case to one of the alphabets. The model trained on the dataset works for the real time case with sufficient accuracy. This shows that the models using keypoint based features are signer independent, as no images from the webcam were used for training purposes. It additionally highlights the performance of MediaPipe to detect hand landmarks' coordinates for real time webcam images.

For continuous sign language recognition we provided a model architecture and the various metrics that can be used to measure performance of models used for continuous sign language recognition in general. At the time of the submission of this thesis, the proposed model could not be trained for train split of the How2Sign dataset but we presented the various pre-processing steps that need to be applied to the How2Sign dataset to be used as input. Researches in the past have worked on datasets of other sign languages and have shown promising results using similar architectures. Some of these work include [88] and [14]. In addition to this, this thesis work can be used as a document of reference for understanding how sign languages are signed; past research works in the field of sign language recognition; the various methods of human pose estimation; the popular open source solutions for detection of relevant keypoints for human pose estimation, namely, OpenPose and MediaPipe and brief informative highlights of the various benchmark datasets for sign language recognition tasks.

4.3 Outlook

This thesis work explored the suitability of using human pose information for sign language recognition tasks. While probing about the feasibility of this approach, we looked at alphabet and sentence level recognition. This section is dedicated to the limitations of the applied approaches and also presents suggestions for future directions in which this work can be

extended and improved further.

While exploring the recognition of single alphabets using keypoints, detected using MediaPipe Hands solution, there were cases where no hand was detected. In this thesis work, we simply discarded these images. Future work can possibly look at the reasons why such a case arise and investigate the ways in which this could be mitigated. One possible way could be using a secondary method for detection of these keypoints, such as OpenPose, in addition to existing MediaPipe Hands solution. Another important direction of improvement could be to use the detected keypoint coordinates to calculate derived features such as size of the hand, distance between the various joints, angle between the different finger positions. These derived features can additionally be used as input to the classification methods. The dataset used only had 1278 images and was only recorded by a single person. This puts a bar on the performance of the models trained on this dataset. We suggest to either extend the dataset to include images from other signers, thus improving not only the number of training samples but also improving the diversity of these samples.

The future works in this area can also tweak around the neural network architecture used for recognition of alphabets from the keypoint information. Additionally, we suggest augmentation techniques such as zooming in/out, small rotation, illumination changes, blurring, etc., to be applied to the images used for recognition tasks. The image dataset used in this thesis work is challenging for recognition of signs, as the images are captured against a cluttered background. Further variations in background of the dataset can help to built better models for recognition tasks.

The methods discussed and implemented in this thesis work rely on the keypoint estimation from MediaPipe Hands solution and OpenPose framework. This thesis work assumes these detected keypoints to be accurate and does not consider the degree of confidence associated with these detection. Future studies can additionally incorporate this confidence score for the detected keypoints and investigate ways in which these scores can be used to built more robust and accurate models.

How2Sign dataset used for continuous sign language recognition in this work is a recent dataset and is still in development stages. This thesis work just focuses on the keypoints detected from the front view videos, the remaining modalities provided with the dataset remain unused. It would be interesting to also collect the keypoint information for the side view videos, and use these new keypoint values in combination with the keypoint coordinates provided originally with the dataset. This combination of the keypoints coordinates collected from different views can help to overcome limitations caused by occlusion, overlapping of body parts and blurry frames in a specific view.

The choice of models selected and the pre-processing steps used for the dataset have a great impact on the recognition quality achieved. Transformer models are a promising candi-

date model for sign language recognition utilizing the human keypoint information. This thesis work only looks at a simple transformer model architecture for the problem in hand. There are a lot of frameworks like huggingface, adapterhub and fairseq that allow selection of pre-trained complex and optimized transformer models. The limited scope of this thesis did not allow testing these frameworks. It would be worth exploring the effect of increasing model complexity on the performance of the recognition tasks.

This work presents some ideas on pre-processing of both keypoints and textual descriptions associated with the sign language sentence videos. There are other available methods for textual pre-processing in the literature that could be applied to the textual subtitles associated with sign language videos of How2Sign dataset. While the thesis discusses ways to mitigate missing keypoint information within the dataset, the methods are not exhaustive and the work does not looks into the impact of applying these methods over the performance of the recognition tasks. It would be constructive to look into the possible solutions for the issue of missing keypoints and compare the performance results obtained by employing the various solutions. Further research is necessary to examine how these keypoints can be utilized to extract derived features such as angle arms make with the torso, degree of head tilt, trajectory of hand motion involved, etc. These features can crucially improve the performance of the models for sign language recognition.

How2Sign is a large dataset with extensive vocabulary size. While on one hand it provides the benefit of modelling larger parts of ASL vocabulary, it, on the other hand, comes with the difficulty of training the models. Due to the large number of training samples, this work could not achieve complete training of the proposed model because of time constraints. Training the model and experimenting with the hyper-parameters of the model were not covered in this thesis work. Future works in this direction can try to train the model on better and more powerful machines, thus providing the ability of performing various experiments and deciding on better architectural parameters for achieving good metric scores.

At the time of this work, the gloss annotations for the various sentences in the How2Sign dataset were not publicly released. Once these gloss annotations are released, studies can be done to detect these gloss annotations first from the detected human poses and then later detect the complete sentences from the detected glosses. This would help improve the discussed solution further. This is majorly because the detection of a complete sentence from the keypoints is a rather long shot. The discussed approach does not look into the individual signs that were made while signing a complete sentence. The incorporation of gloss annotations can help to capture an idea of the different words used in the complete sentence and could possibly be used for sentence construction using natural language processing techniques.

A crucial prospect of sign language recognition problem is the evaluation the recognized text. While for the case of alphabets being recognized, this evaluation is rather straightforward,

the evaluation of recognized sentences is much more complicated. A recognized sentence that uses the same words as the ground truth sentence can be a bad prediction, if the word order is disturbed and the sentence does not make any grammatical and meaningful sense. The sign language recognition tasks usually use metrics used for language translation such as BLEU and ROUGE. While these metrics correlate to human judgements of translation, this correlation is poor for certain cases. A more preferred approach could be to employ a variety of these metrics and evaluate the model results depending upon the scores for these metrics and what information these metrics capture. One of the metric suggested to perform better in automatic translation evaluation is BLEURT. This metric is based on BERT, a pre-trained language representation model.

While the How2Sign dataset provides significant improvements over the existing benchmark datasets in terms of increased number of modalities, larger target vocabulary, signs associated with various topics, etc., it does have its own limitations. The dataset is recorded against a green screen and does not provide good specimens of real world use cases where the backgrounds are noisy and cluttered. Also, the videos are recorded with high quality cameras that focus on the signer. Real world scenarios involve usage of a variety of camera qualities and even possibility of multiple persons being present. It would be interesting to explore how the approach of using human body postures work for real world videos and on datasets used in works such as [16]. Another limitation of the How2Sign dataset is limited ethnic and racial diversity of the signers used for recording of the data. This is important to also evaluate the performance of human pose estimation methods for different skin colours.

The number of signers used for recording of How2Sign dataset is limited to 11. Other datasets for American Sign Language such as MS-ASL [108] include signs by almost 200 signers and should be tried for future works. This perspective of the dataset is important to make the detection model signer independent and model more variations in body sizes. Other possible improvement areas include usage of spatio-temporal features captured from the RGB videos along with the pose information for the detection process. While this would be a complex task, it would help to capture the facial expressions better, in turn, improving the recognition performance. One could also possibly look into applying a model architecture built for a particular sign language to other sign languages, as sign languages are intrinsically same in terms of the body parts and movements involved for making the sign gestures.

This thesis work studied the problem of sign language recognition in detail, and the feasibility of utilizing keypoint information for recognition of American Sign Language. We highlighted the shortcomings of the proposed methods and architectures and provided detailed analysis of the results obtained for the various experiments. The sign language recognition task still remains challenging and the scope of this work is rather limited in terms of the target problem and achieved performance, it is hoped that this work proves helpful in further development and research studies in this field.

List of Figures

List of Tables

Bibliography

[1] 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: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[2] *World Health Organization 2021 Deafness and hearing loss*. 2021. URL: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (visited on 12/15/2021).

[3] A. Graves, A.-r. Mohamed, and G. E. Hinton. "Speech Recognition with Deep Recurrent Neural Networks". In: *CoRR* abs/1303.5778 (2013). arXiv: 1303.5778. URL: http://arxiv.org/abs/1303.5778.

[4] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. "Using multiple sensors for mobile sign language recognition". In: Nov. 2005, pp. 45–52. ISBN: 0-7695-2034-0. DOI: 10.1109/ISWC.2003.1241392.

[5] D. Uebersax, J. Gall, M. van den Bergh, and L. V. Gool. "Real-time sign language letter and word recognition from depth data". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (2011), pp. 383–390.

[6] S. A. Mehdi and Y. N. Khan. "Sign language recognition using sensor gloves". In: *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.* 5 (2002), 2204–2206 vol.5.

[7] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. "American sign language recognition with the kinect". In: *Proceedings of the 13th international conference on multimodal interfaces*. 2011, pp. 279–286.

[8] R. Sutton-Spence and B. Woll. *The Linguistics of British Sign Language: An Introduction*. Cambridge University Press, 1999. ISBN: 9781107494091. URL: https://books.google.de/books?id=0betAQAAQBAJ.

[9] P. Boyes-Braem, R. Sutton-Spence, and R. te Leiden. *The Hands are the Head of the Mouth: The Mouth as Articulator in Sign Languages*. International studies on sign language and the communication of the deaf. Gallaudet University Press, 2001. ISBN: 9783927731837. URL: https://books.google.de/books?id=EyeLQgAACAAJ.

[10] O. M. Sincan, J. C. S. J. Junior, S. Escalera, and H. Y. Keles. *ChaLearn LAP Large Scale Signer Independent Isolated Sign Language Recognition Challenge: Design, Results and Future Research*. 2021. arXiv: 2105.05066 [cs.CV].

[11] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah. "Deep Learning-Based Human Pose Estimation: A Survey". In: *CoRR* abs/2012.13392 (2020). arXiv: 2012.13392. URL: https://arxiv.org/abs/2012.13392.

[12] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann. "Mediapipe hands: On-device real-time hand tracking". In: *arXiv preprint arXiv:2006.10214* (2020).

[13] K. Emmorey. *Language, Cognition, and the Brain: Insights From Sign Language Research*. Nov. 2001.

[14] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden. *Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation*. 2020. arXiv: 2003.13830 [cs.CV].

[15] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu. *Skeleton Aware Multi-modal Sign Language Recognition*. 2021. arXiv: 2103.08833 [cs.CV].

[16] M. Borg and K. P. Camilleri. "Sign Language Detection "in the Wild" with Recurrent Neural Networks". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 1637–1641.

[17] A. Moryossef, I. Tsochantaridis, R. Aharoni, S. Ebling, and S. Narayanan. *Real-Time Sign Language Detection using Human Pose Estimation*. 2020. arXiv: 2008.04637 [cs.CV].

[18] 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 (CVPR)*. June 2018.

[19] 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 (CVPR)*. July 2017.

[20] T. Hanke, M. Schulder, R. Konrad, and E. Jahn. "Extending the Public DGS Corpus in size and depth". In: *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*. 2020, pp. 75–82.

[21] B. G. Gebre, P. Wittenburg, and T. Heskes. "Automatic sign language identification". In: *2013 IEEE International Conference on Image Processing*. IEEE. 2013, pp. 2626–2630.

[22] B. G. Gebre, O. Crasborn, P. Wittenburg, S. Drude, and T. Heskes. "Unsupervised feature learning for visual sign language identification". In: (2014).

[23] C. D. Monteiro, C. M. Mathew, R. Gutierrez-Osuna, and F. Shipman. "Detecting and identifying sign languages through visual features". In: *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE. 2016, pp. 287–290.

[24] G. J. Grimes. *Digital data entry glove interface device*. US Patent 4,414,537. Nov. 1983.

[25] K. Grobel and M. Assan. "Isolated sign language recognition using hidden Markov models". In: *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*. Vol. 1. 1997, 162–167 vol.1. DOI: 10.1109/ICSMC.1997.625742.

[26] M. Assan and K. Grobel. "Video-based sign language recognition using hidden markov models". In: *International Gesture Workshop*. Springer. 1997, pp. 97–109.

[27] I. Imagawa, H. Matsuo, R. Taniguchi, D. Arita, S. Lu, and S. Igi. "Recognition of local features for camera-based sign language recognition system". In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. Vol. 4. 2000, 849–853 vol.4. DOI: 10.1109/ICPR.2000.903050.

[28] O. Mercanoglu, A. O. Tur, and H. Y. Keles. "Isolated Sign Language Recognition with Multi-scale Features using LSTM". In: *2019 27th Signal Processing and Communications Applications Conference (SIU)* (2019), pp. 1–4.

[29] D. Li, C. Rodriguez, X. Yu, and H. Li. "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020, pp. 1459–1469.

[30] L. Pigou, A. van den Oord, S. Dieleman, M. V. Herreweghe, and J. Dambre. *Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video*. 2016. arXiv: 1506.01911 [cs.CV].

[31] J. Huang, W. Zhou, H. Li, and W. Li. "Attention-Based 3D-CNNs for Large-Vocabulary Sign Language Recognition". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.9 (2019), pp. 2822–2832. DOI: 10.1109/TCSVT.2018.2870740.

[32] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li. "Video-based sign language recognition without temporal segmentation". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[33] B. Shi, A. M. Del Rio, J. Keane, J. Michaux, D. Brentari, G. Shakhnarovich, and K. Livescu. "American sign language fingerspelling recognition in the wild". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 145–152.

[34] O. Koller. "Quantitative survey of the state of the art in sign language recognition". In: *arXiv preprint arXiv:2008.09918* (2020).

[35] R. Elakkiya. "Machine learning based sign language recognition: A review and its research frontier". In: *Journal of Ambient Intelligence and Humanized Computing* 12.7 (2021), pp. 7205–7224.

[36] K. M. Lim, A. W. C. Tan, C. P. Lee, and S. C. Tan. "Isolated sign language recognition using convolutional neural network hand modelling and hand energy image". In: *Multimedia Tools and Applications* 78.14 (2019), pp. 19917–19944.

[37] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakas, D. Papazachariou, and P. Daras. "A comprehensive study on sign language recognition methods". In: *arXiv preprint arXiv:2007.12530* 2 (2020).

[38] B. Bauer and H. Hienz. "Relevant features for video-based continuous sign language recognition". In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE. 2000, pp. 440–445.

[39] 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. 07. 2020, pp. 13009–13016.

[40] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. Piater, and H. Ney. "RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus". In: *Language Resources and Evaluation*. Istanbul, Turkey, May 2012, pp. 3785–3789. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/844_Paper.pdf.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[42] B. Garcia and S. A. Viesca. "Real-time American sign language recognition with convolutional neural networks". In: *Convolutional Neural Networks for Visual Recognition* 2 (2016), pp. 225–232.

[43] P. S. Santhalingam, P. Pathak, J. Košecké, H. Rangwala, et al. "Body Pose and Deep Hand-shape Feature Based American Sign Language Recognition". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2020, pp. 207–215.

[44] P. S. Santhalingam, P. Pathak, J. Košecká, H. Rangwala, et al. "Sign language recognition analysis using multimodal data". In: *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2019, pp. 203–210.

[45] M. M. Rahman, M. S. Islam, M. H. Rahman, R. Sassi, M. W. Rivolta, and M. Aktaruzzaman. "A new benchmark on american sign language recognition using convolutional neural network". In: *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE. 2019, pp. 1–6.

[46] K. Bantupalli and Y. Xie. "American sign language recognition using deep learning and computer vision". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 4896–4899.

[47] T. F. Dima and M. E. Ahmed. "Using YOLOv5 Algorithm to Detect and Recognize American Sign Language". In: *2021 International Conference on Information Technology (ICIT)*. IEEE. 2021, pp. 603–607.

[48] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak. "A new 2D static hand gesture colour image dataset for ASL gestures". In: (2011).

[49] S. Sharma and K. Kumar. "ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks". In: *Multimedia Tools and Applications* (2021), pp. 1–13.

[50] C. K. Lee, K. K. Ng, C.-H. Chen, H. C. Lau, S. Chung, and T. Tsoi. "American sign language recognition and training method with recurrent neural network". In: *Expert Systems with Applications* 167 (2021), p. 114403.

[51] A. A. Hosain, P. S. Santhalingam, P. Pathak, H. Rangwala, and J. Kosecka. "Hand Pose Guided 3D Pooling for Word-Level Sign Language Recognition". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 3429–3439.

[52] M. Dantone, J. Gall, C. Leistner, and L. van Gool. "Human Pose Estimation using Body Parts Dependent Joint Regressors". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Portland, OR, USA: IEEE, June 2013, pp. 3041–3048. DOI: 10.1109/CVPR.2013.391.

[53] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. "Using k-poselets for detecting people and localizing their keypoints". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3582–3589.

[54] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. "Deep convolutional neural networks for efficient pose estimation in gesture videos". In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 538–552.

[55] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. "Human Pose Estimation With Iterative Error Feedback". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[56] D. C. Luvizon, H. Tabia, and D. Picard. "Human pose regression by combining indirect part detection and contextual information". In: *Computers & Graphics* 85 (2019), pp. 15–22. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2019.09.002. URL: https://www.sciencedirect.com/science/article/pii/S0097849319301475.

[57] F. Zhang, X. Zhu, and M. Ye. "Fast human pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3517–3526.

[58] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. "Multi-Context Attention for Human Pose Estimation". In: *CoRR* abs/1702.07432 (2017). arXiv: 1702.07432. URL: http://arxiv.org/abs/1702.07432.

[59] G. Hinton, O. Vinyals, and J. Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].

[60] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper/2014/file/e744f91c29ec99f0e662c9177946c627-Paper.pdf.

[61] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. "Convolutional pose machines". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 4724–4732.

[62] Y. Luo, J. Ren, Z. Wang, W. Sun, J. Pan, J. Liu, J. Pang, and L. Lin. "LSTM Pose Machines". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[63] B. Artacho and A. Savakis. "UniPose: Unified Human Pose Estimation in Single Images and Videos". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[64] J. Liang and M. C. Lin. "Shape-aware human pose and shape reconstruction using multi-view images". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4352–4362.

[65] J. Martinez, R. Hossain, J. Romero, and J. J. Little. "A simple yet effective baseline for 3d human pose estimation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2640–2649.

[66] L. Chen, H. Ai, R. Chen, Z. Zhuang, and S. Liu. "Cross-view tracking for multi-human 3d pose estimation at over 100 fps". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 3279–3288.

[67] M. Oudah, A. Al-Naji, and J. Chahl. "Hand gesture recognition based on computer vision: a review of techniques". In: *journal of Imaging* 6.8 (2020), p. 73.

[68] L. Yun and Z. Peng. "An automatic hand gesture recognition system based on Viola-Jones method and SVMs". In: *2009 Second International Workshop on Computer Science and Engineering*. Vol. 2. IEEE. 2009, pp. 72–76.

[69] Y.-T. Chen and K.-T. Tseng. "Multiple-angle hand gesture recognition by fusing SVM classifiers". In: *2007 IEEE International Conference on Automation Science and Engineering*. IEEE. 2007, pp. 527–530.

[70] H.-K. Lee and J.-H. Kim. "An HMM-based threshold model approach for gesture recognition". In: *IEEE Transactions on pattern analysis and machine intelligence* 21.10 (1999), pp. 961–973.

[71] C. Feichtenhofer, A. Pinz, and A. Zisserman. "Convolutional two-stream network fusion for video action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1933–1941.

[72] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. "Long-term recurrent convolutional networks for visual recognition and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

[73] N.-H. Nguyen, T.-D.-T. Phan, G.-S. Lee, S.-H. Kim, H.-J. Yang, et al. "Gesture Recognition Based on 3D Human Pose Estimation and Body Part Segmentation for RGB Data Input". In: *Applied Sciences* 10.18 (2020), p. 6188.

[74] K. Hara, H. Kataoka, and Y. Satoh. "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 6546–6555.

[75] K. Lai and S. N. Yanushkevich. "CNN+RNN Depth and Skeleton based Dynamic Hand Gesture Recognition". In: *2018 24th International Conference on Pattern Recognition (ICPR)*. 2018, pp. 3451–3456. DOI: 10.1109/ICPR.2018.8545718.

[76] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Velez. "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition". In: *Pattern Recognition* 76 (2018), pp. 80–94.

[77] X. S. Nguyen, L. Brun, O. Lézoray, and S. Bougleux. "A neural network based on SPD manifold learning for skeleton-based hand gesture recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12036–12045.

[78] Q. Gao, Y. Chen, Z. Ju, and Y. Liang. "Dynamic Hand Gesture Recognition Based on 3D Hand Pose Estimation for Human-Robot Interaction". In: *IEEE Sensors Journal* (2021).

[79] Z. Xaitqulov. "AN OVERVIEW OF AUTOMATED TRANSLATION AND ITS LINGUISTIC PROBLEMS". In: *Philology Matters* 2021.1 (2021), pp. 139–149.

[80] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[81] I. Sutskever, O. Vinyals, and Q. V. Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

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

[83] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[84] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. "Video action transformer network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 244–253.

[85] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. "Transformer-xl: Attentive language models beyond a fixed-length context". In: *arXiv preprint arXiv:1901.02860* (2019).

[86] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[87] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, et al. "Sign language recognition, generation, and translation: An interdisciplinary perspective". In: *The 21st international ACM SIGACCESS conference on computers and accessibility*. 2019, pp. 16–31.

[88] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho. "Neural sign language translation based on human keypoint estimation". In: *Applied Sciences* 9.13 (2019), p. 2683.

[89] B. Saunders, N. C. Camgoz, and R. Bowden. "Progressive transformers for end-to-end sign language production". In: *European Conference on Computer Vision*. Springer. 2020, pp. 687–705.

[90] B. J. Bahan. "Non-manual realization of agreement in American Sign Language". PhD thesis. Boston University, 1996.

[91] C. Valli, C. Lucas, and K. MULROONEY. *Linguistics of American Sign Language: An Introduction 4th ed Washington*. 2005.

[92] U. v. Agris and K.-F. Kraiss. "Signum database: Video corpus for signer-independent continuous sign language recognition". In: *sign-lang@ LREC 2010*. European Language Resources Association (ELRA). 2010, pp. 243–246.

[93] A. Schembri, J. Fenlon, R. Rentelis, S. Reynolds, and K. Cormier. "Building the British sign language corpus". In: *Language Documentation & Conservation* 7 (2013), pp. 136–154.

[94] M. Zahedi, P. Dreuw, D. Rybach, T. Deselaers, and H. Ney. "Continuous sign language recognition-approaches from speech recognition and available data resources". In: *Second Workshop on the Representation and Processing of Sign Languages: Lexicographic Matters and Didactic Scenarios*. 2006, pp. 21–24.

[95] C. Vogler and C. Neidle. "A new web interface to facilitate access to corpora: development of the ASLLRP data access interface". In: *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*. 2012.

[96] *Roboflow American Sign Language Letters dataset*. 2021. URL: `https://public.roboflow.com/object-detection/american-sign-language-letters` (visited on 07/19/2021).

[97] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze. "How2: a large-scale dataset for multimodal language understanding". In: *arXiv preprint arXiv:1811.00347* (2018).

[98] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. "Panoptic studio: A massively multiview system for social motion capture". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3334–3342.

[99] S. K. Liddell et al. *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press, 2003.

[100] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". In: *IEEE transactions on pattern analysis and machine intelligence* 43.1 (2019), pp. 172–186.

[101] O. Crasborn and H. Sloetjes. "Enhanced ELAN functionality for sign language corpora". In: *6th International Conference on Language Resources and Evaluation (LREC 2008)/3rd Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora*. 2008, pp. 39–43.

[102] 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.

[103] S. Banerjee and A. Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

[104] C.-Y. Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[105] C. van Rijsbergen. *Information Retrieval, 2nd edButterworths*. 1979.

[106] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, et al. "Mediapipe: A framework for building perception pipelines". In: *arXiv preprint arXiv:1906.08172* (2019).

[107] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[108] H. R. V. Joze and O. Koller. "Ms-asl: A large-scale data set and benchmark for understanding american sign language". In: *arXiv preprint arXiv:1812.01053* (2018).