BAŞKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING
DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING

SIGN LANGUAGE RECOGNITION WITH ZERO-SHOT LEARNING

BY

GİRAY SERCAN ÖZCAN

DOCTOR OF PHILOSOPHY THESIS

ANKARA - 2024

BAŞKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING
DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING

SIGN LANGUAGE RECOGNITION WITH ZERO-SHOT LEARNING

BY

GİRAY SERCAN ÖZCAN

DOCTOR OF PHILOSOPHY THESIS

ADVISOR
ASSOC. PROF. DR. EMRE SÜMER

CO-ADVISOR
DR. YUNUS CAN BİLGE

ANKARA - 2024

BAŞKENT UNIVERSITY

INSTITUTE OF SCIENCE AND ENGINEERING

This study, which was prepared by Giray Sercan ÖZCAN, has been approved in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in the Computer Engineering Department by the following committee.

Date of Thesis Defense: 26 / 07 / 2024

Thesis Title: Sign Language Recognition with Zero-Shot Learning

Examining Committee Members	**Signature**
Prof. Dr. Uğur Murat LELOĞLU, THK University	. . . . .
Assoc. Prof. Dr. Emre SÜMER, Başkent University	. . . . .
Assoc. Prof. Dr. Mustafa SERT, Başkent University	. . . . .
Assist. Prof. Dr. Hakan TORA, Biruni University	. . . . .
Assist. Prof. Dr. Çağatay Berke ERDAŞ, Başkent University	. . . . .

APPROVAL

Prof. Dr. Dilek ÇÖKELİLER SERDAROĞLU
Director, Institute of Science and Engineering
Date: . . . . / . . . . / 2024

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: 01/08/2024

Öğrencinin Adı, Soyadı: Giray Sercan ÖZCAN
Öğrencinin Numarası: 21410130
Anabilim Dalı: Bilgisayar Mühendisliği Anabilim Dalı
Programı: Bilgisayar Mühendisliği Doktora Programı
Danışmanın Unvanı/Adı, Soyadı: Doç. Dr. Emre SÜMER
Tez Başlığı: Sıfır-Atış Öğrenmesi ile İşaret Dili Tanıma

Yukarıda başlığı belirtilen Doktora tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam ..... sayfalık kısmına ilişkin .... / .... / 2024 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orjinallik raporuna göre, tezimin benzerlik oranı .....'dır. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar Hariç
3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

"Başkent Üniversitesi Enstitüleri Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını" inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası: .....

ONAY
Öğrenci Danışmanı
Doç. Dr. Emre SÜMER
Tarih: .... / .... / 2024

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis advisor, Assoc. Prof. Dr. Emre SÜMER, and my thesis co-advisor, Dr. Yunus Can BİLGE, for their contributions.

I also want to express my gratitude to the members of my thesis monitoring committee members, Prof. Dr. Uğur Murat LELOĞLU and Assoc. Prof. Dr. Mustafa SERT, for their support and for sharing their valuable insights and suggestions.

I would like to express my heartfelt thanks to my family, who consistently supported me and made every sacrifice throughout the thesis process.

Finally, I would like to thank my colleagues and friends who shared their thoughts and ideas with me throughout the thesis process.

ABSTRACT

Giray Sercan ÖZCAN

SIGN LANGUAGE RECOGNITION WITH ZERO-SHOT LEARNING
Başkent University Institude of Science and Engineering
Department of Computer Engineering
2024

Sign language holds great importance for a specific segment of society. Automating Sign Language Recognition (SLR) using machine learning is crucial for facilitating communication between different segments of society. However, creating the necessary labeled data for this task is very challenging. Furthermore, the evolution and changing meanings of sign language words over time make this field even more difficult. This work presents a novel approach to Zero-Shot Sign Language Recognition (ZSSLR). Using hand and landmark data extracted from the signer's body data, the signer's hand and body have been modeled. To determine which of the extracted and modeled features are more important for this purpose, a data grading method was applied. In Zero-Shot Learning (ZSL), datasets containing descriptions of the movements in sign language videos were used. The results were tested on two benchmarkable ZSL datasets and demonstrated in ZSL and Generalized Zero-Shot Learning (GZSL) settings.

KEYWORDS: Sign Language Recognition, Zero-Shot Sign Language Recognition, Zero-Shot Learning

ÖZET

Giray Sercan ÖZCAN

SIFIR ATIŞ ÖĞRENMESİ İLE İŞARET DİLİ TANIMA
Başkent Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
2024

İşaret dili, toplumun belirli bir kesimi için büyük önem taşımaktadır. İşaret Dili Tanımanın (SLR) makine öğrenmesi kullanılarak otomatikleştirilmesi, toplum kesimlerinin iletişimini kolaylaştırmak için çok önemlidir. Ancak, bu görev için gerekli olan etiketlenmiş verilerin oluşturulması oldukça zordur. Dahası, zaman içinde işaret dili kelimelerinin evrim geçirip anlamlarının değişmesi bu alanı daha da zor hale getirmektedir. Bu çalışma, Sıfır-Shot İşaret Dili Tanıma (ZSSLR) için yenilikçi bir yaklaşım sunmaktadır. İşaretçinin vücut verilerinden çıkarılan el ve landmark verileri kullanılarak, işaretçinin el ve vücudu modellenmiştir. Çıkarılan ve modellenen özniteliklerin bu amaç için hangisinin daha önemli olduğunu belirlemek amacıyla bir veri derecelendirme yöntemi uygulanmıştır. Sıfır-Shot Öğrenmede (ZSL), işaret dili videolarında yapılan hareketlerin tanımlarını içeren veri kümeleri kullanılmıştır. Sonuçlar, iki karşılaştırılabilir ZSL veri kümesinde test edilmiş ve ZSL ve Genel Sıfır-Shot Öğrenme (GZSL) ayarlarında gösterilmiştir.

KEYWORDS: Sign Language Recognition, Zero-Shot Sign Language Recognition, Zero-Shot Learning

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

1D-CNN	1-Dimensional Convolutional Neural Network
2D-CNN	2-Dimensional Convolutional Neural Network
3D-CNN	3-Dimensional Convolutional Neural Network
ALF	Attribute-Level Fusion
ART	Adaptive Resonance Theory
BdSL	Bangla Sign Language
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
BLSTM-NN	Bidirectional Long Short-Term Memory Neural Network
BPN	Back Propagation Network
CLF	Cross-Level feature Fusion
CLIP	Contrastive Language-Image Pretraining
CNN	Convolutional Neural Networks
CorrNet	Correlation Network
CSE	Commonsense Embeddings
CSLR	Continuous Sign Language Recognition
CTC	Connectionist Temporal Classification
DAP	Direct Attribute Prediction
DoF	Degrees of Freedom
DSVTM	Dual Semantic-Visual Transformer Module
FGA	Field Programmable Gate Array
FLF	Feature-Level Fusion
GCN	Graph Convolutional Network
GNN	Graph Neural Networks
GZSL	Generalized Zero-Shot Learning
HaGRID	Hand Gesture Recognition Image Dataset
HCRF	Hidden Conditional Random Fields
HOG	Histogram of Oriented Gradients
HRT	Hybrid Routing Transformer
IMSE	Instance-Motivated Semantic Encoder
InterLSR	Inter-layer Language-Specific Routing
IntraLSR	Intra-layer Language-Specific Routing
ISLR	Isolated Sign Language Recognition
kNN	k-Nearest Neighbor
LDA	Linear Discriminant Analysis
LLE	Logistic Label Embedding
LSTM	Long Short-Term Memory

LSTM-RNN	Long-Short Term Memory Recurrent Neural Network
MLP	Multilayer Perceptron
MS-G3D	Multi-Scale Spatial-Temporal Graph Convolutional Networks
MSLT	Multilingual Sign Language Translation
MViTv2	Multiscale Vision Transformer
MZSL	Multi-label Zero-Shot Learning
PBD	Prototype-Based Detector
PFFN	Position-Wise Feed-Forward Network
PSVMA	Progressive Semantic-Visual Mutual Adaptation
ResNeXt	Residual Networks Next
RGB	Red-Green-Blue
SAE	Semantic Auto-Encoder
SAM-SLR	Skeleton Aware Multi-modal Sign Language Recognition
SL-GCN	Sign Language Graph Convolution Network
SLR	Sign Language Recognition
SMC	Spatial Multi-Cue
SMID	Semantic-Motivated Instance Decoder
SPOT	Selection using Proximal Policy Optimization
SSTCN	Separable Spatial-Temporal Convolution Network
ST-GCN	Spatial-Temporal Graph Convolutional Network
STMC	Spatial-Temporal Multi-Cue
TDK	Turkish Language Association
TMC	Temporal Multi-Cue
VA	Visual Alignment
VAC	Visual Alignment Constraint
VE	Visual Enhancement
ViT	Vision Transformer
ZSSLR	Zero-Shot Sign Language Recognition

1. INTRODUCTION

Sign language serves as a communication tool for deaf individuals. In a way, each person uses their own sign language by making signs, although not necessarily at the level of a deaf person, but since sign languages are more extensive, they fail to communicate with deaf people, leading to a deep communication gap between the deaf and the hearing. This problem is being addressed with sign language recognition systems.

The datasets necessary for training sign language recognition systems are few in number when considering the number of sign languages and the sign words contained within them ([11], [12], [13], [14], [15]). This represents a scarcity of data for machine learning. Therefore, the necessity of a system that can be trained without the need for extensive data and can yield results is apparent. Hence, this study proposes a Zero-Shot Sign Language Recognition (ZSSLR) system.

Sign language can be defined as a language that uses hand movements as a means of conveying meaning in communication. It involves a speaker using hand gestures, fingers, arm or body movements, direction, and facial expressions simultaneously to convey their ideas. Each movement represents a different word [16][17]. Additionally, an alphabet has been established for each letter of the spoken language used in the geography where sign language has developed, to express proper nouns in sign language.

There is no standardization in Turkish sign language, and regional variations are observed. However, the sign language dictionary and the Turkish Sign Language Grammar Book [18] published by the Turkish Language Association (TDK) are examples of standardization efforts.

Just as in spoken languages like Turkish, English, Chinese, etc., sign languages such as American Sign Language (ASL), Indian Sign Language (ISL), Chinese Sign Language (CSL), etc., have their own word/sign structure, syntax, question formations, semantic structures, and grammatical rules. Sign languages have a complex and rich structure like spoken languages and are used to convey poetry, theater, and literary works. There is not just one sign language in the world; many sign languages exist, and as mentioned earlier, these languages can vary according to the geography of the region. Taking these points into account, it is believed that the world hosts over 100 sign languages, each with many distinct dialects [19], and each language comprising over 3000 words [16][17]. Knowing one sign language does not mean knowing others.

The movements made to perform sign language are divided into static and dynamic.

This presents various challenges from a computer vision perspective. For example, while a single image may be sufficient for static movements, dynamic movements require consideration of the direction the body is facing.

This thesis aims to recognize previously unseen classes by using textual descriptions and visual representations of the classes. The visual representation models spatial dependencies, such as where the hand is located, and temporal dependencies such as movements in sign language videos by using hand and landmark data extracted from the body data stream, in addition to the body data itself. Residual Networks Next (ResNeXt) [20], Spatial-Temporal Graph Convolutional Network (ST-GCN) [10], Long Short-Term Memory (LSTM), and the recently proposed Multiscale Vision Transformers (MViTv2) [21] modules were utilized for achieving this. The ResNeXt architecture, developed for recognizing hand gestures, uses the HaGRID [1] dataset to model spatial dependencies. Meanwhile, ST-GCN works on hand landmarks to model hand movements, which are critical for sign language, both spatially and temporally. The MViTv2 model, developed as an action recognition architecture, is of great importance for modeling advanced spatial and temporal dependencies.

The hand and body regions are critical for SLR as even the smallest movement changes in these areas can drastically change a sign's meaning. Therefore, our framework includes a component dedicated to analyzing the hand and body segments extracted from videos. MViTv2 and ResNeXt models are used for extracting features from these segments. To integrate these features into the system, methods like Bidirectional Long Short-Term Memory (Bi-LSTM), 1-Dimensional Convolutional Neural Network (1D-CNN), and self-attention were employed. Landmark data is processed using Bi-LSTM and ST-GCN. Given the substantial data from diverse sources, data grading were implemented to efficiently manage and prioritize relevant information. Contrastive Language-Image Pretraining (CLIP) [22] architecture were used for extracting semantic information from textual class definitions, mapping visual embeddings to their nearest semantic classes. This mapping is intended to show the closest resemblance of the extracted information to specific classes. This thesis goal is to align the learned visual representation with its textual counterpart. ZSSLR benchmark datasets ASL-Text [2] and MS-ZSSLR-W/C [3] were used for rigorously evaluated this approach, demonstrating its effectiveness.

1.1. Contribution of The Thesis

Due to the large number of sign languages and the abundance of sign words within these languages, there has been a lack of data, prompting the development of a zero-shot learning system for sign language recognition. The key innovations of this study are as follows: (1) this thesis is the first to apply the advanced feature extraction capabilities of MViTv2 and ResNeXt to tackle the ZSSLR problem domain, introducing a new method in

this field, (2) this study is the first in using the ST-GCN method for ZSSLR, showcasing its potential to capture complex spatial and temporal relationships, (3) first to utilize the CLIP architecture for extracting textual vectors essential for effective ZSSLR, (4) this study is the first to integrate these advanced techniques, combining MViTv2, ResNeXt, ST-GCN, and CLIP.

1.2. Structure of The Thesis

In the introduction section, the purpose and the summary of the thesis is described. Section two offers a review of related works in the literature. Section three outlines the problem definition and presents the baseline study. The approach for tackling the ZSSLR problem is detailed on section four. Section five discusses the datasets and their characteristics, as well as provides an extensive evaluation and analysis of our models. Lastly, the goal of this thesis, results and future work are discussed in conclusion section.

2. LITERATURE REVIEW

2.1. Sign Language Recognition

SLR can be categorized into two main types: Isolated SLR (ISLR) [23] and Continuous SLR (CSLR)[24]. ISLR is concerned with identifying individual signs or gestures presented separately, which simplifies the recognition task by minimizing variability and dependence on context. Conversely, CSLR focuses on recognizing sign language in natural, continuous sequences. This requires identifying signs both in isolation and within the context of preceding and succeeding signs, adding complexity due to the variability in sign transitions and the impact of non-manual features.

Alphabet recognition and production of alphabetical signs were conducted by Hruz et al. [25]. It was one of the first studies to use a single webcam for data collection. Kindiroglu et al.[26] improved upon the work of [25]. The dataset from [25] was expanded to create a richer dataset. An adaptive skin color-based alphabet recognition system was developed. Two algorithms were used for classification: Hidden Markov Model (HMM) and k-Nearest Neighbor (kNN).

In their study Rivera et al. [27], a Field Programmable Gate Array (FGA) was used to develop a sign language alphabet translation system. Data acquisition was performed using a neuromorphic camera. This camera detects dynamic changes in brightness in the scene and sends data whenever there is any change. The feature extraction phase was carried out using digital image processing techniques aimed at eliminating unnecessary information and reducing the system's computational cost.

Bhuiyan et al. conduct[28] an alphabet recognition study on ASL for human-robot interaction. Unlike other studies, the Adaptive Resonance Theory (ART) neural network method was used for classification. Data were collected with a standard digital camera under different lighting conditions and complex backgrounds. However, only the hand was visible in the image frame, and static frames were used.

Kumar et al. [29] developed an ISLR for words using sensors. Leap Motion sensors and Microsoft Kinect were used to detect finger and palm positions. The Leap Motion sensor was positioned just below the hands, while the Kinect sensor was placed directly opposite the signer to capture horizontal and vertical finger positions. They extracted features from the sensor data and performed recognition using both HMM and Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN) based sequential classifiers.

Research in ISLR and CSLR can be arranged into two main categories: those that

rely on manually extracted features ([30], [31], [32], [33], [34], [35], [9]) and those that utilize deep learning architectures ([8], [36], [37], [38]). Studies involving manually extracted features often employed techniques like HMM and Hidden Conditional Random Fields (HCRF). One of the pioneering real-time studies by Starner et al. [39] implemented an HMM-based approach for CSLR using a wearable camera. Recent research includes works that operate under weak supervision ([40], [41], [42]).

More recent works like [43] propose a Skeleton Aware Multi-modal SLR (SAM-SLR) framework that integrates multi-modal information from RGB, depth, and skeleton data. They use a Sign Language Graph Convolution Network (SL-GCN) to model dynamic skeleton information and a Separable Spatial-Temporal Convolution Network (SSTCN) for detailed skeleton features. They are the first to do the construction of a skeleton graph using whole-body keypoints, the development of SSTCN and SL-GCN models, and the integration of these with RGB and depth data.

Lee et al. [44] address the problem of developing an effective and real-time educational application for learning sign language. It utilizes a Leap Motion Controller to capture hand gestures and employs a Long-Short Term Memory Recurrent Neural Network (LSTM-RNN) combined with a kNN method for classifying the sign language alphabet. Their contributions include a high recognition accuracy on sign language alphabets, the integration of a real-time sign recognition system into an interactive learning game, and the enhancement of sign language learning through an engaging game-based approach.

Hu et al. [45] propose a ISLR framework for addressing the challenge of recognizing sign language using hand gestures, which often have limited overfitting and interpretability issues due to limited training data. To solve this, the authors introduce a hand-model-aware framework that integrates a hand prior to enhance the recognition process. This framework includes a visual encoder, a hand-model-aware decoder, and an inference module, which work together to transform hand sequences into latent semantic features, refine them with a statistical hand model, and enhance spatio-temporal pose representations for recognition. The contributions of the paper include introducing the hand prior, constructing a hand-model-aware framework, and achieving state-of-the-art (SOTA) performance on four benchmark datasets.

Hu et al. [46] propose a Correlation Network (CorrNet) that includes a correlation module to compute correlation maps between identification module and consecutive frames to emphasize body trajectories within these maps. They use 2-Dimensional Convolutional Neural Network (2D-CNN) for capturing frame-wise features from input frames and 2-Dimensional Convolutional Neural Network (1D-CNN) for temporal modeling. Connec-

tionist Temporal Classification (CTC) lossis utilized for predicting the probability of target gloss sequences, aligning target sentences with input frames.

Zhou et al. [47] recommend a Spatial-Temporal Multi-Cue (STMC) method for SLR by capturing and integrating multiple visual cues, such as hand shapes, facial expressions, and body posture. Their method includes three modules: Spatial Multi-Cue (SMC), Temporal Multi-Cue (TMC) and sequence learning module.

Min et al. [15] address the challenge of overfitting in CSLR models, which hampers the adequate training of the feature extraction component. They use 2D-CNN, 1D-CNN and Bi-directional Long Short-Term Memory (Bi-LSTM) for temporal modeling and Visual Alignment Constraint (VAC) by Visual Enhancement (VE) and Visual Alignment (VA) loss. The VAC aims to improve the generalization capabilities of the visual feature extractor by providing alignment supervision that guides the organization of the feature space.

Tunga et al. [48] propose an architecture for ISLR by effectively capturing both spatial and temporal dependencies from pose information in sign language videos. They utilize pose-based Graph Convolutional Network (GCN), BERT [49] for temporal modeling and Position-Wise Feed-Forward Network (PFFN).

Vazquez et al. [50] utilize Multi-Scale Spatial-Temporal Graph Convolutional Networks (MS-G3D) for capturing the complex and dynamic nature of sign language. Wei et al. [51] introduce a method for CSLR by utilizing cross-lingual data, thereby addressing the critical challenge of data scarcity in SLR.

Boháček et al. [52] present a system for word-level SLR using a Transformer model, focusing on low computational cost to enable usage on hand-held devices. They use 2D landmark locations for body pose estimation, with a robust pose normalization scheme and several augmentations, including sequential joint rotation augmentation.

Sreemathy et al. [53] propose an automated SLR system designed to enhance the cognitive skills of hearing-impaired children through the use of artificial intelligence. They utilize HSV color space and histogram equalization for preprocessing, Histogram of Oriented Gradients (HOG) for feature extraction and Back Propagation Network (BPN) and deep learning models for classification.

Kothadiya et al. [54] introduce a Transformer-based architecture designed for recognizing static sign language. They employ a Vision Transformer (ViT) that divides images into patches, which are then processed by a Transformer Encoder with four self-attention

layers and a Multilayer Perceptron (MLP) for classification.

Shin et al. [55] propose a multi-branch network combining Convolutional Neural Networks (CNN) and Transformers SLR, addressing the challenges of light illumination and background complexity. They use an initial grain module to extract features, followed by parallel feature extraction using CNN and transformers. The extracted features are then merged and processed through a classification module that incorporates global average pooling, fully connected layers, and softmax activation.

Bora et al. [56] aim to develop a real-time recognition system for SLR using the MediaPipe framework and deep learning. The approach involves creating a dataset of 2094 data points for nine static sign language gestures using both 2D and 3D images, extracting hand landmarks with MediaPipe, and training a feedforward neural network with the extracted landmarks.

2.2. Zero-Shot Learning

Larochelle et al. [57] set the stage for ZSL research, with Palatucci et al. [58] and Lampert et al. [59] conducting the first key studies. They introduced a method that transfers knowledge to unseen classes through shared attributes between observed and unobserved classes. Since these initial works, numerous classification studies in ZSL have emerged, using methods optimized for various goals ([60], [61], [62], [63], [64]).

Recent studies such as [65] present a Multi-label Zero-Shot Learning (MZSL) model using GCN to recognize novel categories with no annotated training data. This model constructs a label relation graph using label co-occurrences and semantic similarities, and then applies GCN to learn label semantic embeddings. It also employs an attention network to capture local and global visual features of objects. The key innovations of this paper are the integration of GCNs to explore label correlations, an attention mechanism for adaptive feature learning, and the use of unlabeled data to reduce bias towards seen labels.

Chen et al. [66] introduce TransZero, an attribute-guided Transformer network designed to improve ZSL by refining visual features and localizing object attributes for more discriminative visual embeddings. This model features a feature augmentation encoder designed to mitigate cross-dataset bias and minimize intertwined region relationships, alongside a visual-semantic decoder that focuses on developing visual features enhanced by locality and guided by semantic attributes. It incorporates an attribute-based cross-entropy loss, attribute regression loss, and self-calibration loss to optimize performance. Their contributions are the use of an attribute-guided Transformer to enhance visual feature transferability and attribute localization, and the incorporation of feature augmentation and visual-semantic

interaction to address cross-dataset bias and improve discriminative region features.

Roy et al. [67] explore the use of commonsense knowledge from ConceptNet to enhance ZSL by generating Commonsense Embeddings (CSE) to improve the association between visual and semantic embeddings. They employ a GCN-based autoencoder to encode commonsense knowledge from ConceptNet, creating CSE of class labels. These embeddings are then fused with existing semantic embeddings which is human-defined attributes and distributed word embeddings.The key contributions of this paper are the introduction of a GCN-based autoencoder for generating CSE and the effective fusion of these embeddings with traditional semantic embeddings highlight the potential of commonsense knowledge in enhancing ZSL tasks.

Wang et al. [68] introduce a domain-aware multi-modality fusion network to address the challenges of Generalized Zero-Shot Learning (GZSL), specifically focusing on overcoming the bias problem where unseen samples are misclassified into seen classes. They develop a two step model: initially, a local neighborhood-based gating model identifies and segregates seen and unseen samples, effectively simplifying the GZSL challenge into a straightforward ZSL and supervised classification problem. Subsequently, a GCN-based model is employed to fuse multiple semantic modalities to enhance the ZSL task. Their innovation includes the introduction of a local neighborhood-based domain-aware mechanism for efficient domain detection and a GCN-based multi-modality fusion network to merge different semantic representations for creating effective discriminative classifiers.

Gupta et al. [69] address the challenge of classifying images into multiple unseen categories in a ZSL setting using a generative approach. They introduce three fusion approaches for synthesizing multi-label features: Feature-Level Fusion (FLF), Attribute-Level Fusion (ALF), and Cross-Level feature Fusion (CLF). The ALF method generates a global image-level embedding, FLF synthesizes features from class-specific embeddings and integrates them, and CLF combines both ALF and FLF to enhance label dependency and class-specific discriminability.

Cheng et al. [70] introduce the Hybrid Routing Transformer (HRT) to address ZSL by establishing a stronger semantic alignment between attribute vectors and visual features through a transformer-based architecture. HRT combines dynamic top-down and bottom-up routing pathways in its encoder to produce attribute-aligned visual features and uses static routing in its decoder to translate these features into class predictions. The model includes an active attention mechanism that integrates dynamic routing for both visual and semantic information.

Gowda et al. [71] propose an approach for selecting synthetic features that enhance classification performance, rather than just generating real-looking samples. Their approach, named Selection using Proximal Policy Optimization (SPOT), employs a transformer-based selector that is trained via reinforcement learning to identify the most informative synthetic features. This selection is based on the validation classification accuracy of the observed classes.

Guo et al. [72] present a novel approach to ZSL by redefining it as a sample-level graph recognition task, which enhances the accuracy and robustness in identifying unseen classes. Their method involves breaking down each input sample into fine-grained components and creating a graph structure for each to illustrate the relationships among these elements. Graph Neural Networks (GNN) are subsequently employed to align these sample-level graphs with semantic space, utilizing correlations between elements and semantics as well as local sub-structural data.

Liu et al. [73] introduce the Progressive Semantic-Visual Mutual Adaptation (PSVMA) network, designed to progressively align semantic attributes with visual features to diminish semantic ambiguity and improve knowledge transferability. The PSVMA network utilizes a Dual Semantic-Visual Transformer Module (DSVTM), which includes an Instance-Motivated Semantic Encoder (IMSE) and a Semantic-Motivated Instance Decoder (SMID). The IMSE adapts shared attributes across different visual features, converting mismatched semantic-visual pairs into aligned ones. Meanwhile, the SMID facilitates cross-domain interactions and refines visual representations by embedding significant semantic features into visual patches.

2.3. Zero-Shot Sign Language Recognition

Bilge et al. [2] were the first to define the ZSSLR problem and proposed a solution by comparing various methods. Their architecture can be seen on Figure 2.1. They created the ASL-Text dataset by combining sign language videos with descriptions of the movements from sign language dictionaries. The proposed model leverages a bi-linear compatibility function to associate video and class representations by embedding videos and textual descriptions of sign classes into respective feature spaces using a 3-Dimensional Convolutional Neural Network (3D-CNN) and BERT [49] model, and then learns a compatibility matrix through Logistic Label Embedding (LLE) optimized with cross-entropy loss and L2-regularization. In their subsequent study [3], the same authors expanded the ASL-Text dataset to include binary attribute matrices and created two datasets, MS-ZSSLR-W and MS-ZSSLR-C. Their model is illustrated in Figure 2.2. In addition to the methods from their previous work, they experimented with shift-based CNN [74], 3D-CNN, and LSTM methods on these datasets.

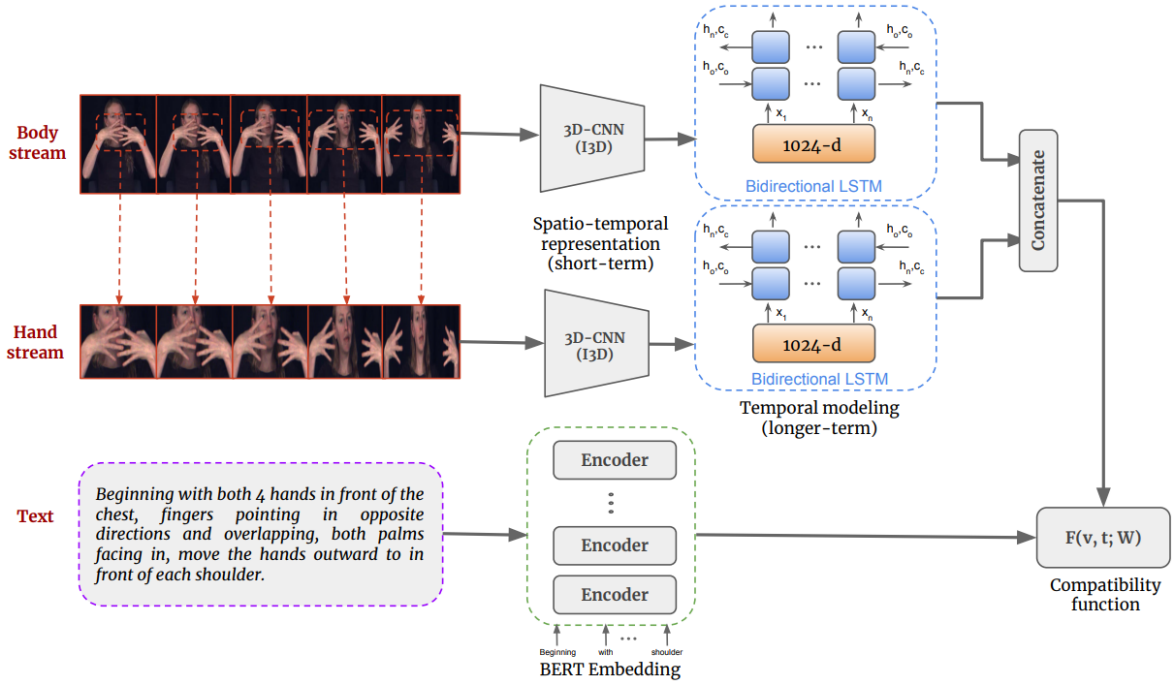


Figure 2.1: Bilge et al. [2] architecture

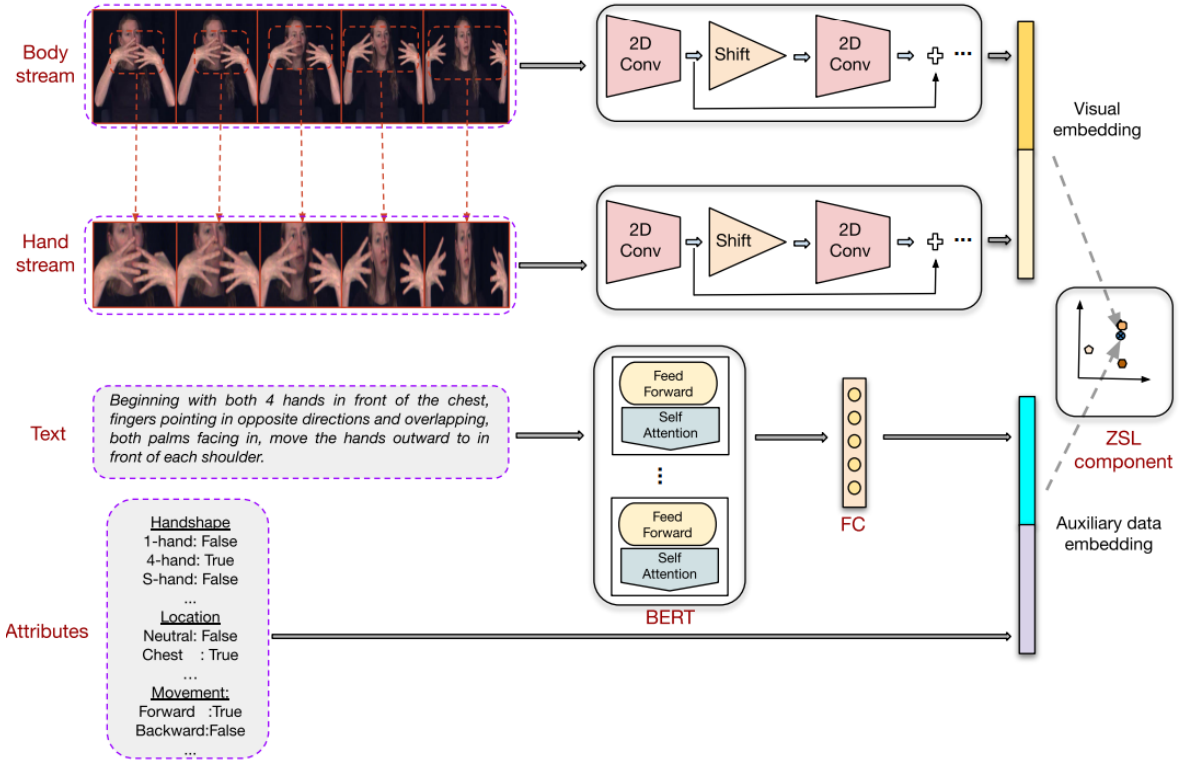


Figure 2.2: Bilge et al. [3] architecture

Nihal et al. [4] propose two approaches for the automatic recognition of Bangla Sign Language (BdSL) alphabets: a traditional transfer learning approach and a contemporary ZSL approach to recognize both seen and unseen signs. Their architecture which consists of four components can be seen in Figure 2.3. The transfer learning method utilizes the pre-trained DenseNet201 for extracting features and Linear Discriminant Analysis (LDA)

for classification and ZSL approach employs a set of semantic descriptors specific to BdSL. These attributes include details about finger Degrees of Freedom (DoF), position, view, and orientation. Direct Attribute Prediction (DAP) method is used in training, where each semantic attribute is learned via a classifier. DenseNet201 features are used as input, and the model is trained to predict the semantic attributes corresponding to each class.

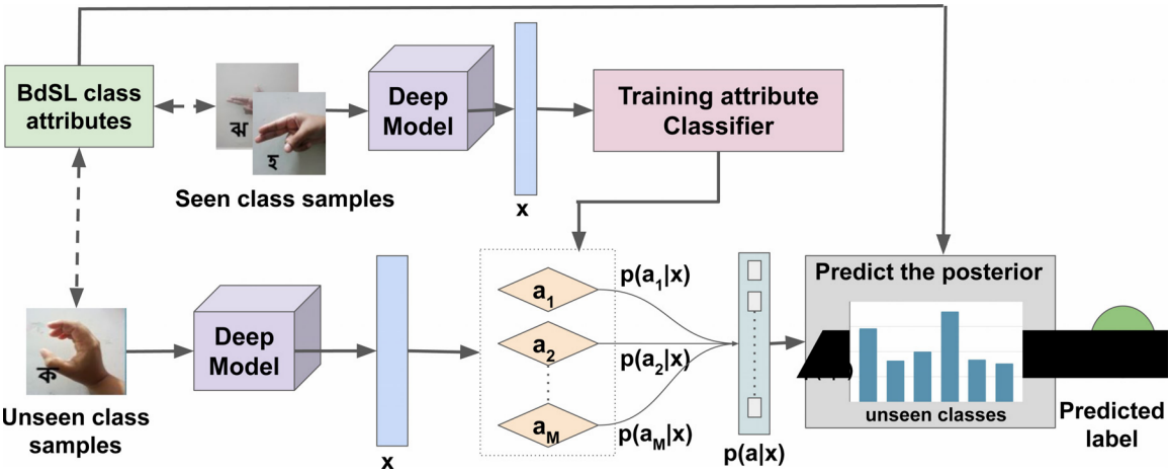


Figure 2.3: Nihal et al. [4] model

Wu et al. [5] propose a framework that includes two primary elements: a Prototype-Based Detector (PBD) and a zero-shot label predictor. Their architecture can be seen in Figure 2.4. This approach aims to recognize both observed and unobserved hand gesture classes by integrating prototype learning and semantic feature mapping. PBD utilizes a multi-layer Bi-LSTM network to extract features from hand gesture sequences. This branch determines if a sample belongs to a seen or unseen category by comparing the distance of the sample's feature representation to learned class prototypes. Zero-shot label predictor uses a Semantic Auto-Encoder (SAE) to map feature representations to a semantic space. It predicts labels for samples classified as unseen by mapping the extracted features to semantic attributes. Input sequences are recorded using a Leap Motion Controller, which tracks the palm center, hand direction, and positions of skeletal joints.

Yin et al. [6] address the problem of building efficient and scalable Multilingual Sign Language Translation (MSLT) systems. Their model is illustrated in Figure 2.5. They use a transformer-based model, which is well suited for sequence-to-sequence tasks. For video embeddings, they employ a CNN to extract features from individual frames, subsequently mapping these features to a denser space using a linear layer. A multilingual sub-word segmentation model is used for text embedding. Sub-word embeddings are initialized using pre-trained embeddings and adjusted through a linear layer. They introduce two dynamic routing mechanisms: Intra-layer Language-Specific Routing (IntraLSR) and Inter-layer Language-Specific Routing (InterLSR). IntraLSR manages the data flow between shared and language-specific parameters within a transformer layer, while InterLSR regulates the extent of param-

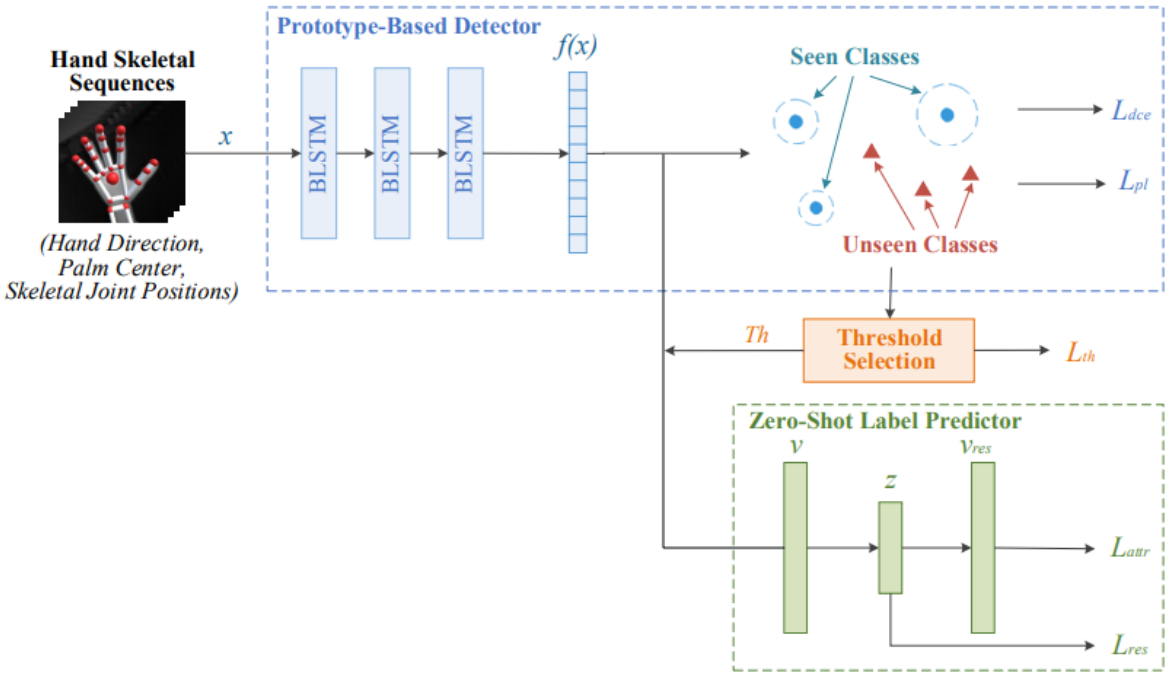
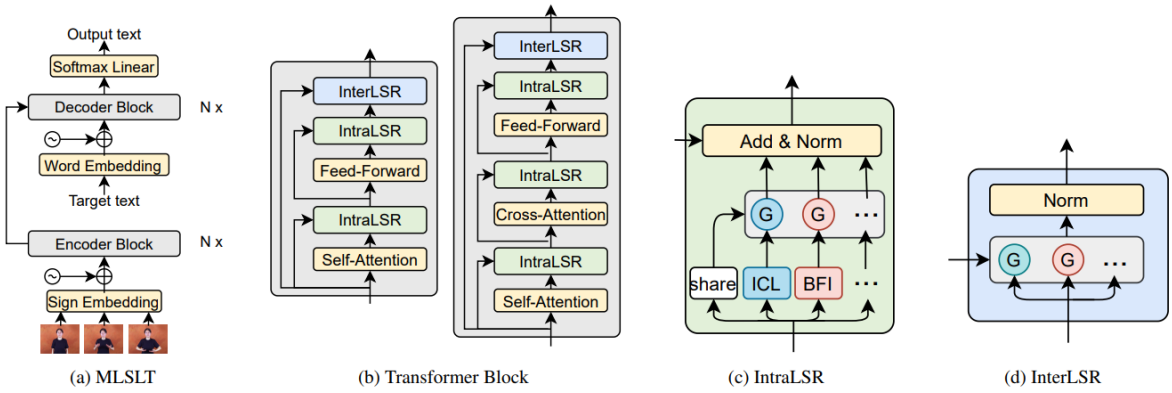


Figure 2.4: Wu et al. [5] model



(a) MLSLT (b) Transformer Block (c) IntraLSR (d) InterLSR

Figure 2.5: Yin et al. [6] model

eter sharing across different languages between transformer layers.

3. BACKGROUND

3.1. Sign Language Recognition

SLR spans across the fields of machine learning, computer vision, and natural language processing. It is dedicated to the automated interpretation of sign languages, which are complete, structured natural languages with their own syntax and grammar. These languages typically utilize hand shapes, orientations, movements, facial expressions, and body postures for conveying messages. An example of sign language is depicted in Figure 3.1. The primary objective of SLR is to enhance communication between deaf or hard-of-hearing individuals and those who are not proficient in sign language, thereby increasing accessibility and inclusivity in various social, educational, and professional environments.

Figure 3.1: Sign language example

SLR systems begin by capturing visual data using cameras or sensors, which is then processed to identify and isolate key features such as movements, hand shapes, facial expressions, and body postures. Advanced computer vision technologies, such as deep learning methods including CNNs and RNNs [9], are employed to examine these features and recognize patterns associated with specific signs. These patterns are then converted into their corresponding meanings or equivalents in spoken language.

The field of SLR faces several challenges, including the variability of signing styles across individuals, the complexity of simultaneous hand and body movements, and the need

to incorporate contextual information. Another significant challenge is the creation of large, annotated datasets that reflect the diversity of sign languages, which is both crucial and difficult. Despite these obstacles, advancements in deep learning and the availability of enhanced computational resources have led to notable progress in the field, resulting in more precise and reliable SLR systems.

The potential applications of SLR are vast, ranging from automated sign language interpreters [75] and educational resources for learning sign language [76] to improved human-computer interactions for the deaf community. Ongoing advancements in SLR technology continue to promote greater inclusion and accessibility, contributing significantly to a more equitable society.

3.2. Sign Language Recognition Datasets

Studies in the field of SLR are divided into three types based on the variety of datasets they focus on: letter recognition [77], word recognition [78], and continuous sign language recognition [9]. Alphabet recognition attempts to recognize a single letter in sign language, word recognition attempts to recognize a single word, and continuous sign language recognition aims to recognize one or more sentences expressed in sign language as used in real life. Examples of these datasets will be given in the following paragraphs.

An example of a dataset [7] used for letter recognition is given in Figure 3.2. Here, samples from the 5 instances for each letter are provided. Sign language letter datasets typically include attributes such as images or videos of hand gestures representing letters of the alphabet. These datasets capture variations in hand shapes, orientations, and movements, often under different lighting conditions and backgrounds. Key attributes include the position of the hand relative to the body, finger configurations, and sometimes the use of both static and dynamic gestures.

Figure 3.2: Sign language letter recognition dataset example [7]

Sign language word datasets are essential for building effective SLR systems. An

illustration of such a dataset [8] used for word recognition is presented in Figure 3.3, featuring signers in varying backgrounds, lighting conditions, and appearances. These datasets are typically composed of video recordings or image sequences that depict individuals executing sign language gestures. Each dataset entry is labeled with the corresponding sign language words or phrases, serving as a ground truth for training and testing machine learning models. Key attributes of these datasets include a wide variety of signers to accommodate differences in signing styles and speeds, high-definition video to capture clear hand and finger movements, and an extensive vocabulary that encompasses a broad range of sign language words.

Figure 3.3: Example of sign language word recognition dataset [8]

Figure 3.4: Example of continuous SLR dataset [9]

Datasets for continuous SLR are vital for progressing the automated interpretation of seamless sign language interactions. An example of a dataset [9] used for this, taken from real-life scenarios, is shown in Figure 3.4. This dataset is derived from sign language videos featured on German television. These datasets usually consist of video recordings showing signers executing long sequences of signs in a manner that reflects natural, fluid communication. Important features of these datasets include high temporal resolution to capture the smooth transitions between signs and precise annotations that mark the beginning and end of each sign in the ongoing sequence. Such annotations are critical for training models to accurately segment and identify individual signs. The datasets typically include several signers to incorporate variation in signing styles, speeds, and personal nuances, which helps improve the generalizability of recognition systems. High-definition video quality is crucial for clearly depicting complex hand, facial, and body movements. Additionally,

extensive metadata, including details about the signers, recording conditions, and contextual background, aids in developing robust models suited for various real-world environments. The presence of multiple sign languages and dialects in these datasets also enhances their applicability across different linguistic and cultural settings.

3.3. Zero-Shot Sign Language Recognition

Unlike traditional SLR systems that rely on a large amount of training data to make predictions, ZSSLR attempts to define unseen classes during the training phase. The overall ZSSLR approach is illustrated in Figure 3.5. In this approach, visual data and sign language class descriptions are taken as input, and features are extracted from these inputs using various methods. Extracted features presents two feature spaces: a visual feature space and a semantic feature space. The visual feature space is transformed into the semantic feature space by calculating the compatibility matrix with various ZSL recognition methods found in the literature and performing matrix multiplication with the visual feature matrix. The closest class within the semantic feature space, pre-positioned by semantic feature extraction, is predicted as our class. Thus, the system can identify classes that were not seen during the training phase.

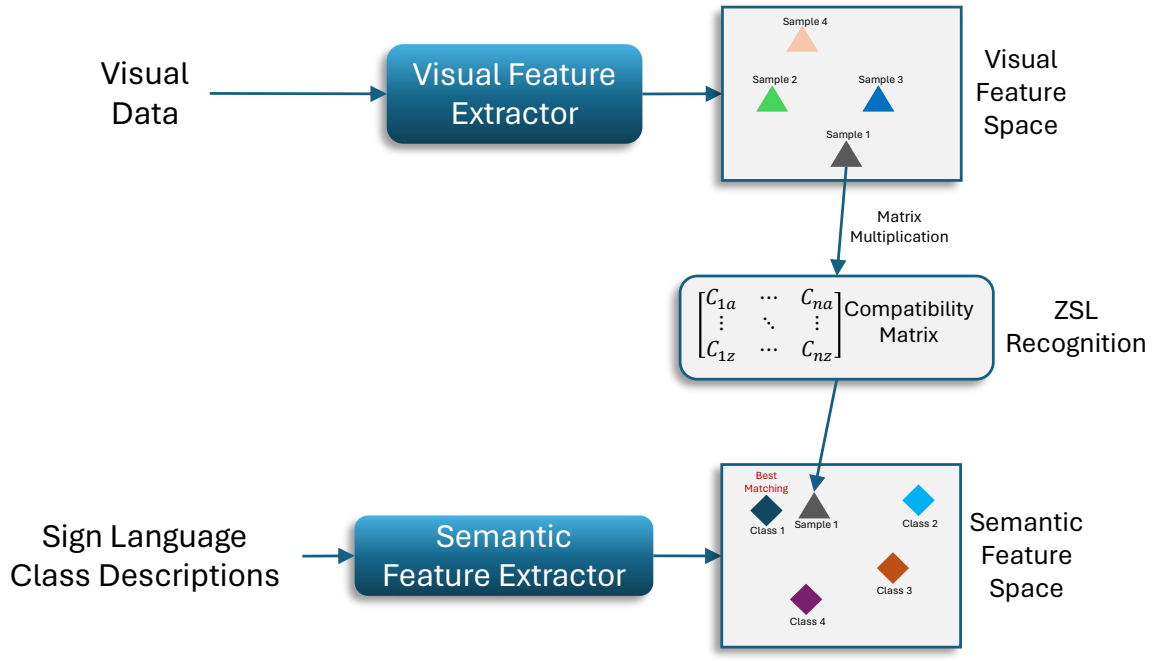


Figure 3.5: General ZSSLR approach

3.4. Difference Between ZSL and GZSL Settings

The thesis results are analyzed in two different frameworks: ZSL and GZSL, making it essential to comprehend the distinctions between them. ZSL focuses on classifying items from classes not present during the training period, utilizing supplementary data such as attributes or textual descriptions to extrapolate knowledge about these unseen classes. For example, a model trained with images of cats, dogs, and birds, but not horses, can still

recognize a horse given sufficient semantic details about horses. On the other hand, GZSL enhances ZSL's capabilities by identifying both unseen and seen classes in the same test set. Using the previous example, a GZSL model would correctly classify images of known classes like cats, dogs, and birds, and also categorize an unseen class like horses, thus distinguishing between all classes regardless of their exposure during training. Essentially, ZSL is designed to identify classes unseen during training, whereas GZSL is developed to recognize both seen and unseen classes using the same training data and semantic input.

4. METHODS

In this section, the ZSSLR methods, for which better results were achieved by developing visual and textual representations, will be explained. As seen in Figure 4.1, the applied ZSSLR method encompasses three distinct stages. Initially, the visual embedding stage involves obtaining the visual representation through various techniques. Following this, the auxiliary embedding stage is concerned with acquiring the textual representation using a specific architecture. The final stage, known as the ZSL recognition method, employs transfer learning. Once both the visual and auxiliary embeddings are obtained, classification is conducted utilizing the ZSL recognition method. Details of the methods performed in these stages can be found in the following paragraphs.

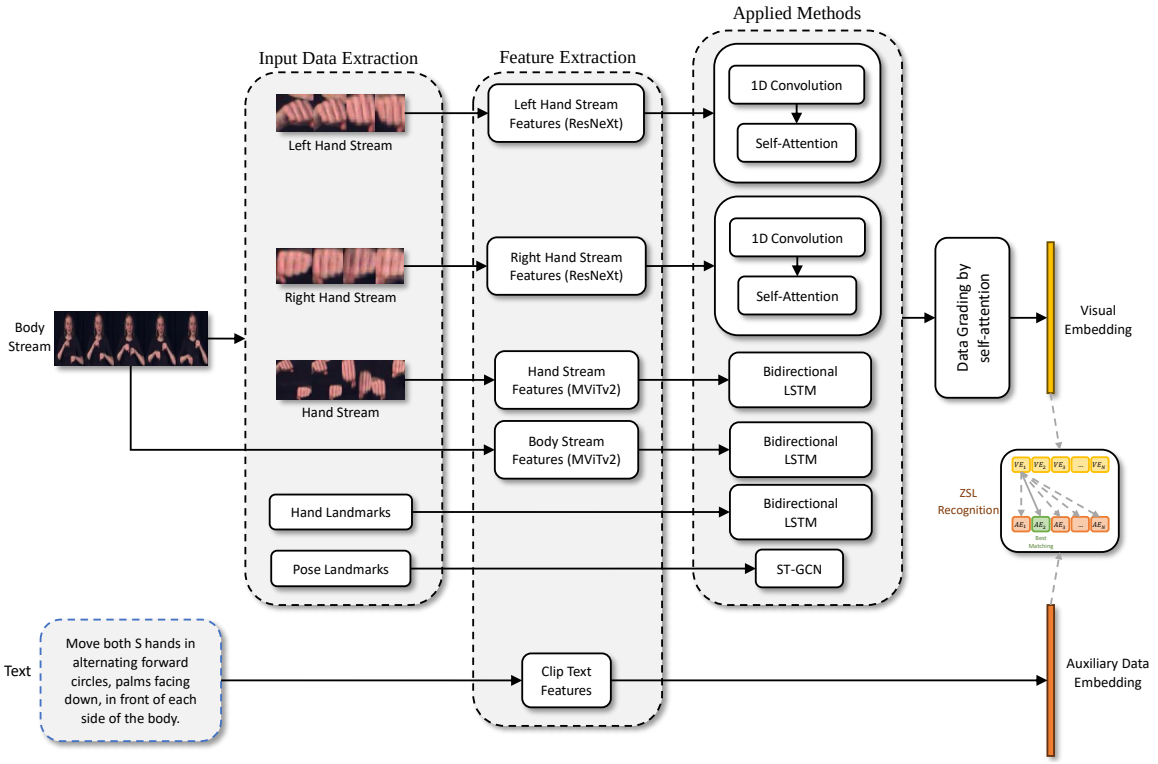


Figure 4.1: General workflow

4.1. Hand and Pose Based Feature Representation

The aim here is to achieve a feature representation that will ensure the distinction between classes. For this purpose, a method has been developed that uses body stream, individual streams for the right and left hands, stream that includes both hands, as well as pose and hand landmarks. This thorough approach ensures a detailed understanding of the data, leading to better feature representation.

Due to its *state-of-the-art* (SOTA) success in recent benchmarks, MviTv2 [21] has been used for feature extraction from body and combined hand streams. It is believed that

using such an architecture for feature extraction will improve performance. After extracting these features, Bi-LSTM is applied to capture temporal dependencies. For the separate right and left hand streams, the ResNeXt [20] architecture, pre-trained on the HaGRID[1] dataset, is applied to extract different features of individual single hands. The ResNeXt architecture trained on the HaGRID dataset is chosen because it was pre-trained on hand images and can model hand positions well. Features are extracted from each frame of the video and stacked. To extract the features that best match the overall model derived from these stacked images, 1D-CNN and self-attention mechanisms are applied.

MediaPipe [79] was used for the extraction of right, left and both hand streams and landmark data. It is an advanced open-source framework designed for the creation of multimodal applied machine learning pipelines, encompassing audio, video, and sensor data processing. It is particularly renowned for its effectiveness in media processing and computer vision applications. The framework is distinguished by its modular and graph-based architecture, enabling the efficient flow and transformation of data through various processing nodes. This architectural design ensures a high degree of customizability and scalability, catering to a wide spectrum of applications. Moreover, it offers extensive cross-platform support, including desktop, mobile, and edge devices, facilitated by its platform-agnostic APIs and optimized utilization of hardware acceleration, such as GPU processing. Additionally, it is equipped with a variety of pre-built solutions and machine learning models for standard tasks like face detection, hand tracking, and pose estimation, which are optimized for performance and ease of integration. For extraction processes, MediaPipe's hand and pose landmarks extraction capabilities were used, as shown in Figure 4.2 and Figure 4.3, and a modified extraction process for stream and landmark data.

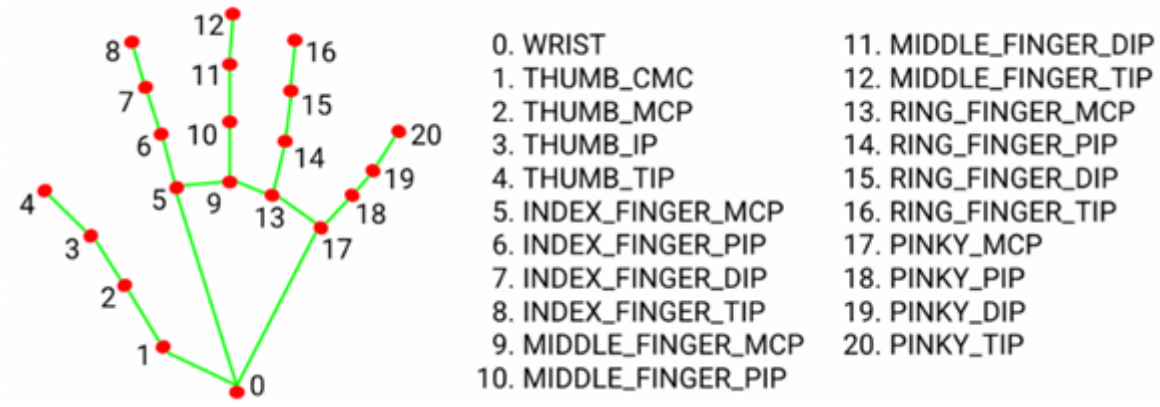


Figure 4.2: Extracted hand landmarks and their indices

For single hand streams such as the right or left hand, a bounding box was determined for each frame of the video by taking the largest and smallest values of the landmarks on the x or y axes. The same algorithm was also used when extracting streams involving two hands. For pose landmarks, those corresponding to the shoulder, arm, and hand landmarks seen in

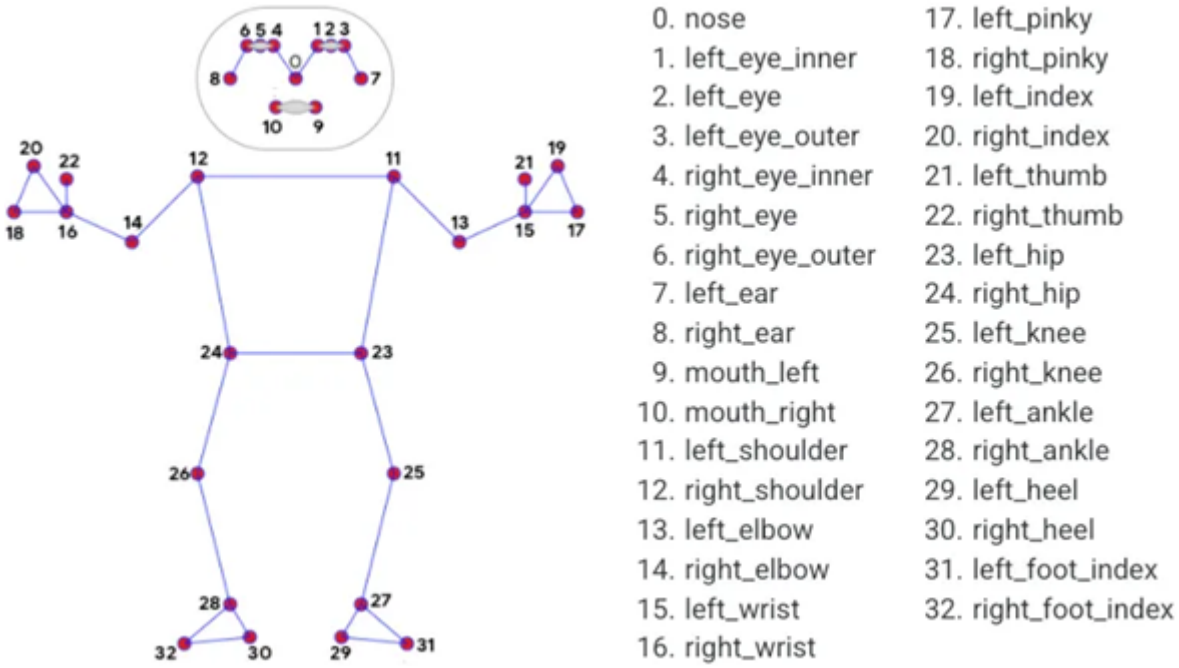


Figure 4.3: Extracted pose landmarks and their indices

Figure 4.3, specifically numbers 11-22, were used. For hand landmarks, all landmarks of both hands seen in Figure 4.2 were combined.

HaGRID [1], is a specialized dataset designed explicitly for the task of hand gesture recognition. It comprises a large collection of annotated images that capture a diverse range of hand gestures, making it a valuable resource for training and evaluating hand gesture recognition models. The dataset is characterized by its variety both in terms of the gestures it includes and the conditions under which the images were captured. This includes variations in background, lighting conditions, and hand positions, as well as differences in hand shapes and sizes among individuals. Such diversity ensures that models trained on HaGRID can generalize well across a wide range of real-world scenarios. The dataset includes annotations that not only label the type of gesture but also provide bounding box information for the location of the hands in each image. This level of detail is crucial for training accurate and efficient hand gesture recognition models, particularly in applications where precise hand position and movement are critical.

In implementing the ResNeXt [20] architecture for processing the HaGRID [1] dataset, specific adaptations are typically made to cater to the nuances of hand gesture recognition. ResNeXt, known for its modularized architecture that extends the ResNet [80] model by integrating parallel convolutional pathways within its layers, offers enhanced feature extraction capabilities. This architecture, characterized by its 'cardinality' or the number of parallel paths, allows for more complex and varied feature representations, which is beneficial for the intricate task of hand gesture recognition. By leveraging these parallel pathways, the

Table 4.1: F1-scores of the evaluated models on HaGRID [1]

Model	F1-score
MobileNetV3_small	86.4
MobileNetV3_large	91.9
VitB16	91.1
ResNet18	97.5
ResNet152	95.5
ResNeXt50	**98.3**
ResNeXt101	97.5

ResNeXt model can effectively capture the subtleties and variations in hand gestures presented in the HaGRID dataset. Each pathway can learn different aspects of the gestures, such as the orientation of fingers, the shape of the hand, and the contextual information from the surrounding environment. This leads to a more robust and nuanced understanding of hand gestures, enabling the development of highly accurate recognition systems. When applied to HaGRID, a ResNeXt-based model can exploit its deep and complex architecture to effectively handle the variability and complexity inherent in human hand gestures, making it an ideal choice for advanced gesture recognition tasks. The models tested on the HaGRID dataset by Alexander et al.[1] are shown in Table 4.1.The reason for choosing the ResNeXt50 model among these models is that it has the highest F1-score. The 18 classes in the HaGRID dataset can be seen in Figure 4.4.

Figure 4.4: HaGRID [1] classes

MViTv2 [21] is an advanced neural network architecture that builds upon its predecessor, MViT (Multiscale Vision Transformers) [81], which was designed for visual recognition tasks. This architecture is specifically engineered to address the computational inefficiencies and scalability limitations of the original MViT model. MViTv2 introduces key modifications such as a more efficient tokenization process, refined multiscale feature extraction, and enhanced attention mechanisms. The fundamental principle of MViTv2 lies in processing visual inputs at multiple scales, enabling the model to capture both fine-grained details and global contextual information effectively. This multiscale strategy is vital for handling complex visual challenges such as object detection, image classification, and segmentation.

In terms of operational mechanics, MViTv2 [21] starts by partitioning the input image into a sequence of non-overlapping patches. These patches are then linearly embedded into tokens. Unlike its predecessor, MViTv2 employs a hierarchical structure where the number of tokens is progressively reduced while increasing their feature dimensions at deeper layers of the network. This hierarchical tokenization allows for efficient computation and reduces the memory footprint, addressing a significant challenge in the original MViT design. The core of MViTv2 is its enhanced transformer block, which incorporates an improved attention mechanism. This mechanism dynamically adjusts the attention span at different layers, allowing the model to focus on more relevant features at various scales. The network also integrates a novel pooling strategy to fuse multiscale features effectively. This combination of hierarchical tokenization, dynamic attention, and multiscale pooling results in a powerful and computationally efficient architecture, making MViTv2 particularly adept at handling a wide range of visual recognition tasks with high accuracy and speed.

Pose landmarks have been processed with ST-GCN [10] to better extract spatial and temporal relationships. Hand landmarks, on the other hand, have been further processed with Bi-LSTM to improve the model's understanding of temporal information. ST-GCN uses skeleton sequences to create a spatio-temporal graph and applies graph convolutional operations on this graph to capture dynamic human movements. This method teaches the model the temporal dependencies of hand and pose joints more effectively and better determines their positional changes.

ST-GCN [10] is a pioneering approach in the domain of graph-based neural networks, specifically tailored for analyzing spatial-temporal data, such as human actions captured through skeletal landmarks. The fundamental architecture of ST-GCN is designed to process data that is inherently graph-structured and evolves over time. As can be seen on Figure 4.5, blue dots represent the body joints, the edges within the body, connecting these joints, are determined by the inherent linkages in the human anatomy. Additionally, edges between frames link identical joints across successive frames. This graph structure encapsulates the spatial configuration of the body joints, while the sequence of movements over time forms the temporal dimension. The innovation of ST-GCN lies in its ability to simultaneously process and learn from both these spatial and temporal dimensions, which is crucial for accurately recognizing and interpreting human actions.

The operation of ST-GCN [10] on landmark data involves a series of spatial and temporal graph convolutional layers. In the spatial domain, ST-GCN applies graph convolution to the skeletal graph at each time frame. This convolutional process involves aggregating features from neighboring nodes based on the graph structure, allowing the network to learn spatial features representative of the body's pose and configuration at each moment. In ad-

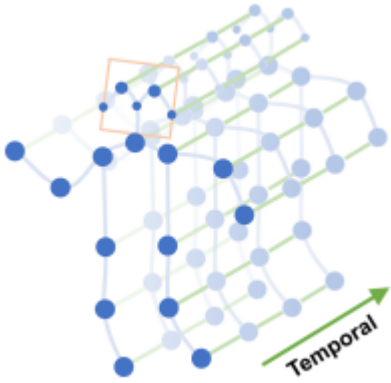


Figure 4.5: ST-GCN [10] nodes and joints

dition to spatial convolutions, the model also includes temporal convolutions that aggregate features across consecutive time frames. This temporal aspect enables ST-GCN to capture the dynamics of human motion over time. The interplay between spatial and temporal convolutions allows ST-GCN to construct a comprehensive feature representation of human actions, accounting for both the position and connectivity of joints at each moment and their evolution throughout the action sequence. This dual convolution approach makes ST-GCN particularly effective for tasks like action recognition, gait analysis, and any other applications where understanding the complex interplay between spatial configurations and temporal dynamics of human movements is critical. The working logic of ST-GCN on landmark data can be seen on Figure 4.6. Several layers of ST-GCN will be utilized, progressively creating more advanced feature maps on the graph.

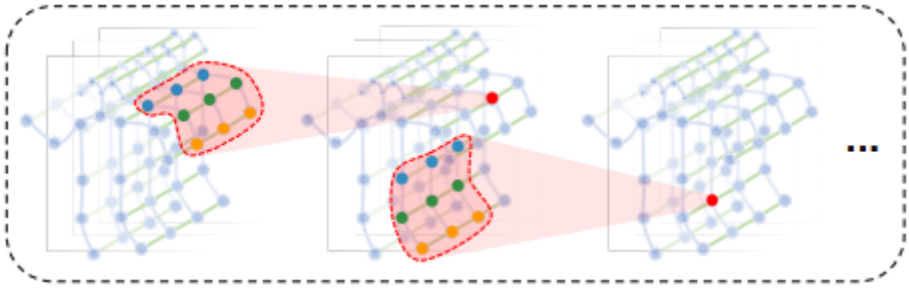


Figure 4.6: ST-GCN [10] working logic

4.2. Hand and Pose Based Feature Selection

The purpose of the applied hand and pose-based feature selection is to enable the model to select the best hand and pose features that can improve the overall model performance. To process the ResNeXt [20] features, combination of 1D-CNN and self-attention are utilized. Given S is the size of the kernel and c_{in} is the input channel id, 1D-CNN output Y at the output channel j for a given position t, $y_j[t]$ can be calculated as:

$$y_j[t] = g(b_j + \sum_{i=0}^{c_{in}-1} \sum_{h=0}^{S-1} W_{j,i,m} \cdot E_i[t+h]) \tag{4.1}$$

where b_j is the bias term for output channel j, g stands for the activation function, $W_{j,i,m}$ is the weights of the kernel at output channel j, input channel i and kernel position m. $E_i[t+h]$ is the ResNeXt feature matrix at channel i and position $t+h$.

Self-attention query Q, key K and value V from 1D-CNN output Y are calculated as:

$$Q = YW_Q, K = YW_K, V = YW_V \tag{4.2}$$

where W_Q, W_K and W_V is the learnable projection matrices for query, key and value, respectively. The self-attention output O are calculated as follows:

$$O = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{4.3}$$

where K^T is the transpose of K weight matrice and d_K is the dimension of key vectors. The $softmax$ function is applied to convert scores into probabilities, ensuring that the sum of the probabilities equals 1. O_r and O_l output matrices were extracted from E_r right hand, and E_l left hand ResNeXt [20] features, respectively.

Bi-LSTM method enables the network to capture context from both past and future states, enhancing its capability to learn spatial and temporal sequence dependencies. It is applied to MViTv2 [21] body and hand features, resulting in B_b and B_h, respectively, which were extracted with stride 4 and stacked. Moreover, Bi-LSTM is utilized to extract essential information from hand landmarks, producing B_{Hl}.

ST-GCN [10] is employed to capture spatial and temporal dependencies from landmark data. Initially, the graph $G = (V, E)$ were constructed, where V represents a set of vertices and E represents a set of edges. This process is repeated for each time frame to incorporate temporal data. In ST-GCN, three methods are employed to determine the order of vertex multiplications, involving subsets of vertices known as layers. For calculating the spatial-temporal graph convolution $\mu^{(l+1)}$ at label $(l+1)$:

$$\mu^{(l+1)} = \sum_{n=1}^{N} \Theta_n^{(l)} \left(D_n^{-\frac{1}{2}} A_n D_n^{-\frac{1}{2}}\right) \mu^{(l)} \tag{4.4}$$

Here, A_n is the adjacency matrix between vertices representing different types of relationships in layer l, $\mu^{(l)}$ denotes the feature matrix at layer l, D_n is the diagonal degree matrix corresponding to A_n, $\Theta_n^{(l)}$ represents the weights at layer l for the n-th type of connection, and N is the total number of subgraphs considered.

4.3. Data Grading Methodology

Data grading has been employed due to the extraction of features using various methods from a wide array of data types, including landmark and video data. This process allows the model to identify and select the most effective features for recognition task. Similar to Equation 4.3, self-attention were applied with a slight modification. Initially, the features derived from different methods are concatenated to form Z:

$$Z = Join(O_r, O_l, B_b, B_h, B_{Hl}, \mu) \tag{4.5}$$

where O_r and O_l represent the 1D-CNN and self-attention output from the right and left hand ResNeXt [20] features, respectively. B_b and B_h denote the Bi-LSTM output from the body and hand MViTv2 [21] features, while B_{Hl} and μ signify the Bi-LSTM and ST-GCN [10] output from the hand and pose landmarks, respectively. Therefore, $Z \in \mathbb{R}^{p \times d}$, where p indicates the number of methods and d represents the number of features.

Learnable weight matrices W_ϕ, W_κ, and W_ψ are incorporated to transform input features into query ϕ, key κ, and value ψ components, which are essential for self-attention. These matrices are designed to dynamically capture and model the relationships between different elements within the data. The query matrix W_ϕ identifies the elements to focus on, the key matrix W_κ determines the compatibility or relevance of the elements, and the value matrix W_ψ adjusts the output based on the identified relationships. By learning these matrices during the training process, the model can adaptively weigh the importance of various data features, allowing for a more nuanced and accurate representation of complex data types. This dynamic weighting is particularly beneficial in handling heterogeneous data sources, as it enables the model to prioritize relevant information, improving the overall performance of feature extraction and subsequent tasks in ZSSLR. These matrices project d-dimensional feature vectors into d'-dimensional vectors as specified in Equation 4.2, substituting Y with the concatenated feature matrix Z, such that $\phi = ZW_\phi$, $\kappa = ZW_\kappa$, and $\psi = ZW_\psi$. Here, ϕ, κ, and ψ belong to $\mathbb{R}^{p \times d'}$. The attention weights are computed by normalizing the dot products of ϕ and κ using the $softmax$ function, which highlights the significance of each feature:

$$A = softmax\left(\frac{\phi\kappa^T}{\sqrt{d'}}\right) \tag{4.6}$$

where $A \in \mathbb{R}^{p \times p}$ represents the attention matrix, with each element a_{ij} indicating the influence of feature j on feature i. Subsequently, the value matrices are combined with the attention weights:

$$\nu = A\psi \tag{4.7}$$

where $\nu \in \mathbb{R}^{p \times d'}$ is the matrix of weighted features post-attention, forming visual embedding matrix. This encapsulates thesis approach for visual embedding learning.

4.4. Sign Description Modelling

The language embedding vector matrix for all classes, which stand for as L, is derived from the written descriptions of visual signs using the advanced language model CLIP [22], as illustrated in Figure 4.7. CLIP is a powerful model designed for ZSL, utilizing natural language to associate with visual concepts it has either learned in the past or can describe new ones. This functionality enables CLIP to be applied to a wide range of tasks without needing specific prior training, representing a notable improvement over traditional models such as word2vec [82], GloVe [83], and BERT [49]. Unlike these earlier models, CLIP's key strengths are its context-aware embeddings and versatility. Moreover, it is important to note that the training of these embeddings, alongside the compatibility function, is executed in a seamless end-to-end process.

CLIP [22], developed by OpenAI, is an advanced neural network architecture that marks a significant progression in zero-shot learning technologies. It is engineered to integrate and understand both textual and visual data seamlessly. The architecture includes two main components: a text encoder and an image encoder. The text encoder, usually based on Transformer technology, transforms input text into vectors within a high-dimensional space. The image encoder, which may be either a Vision Transformer (ViT) [84] or a ResNet-based [80] model, processes images to map them into the same vector space. The fundamental concept of CLIP involves training these encoders together using a contrastive learning approach. This method trains the model to link images with relevant textual descriptions effectively. The training process uses a large, varied dataset of image-text pairs, teaching the model to increase similarity between vectors of matching pairs and decrease similarity for non-matching pairs.

The capability of feature extraction in CLIP [22] significantly enhances the accuracy of zero-shot learning models. CLIP achieves this by leveraging its generalized representations of images and text. Since both encoders map inputs into a shared embedding space, CLIP can compare and relate any given text to any image, even if the specific content was not part of its training data. For instance, when presented with a new category described in text, CLIP can effectively identify and relate this description to relevant images, despite never having seen labeled examples of this category. This flexible and robust feature extraction enables CLIP to perform remarkably well in zero-shot scenarios, where traditional models would struggle without explicit training data. This approach also allows for a wide range of applications, from content-based image retrieval to novel object recognition, making CLIP a versatile tool in bridging the gap between visual and linguistic understanding.

The respective encoders generate the feature embeddings for both the image and the possible texts. These embeddings' cosine similarity is computed, adjusted by a temperature

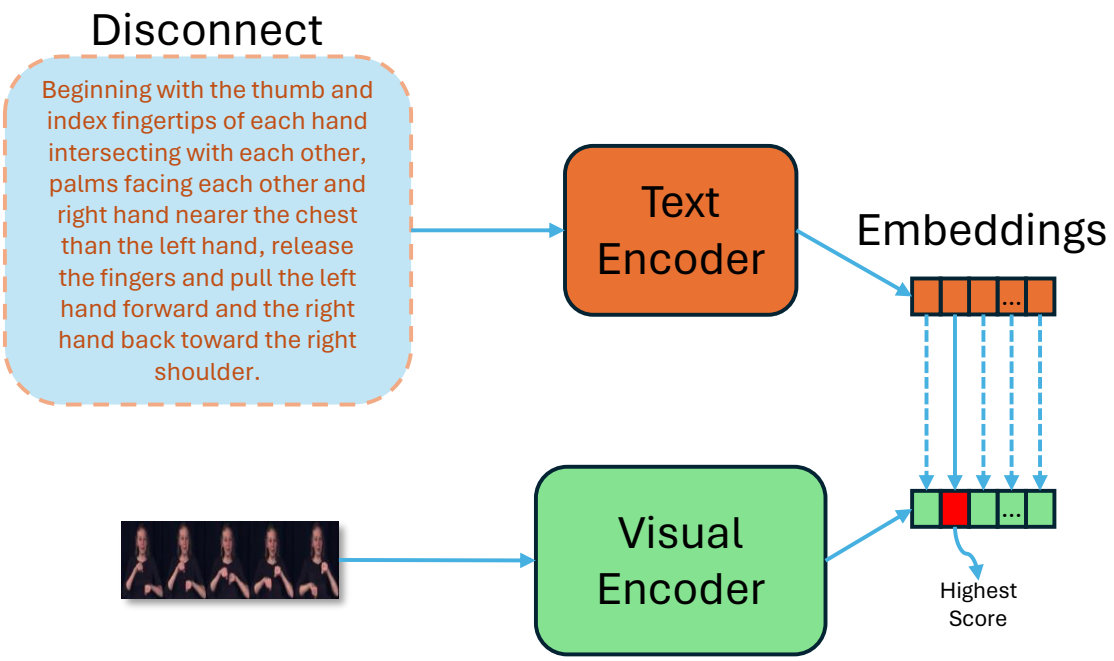


Figure 4.7: CLIP architecture workflow

parameter, and converted into a probability distribution using a softmax function. This setup functions as a multinomial logistic regression classifier that employs L2-normalized inputs and weights, without a bias, and incorporates temperature scaling. This method is known as the contrastive approach.

4.5. Zero-shot Sign Recognition

Label embedding-based approach [85], [86] was adopted as a component for ZSL, as depicted in Figure 4.1. Specifically, LLE [3] focuses on determining a compatibility matrix that adjusts video feature embeddings with semantic class embeddings, such as textual descriptions. This method employs a logistic model to predict the likelihood of a video belonging to a particular class, utilizing a cross-entropy loss function for optimization. Regularization is applied to prevent overfitting, thereby promoting the acquisition of generalized representations that bridge the inconsistency between observed and unobserved sign classes.

First, the compatibility score s was computed between the visual embedding matrix ν and the learned language embedding of label l:

$$s(\nu, l) = \alpha(W\nu + \rho)^T y_l \tag{4.8}$$

where W represents the learnable compatibility matrix, α represents the non-linear activation function, ρ is the bias term, and y_l is the label embedding at layer l. Next, the probability of

each label was calculated by:

$$P(l|\nu) = \frac{\exp(s(\nu, l))}{\sum_{l' \in L} \exp(s(\nu, l'))} \tag{4.9}$$

where L comprises both seen and unseen labels. The sample is then classified into the class that has the highest probability.

5. EXPERIMENTS

5.1. Implementation Details

In the latest experiment, the proposed architecture, which is the applied method, is shown in Figure 4.1. In this section, the details of the experiments conducted will be provided.

First, input data were extracted from the MS-ZSSLR-W/C [3] and ASL-Text [2] datasets. Using MediaPipe [79], separate streams were extracted from the body stream, including left and right hands, a combined stream of both hands, and pose and hand landmark data. Features were extracted and stacked for each frame from the different streams of the left and right hands using the ResNeXt [20] architecture pre-trained on the HaGRID [1] dataset, with the frame count kept at 32 and a stride of 4. Similarly, features were extracted from the body and combined hand streams using the MViTv2 [21] architecture. Pre-trained ResNeXt50 model were employed, which achieved a 98.3 F1 score on the HaGRID dataset [1]. Additionally, MViTv2-S pre-trained model were used, sized at $16x4$, which attained an 81.0% accuracy on the Kinetics-400 dataset [87].

To optimally select the features that could best impact overall model performance, 1D-CNN and self-attention were applied to the ResNeXt [20] features. Bi-LSTM was applied to the body and combined hand streams to extract spatial and temporal features and ensure better model alignment. Additionally, while Bi-LSTM was applied to the hand landmarks, ST-GCN [10] was applied to the pose landmarks. Only shoulder and hand landmarks were taken from MediaPipe for the pose landmarks. The adjacency matrices for the landmark nodes required for ST-GCN were manually extracted from the MediaPipe documentation and integrated into the model. Finally, due to the large amount of data and methods used, data grading was performed with self-attention.

To extract textual features, CLIP [22] architecture was used, which focuses on sentence-based features with a length of 768. Given the CLIP implementation's restriction of 77 words per class without fine-tuning, maximum number of words in sign class descriptions were limited to 77. CLIP-ViT-L/14-336px model were utilized, known for its superior performance across various datasets [22]. Due to the variations in the results obtained from ZSL, five experiments were conducted with the same parameters, and the average was taken.

GZSL experiments were conducted on all datasets using the architecture shown in Figure 4.1. For this purpose, the datasets were partitioned separately, which will be detailed in the dataset section. Additionally, ZSL experiments were conducted using the binary attributes [3] of the datasets. In these experiments, the CLIP [22] architecture's contrastive

approach was tested, but promising results were not obtained.

The model's performance was assessed using the top-n accuracy metric, which considers a prediction correct if the actual label is within the top-n predictions sorted by confidence. Specifically, Top-1 accuracy were used to determine if the model's highest-confidence prediction matches the correct answer exactly. Additionally, Top-2 and Top-5 accuracy metrics were utilized to check if the true label is within the model's top two or five predictions, respectively. This approach is necessary due to the large number of classes, totaling 50, in the test split of our datasets. To further evaluate performance, the Area Under the Curve (AUC) metric was utilized to differentiate between positive and negative classes. For this multi-class classification task, One-vs-Rest (OvR) approach was adopted for calculating the AUC score. In this method, the class of interest is treated as the positive class while all other classes are considered negative. Average AUC score was reported across all classes. Additionally, to provide a clearer understanding of our method's effectiveness, changes in classifier loss and performance over epochs were graphically depicted.

5.2. Datasets

In this study, three datasets were used: ASL-Text [2], MS-ZSSLR-W, and MS-ZSSLR-C [3]. The ASL-Text dataset was derived from the ASLLVD [88] dataset. From ASLLVD, 250 classes with the highest number of signers and samples were selected. These classes were split into training, validation, and test sets, and the necessary textual descriptions for Zero-Shot Learning (ZSL) were extracted for each class. In Figure 5.1, visual examples of the ASL-Text dataset and class definitions can be seen. The class definitions include expressions like "S Hand" and "O Hand," which are terms found in the American Sign Language Hand Shape [89] Dictionary.

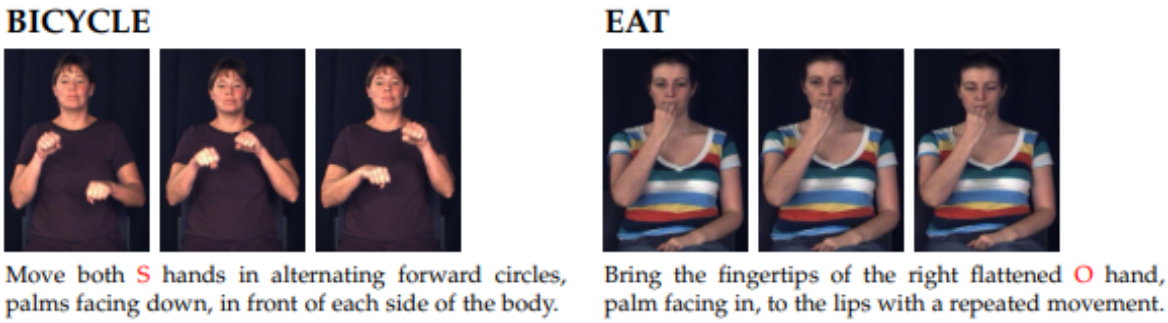


Figure 5.1: Examples of ASL-Text [2] dataset visual samples and class descriptions

The MS-ZSSLR-W/C datasets originated from the MS-ASL [90] dataset. Similarly to the ASL-Text dataset creation, 200 classes with the most signers and examples were chosen from the MS-ASL dataset. Textual descriptions for each class were extracted while dividing the dataset into training, validation, and test sets. The MS-ZSSLR-W dataset includes linguistic variations, making it more effective for modeling real-life scenarios. In contrast, the

MS-ZSSLR-C dataset lacks these variations, providing a benchmark in a more controlled and pristine environment. As can be seen in Figure 5.2, the visual examples in MS-ZSSLR-W/C were taken in an uncontrolled environment.

Figure 5.2: Examples of MS-ZSSLR-W/C [3] visual samples and class descriptions

The ASL-Text dataset contains 250 classes for ZSL experiments, which are divided into 170, 30, and 50 classes for training, validation, and testing, respectively. The MS-ZSSLR-W/C datasets contain 200 classes for ZSL experiments, divided into 120, 30, and 50 classes for training, validation, and testing, respectively. For GZSL experiments, the validation and test splits from the ZSL experiments were retained, with additional classes added. For the ASL-Text dataset, 170, 200, and 220 classes were used for training, validation, and testing, respectively, in GZSL experiments. For the MS-ZSSLR-W/C datasets, 120, 150, and 170 classes were used for training, validation, and testing, respectively, in GZSL experiments. In the MS-ZSSLR-W/C experiments conducted using binary attributes, the binary attributes were extracted from the American Sign Language Hand Shape [89] Dictionary.

In ZSL setting the ASL-Text dataset is composed of 1188 videos for training, 151 for validation, and 259 for testing. In contrast, the MS-ZSSLR-W/C datasets are substantially larger. The MS-ZSSLR-W dataset consists of 5862 videos for training, 1153 for validation, and 1846 for testing. Similarly, the MS-ZSSLR-C dataset includes 7303 training videos, 1368 validation videos, and 1779 test videos. Overall, the ASL-Text dataset comprises 1598 samples, the MS-ZSSLR-W dataset contains 8861 samples, and the MS-ZSSLR-C dataset includes 10450 samples. The training set for all these datasets was divided in the ratio of 6:2:2 for the train, validation, and test sets in the GZSL setting.

Unlike regular learning datasets, in datasets created for ZSL, the validation and test classes are different. While creating these datasets, classes containing similar actions were chosen for the selection of validation and test classes.

Providing text descriptions during the testing phase might be seen as a limitation for real-world applications. However, these descriptions are extracted beforehand, so they can

be considered somewhat like class labels, even if not exactly so. Moreover, extracting descriptions is easier than labeling dozens of videos because they are extracted for classes, not for individual samples. Although providing text descriptions during the testing phase is indeed a limitation, it can be considered a smaller restriction compared to labeling dozens of videos.

5.3. Results

Various experiments were conducted using different data sources to assess the effectiveness of the proposed model for ZSL setting. These sources are detailed in Table 5.1. In the first experiment, only MViTv2 [21] hand features were utilized. The second experiment focused solely on MViTv2 body features. The third experiment combined MViTv2 hand and body features. The fourth experiment incorporated MViTv2 features along with pose and hand landmark data. In the fifth experiment, the landmark data were excluded, and ResNeXt [20] features for the right and left hands were added. The sixth experiment reintroduced pose landmark data in addition to the features used in the fifth experiment. Finally, the seventh experiment employed all available data sources.

Table 5.1: Landmarks and features used in experiments

Experiment #	MViTv2 Body Stream Features	MViTv2 Hand Stream Features	Pose Landmarks	Hand Landmarks	Right Hand ResNeXt Features	Left Hand ResNeXt Features
#1		✓				
#2	✓					
#3	✓	✓				
#4	✓	✓	✓	✓		
#5	✓				✓	✓
#6	✓		✓		✓	✓
#7	✓	✓	✓	✓	✓	✓

The results from the ASL-Text dataset in the ZSL setting are shown in Table 5.2. In Experiment#1, where only MViTv2 hand features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 17.62, 26.78, and 54.67, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 22.46, 30.77, and 47.18, respectively with an average AUC score of 0.68. The higher test results compared to the validation results suggest that the discrepancy may be due to the test set results being randomly assigned. This outcome could indicate that the model's performance on the test set is not necessarily reflective of its true ability to generalize to unseen classes but rather a consequence of the specific data points included in the test set.

In Experiment#2, where only MViTv2 body features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 23.87, 31.2, and 50.84, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 18.12, 26.8, and 45.33, respectively with an average AUC score of 0.68. In this experiment, the test results did not exceed the validation results, indicating that the anomaly observed in Experiment#1, where the test results were

Table 5.2: Results obtained from ASL-Text dataset

Experiment #	Val (30 classes)			Test (50 classes)			Average AUC Score
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	
#1	17.62	26.78	54.67	22.46	30.77	47.18	0.68
#2	23.87	31.2	50.84	18.12	26.8	45.33	0.68
#3	23.58	34.96	**60.91**	21.89	30.53	**50.71**	0.74
#4	**26.09**	**35.51**	57.95	21.22	30.38	45.48	0.72
#5	21.98	32.36	56.49	17.89	27.55	43.11	0.73
#6	23.98	34.18	56.11	18.52	29.56	48.15	0.71
#7	24.31	34.8	59.64	**22.76**	**32.83**	50.15	0.71

unexpectedly high, is not present here. Therefore, the previous concerns about the random assignment of test results do not apply to this experiment. However, despite the more consistent performance between the validation and test sets, it is important to note that the test results still did not surpass the performance of the current SOTA methods. This suggests that while the model's generalization capability may be more reliable in this experiment, there is still room for improvement in achieving or exceeding the performance levels set by existing approaches.

In Experiment#3, where MViTv2 hand and body features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 23.58, 34.96, and 60.91, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 21.89, 30.53, and 50.71, respectively with an average AUC score of 0.74. The incorporation of hand features alongside body features produced validation results that were comparable to those observed in Experiment#2, indicating a consistent performance during the validation phase. However, the inclusion of hand features significantly enhanced the model's performance on the test set, leading to better outcomes than those achieved in the previous experiment. This improvement in the test set results suggests that the combined use of hand and body features enables the model to capture more nuanced and discriminative information, which is particularly effective in handling unseen data during testing.

In Experiment#4, which included hand and body features along with hand and pose landmarks, the results on the validation set for Top-1, Top-2, and Top-5 were 26.09, 35.51, and 57.95, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 21.22, 30.38, and 45.48, respectively with an average AUC score of 0.72. The inclusion of landmark data resulted in the highest success rates observed in the validation set across all experiments. This indicates that the model was able to leverage the additional spatial information provided by the landmarks to improve its accuracy during validation. However, despite this notable improvement in the validation phase, the addition of landmark data did not lead to a significant change in the test set results. This suggests that while landmark data can enhance

performance during model tuning and validation, its impact may be limited when applied to unseen test data. The lack of substantial improvement in the test set indicates that other factors, such as overfitting to the validation set or insufficient diversity in the training data, may be influencing the model's ability to generalize effectively.

In Experiment#5, where only MViTv2 body features and ResNeXt hand features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 21.98, 32.36, and 56.49, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 17.89, 27.55, and 43.11, respectively with an average AUC score of 0.73. The model was enhanced by incorporating the ResNeXt architecture in addition to the MViTv2 body features that were exclusively used in Experiment-2. Despite this combination of ResNeXt with MViTv2, the results on the test set remained similar to those obtained in Experiment#2, where only MViTv2 body features were employed. This outcome suggests that the addition of ResNeXt did not lead to a significant improvement in the model's ability to generalize to unseen data, at least in the context of this specific experiment. While the ResNeXt architecture is known for its robustness and ability to capture complex patterns, its integration with MViTv2 body features did not translate into a noticeable enhancement in test performance. This indicates that, for this task, the features extracted by MViTv2 may already be capturing most of the critical information needed for recognition, and the ResNeXt architecture did not substantially augment this capability.

In Experiment#6, where pose landmarks were added to the resources used in Experiment#5, the results on the validation set for Top-1, Top-2, and Top-5 were 23.98, 34.18, and 59.64, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 18.52, 29.56, and 48.15, respectively with an average AUC score of 0.71. The model was evaluated using a combination of ResNeXt features, MViTv2 body features, and pose landmarks. However, the results obtained were similar to those from Experiment#2, where only MViTv2 body features were utilized. This similarity in outcomes suggests that the additional ResNeXt features and pose landmarks did not contribute significantly to improving the model's performance on the test set. The lack of substantial improvement indicates that the information captured by ResNeXt and pose landmarks may be redundant or less relevant in the context of this particular task. Additionally, due to the small size of the dataset, the spatial and temporal dependencies captured by the ResNeXt and pose landmark data are minimal.

Experiment-7, which integrated all available data and features, emerged as the most successful approach in the series of experiments, achieving the highest performance on the test dataset. The model's comprehensive use of diverse data sources and feature types allowed it to capture a wide range of discriminative information, leading to superior results. For the validation dataset, it achieved Top-1, Top-2, and Top-5 accuracies of 24.31, 34.8, and

59.64, respectively. On the test dataset, the corresponding accuracies were 22.76, 32.83, and 50.15. Additionally, the average AUC score for the test dataset was 0.71.

The results on the MS-ZSSLR-W dataset are shown in Table 5.3. In Experiment#1, where only MViTv2 hand features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 21.01, 32.31, and 58.09, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 10.71, 18.4, and 35.37, respectively with an average AUC score of 0.71. This outcome suggests that relying solely on hand features may not provide sufficient discriminative power for the task of ZSSLR. The limited performance indicates that while hand movements are crucial for understanding sign language, they may not capture the full range of necessary information on their own, leading to suboptimal results. The findings from this experiment highlight the importance of considering additional features or data modalities to improve the model's accuracy and robustness, as the hand features alone do not seem to offer a comprehensive solution for effective ZSSLR on the MS-ZSSLR-W dataset.

Table 5.3: Outcomes derived from MS-ZSSLR-W dataset

Experiment #	Val (30 classes)			Test (50 classes)			Average AUC Score
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	
#1	21.01	32.31	58.09	10.71	18.4	35.37	0.71
#2	24.13	34.81	59.02	13.02	21.26	38.21	0.72
#3	25.09	37.44	63.69	14.11	23.74	42.6	0.74
#4	28.15	**42.83**	**67.89**	14.37	23.26	41.3	0.74
#5	26.68	39.03	62.65	14.43	22.56	40.33	0.71
#6	25.39	38.57	65.96	14.37	22.38	39.15	0.72
#7	**28.7**	42.63	67.67	**14.86**	**24.6**	**43.07**	0.74

In Experiment#2, where only MViTv2 body features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 24.13, 34.81, and 59.02, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 13.02, 21.26, and 38.21, respectively with an average AUC score of 0.72. The exclusive use of MViTv2 body features led to a noticeable improvement in results compared to Experiment#1, where only hand features were utilized. This enhancement underscores the critical role that body features play in ZSSLR. Body features capture a wider range of motion and spatial relationships, which appear to be more informative and relevant for distinguishing between different signs in a ZSL framework. The improved performance suggests that the body's overall movement, posture, and dynamics provide a more comprehensive representation of sign language gestures than hand movements alone.

In Experiment#3, where MViTv2 hand and body features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 25.09, 37.44, and 63.69, respectively. On

the test set, the results for Top-1, Top-2, and Top-5 were 14.11, 23.74, and 42.6, respectively with an average AUC score of 0.74. Using hand features alongside body features dramatically improved the test set results. The integration of hand features alongside body features led to a dramatic improvement in test set results. This significant enhancement highlights the complementary nature of hand and body features in ZSSLR. While body features alone capture the general structure and movement of signs, the addition of hand features provides critical fine-grained details that are essential for accurately distinguishing between similar gestures. The synergy between hand and body features allows the model to build a more comprehensive and nuanced understanding of each sign, leading to more precise predictions on the test set. This experiment underscores the importance of multi-modal feature integration in ZSSLR, demonstrating that the combination of detailed hand movements with broader body dynamics can substantially boost the model's generalization capabilities and overall performance.

In Experiment#4, which included hand and body features along with hand and pose landmarks, the results on the validation set for Top-1, Top-2, and Top-5 were 28.15, 42.83, and 67.89, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 14.37, 23.26, and 41.3, respectively with an average AUC score of 0.74. The inclusion of hand and pose landmark data alongside MViTv2 hand and body features resulted in a significant increase in performance on the validation set compared to Experiment#3. This improvement suggests that the landmark data, which provides precise information about key points of the hand and body, enhances the model's ability to accurately interpret and classify signs during the validation phase. Landmarks offer a detailed understanding of the spatial relationships and movement patterns that are crucial for distinguishing between similar gestures. However, despite the notable improvement in the validation set, the test set results remained approximately the same as those observed in Experiment#3. This indicates that while the landmark data helps the model perform better on known data, it does not translate into better generalization to unseen data in the test set. The findings suggest that the model may be benefiting from the landmark information during training and validation, but this advantage does not carry over to the test set, possibly due to overfitting or the limited variability in the test data.

In Experiment#5, where only MViTv2 body features and ResNeXt hand features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 26.68, 39.03, and 62.65, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 14.43, 22.56, and 40.33, respectively with an average AUC score of 0.71. The combination of MViTv2 body features with ResNeXt hand features resulted in improved performance compared to Experiment#2, where only MViTv2 body features were utilized. This improvement highlights the added value that the ResNeXt hand features bring to the model, likely due to

their ability to capture fine-grained details of hand movements, which are crucial for accurate SLR. However, despite this enhancement over Experiment#2, the results were similar to those obtained in Experiments #3 and #4, where hand features and landmark data were also incorporated. This suggests that while the addition of ResNeXt hand features contributes positively to the model's performance, it does not provide a significant advantage over the previously explored combinations of hand and body features.

In Experiment#6, which added pose landmarks to the resources used in Experiment-5, the results on the validation set for Top-1, Top-2, and Top-5 were 25.39, 38.57, and 65.96, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 14.37, 22.38, and 39.15, respectively with an average AUC score of 0.72. Similar results were obtained compared to Experiment#5, indicating that the use of pose features provided similar results. Despite the inclusion of pose landmarks data source, the results were similar to those obtained in Experiment#5, indicating that the pose landmarks did not significantly impact the model's performance. This outcome suggests that while pose landmarks offer detailed information about the body's key points, in this particular setup, they did not contribute additional discriminative power beyond what was already captured by the existing body and hand features. The similarity in results across these experiments implies that the pose landmarks, although potentially valuable, may overlap with the information provided by the MViTv2 body features, leading to diminishing returns in terms of performance gains. Furthermore, this decrease in performance may be due to MediaPipe, used for extracting landmark data, not being able to detect the hand or pose movements of the sign language performer in each frame resulting in an empty array.

Experiment#7, which utilized all available data and features, produced the best outcomes on the test dataset. For the validation dataset, it achieved Top-1, Top-2, and Top-5 accuracies of 28.75, 42.63, and 67.67, respectively. On the test dataset, the performances were 14.86, 24.6, and 43.07 for Top-1, Top-2, and Top-5 accuracies, respectively, with an average AUC score of 0.74. This comprehensive approach enabled the model to capture a wide array of discriminative information, leading to superior performance compared to the previous experiments. The success of this experiment underscores the value of integrating diverse data modalities, as each contributes unique insights that enhance the model's ability to generalize to unseen data.

Table 5.4 illustrates the results on the MS-ZSSLR-C dataset. In Experiment#1, where only MViTv2 hand features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 22.84, 35.68, and 60.77, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 14.27, 23.95, and 43.74, respectively with an average AUC score of 0.73. The lowest performance among the experiments was in Experiment#1. This suggests

that hand features alone do not capture enough of the essential information required for accurate sign language recognition in this particular dataset. The model's limited ability to distinguish between different signs indicates that the nuances of hand movements, when not complemented by additional data such as body features or pose landmarks, are insufficient for achieving strong recognition performance.

Table 5.4: Results on the MS-ZSSLR-C dataset

Experiment #	Val (30 classes)			Test (50 classes)			Average AUC Score
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	
#1	22.84	35.68	60.77	14.27	23.95	43.74	0.73
#2	23.97	36.57	59.98	18.16	27.69	45.48	0.77
#3	26.62	40.43	66.28	17.22	28.77	49.4	0.76
#4	30.64	44.72	70.07	18.38	**30.33**	50.6	0.75
#5	28.14	42.01	67.52	17.88	27.73	46.62	0.73
#6	29.19	43.74	68.18	18.48	28.24	45.96	0.77
#7	**31.73**	**47.23**	**72.6**	**18.67**	30.29	**50.92**	0.76

In Experiment#2, where only MViTv2 body features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 23.97, 36.57, and 59.98, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 18.16, 27.69, and 45.48, respectively with an average AUC score of 0.73. The use of only MViTv2 body features improved performance. This enhancement suggests that body features capture a broader range of motion and spatial information, which is crucial for understanding the full context of sign language gestures. By focusing on the overall body movements, the model was able to better interpret the dynamics of each sign, leading to more accurate predictions.

In Experiment#3, where MViTv2 hand and body features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 26.62, 40.43, and 66.28, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 17.22, 28.77, and 49.4, respectively with an average AUC score of 0.76. This approach led to improvements in performance on the validation set, suggesting that the integration of hand features with body features provided a more detailed and nuanced understanding of the sign language gestures during the validation phase. However, this initial success did not carry over to the test set, where a decrease in performance was observed. The drop in test set accuracy indicates that the added hand features may have introduced complexity or noise that hindered the model's ability to generalize to unseen data.

In Experiment#4, which included hand and body features along with hand and pose landmarks, the results on the validation set for Top-1, Top-2, and Top-5 were 30.64, 44.72, and 70.07, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 18.38, 30.33, and 50.6, respectively with an average AUC score of 0.75. The model incorporated

hand and body features from MViTv2 along with hand and pose landmarks, leading to a significant increase in performance on the validation set compared to Experiments 2 and 3. The addition of landmark data provided the model with precise information about key points and spatial relationships within the gestures, enhancing its ability to accurately classify signs during the validation phase. This improvement underscores the value of landmark data in capturing critical details that might be missed by the broader features alone. However, despite the substantial gains observed in the validation set, the outcomes on the test set remained similar to those seen in the previous experiments. This suggests that while the landmark data helps the model perform better on familiar data, it does not translate into better generalization to unseen data.

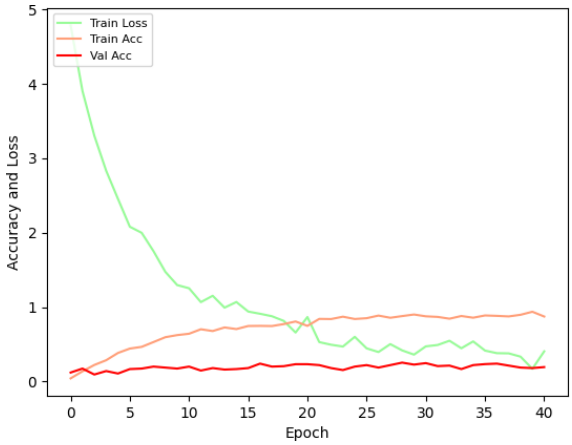
In Experiment#5, where only MViTv2 body features and ResNeXt hand features were used, the results on the validation set for Top-1, Top-2, and Top-5 were 28.14, 42.01, and 67.52, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 17.88, 27.73, and 46.62, respectively with an average AUC score of 0.73. Compared to Experiment#2, the performance increased on the validation set, while similar results were obtained on the test set. The inclusion of ResNeXt hand features added a layer of detailed information about hand movements, contributing to a more comprehensive understanding of the gestures during the validation phase. However, despite these gains in the validation set, the results on the test set remained similar to those achieved in Experiment-2. This indicates that while the addition of ResNeXt hand features helps the model perform better on known data, it does not necessarily improve its generalization to unseen data.

In Experiment#6, which added pose landmarks to the resources used in Experiment#5, the results on the validation set for Top-1, Top-2, and Top-5 were 29.19, 43.74, and 68.18, respectively. On the test set, the results for Top-1, Top-2, and Top-5 were 18.48, 28.24, and 45.96, respectively with an average AUC score of 0.77. Compared to Experiment#5, where the only difference was the use of pose landmarks, the performance increased on the validation set, while the results on the test set remained approximately similar. The improved performance on the validation set can be attributed to the inclusion of pose landmarks, which provided the model with additional spatial information. This enhancement allowed the model to better capture the nuances of sign language gestures, thereby improving its classification accuracy during training. The comparable results obtained from the test set, relative to Experiment#5, suggest that while the inclusion of pose landmarks improved the model's performance on familiar data, they did not significantly enhance its ability to generalize to new, unseen data.

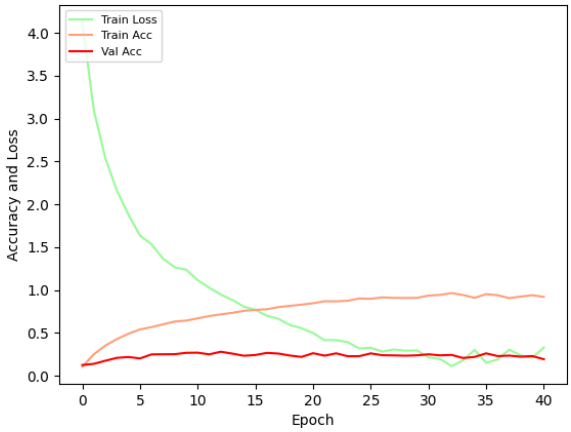
Similar to the observations from the MS-ZSSLR-W experiments, Experiment#7, which utilized all features and data, achieved the highest performance. For the validation dataset,

the Top-1, Top-2, and Top-5 accuracies were 31.73, 47.23, and 72.6, respectively. On the test dataset, the performances were 18.67, 30.29, and 50.92 for Top-1, Top-2, and Top-5 accuracies, respectively, with an average AUC score of 0.76. This outcome highlights that integrating diverse feature sets is crucial for capturing the full complexity of sign language gestures, leading to superior generalization and recognition capabilities.

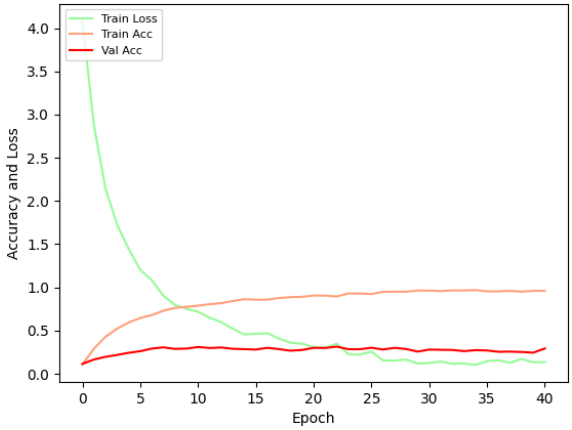
The accuracy and loss graphs for Experiment#7 on the ASL-Text, MS-ZSSLR-W, and MS-ZSSLR-C datasets are shown in Figure 5.3a, Figure 5.3b ,and Figure 5.3c, respectively. These figures demonstrate that as the epochs advance, both the training and validation performance improve while the loss decreases. This trend indicates that the models are effectively learning.



(a) Accuracy and loss graphics on ASL-Text



(b) Accuracy and loss graphics on MS-ZSSLR-W



(c) Accuracy and loss graphics on MS-ZSSLR-C

Figure 5.3: Accuracy and loss graphics on datasets

The comparison of the results for ZSL setting obtained from the experiments conducted on all datasets with SOTA baseline approaches and differences between SOTA and our approach are presented in Table 5.5. The results from the test datasets of Experiment#7 were used for this comparison. This observation highlights the enhanced outcomes achieved by the studies introducing the MViTv2 model [21]. Additionally, the extraction of text features

using the CLIP architecture, which was trained using a zero-shot method, as opposed to the BERT architecture trained with supervised methods in the baseline study [3], improves the adaptability of this architecture to other zero-shot frameworks. This enhancement is identified as a key contributor to the increased success reflected in this table. The integration of the ST-GCN method, applied for the first time to the pose landmarks in this study, and the Bi-LSTM method applied to hand landmarks, has resulted in a greater increase in performance compared to the method using ResNeXt features extracted from separate right and left hand videos, previously trained on the HaGRID dataset. This is attributed to the fact that the HaGRID dataset consists exclusively of high-resolution videos, whereas the datasets used in the present study are low-resolution.

Examining Table 5.5, it is evident that our study demonstrates superior performance. In the ASL-Text dataset, the performance gap between the baseline and our approach is minimal, likely due to the dataset having fewer training samples per class compared to others. However, for the MS-ZSSLR-W dataset, our approach nearly doubled the performance, and for the MS-ZSSLR-C dataset, it achieved more than double the performance.

Table 5.5: Contributions on datasets for ZSL setting

	Study	Top-1	Top-2	Top-5
Baseline	MS-ZSSLR-W [3]	8.7	15.7	30.3
	MS-ZSSLR-C [3]	7.01	11.5	22.1
	ASL-Text [2]	20.9	32.5	**51.4**
Ours	MS-ZSSLR-W	**14.86**	**24.6**	**43.07**
	MS-ZSSLR-C	**18.67**	**30.29**	**50.92**
	ASL-Text	**22.76**	**32.83**	50.15
Improvement	MS-ZSSLR-W	+6.16	+8.9	+12.77
	MS-ZSSLR-C	+11.66	+18.79	+28.82
	ASL-Text	+1.89	+0.33	-1.25

The results obtained using binary attributes for ZSL setting are shown in Table 5.6. In the validation set of the MS-ZSSLR-W dataset, the Top-1, Top-2, and Top-5 accuracies were 38.8, 54.55, and 76.11, respectively. In the test set of same dataset, the results were 24.41, 37.74, and 57.11 for Top-1, Top-2, and Top-5, respectively. In the validation set of the MS-ZSSLR-C dataset, the performances achieved were 45.26 for Top-1 accuracy, 61.11 for Top-2 accuracy, and 81.41 for Top-5 accuracy. In the test set of same dataset, the results were 33.66, 48.14, and 69.13 for Top-1, Top-2, and Top-5, respectively. When examining Tables 5.3 and 5.4, where experiments without using binary attributes were conducted, it can be seen that the use of binary features has led to significant improvements in the results. This indicates that not only improving the visual space in ZSL but also enhancing the semantic space contributes to better outcomes.

Table 5.6: Results on datasets with binary attribute usage

Dataset	Val (30 classes)			Test (50 classes)		
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
MS-ZSSLR-W	38.8	54.55	76.11	24.41	37.74	57.11
MS-ZSSLR-C	45.26	61.11	81.41	33.66	48.14	69.13

Experiments conducted using the CLIP architecture's contrastive approach for ZSL setting, both with and without binary attributes, are shown in Table 5.7. These results indicate that the contrastive approach does not perform well in this domain. Results that are higher compared to their use of attribute category are highlighted in bold.

In the experiments conducted on the MS-ZSSLR-W dataset using binary attributes with the contrastive approach, the results for Top-1, Top-2, and Top-5 were 23.09, 35.13, and 54.43, respectively, while in the LLE method, the results were 24.41, 37.74, and 57.11, respectively. Without using binary attributes, the contrastive approach yielded results of 15.48, 24.09, and 41.13 for Top-1, Top-2, and Top-5, respectively, whereas the LLE method yielded results of 14.86, 24.6, and 43.07 for Top-1, Top-2, and Top-5, respectively.

In the MS-ZSSLR-C dataset experiments using binary attributes with a contrastive approach, the Top-1, Top-2, and Top-5 accuracies were 30.47, 44.02, and 63.77, respectively. For the LLE method under the same conditions, the accuracies were 33.66, 48.14, and 69.13, respectively. Without binary attributes, the contrastive approach produced Top-1, Top-2, and Top-5 accuracies of 18.61, 29.65, and 51.07, respectively, while the LLE method achieved 18.67, 30.29, and 50.92 for Top-1, Top-2, and Top-5, respectively.

When examining Table 5.7, it is clear that using LLE yields better results when binary attributes are used. When binary attributes are not used, the results are approximately similar, but the LLE algorithm still performs slightly better. It can be concluded that when textual representations are enhanced with binary attributes, the LLE algorithm is better suited for this domain compared to the contrastive approach because it provides a direct, discriminative mapping between visual features and semantic descriptions, aligning well with the classification goal. Contrastive learning, while powerful, may require more careful tuning and a larger dataset to achieve comparable performance.

The results obtained from the datasets for the GZSL setting are shown in Table 5.8. In the ASL-Text dataset, the validation set achieved Top-1, Top-2, and Top-5 accuracies of 36.2, 44.94, and 56.95, respectively. In the test set, these figures were 31.02, 42.13, and 55.17. For the MS-ZSSLR-W dataset, the validation set saw Top-1, Top-2, and Top-5 accuracies of 38.74, 49.5, and 62.87, respectively, while the test set recorded accuracies of 33.85, 43.68,

Table 5.7: Results with/without using attributes and contrastive learning approach

Setting	Method	Metric	MS-ZSSLR-W	MS-ZSSLR-C
W/ Attributes	Contrastive	Top-1	23.09	30.47
		Top-2	35.13	44.02
		Top-5	54.43	63.77
	LLE	Top-1	**24.41**	**33.66**
		Top-2	**37.74**	**48.14**
		Top-5	**57.11**	**69.13**
W/O Attributes	Contrastive	Top-1	**15.48**	18.61
		Top-2	24.09	29.65
		Top-5	41.13	**51.07**
	LLE	Top-1	14.86	**18.67**
		Top-2	**24.6**	**30.29**
		Top-5	**43.07**	50.92

and 56.9. In the MS-ZSSLR-C dataset, the validation set results were 49.45, 59.23, and 70.76 for Top-1, Top-2, and Top-5, respectively, and the test set results were 42.73, 51.95, and 63.67.

Table 5.8: Results on datasets for GZSL setting

Dataset	Val			Test		
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
ASL-Text	36.2	44.94	56.95	31.02	42.13	55.17
MS-ZSSLR-W	38.74	49.5	62.87	33.85	43.68	56.9
MS-ZSSLR-C	49.45	59.23	70.76	42.73	51.95	63.67

Table 5.9: Contributions on datasets for GZSL setting

Study		Top-1	Top-2	Top-5
Baseline	MS-ZSSLR-C [3]	33.4	40.5	48.8
	ASL-Text [2]	22.5	32.5	45.6
Ours	MS-ZSSLR-C	**42.73**	**51.95**	**63.67**
	ASL-Text	**31.02**	**42.13**	**55.17**
Improvement	MS-ZSSLR-C	+9.33	+11.45	+14.87
	ASL-Text	+8.52	+9.63	+9.57

The comparison of the results obtained for the GZSL setting with the SOTA method and the improvements made are shown in Table 5.9. These results were obtained using only the architecture of Experiment#7 on the GZSL data. When examining the comparison in the table, it can be clearly seen that, similar to the ZSL setting, the combination of MViTv2, ResNeXt, ST-GCN, and CLIP architectures improves performance.

In GZSL experiments, the model achieved results that surpassed those obtained in all previous experiments, highlighting the effectiveness of the GZSL approach in handling the complexities of ZSSLR. This significant improvement demonstrates the model's enhanced ability to generalize across both seen and unseen classes, effectively bridging the gap between traditional ZSL and practical application scenarios where a mix of familiar and unfamiliar signs are encountered.

6. DISCUSSION

An ablation study is conducted to better understand the contribution of different components to the overall model in a complex setup. By removing specific components of the model, the impact of that component on the overall performance can be observed. This helps in identifying which components are necessary or unnecessary for achieving the desired performance.

The performance outcomes for ZSL setting of Experiment#1 in the ASL-Text, which relied solely on hand features, are comparable to those of Experiment#7, where we used all data and features, as indicated in Figure 5.2. Nonetheless, Experiment#1 gives the weakest validation scores among the evaluations conducted on the ASL-Text. This is due to the fewer training examples in ASL-Text relative to other datasets in this research, as well as the constraints of only using hand features, highlighting challenges of data insufficiency and limited representation. This observation becomes clearer when noting that the test outcomes of Experiment#1 in the MS-ZSSLR-W/C datasets do not exceed the validation results. It could be argued that the test results in Experiment#1 of ASL-Text were coincidental since they exceeded the validation outcomes. Within the MS-ZSSLR-W/C datasets, as detailed in Table 5.3 and 5.4, Experiment#1 yielded the lowest test scores. Nevertheless, given the ample samples in the training sets of these datasets, this poor performance is likely due to the insufficient representation of the data alone.

Inconsistent outcomes were observed across all datasets; however, Experiment #7, which utilized all data sources, delivered the strongest validation and test results in the MS-ZSSLR-W and MS-ZSSLR-C datasets. When comparing Experiment#5 and Experiment#6, where the only variation was the inclusion of pose landmark data, it is noted that the application of ST-GCN either enhanced performance or produced results very similar to those without it in the MS-ZSSLR-W dataset. Among the first four experiments, the most favorable test outcomes were recorded in Experiment#4, which incorporated ST-GCN for pose landmarks.

7. CONCLUSION

This thesis aims to perform ZSSLR by developing visual representations using hand, pose, and body data, and textual representations using binary attributes. Unlike other studies, this work includes the use of ST-GCN for pose landmark data, the application of MViTv2 and the ResNeXt architecture pre-trained on the HaGRID dataset for feature extraction, the use of Bi-LSTM and self-attention methods, the extraction of textual features using CLIP, and the examination of the contrastive approach. By combining these methods, performance improvements have been achieved for the MS-ZSSLR-W, MS-ZSSLR-C, and ASL-Text datasets in both ZSL and GZSL settings. These results indicate that the ZSSLR field requires further research. The main challenge in ZSSLR is how well the visual and textual representations are created and how well they suit the applied ZSL method. This difficulty can be mitigated by bridging the gap between seen and unseen classes in the textual and visual domains. The lack of a direct relationship between observed and unobserved classes complicates this challenge, necessitating the application of different methods for knowledge transfer and generalization.

SLR has made significant progress in recent years, leveraging advances in computer vision, deep learning, and natural language processing. However, there are still numerous challenges and opportunities for future research that can further enhance the robustness and applicability of SLR systems.

Various methods can be applied to improve visual and textual representations. Different architectures developed for classifying hand signs, poses, and facial movements can be used to enhance visual representations. Additionally, different tools can be used to extract hand and pose landmark data, and various GCN architectures can be combined and utilized.

While the ResNeXt architecture has provided a strong and reliable baseline, we believe that exploring alternative architectures could yield even better performance. Specifically, ViT architecture warrants investigation due to its proven success in various image recognition tasks, particularly for its ability to capture long-range dependencies and contextual information. Additionally, advanced CNN architectures, such as DenseNet and EfficientNet, can be explored for their sophisticated feature extraction capabilities. These architectures utilize distinct approaches to handle the complexity and variability inherent in sign language gestures, which could lead to improved recognition accuracy.

Current SLR systems often struggle with generalization across different signers and environments. Future research should focus on developing models that are invariant to variations in signer appearance, signing speed, and environmental conditions. Techniques such

as domain adaptation, transfer learning, and few-shot learning could be explored to improve generalization capabilities.

Sign languages are not universal; each has its own linguistic structure and cultural context. Future research should aim to develop models that can recognize and translate multiple sign languages, taking into account their unique characteristics. This includes creating large, diverse datasets that represent various sign languages and cultures.

Developing standardized and robust evaluation metrics is essential for assessing the performance of SLR systems. Future work should focus on creating benchmarks that reflect real-world challenges, including signer variability, noise, and occlusion. Collaborative efforts within the research community can facilitate the establishment of such standards.

As SLR technology becomes more widespread, it is crucial to address ethical considerations, such as privacy and user consent. Future research should also emphasize user-centric design, ensuring that SLR systems are accessible, inclusive, and tailored to the needs of diverse user groups, including those with different levels of hearing ability

REFERENCES

[1] A. Kapitanov, A. Makhlyarchuk, and K. Kvanchiani, "Hagrid-hand gesture recognition image dataset," *arXiv preprint arXiv:2206.08219*, 2022.

[2] Y. C. Bilge, N. Ikizler-Cinbis, and R. G. Cinbis, "Zero-shot sign language recognition: Can textual data uncover sign languages?," *British Machine Vision Conference (BMVC)*, 2019.

[3] Y. C. Bilge, R. G. Cinbis, and N. Ikizler-Cinbis, "Towards zero-shot sign language recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1217–1232, 2022.

[4] R. A. Nihal, S. Rahman, N. M. Broti, and S. A. Deowan, "Bangla sign alphabet recognition with zero-shot and transfer learning," *Pattern Recognition Letters*, vol. 150, pp. 84–93, 2021.

[5] J. Wu, Y. Zhang, and X. Zhao, "A prototype-based generalized zero-shot learning framework for hand gesture recognition," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 3435–3442, IEEE, 2021.

[6] A. Yin, Z. Zhao, W. Jin, M. Zhang, X. Zeng, and X. He, "Mlslt: Towards multilingual sign language translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5109–5119, 2022.

[7] N. Pugeault and R. Bowden, "Spelling it out: Real-time asl fingerspelling recognition," in *2011 IEEE International conference on computer vision workshops (ICCV workshops)*, pp. 1114–1119, IEEE, 2011.

[8] 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*, pp. 1459–1469, 2020.

[9] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015.

[10] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[11] U. Nandi, A. Ghorai, M. M. Singh, C. Changdar, S. Bhakta, and R. Kumar Pal, "Indian sign language alphabet recognition system using cnn with diffgrad optimizer and

stochastic pooling," *Multimedia Tools and Applications*, vol. 82, no. 7, pp. 9627–9648, 2023.

[12] J. Gangrade and J. Bharti, "Vision-based hand gesture recognition for indian sign language using convolution neural network," *IETE Journal of Research*, vol. 69, no. 2, pp. 723–732, 2023.

[13] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K.-H. Park, and J. Kim, "An integrated mediapipe-optimized gru model for indian sign language recognition," *Scientific Reports*, vol. 12, no. 1, p. 11964, 2022.

[14] S. Katoch, V. Singh, and U. S. Tiwary, "Indian sign language recognition system using surf with svm and cnn," *Array*, vol. 14, p. 100141, 2022.

[15] Y. Min, A. Hao, X. Chai, and X. Chen, "Visual alignment constraint for continuous sign language recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11542–11551, 2021.

[16] R. Hills, D. Renner, P. S. Lott, and C. Valli, *The Gallaudet dictionary of American sign language*. Gallaudet University Press, 2021.

[17] D. Brien, B. D. Association, *et al.*, "Dictionary of british sign language/english," *(No Title)*, 1992.

[18] H. Dikyuva, B. Makaroğlu, and E. Arık, "Türk işaret dili dilbilgisi kitabı," *Aile ve Sosyal Politikalar Bakanlığı Yayınları: Ankara*, 2015.

[19] M. P. Lewis and F. Gary, "Simons, and charles d. fennig (eds.). 2013," *Ethnologue: Languages of the world,*, pp. 233–62, 2015.

[20] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500, 2017.

[21] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4804–4814, 2022.

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021.

[23] J. Duan, S. Zhou, J. Wan, X. Guo, and S. Li, "Multi-modality fusion based on consensus-voting and 3d convolution for isolated gesture recognition," *arXiv preprint arXiv:1611.06689*, 2016.

[24] J. Pu, W. Zhou, and H. Li, "Dilated convolutional network with iterative optimization for continuous sign language recognition.," in *IJCAI*, vol. 3, p. 7, 2018.

[25] M. Hrúz, P. Campr, E. Dikici, A. A. Kındıroğlu, Z. Krňoul, A. Ronzhin, H. Sak, D. Schorno, H. Yalçın, L. Akarun, *et al.*, "Automatic fingersign-to-speech translation system," *Journal on Multimodal User Interfaces*, vol. 4, pp. 61–79, 2011.

[26] A. A. Kindiroglu, H. Yalcin, O. Aran, M. Hruz, P. Campr, L. Akarun, and A. Karpov, "Automatic recognition fingerspelling gestures in multiple languages for a communication interface for the disabled," *Pattern Recognition and Image Analysis*, vol. 22, pp. 527–536, 2012.

[27] M. Rivera-Acosta, S. Ortega-Cisneros, J. Rivera, and F. Sandoval-Ibarra, "American sign language alphabet recognition using a neuromorphic sensor and an artificial neural network," *Sensors*, vol. 17, no. 10, p. 2176, 2017.

[28] M. A.-A. Bhuiyan, "Recognition of asl for human-robot interaction," *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY*, vol. 17, no. 7, pp. 66–71, 2017.

[29] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "A multimodal framework for sensor based sign language recognition," *Neurocomputing*, vol. 259, pp. 21–38, 2017.

[30] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A hand gesture interface device," *ACM Sigchi Bulletin*, vol. 18, no. 4, pp. 189–192, 1986.

[31] S. Tamura and S. Kawasaki, "Recognition of sign language motion images," *Pattern Recognition*, vol. 21, no. 4, pp. 343–353, 1988.

[32] M. B. Waldron and S. Kim, "Isolated asl sign recognition system for deaf persons," *IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 3, pp. 261–271, 1995.

[33] M. W. Kadous *et al.*, "Machine recognition of auslan signs using powergloves: Towards large-lexicon recognition of sign language," in *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, vol. 165, pp. 165–174, DE Wilmington, 1996.

[34] M. Zahedi, D. Keysers, T. Deselaers, and H. Ney, "Combination of tangent distance and an image distortion model for appearance-based sign language recognition," in *Pattern Recognition: 27th DAGM Symposium, Vienna, Austria, August 31-September 2, 2005. Proceedings 27*, pp. 401–408, Springer, 2005.

[35] H. Cooper and R. Bowden, "Sign language recognition using boosted volumetric features," in *Proceedings of the IAPR Conference on Machine Vision Applications*, pp. 359–362, 2007.

[36] O. Koller, O. Zargaran, H. Ney, and R. Bowden, "Deep sign: Hybrid cnn-hmm for continuous sign language recognition," in *Proceedings of the British Machine Vision Conference 2016*, 2016.

[37] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4297–4305, 2017.

[38] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2306–2320, 2019.

[39] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.

[40] S. Albanie, G. Varol, L. Momeni, T. Afouras, J. S. Chung, N. Fox, and A. Zisserman, "Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 35–53, Springer, 2020.

[41] L. Momeni, G. Varol, S. Albanie, T. Afouras, and A. Zisserman, "Watch, read and lookup: learning to spot signs from multiple supervisors," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[42] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10023–10033, 2020.

[43] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, "Skeleton aware multi-modal sign language recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3413–3423, 2021.

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

[45] H. Hu, W. Zhou, and H. Li, "Hand-model-aware sign language recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1558–1566, 2021.

[46] L. Hu, L. Gao, Z. Liu, and W. Feng, "Continuous sign language recognition with correlation network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2529–2539, 2023.

[47] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for sign language recognition and translation," *IEEE Transactions on Multimedia*, vol. 24, pp. 768–779, 2021.

[48] A. Tunga, S. V. Nuthalapati, and J. Wachs, "Pose-based sign language recognition using gcn and bert," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 31–40, 2021.

[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[50] M. Vázquez-Enríquez, J. L. Alba-Castro, L. Docío-Fernández, and E. Rodríguez-Banga, "Isolated sign language recognition with multi-scale spatial-temporal graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3462–3471, 2021.

[51] F. Wei and Y. Chen, "Improving continuous sign language recognition with cross-lingual signs," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23612–23621, 2023.

[52] M. Boháček and M. Hrúz, "Sign pose-based transformer for word-level sign language recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 182–191, 2022.

[53] R. Sreemathy, M. Turuk, I. Kulkarni, and S. Khurana, "Sign language recognition using artificial intelligence," *Education and Information Technologies*, vol. 28, no. 5, pp. 5259–5278, 2023.

[54] D. R. Kothadiya, C. M. Bhatt, T. Saba, A. Rehman, and S. A. Bahaj, "Signformer: Deepvision transformer for sign language recognition," *IEEE Access*, vol. 11, pp. 4730–4739, 2023.

[55] J. Shin, A. S. Musa Miah, M. A. M. Hasan, K. Hirooka, K. Suzuki, H.-S. Lee, and S.-W. Jang, "Korean sign language recognition using transformer-based deep neural network," *Applied Sciences*, vol. 13, no. 5, p. 3029, 2023.

[56] J. Bora, S. Dehingia, A. Boruah, A. A. Chetia, and D. Gogoi, "Real-time assamese sign language recognition using mediapipe and deep learning," *Procedia Computer Science*, vol. 218, pp. 1384–1393, 2023.

[57] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks.," in *AAAI*, vol. 1, p. 3, 2008.

[58] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[59] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958, IEEE, 2009.

[60] T. Mensink, E. Gavves, and C. G. Snoek, "Costa: Co-occurrence statistics for zero-shot classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2441–2448, 2014.

[61] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong, "Transductive multi-view embedding for zero-shot recognition and annotation," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pp. 584–599, Springer, 2014.

[62] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, pp. 2152–2161, PMLR, 2015.

[63] X. Xu, T. Hospedales, and S. Gong, "Semantic embedding space for zero-shot action recognition," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 63–67, IEEE, 2015.

[64] J. Lei Ba, K. Swersky, S. Fidler, *et al.*, "Predicting deep zero-shot convolutional neural networks using textual descriptions," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4247–4255, 2015.

[65] G. Ou, G. Yu, C. Domeniconi, X. Lu, and X. Zhang, "Multi-label zero-shot learning with graph convolutional networks," *Neural Networks*, vol. 132, pp. 333–341, 2020.

[66] S. Chen, Z. Hong, Y. Liu, G.-S. Xie, B. Sun, H. Li, Q. Peng, K. Lu, and X. You, "Transzero: Attribute-guided transformer for zero-shot learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 330–338, 2022.

[67] A. Roy, D. Ghosal, E. Cambria, N. Majumder, R. Mihalcea, and S. Poria, "Improving zero-shot learning baselines with commonsense knowledge," *Cognitive Computation*, vol. 14, no. 6, pp. 2212–2222, 2022.

[68] J. Wang, X. Wang, and H. Zhang, "Domain-aware multi-modality fusion network for generalized zero-shot learning," *Neurocomputing*, vol. 488, pp. 23–35, 2022.

[69] A. Gupta, S. Narayan, S. Khan, F. S. Khan, L. Shao, and J. van de Weijer, "Generative multi-label zero-shot learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[70] D. Cheng, G. Wang, B. Wang, Q. Zhang, J. Han, and D. Zhang, "Hybrid routing transformer for zero-shot learning," *Pattern Recognition*, vol. 137, p. 109270, 2023.

[71] S. N. Gowda, "Synthetic sample selection for generalized zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 58–67, 2023.

[72] J. Guo, S. Guo, Q. Zhou, Z. Liu, X. Lu, and F. Huo, "Graph knows unknowns: Reformulate zero-shot learning as sample-level graph recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 7775–7783, 2023.

[73] M. Liu, F. Li, C. Zhang, Y. Wei, H. Bai, and Y. Zhao, "Progressive semantic-visual mutual adaption for generalized zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15337–15346, 2023.

[74] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7083–7093, 2019.

[75] T. Yuan, S. Sah, T. Ananthanarayana, C. Zhang, A. Bhat, S. Gandhi, and R. Ptucha, "Large scale sign language interpretation," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pp. 1–5, IEEE, 2019.

[76] K. Karpouzis, G. Caridakis, S.-E. Fotinea, and E. Efthimiou, "Educational resources and implementation of a greek sign language synthesis architecture," *Computers & Education*, vol. 49, no. 1, pp. 54–74, 2007.

[77] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 44–52, 2015.

[78] S. Masood, A. Srivastava, H. C. Thuwal, and M. Ahmad, "Real-time sign language gesture (word) recognition from video sequences using cnn and rnn," in *Intelligent Engineering Informatics: Proceedings of the 6th International Conference on FICTA*, pp. 623–632, Springer, 2018.

[79] 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," *arXiv preprint arXiv:1906.08172*, 2019.

[80] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[81] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6824–6835, 2021.

[82] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[83] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[84] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[85] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 819–826, 2013.

[86] J. Weston, S. Bengio, and N. Usunier, "Large scale image annotation: learning to rank with joint word-image embeddings," *Machine Learning*, vol. 81, pp. 21–35, 2010.

[87] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[88] C. Neidle, A. Thangali, and S. Sclaroff, "Challenges in development of the american sign language lexicon video dataset (asllvd) corpus," in *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC*, Citeseer, 2012.

[89] R. A. Tennant and M. G. Brown, *The American sign language handshape dictionary*. Gallaudet University Press, 1998.

[90] H. R. Vaezi Joze and O. Koller, "Ms-asl: A large-scale data set and benchmark for understanding american sign language," *arXiv e-prints*, pp. arXiv–1812, 2018.