SIGN LANGUAGE RECOGNITION USING DINO AND HEATMAP TECHNIQUES

by

Onur Şero

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University

2025

ACKNOWLEDGEMENTS

Firstly, I would like to thank my thesis advisor, Prof. Lale Akarun, for her expertise, patience, and continuous guidance during my master's studies. Her guidance has given me the confidence and knowledge necessary to navigate the challenges during my research. I am greatly thankful for her consistent support and mentorship.

I would also like to thank Assist. Prof. İnci M. Baytaş and Assist. Prof. Furkan Kıraç for participating in my thesis jury and sharing their valuable comments and recommendations.

I am very thankful to Oğulcan Özdemir for his continuous support, invaluable insights, and encouragement throughout my thesis. I also extend my deepest gratitude to all faculty members of the Computer Engineering Department for providing an excellent learning experience.

I would also like to thank my partner, Buse Eryiğit, for supporting me throughout my master's studies. She was there for me during all the sleepless nights. I would also like to thank my family for their endless support, motivating words, and kindness. Things would not be possible without them.

ABSTRACT

SIGN LANGUAGE RECOGNITION USING DINO AND HEATMAP TECHNIQUES

Deaf people mainly use sign language as their primary method of communication. However, learning sign language can be challenging because it requires considerable time and effort, particularly for those with hearing ability. Sign Language Recognition (SLR) involves identifying and translating individual signs into corresponding spoken language words, known as glosses. SLR approaches help communication problems between the deaf and hearing communities. This study introduced an approach that combines self-supervised learning Distillation with No Labels (DINO) with heatmap-based 3-dimensional Convolutional Neural Network (3D CNN) cue representation techniques to improve recognition accuracy. The proposed approach uses DINO to focus on learning visual cues from hands and faces through self-supervised learning. Heatmap-based modeling is applied to extract upper body posture and movement. The approach includes the proposed CueSeqNet model for body part-specific cue processing and the proposed HeatPose2D/3D model, which applies CNNs to interpret joint and limb heatmaps. An introduced mixed modeling strategy, CueHeat2D/3D, combines these cues for improved recognition. We tested our models with the isolated Turkish SLR dataset BosphorusSign22k. The proposed model achieves 91.71% classification accuracy. Our study shows that self-supervised learning DINO with a 3D heatmap gives competitive results in isolated SLR.

ÖZET

DINO VE ISI HARİTASI TEKNİKLERİ İLE İŞARET DİLİ TANIMA

Konuşma engelli insanlar iletişim kurmak için temel yöntem olarak işaret dillerini kullanmaktadırlar. Ancak işaret dilini öğrenmek, özellikle işaret dili bilmeyenler için önemli miktarda zaman ve çaba gerektirdiğinden zorlu olabilmektedir. Bu yöntemler işaret dili görüntülerindeki işaretçilerin yaptığı bireysel işaretleri tanımlamak ve ardından bu tanımlanan işaretleri konuşulan dil ifadelerine çevirmeyi amaçlamaktadır. İDT yaklaşımları, konuşma engelli toplum ile konuşma engeli bulunmayan kişiler arasındaki iletişim sorunlarının çözümüne yardımcı olmaktadır. Bu çalışmada, tanıma doğruluğunu artırmak için kendi kendine denetimli öğrenme yöntemi olan Etiketsiz Öz Damıtma ile ısı haritası tabanlı 3 boyutlu Evrişimsel Sinir Ağları (3B ESA) öznitelik çıkarma tekniklerini birleştiren bir yaklaşım tanıtılmıştır. Önerdiğimiz yaklaşımda, kendi kendine denetimli öğrenme yoluyla el ve yüz görsel ipuçlarını öğrenmeye odaklanmak için Etiketsiz Öz Damıtma yöntemi kullanılmıştır. Üst vücut duruşunu ve hareketini çıkarmak için ısı haritası tabanlı modelleme uygulanmıştır. Yaklaşım, el ve yüz ipucu işleme için önerilen CueSeqNet modelini ve eklem ve uzuv ısı haritalarını yorumlamak için ESA'ları uygulayan HeatPose2D/3D modelini içermektedir. Tanıtılan karma modelleme stratejisi CueHeat2D/3D, bu ipuçlarını daha iyi tanıma için birleştirmektedir. Bu tezde önerilen modeller yalıtılmış Türk İDT veri seti olan BosphorusSign22k veri seti ile test edilmiştir. Önerdiğimiz model %91.71 sınıflandırma başarımı elde etmiştir. Çalışmamız, yalıtılmış İDT alanında 3B ısı haritası ile birleştirilen kendi kendine denetimli öğrenme yaklaşımı olan Etiketsiz Öz Damıtma yönteminin rekabetçi sonuçlar verdiğini göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c_a	Confidence scores of joint a
$D(a, b)$	Distance between point a and b
$\mathbf{J}_{kij}$	Joint heatmap intensity at pixel (i, j) for the k-th joint
$\mathbf{L}_{kij}$	Limb heatmap intensity at pixel (i, j) for the k-th limb
$\text{seg}[a, b]$	Segment between joints a and b
i, j	Pixel coordinates
σ	Variance of the Gaussian component
exp	Exponential function

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
ASL	American Sign Language
AUTSL	Ankara University Turkish Sign Language Dataset
BERT	Bidirectional Encoder Representations from Transformers
BLSTM	Bidirectional Long-Short Term Memory
BSL	British Sign Language
BYOL	Bootstrap Your Own Latent
CNN	Convolutional Neural Network
CueHeat2D	Cue-based 2D Heatmap Model
CueHeat3D	Cue-based 2D Heatmap Model
CueSeqNet	Cue Sequence Network
DAN	Distract Your Attention Network
DINO	Distillation with No Labels
FER	Facial Expression Recognition
FPM	Feature Pooling Module
GCN	Graph Convolutional Network
HAR	Human Action Recognition
HeatPose2D	Heatmap-based 2D Pose Model
HeatPose3D	Heatmap-based 2D Pose Model
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HOF	Histogram of Optical Flow
I3D	Inflated 3D Convolutional Network
IDT	Improved Dense Trajectories
LSTM	Long Short-Term Memory
MAE	Masked Autoencoder
MBH	Motion Boundary Histogram

MC3	Mixed Convolution 3D
MC-LSTM	Multi-Cue Long Short-Term Memory
MHI	Motion History Image
MV-LSTM	Multi-View Long Short-Term Memory
PCA	Principal Component Analysis
RGB	Red Green Blue
RNN	Recurrent Neural Network
SF	Score Fusion
SL	Sign Language
SLR	Sign Language Recognition
SSL	Self-supervised Learning
ST	Spatio-Temporal
ST-GCN	Spatial-Temporal Graph Convolutional Network
SVM	Support Vector Machine
TAF	Temporal Accumulative Features
TID	Turkish Sign Language
TSL	Turkish Sign Language
ViT	Vision Transformer

1. INTRODUCTION

Sign language is the primary means of communication for deaf people in their daily lives [1]. In contrast to spoken languages that utilize audio elements, sign languages depend on visual cues [2]. Every deaf community has developed a distinct sign language independent of the spoken language. For example, there are British Sign Language (BSL) and American Sign Language (ASL). The sign language of Türkiye is Turkish Sign Language (Türk İşaret Dili-TİD).

However, mastering these sign languages poses a major challenge and requires a significant time commitment from those with hearing ability. Sign Language Recognition (SLR) is recognizing individual signs and then converting those signs into corresponding words from spoken language, called glosses [3]. Sign Language Recognition (SLR) techniques play a significant role in communication between the deaf community and hearing individuals. The need for SLR comes from the communication barriers between those with hearing disabilities and those unfamiliar with sign language. By automating the recognition process, SLR systems can enhance access to essential services in sectors like healthcare, education, and public services [4].

Recent advancements in deep learning have markedly enhanced Sign Language Recognition (SLR) through the utilization of robust models, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Vision Transformers (ViTs) [5], and Graph Convolutional Networks (GCN) [6–8]. Recently, Graph Convolutional Networks (GCNs) have been investigated as an efficient method for processing skeletal and trajectory-based features, as they may represent the interrelations among various keypoints in a signer's body, including hand, arm, and face [8]. However, they provide effective methods for extracting information from sign language videos; large amounts of labeled data and processing power are frequently needed. Because of their high resource requirements, they may not be useful in some cases, such as real-time applications.

Self-supervised learning (SSL) has been investigated recently as a potential solution to these issues. SSL derives knowledge from the data on its own, without any external supervision. Many unlabeled videos can be used in models thanks to the valuable characteristics of SSL. The DINO (Distillation with No Labels) [9] framework is one of the preferred SSL techniques for images. DINO compares different variations of the same image to learn powerful representations without requiring manual labeling. Heatmap techniques are additional helpful tools that analyze body pose information, which is a critical feature for robust sign language recognition. Heatmaps provide a simple, color-coded representation of important aspects of an image, such as the locations of the hands, face, and upper body joints. PoseConv3D [10] is a heatmap-based model that captures the dynamics of skeletal keypoints over time to learn spatiotemporal features efficiently. Stable feature extraction is ensured by its robustness against pose estimation noise, which reduces errors caused by imprecise keypoint detection. This makes it especially appropriate for human action recognition, where accurate motion representation is crucial [10].

1.1. Thesis Focus and Contributions

Human Action Recognition (HAR) is the process of classifying human actions. Because both HAR and SLR rely on recognizing human motion and gestures, they have a close methodological relationship [11]. In both fields, the main difficulty is effectively capturing and analyzing spatiotemporal movement patterns, which are necessary to differentiate between various signs or actions. Self-supervised learning (SSL) in the vision domain has lately demonstrated excellent performance in classification problems [12]. DINO, a self-supervised learning approach, showed great success in human activity recognition (HAR) by obtaining effective cue representation without needing labeled data. Moreover, heatmap-based methodologies like PoseConv3D have demonstrated significant success in human activity recognition (HAR) by analyzing skeleton and joint movements, thereby facilitating robust spatiotemporal feature encoding. These advancements improved recognition accuracy, making them advantageous for HAR and SLR applications.

This thesis examines the combination of self-supervised learning techniques with heatmap-based models and proposes a framework for Sign Language Recognition. The suggested method utilizes the SSL technique's capacity to understand cue representations in video sequences and heatmaps to capture spatiotemporal relations from skeleton keypoints, helping to improve recognition performance in sign recognition. The contributions of this thesis are summarized as follows.

- We successfully adapted DINO self-supervised learning and heatmap-based methods to the Sign Language Recognition (SLR) field. We create an SLR model that utilizes DINO to extract meaningful representation from key body points, specifically hands, and face while utilizing Long Short-Term Memory (LSTM) for temporal modeling. Furthermore, we present a heatmap-based 3D CNN architecture to capture upper body posture and movement effectively. We assess the contribution of each component by conducting experiments with various models, including a 2D CNN combined with LSTM, to get insights into the impact of different modeling strategies.
- We examined various cue representations to assess the effect of each feature on Sign Language Recognition (SLR). We first assess single cue representations independently and then analyze their combinations to evaluate their complementary effects. To enhance the evaluation of self-supervised learning techniques, we exchange DINO-based feature extraction with alternative methods, such as Distract Your Attention Network (DAN) [13] for facial feature extraction and DeepHand [14] for hand pose representation. This analysis helps us to determine the best feature selection for reliable SLR performance.
- To improve performance while ensuring computational efficiency, we implemented mixed preprocessing techniques, such as active frame extraction and frame sampling. Active frame extraction helps the model process only the most informative frames where the main action is taken. Therefore, it reduces redundant computations. On the other hand, frame sampling reduces computation by selecting a subset of frames while retaining temporal flow. These optimizations enhance the model's efficiency, making it more appropriate for complex SLR applications.

1.2. Organization of the Thesis

The rest of the thesis is organized as follows: In Chapter 2, we review the literature for self-supervised learning, heatmap-based feature extraction, and sign language recognition. Chapter 3 explains cue representation methods with different modeling strategies in our proposed frameworks. In Chapter 4, we present the experiments for sign language recognition and analyze their results. In Chapter 5, we present our conclusion.

2. RELATED WORK

2.1. Self-supervised Learning

Self-supervised learning (SSL) has received much attention in recent years due to the increasing data demands of deep learning models, as manual labeling of all data is not always feasible in practice [6]. SSL methods create models with cue representations that can be used for various tasks, such as computer vision. Self-supervised learning can be divided into two main categories: generative and discriminative [15, 16].

Generative SSL techniques rely on predicting and reconstructing some part of the input data to learn a better representation [17]. Contrastive methods in discriminative self-supervised learning try to distinguish between similar and different samples in a dataset by minimizing the distance between positive pairs (augmented versions of the same image) and pushing negative pairs (different images) [18, 19]. Unlike contrastive methods, some other self-supervised learning methods do not use negative samples. These methods primarily depend on customized strategies [20, 21].

2.1.1. MAE: Masked Autoencoder

Masked Autoencoders (MAE) [22] are an effective self-supervised learning (SSL) method in the field of computer vision. Motivated by the success of masked language modeling methods like BERT [23], MAE broadens the masking principle to the image domain. MAE's fundamental approach is to mask a significant portion of the input data at random and train the model to reconstruct the missing content, which encourages the learning of rich, high-level representations [22]. The MAE methodology demonstrates that masking 75% image patches enables the model to concentrate on global semantic structures instead of small details. Their architecture uses an asymmetric encoder-decoder framework, where the encoder works on visible patches while the lightweight decoder reconstructs the masked areas to optimize computational efficiency.

Subsequent research has examined the effectiveness of MAEs through their adaptation. In [24], modified MAEs emphasize the model's capacity to capture temporal dynamics via spatiotemporal masking techniques for video representation learning. Furthermore, Curriculum-Learned Masked Autoencoders (CL-MAE) [25] are introduced as a curriculum learning technique that modifies the masking strategy during training to gradually increase task complexity, thus improving the model's capacity for representation learning. The MAE framework shows effectiveness across various domains. It establishes itself as a self-supervised learning technique to extract more generalized and robust cue representations from images without requiring extensive labeled data.

2.1.2. BYOL: Bootstrap Your Own Latent

Bootstrap Your Own Latent (BYOL) [26] is a self-supervised learning (SSL) framework that has shown great performance in learning robust data representations without the need for negative samples. This behavior distinguishes BYOL from other discriminative self-supervised learning approaches such as MAE [22]. BYOL employs two neural networks, which consist of an online network and a target network. Both networks share the same architecture, typically based on CNN (ResNet-50) [27]. These two networks engage in interaction and learning processes with each other.

BYOL is based on only positive pairs, which are different augmented versions of the same image. Only the online network is trained to predict how a positive pair of the same image will be represented by the target network. At the same time, the target network is simultaneously updated using an exponential moving average (EMA) of the online network's parameters [26]. However, there is no explicit algorithm to avoid collapsing, which means outputting the same representation for all images. Commonly, negative samples are used to avoid collapsed solutions. In order to prevent collapse, BYOL uses a slow-moving average and adds another predictor on top of the online network [26].

In order to improve self-supervised learning, BYOL uses a series of transformations in the same augmentation pipeline as SimCLR [16]. After a random patch of

the image is chosen and enlarged to 224 × 224, a random horizontal flip is performed. Then, color distortions, such as changes in hue, saturation, contrast, and brightness, are applied in random order. Gaussian blur and solarization are applied as the last step for the augmentation pipeline, where solarization inverts pixel intensities above a certain threshold. These changes help the model to learn features from across various augmented versions of the same image, ensuring that it learns robust, invariant representations [26].

2.1.3. DINO: Distillation with No Labels

Self Distillation with No Labels (DINO) [9] is a self-supervised learning (SSL) method that makes use of knowledge distillation ideas without the necessity of labeled data. It is based on earlier discriminative learning techniques and self-distillation processes. Unlike traditional contrastive SSL methods that require negative samples, such as SimCLR [16] and MoCo [18], DINO uses a non-contrastive approach that is similar to BYOL [26], but it does not require an additional predictor head. Like BYOL,

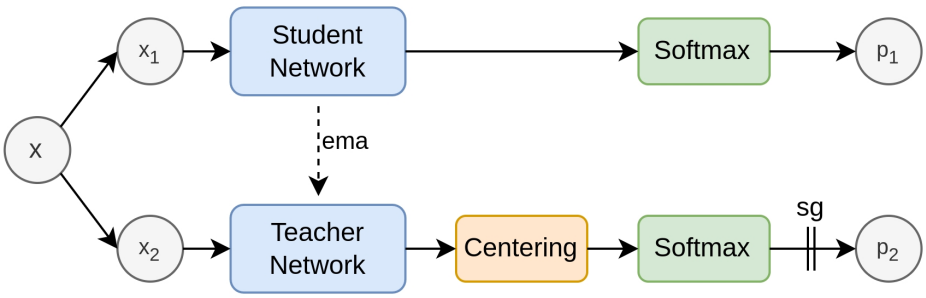

Figure 2.1. Overview of DINO architecture.

DINO also consists of two neural networks. DINO has a teacher-student architecture where both networks share the same architecture. The neural networks are composed of a backbone ViT [5] or ResNet [27]. The main idea is to use several augmented versions of the same input data to enable the student network to match the output representations of the teacher network. Using a momentum encoder [18], the student network learns to match the probability distribution of the teacher's output. Based on the exponential moving average (EMA) of the student, the teacher network updates

its parameters gradually instead of being trained via gradient descent. It differs from conventional knowledge distillation approaches [28], where a fixed pre-trained teacher guides a smaller student model.

DINO uses a multi-crop approach [29] to expose the model to several viewpoints of an image to enhance self-supervised learning. This method consists of two main crop strategies: global and local crops. While local crops make up a small portion of the original image (less than 50%), global crops make up more than half of the original image. While the student network is trained with both global and local crops, the teacher network uses just the global crops, resulting in encouraging local-to-global feature alignment [9]. It is demonstrated that this multi-crop enhances the feature correspondences. DINO also uses centering and sharpening operations for the teacher's outputs before calculating the cross-entropy loss. Sharpening keeps the model from collapsing into trivial solutions with a uniform distribution, while centering prevents the dominance of any single dimension, which is the opposite of sharpening [9].

A key advantage of DINO over other self-supervised learning methods is its ability to produce high-quality semantic representations, particularly when applied to Vision Transformers (ViTs) [5]. The self-attention maps of DINO-trained with ViT automatically highlight distinct objects in an image without any manual annotations. Such properties are neither observed in supervised ViTs, where training focuses on classification tasks, nor in CNN-based self-supervised models [9]. This suggests that combining self-supervised learning with the global attention mechanisms of transformers enables a richer and more structured understanding of visual scenes.

DINOv2 [12] is also one of the most preferred SSL methods for computer vision. DINOv2 brings some improvements over DINO by focusing on efficiency and robustness in self-supervised learning. DINOv2 ensures a better degree of generalization by filtering and rebalancing data from several sources using an automated data curating pipeline. This curated dataset (LVD-142M) [12] improves feature quality over standard datasets like ImageNet-1k [30]. DINOv2 also improves training stability and feature diversity with architectural refinements, including Sinkhorn-Knopp centering [29] and

KoLeo regularization [31]. DINOv2 demonstrates substantial performance gains in various downstream tasks, surpassing both DINO and previous self-supervised learning methods. DINOv2 also shows strong video classification capabilities like action recognition with Kinetics-400 [32], UCF-101 [33], and Something-Something v2 [34] datasets, despite being trained only on images, not videos.

2.2. Heatmap

Because it provides a good representation of motion dynamics, heatmap-based classification has evolved into one of the essential techniques in action recognition. Early techniques such as PoTion [35] introduced heatmap aggregation to encode pose motion, while later developments such as PoseConv3D [10] are based on 3D heatmap volumes to improve spatiotemporal feature learning. To improve the representation of human actions, models such as PA3D [36] and DynaMotion [37] used dynamic motion encoding and temporal pose convolutions. These techniques show how well heatmap-based representations capture detailed motion patterns, outperforming conventional coordinate-based models and enhancing RGB and optical flow modalities for more robust action identification.

2.2.1. PoTion

Pose motion representation (PoTion) [35] introduced an approach that encodes motion by aggregating pose heatmaps over time into a single compact representation. This method colorizes the aggregated heatmaps to preserve temporal ordering, allowing standard Convolutional Neural Networks (CNNs) to process motion information without requiring explicit temporal modeling like RNNs or LSTMs.

The PoTion framework consists of three main steps. In the first step, the model extracts joint heatmaps from each frame using a pose estimator, where the estimator shows the joint's estimated probability at each pixel. Then, it accumulates these heatmaps over time, where each pixel records the presence of a joint across frames. Finally, temporal order is encoded through colorization, where different color channels

(RGB) are assigned to different time steps. By transforming sequential pose information into a more compact image-like structure, PoTion enables CNN-based classification without requiring handcrafted temporal descriptors or recurrent models [35].

A key advantage of PoTion is its robustness for pose estimation errors. Unlike coordinate-based skeleton models that depend on precise joint locations, PoTion operates on heatmaps, making it less sensitive to noise in pose detection. Moreover, the colorization scheme effectively keeps motion cues, ensuring that the temporal sequence of actions is preserved within a compact spatial representation [35]. PoTion provides a robust baseline for heatmap-based action recognition by including an efficient method to represent pose motion in a compact form. Its capacity to translate raw pose sequences into CNN-friendly inputs eventually influenced field research.

2.2.2. PoseConv3D

Traditionally, pose-based action recognition has depended on skeletal representations formed via graph-based methods. These techniques lack interoperability with traditional convolutional neural network architectures. Furthermore, they often suffer from pose estimation errors. By coding human motion as a sequence of stacked joint heatmaps (3D heatmap volume) rather than coordinate-based skeleton graphs, PoseConv3D [10] addresses these issues. This method allows for the use of 3D convolutional networks (3D CNNs), therefore enhancing the capacity of the model to learn spatiotemporal relationships.

The PoseConv3D framework starts with 2D pose extraction, which extracts joint coordinates using a pose estimator (HRNet) [38]. After pose extraction, 2D poses are converted to 3D Heatmap volumes. First, a joint heatmap is created by aggregating Gaussian maps centered at every joint. These heatmaps are then stacked along the temporal dimension to create a 3D heatmap volume, which is used as input for a 3D CNN. However, two techniques are used to reduce the redundancy of 3D heatmap volumes before directly employing them. Heatmaps are cropped using the smallest bounding box covering all 2D poses over frames (Subjects-Centered Cropping). Then,

a subset of frames is chosen using Uniform Sampling [39] to lower the volume along the temporal dimension [10].

Leveraging the benefits of convolutional feature learning, PoseConv3D enables the network to learn spatiotemporal relationships directly from data, resulting in being more robust to pose estimation noise [10]. It differs from traditional graph convolutional networks (GCNs) [40], which explicitly define spatial connections between joints. PoseConv3D offers significant advantages in terms of scalability and fusion capabilities. Early fusion stages allow the 3D heatmap volume to be easily merged with other modalities, such as RGB or optical flow, providing great flexibility for multi-stream action recognition [10].

2.3. Sign Language Recognition

2.3.1. Hand-crafted Based Methods

Action recognition in videos has advanced significantly with the creation of strong feature descriptors. In early research for video-based action recognition, hand-crafted feature descriptions have become more popular. Among these, Histogram of Oriented Gradients (HOG) [41], Histogram of Optical Flow (HOF) [42], Motion Boundary Histograms (MBH) [43], and Improved Dense Trajectories (IDT) [44] have been widely used since their efficiency in capturing spatial and motion-related features of actions. Later, they were adapted for sign language recognition [11].

The Histogram of Orientated Gradients (HOG) [41] was initially introduced for object detection and then modified for action recognition and sign language recognition by extracting spatial information from image sequences. It calculates the distribution of gradient orientations in specific areas, creating a robust descriptor that encapsulates shape information. As a result, it is employed in sign language recognition by extracting discriminative features from the hands and faces of signers, which are critical for interpreting signs [11].

Histogram of Optical Flow (HOF) [42] descriptors are very similar to HOG [41] descriptors, but they extend by incorporating motion information using optical flow. It constructs histograms based on the distribution of motion differences between consecutive frames, providing a compact representation of temporal dynamics [42]. Because of their ability to capture movement dynamics well, they are commonly used for action recognition [42] and sign language recognition [11] in early research.

Motion Boundary Histograms (MBH) [43] descriptors are originally proposed for human detection. MBH is based on histograms of the derivatives of the horizontal and vertical components of optical flow [43]. It highlights motion boundaries and suppresses global motion effects successfully. This helps MBH to be effective in situations with camera motion. Therefore, MBH is one of the hand-crafted features that are used in action recognition [44] and sign language recognition [11].

Improved Dense Trajectory (IDT) [44] is an optical flow-based approach for tracking densely sampled points across frames. By lowering the effect of global scene motion, IDT includes camera motion adjustment, thus enhancing trajectory stability. IDT offers a dense representation of both appearance and motion by extracting several descriptors, including HOG, HOF, and MBH. This methodology has shown great performance in action recognition, surpassing several past hand-crafted techniques [44].

In BosphorusSign22k [11], IDT [44] has been used as a baseline for sign language detection, showing competitive performance against deep learning models, including 3D ResNets [27]. Principal Component Analysis (PCA) is applied for each component, and then the Gaussian Mixture Model (GMM) is used for feature clustering processes. As a final step, Fisher Vectors (FVs) [45] were computed for each component using PCA and GMM parameters. For the classification, Linear Support Vector Machines (SVMs) are used with different settings based on the concatenation of FVs outputs [11].

2.3.2. Deeplearning Based Methods

Sincan and Keles *et al.* [46] created a sign language recognition system based on 2D Convolutional Neural Networks (CNNs) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal modeling. Their methodology combines VGG16-based [47] CNNs to extract important spatial features from RGB frames, followed by unidirectional and bidirectional LSTMs that capture motion dynamics across frames. They use feature pooling techniques with a module called Feature Pooling Module (FPM) to improve temporal information and add an attention mechanism that targets the most informative frames in a sign sequence. Moreover, they evaluate multiple modalities to see the contributions of models. They get the best results with the CNN + FPM + BLSTM + Attention model, which is 62% accuracy with the balanced test set for Ankara University Turkish Sign Language Dataset (AUTSL) [46].

Joze and Koller [48] evaluated multiple deep learning architectures to build strong sign language recognition models, including 2D CNNs, LSTM networks, and 3D CNNs. The authors propose to extend 2D convolutional filters into 3D using the Inflated 3D (I3D) [49] architecture that facilitates effective spatiotemporal feature extraction from video sequences. The I3D model is pre-trained on the Kinetics human action dataset [32] and then fine-tuned on MS-ASL to accurately capture hand movements and facial expressions, which are crucial to convey sign meaning. Furthermore, they investigate the effectiveness of 2D CNNs (VGG16) with LSTMs, while LSTMs capture the temporal relationships of sign gestures. Their assessments indicate that I3D surpasses conventional 2D CNN-LSTM methods, achieving better results in signer-independent situations [48] in the RWTH-PHOENIX-Weather 2014 dataset.

Özdemir *et al.* [11] introduces 3D ResNet [27] with mixed 2D-3D convolutions to extract spatio-temporal features from Turkish Sign Language (TID) videos in the BosphorusSign22k dataset. For 3D convolution 2, two residual blocks are used, and for 2D convolution 3, two residual blocks are used. Then, a fully connected layer is used as a classification layer. This method is set as the second baseline for BosphorusSign22k.

However, it does not exceed the accuracy of Improved Dense Trajectories (IDT) [44], which remains a more efficient baseline for the BosphorusSign22k dataset [11].

Amorim [50] introduces a Spatial-Temporal Graph Convolutional Network (ST-GCN) [40] to represent sign language movements through skeleton models of signers. Their methodology develops a graph-based model wherein nodes represent key body joints, and edges show spatial and temporal relations among these joints over video frames. The skeletal data is obtained from the American Sign Language Lexicon Video Dataset (ASLLVD) [51] through a position estimation technique that monitors body joints during each signing session. Their findings indicate that ST-GCN proficiently captures the spatio-temporal dynamics of sign language, providing a solid foundation for skeletal-based sign recognition [50].

Özdemir [8] introduces a framework that combines Spatial-Temporal Graph Convolutional Networks (ST-GCNs) [40] with Multi-Cue Long Short-Term Memory Networks (MC-LSTMs) to analyze both manual and non-manual articulators for isolated sign language recognition. To learn spatio-temporal dependencies, the ST-GCN is trained on upper body and hand skeleton joints, which make a total of 35 unique joints. Pre-trained CNNs extract features from hand shape with DeepHand [14] and facial expression cues via Distract Your Attention Network (DAN) [13]. The MC-LSTM introduces an adaptive fusion mechanism that dynamically determines the contribution of each visual cue at each time step. The designed MC-LSTM is an extension of Multi-View LSTMs (MV-LSTMs) [52]. The proposed model is assessed on BosphorusSign22k [11] and AUTSL [46]. It shows the integration of various visual cues is important for recognition accuracy [8].

2.3.3. Sign Language Datasets

Every Deaf community has developed a distinct sign language like the spoken language, like British Sign Language (BSL) and Turkish Sign Language (TSL). Consequently, an SLR model for a particular language must be trained using a dataset of that sign language (SL). The type of annotation used in the dataset is one of the

important characteristics of SL datasets, in addition to its language. Isolated sign language datasets possess gloss-level labels, whereas continuous sign language datasets contain sentence-level labels [53].

Sincan and Keles *et al.* [46] published the isolated AUTSL dataset. The dataset consists of 226 Turkish Sign Language signs executed by 43 signers, recorded with RGB videos at 512×512 resolution. It is split into a signer-independent split. AUTSL includes dynamic backgrounds and moving objects, making it a challenging benchmark for isolated sign recognition.

Joze and Koller [48] introduced the isolated MS-ASL dataset, which consists of 25,513 annotated videos and 1,000 unique American Sign Language (ASL) signs, performed by 222 signers.

Forster *et al.* [54] introduced the RWTH-PHOENIX-Weather 2014 dataset. It is a continuous German Sign Language dataset that was collected from a German TV station's weather forecast program. It was recorded from nine native signers, with a resolution of 210×260 pixels at 25 frames per second (fps).

Özdemir *et al.* [11] introduced the BosphorusSign22k dataset. The dataset is recorded with RGB videos at 1920×1080 resolution at 30 frames per second (fps). The dataset has 22,542 video clips and 744 Turkish sign glosses.

3. METHODOLOGY

Human Action Recognition (HAR) denotes a procedure for classifying human movements. Both Human Activity Recognition (HAR) and Sign Language Recognition (SLR) depend on the detection of human motion and gestures, indicating a significant methodological correlation between these domains. A fundamental problem in both fields is the effective extraction and interpretation of spatiotemporal movement patterns, which is crucial for differentiating between various actions or signals. Many successful methods have been proven in the action recognition domain, and they can be adapted to the sign language domain. We adapted DINO and heatmap techniques originally used in the field of action recognition to sign language recognition (SLR). Sign language mainly relies on hand gestures, upper body motions, and facial expressions to convey meaning. As a result, a good sign language recognition model should examine every cue related to a sign.

In this chapter, we introduced an SLR framework, illustrated in Figure 3.1. It starts with the preprocessing of sign language videos and continues with cue representation. These cue representations are extracted for the spatial dynamics of hands, faces, and upper body pose. Then, the cue representations are given to a temporal modeling layer to learn the temporal dynamics of signers' actions.

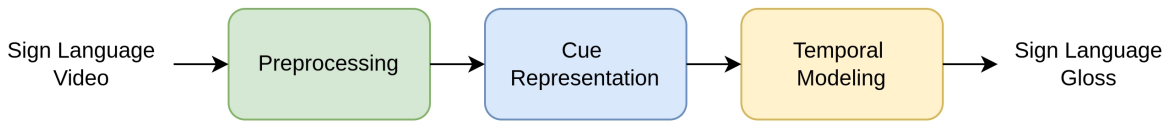

Figure 3.1. Our proposed framework for isolated SLR.

3.1. Preprocessing

To improve computing efficiency, we aimed to use only the most informative parts of the body. The initial step involves extracting key regions, which are hands and faces

because signs are closely represented by hand shapes and facial expressions. We crop the RGB frames according to pose predictions derived from OpenPose [55]. OpenPose provides 25 keypoints for the body, 21 keypoints for each hand, and 70 keypoints for the face. The face area is cropped by creating a square bounding box that contains all detected facial keypoints. The same idea is applied to both hands. The crop size is set to 256×256 pixels. The extracted hand and face regions will subsequently be used for feature extraction. By extracting face and hand regions, we help our feature extractor focus on the most informative areas for SLR rather than using the whole frame.

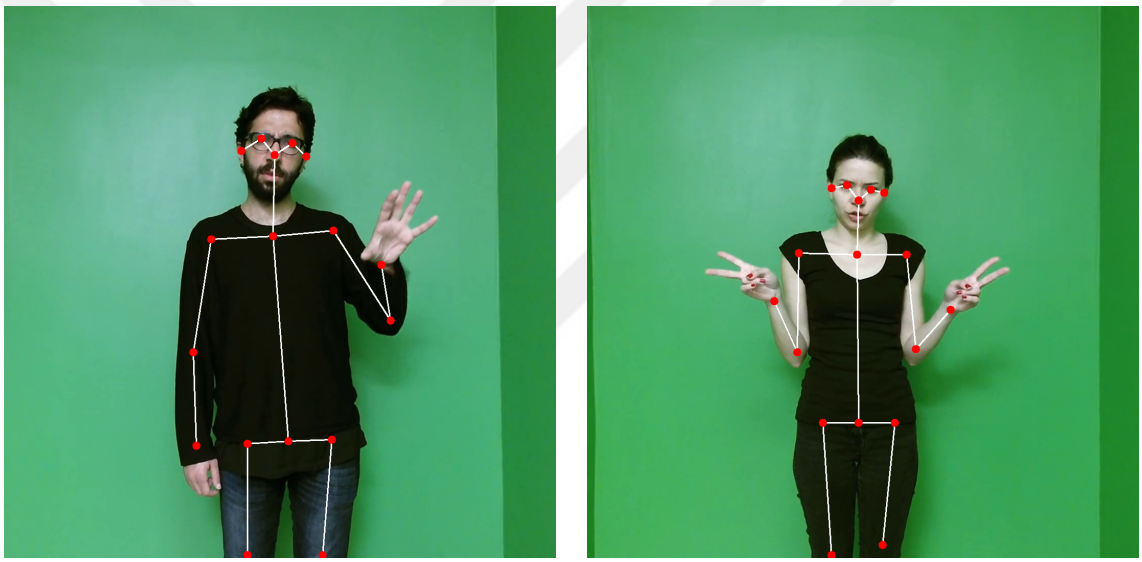

Figure 3.2. OpenPose upper body poses on BosphorusSign22k dataset.

Our methodology focuses on temporally dense regions by establishing an active window corresponding to the time segment where the dominant hand displays movement. In our research dataset (BosphorusSign22k) [11], the left hand is the dominant hand; hence, active frames were identified by observing the movement of the signers' left hand. A frame is considered active when the left hand exceeds a specified vertical threshold within the space between the hips and nose. To optimize feature extraction efficiency, stationary frames are eliminated from the network input, keeping just those within the specified active frames. By isolating inactive frames, the methods enhance the system's robustness while reducing computational overhead. As a result, we improved the overall efficiency of the recognition system.

Processing every frame from the video sequences generates much computational load. Moreover, redundant frames hardly help recognition since the movement across consecutive frames is minimal. To mitigate these issues, we used a frame sampling technique, choosing one out of every two frames. Several studies were carried out to determine the ideal sample frequency, which led to the selection of a sampling frequency of 2. This strategy not only reduces the computational burden but also enhances the efficiency of the system.

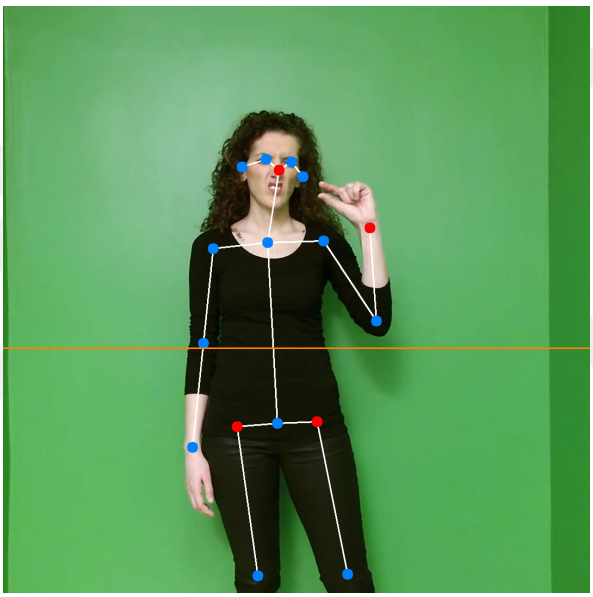

Figure 3.3. OpenPose upper body poses with a horizontal threshold (orange) on BosphorusSign22k dataset.

3.2. Cue Representations

Sign language relies on hand gestures, upper body movements, and facial expressions to convey meaning. Therefore, for cue representation, we focused on hand shapes, facial semantics, and upper body pose. Each of these features contributes differently to recognizing signs.

For hand, face, and upper body pose feature extraction, we used a pre-trained self-supervised learning framework called DINOv2 [12], which is an improved version of Self-Distillation with No Labels (DINO) [9]. DINOv2 implements a teacher-student

architecture where both networks have the same architecture. The main idea of DINOv2 is to be trained with different augmented versions of the same input data to allow the student network to align with the probability distribution of the teacher's output. Only student parameters are updated during back-propagation. The teacher network's parameters are updated based on the exponential moving average (EMA) of the student network's parameters [9].

DINOv2 uses a multi-crop augmentation [29] to enhance self-supervised learning. This approach includes two main crop strategies, which are called global and local crops. Local crops contain a minor part of the original image (around 50%), but global crops comprise more than half of the original image. The student network is trained using both global and local crops, whereas the teacher network uses just global crops [9].

One of the main advantages of using DINOv2 as a feature extractor is that DINOv2 is pre-trained on Vision Transformers (ViT) [5] features. Vision Transformers utilize self-attention throughout the image, enabling the capture of long-range dependencies. This design enables DINOv2 to learn meaningful visual representations with both global context and fine details. There are several publicly available pre-trained DINOv2 versions based on the architecture of ViTs. The embedding dimensions for ViT-S, ViT-B, ViT-L, and ViT-g are 384, 768, 1024, and 1536 respectively [12]. In our models, we used different sizes of ViT to see the effect of ViTs.

Table 3.1. Embedding dimensions for different ViT models.

Model	**Embedding Dimension**
ViT-S	384
ViT-B	768
ViT-L	1024
ViT-g	1536

To visualize embeddings, we use patches of ViT architecture. First, we calculate PCA for the patches of the images. Then, we normalized the values based on min-

max scaling; the data is scaled to a range between 0 and 1. After the normalization, the threshold is applied to the first PCA to remove the background. We define the threshold value as 0.5. After thresholding, we applied a second PCA. Then, we assign three different colors (red, green, and blue) to the first three components of the second PCA. This visualization is proposed in [12]. The visualization of a left hand, right hand, and face is shown in Figure 3.4.

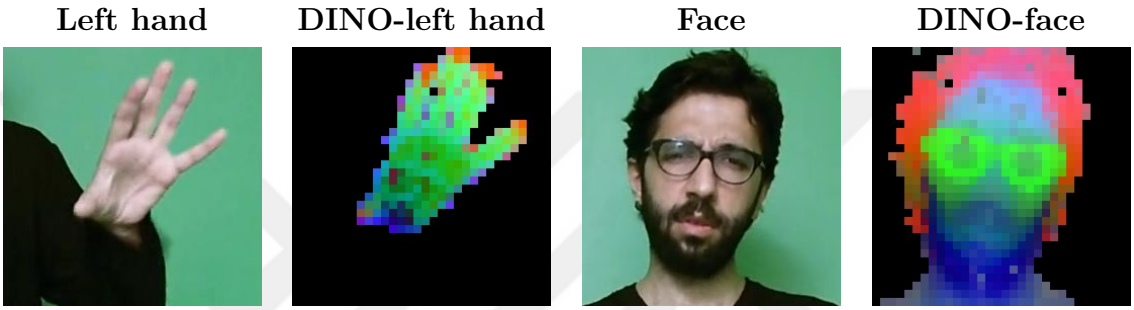

Figure 3.4. Feature visualization of the first PCA components of DINO.

In the context of DINOv2, intermediate ViT layers have demonstrated the ability to capture rich positional and semantic information. Early layers of DINOv2 carry more positional information than the last layer, while deeper layers contain less positional and more semantic information [56]. This paper [56] used intermediate layers for estimating the object pose from image-to-model 2D-3D correspondences. As one of our extraction methods, we use some intermediate layers of DINOv2 to see how both positional and semantic information contribute to recognizing sign language.

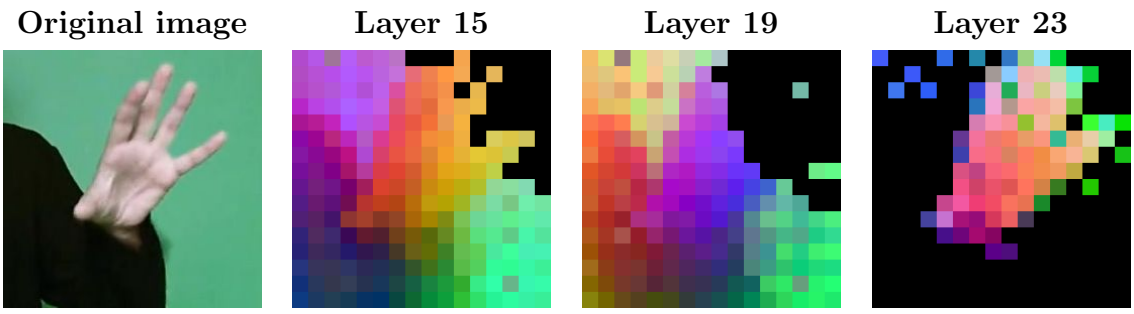

Figure 3.5. Feature visualization with PCA components across different layers after applying threshold as 0.22.

DINOv2 is trained over the LVD-142M dataset [12], which is a large-scale, curated dataset designed to improve self-supervised learning in computer vision by providing a good balance of different types of images. However, our DINOv2 feature extraction method focuses only on hand and face images. Therefore, we continue training for pre-trained DINOv2 to analyze if we can get better DINOv2 features. Our aim was to get two specialized DINOv2 networks for hands and faces. We continued to train DINOv2 with both hand images and created the DINOv2-H model [57]. We applied the same idea for faces and produced the DINOv2-F model [57]. In our experiments, we used the DINOv2-H and DINOv2-F models to extract features from hand and face images.

In addition to DINOv2 for feature extraction, we used a pre-trained DeepHand [14] model for hand feature extraction. DeepHand is based on CNN with Hidden Markov Models (HMMs). The framework was trained on over one million hand images sourced from various publicly available sign language datasets, Danish sign language [58], New Zealand sign language [59], and RWTH-PHOENIX-Weather 2014 [54]. The DeepHand model extracts 1024-dimensional feature maps from hands. In our experiments, we extracted 1024-dimensional hand features from the BosphorusSign22k [11] dataset via DeepHand [14].

Besides DINOv2, we used a pre-trained Distract Your Attention Network (DAN) [13] model for facial feature extraction. DAN is a deep learning architecture developed for Facial Expression Recognition (FER). There are publicly available three pre-trained models based on the datasets: AffectNet [60], RAF-DB [61], and SFEW 2.0 [62]. We used a DAN model pre-trained with AffectNet as a feature extractor for facial images. The dimension of DAN features is 512 [13].

The feature extraction pipeline includes DINOv2, DeepHand, and DAN techniques to generate hand and face representations. However, upper body pose is also crucial in sign language recognition. To extract information from upper body poses, we adapt the PoseConv3D [10] mechanism to recognize sign language. First, we extracted 25 keypoints for the body via OpenPose [55]. However, not all body keypoints have

an effect on sign language representation. Therefore, we select upper body keypoints starting from the hips, which make a total of 13 keypoints. Each keypoint consists of locations (x, y) and confidence score (c). Following keypoint extraction, 2D postures are transformed into 2D heatmap volumes. A joint heatmap is generated by combining Gaussian maps centered on each joint. The Gaussian maps are calculated as follows:

$$\mathbf{J}_{kij} = exp(-\frac{(i - x_k)^2 + (j - y_k)^2}{2\sigma^2}) * c_k, \tag{3.1}$$

σ regulates the variance of Gaussian maps, (x_k, y_k) denotes the position of the k joint, and c_k represents the confidence score of the k joint [10]. The limb heatmap can also be created with this formula:

$$\mathbf{L}_{kij} = exp(-\frac{\mathcal{D}((i, j), \text{seg}[a_k, b_k])^2}{2\sigma^2}) \cdot \min(c_{a_k}, c_{b_k}). \tag{3.2}$$

The k limb is located between two joints, a_k and b_k. The function $\mathcal{D}$ computes the distance from the point (i, j) to the segment defined by the endpoints $[(x_{a_k}, y_{a_k}), (x_{b_k}, y_{b_k})]$ [10].

If the architecture has a 2D CNN model, the heatmaps are directly used as input for a 2D CNN. For the 3D CNN architecture, after creating heatmaps for each frame, these heatmaps are stacked along the temporal dimension to form a 3D heatmap volume. Then, this 3D heatmap volume is fed into a 3D CNN. The dimension of the heatmap is defined as 64×64 pixels.

DINOv2 is also used to extract upper body pose cue representation as an alternative to heatmap-based approaches for upper body pose. The upper body is defined based on the 13th OpenPose key points, similar to the heatmap-based approach.

We used offline feature extraction for all feature extraction methods to reduce computational load during inference. By processing features in advance, we minimized redundant computations. It helps us examine different modalities with faster execution.

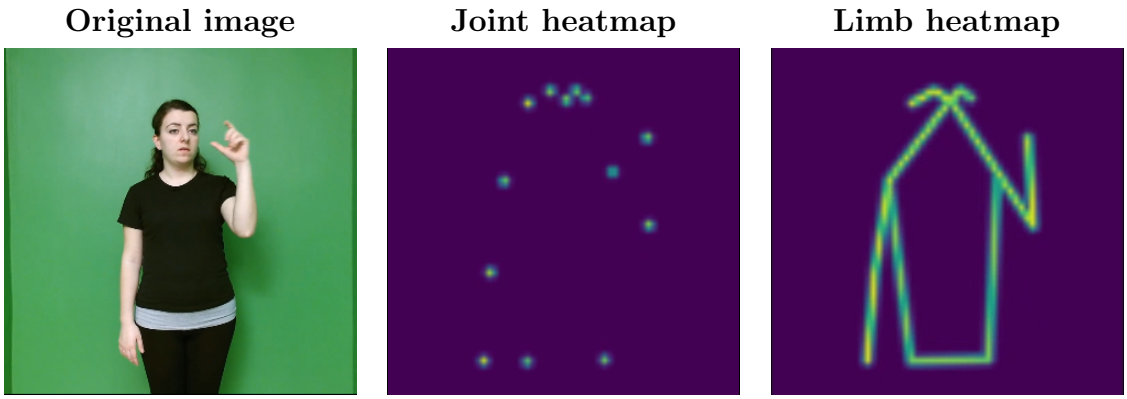

Figure 3.6. Heatmap Visualization. Left to right; original image, joint heatmap, limb heatmap.

3.3. Temporal Modeling

Based on the feature extraction methods, we defined different models. To see the contribution of each feature extraction method, we first create single modality architectures. Modalities focus on visual cues of body parts (hands and faces) and upper body poses (heatmaps) separately. Then, to achieve the full potential of features, we fuse different backbones. By combining multiple modelings, we aimed to use all valuable information available, like hand shapes, facial expressions, and upper body poses.

3.3.1. Body Part-based Modeling

As a first model, we introduced a model called Cue Sequential Network (CueSeqNet). Only body parts (hands and/or face) are used as input for our network. We aimed to maintain simplicity while capturing all essential body parts crucial for sign language recognition. We used the DINOv2 model to extract left hand, right hand, and face representations. In addition to DINOv2, DeepHand feature extraction is applied to the left hand and right hand to get hand representations. We also applied the DAN model to analyze facial expressions.

Based on the experiment settings, cue representations of body parts were concatenated for each active frame in the video. Then, these dense cue representations were fed into the LSTM model to get temporal information. As a final step, we introduced a Multi-Layer Perceptron (MLP) classification header on top of the LSTM to classify signs.

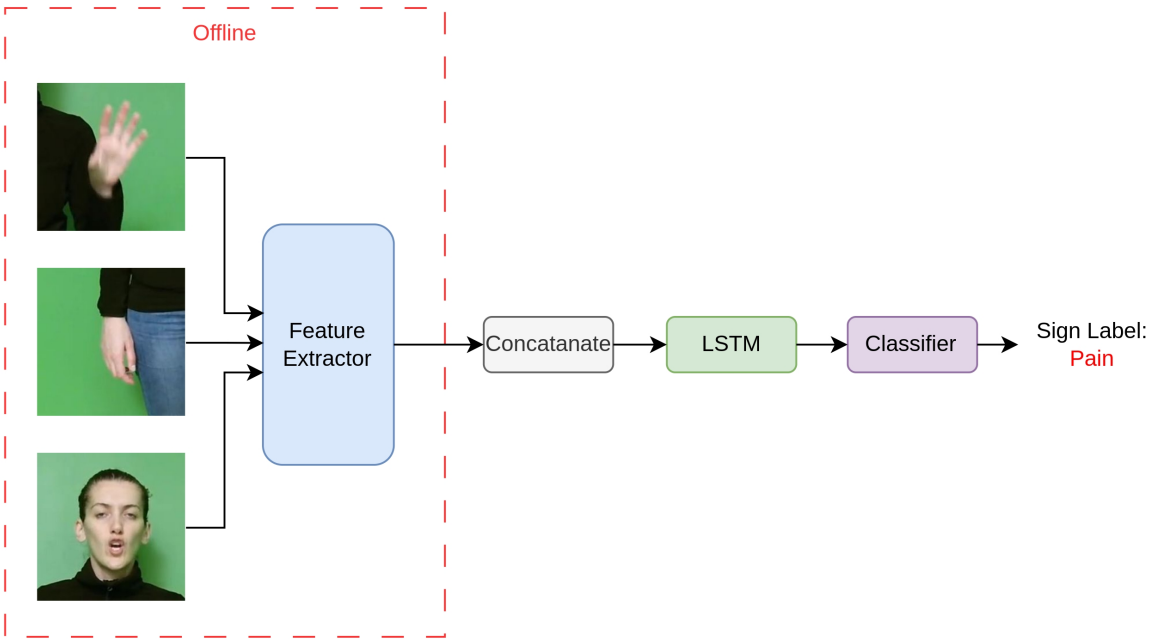

Figure 3.7. CueSeqNet: Body part-based modeling with LSTM.

3.3.2. Heatmap Based Modeling

In the second model, we shifted our focus from hand shape and facial expressions to upper body gestures, as they play a crucial role in understanding sign language. Our proposed models process upper body gestures via heatmaps. For each active frame, heatmaps were created based on upper body keypoints extracted via the OpenPose pose estimator. To create heatmaps, we select keypoints located above the hips, which is 13 out of 25 total body keypoints.

We created two different models to process heatmaps: Heatmap Pose 2D (HeatPose2D) and Heatmap Pose 3D (HeatPose3D). In the HeatPose2D model, we used a pre-trained 2D CNN (ResNet18) model to analyze heatmaps for each frame in the

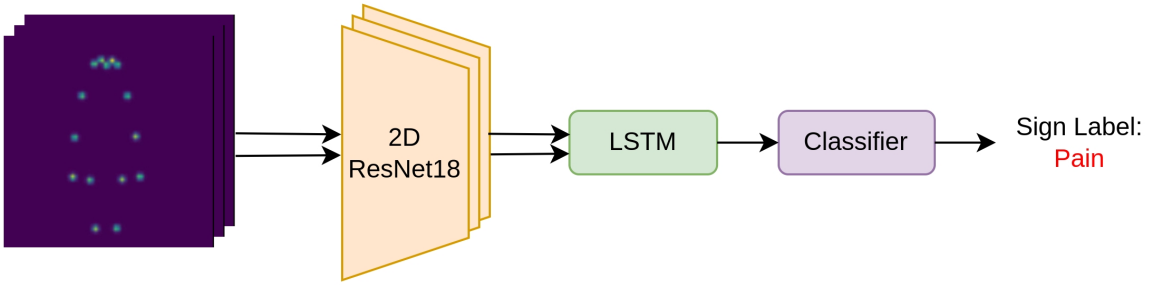

Figure 3.8. HeatPose2D: Heatmap based modeling with 2D CNN.

videos. Via 2D CNN models, spatial information of upper body pose is extracted. This representation is then fed into the LSTM model to capture upper body movements over time. As a final step, an MLP classification header was added after LSTM to classify signs. As a second heatmap modeling (HeatPose3D), we employed a 3D CNN (3D ResNet18) model instead of a 2D CNN with LSTM as proposed in [10]. For HeatPose3D, heatmaps were stacked for each active frame to create a 3D heatmap volume. Then, a 3D CNN model was trained with these 3D heatmap values to learn spatiotemporal relations for upper body pose. Finally, an MLP classifier is used to detect signs.

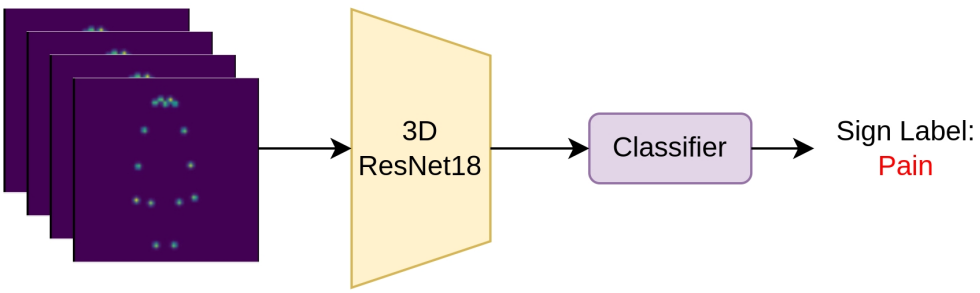

Figure 3.9. HeatPose3D: Heatmap based modeling with 3D CNN.

3.3.3. Mixed Modeling

Sign language conveys meaning through a combination of hand movements, facial expressions, and upper body gestures. Therefore, in the third modeling, we created more complex modeling, a combination of multiple modalities. We combined body part-based modeling with heatmap-based modeling. We arranged two different

setups for mixed modeling, Cue Heatmap 2D (CueHeat2D) and Cue Heatmap 3D (CueHeat3D).

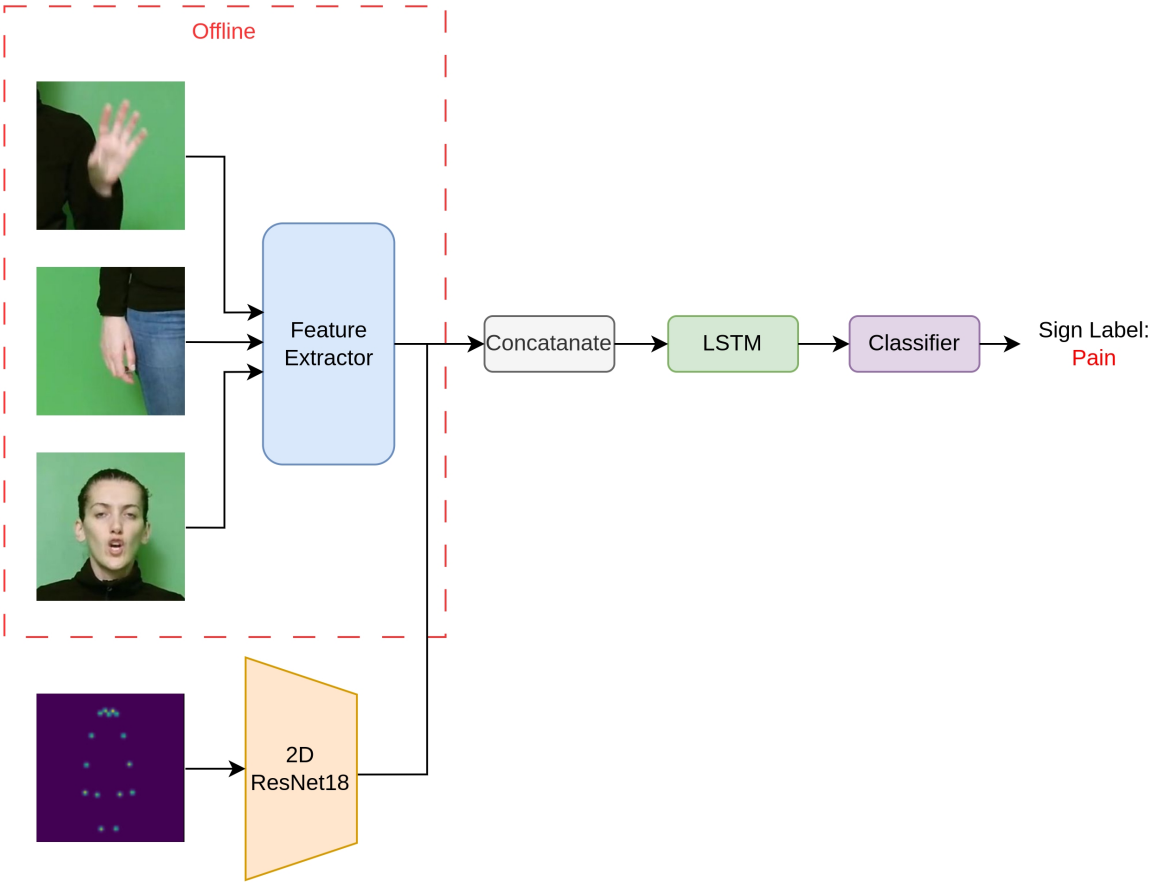

Figure 3.10. CueHeat2D: Mixed modeling with 2D CNN and LSTM.

In the first setup with CueHeat2D, visual cues are directly used, while heatmaps are processed with a 2D CNN to extract upper body pose information. Upper body pose information was extracted via a 2D CNN. The extracted upper body pose information was then concatenated with visual cues and fed into the LSTM network to learn movement. Lastly, an MLP classifier was employed for sign detection.

In the second setup with CueHeat3D, the idea is similar to the first setup, except using 3D CNN instead of 2D CNN. Visual cues for body part features are processed via the LSTM network, while upper body pose information is analyzed via 3D CNN. The visual cue representations are concatenated and then processed with the LSTM network. However, pre-calculated heatmaps are processed directly with 3D CNN to

get spatiotemporal information. As a late fusion of these two modalities, outputs are concatenated and fed into an MLP classifier for the final result. This architecture is our main proposed model called CueHeat3D.

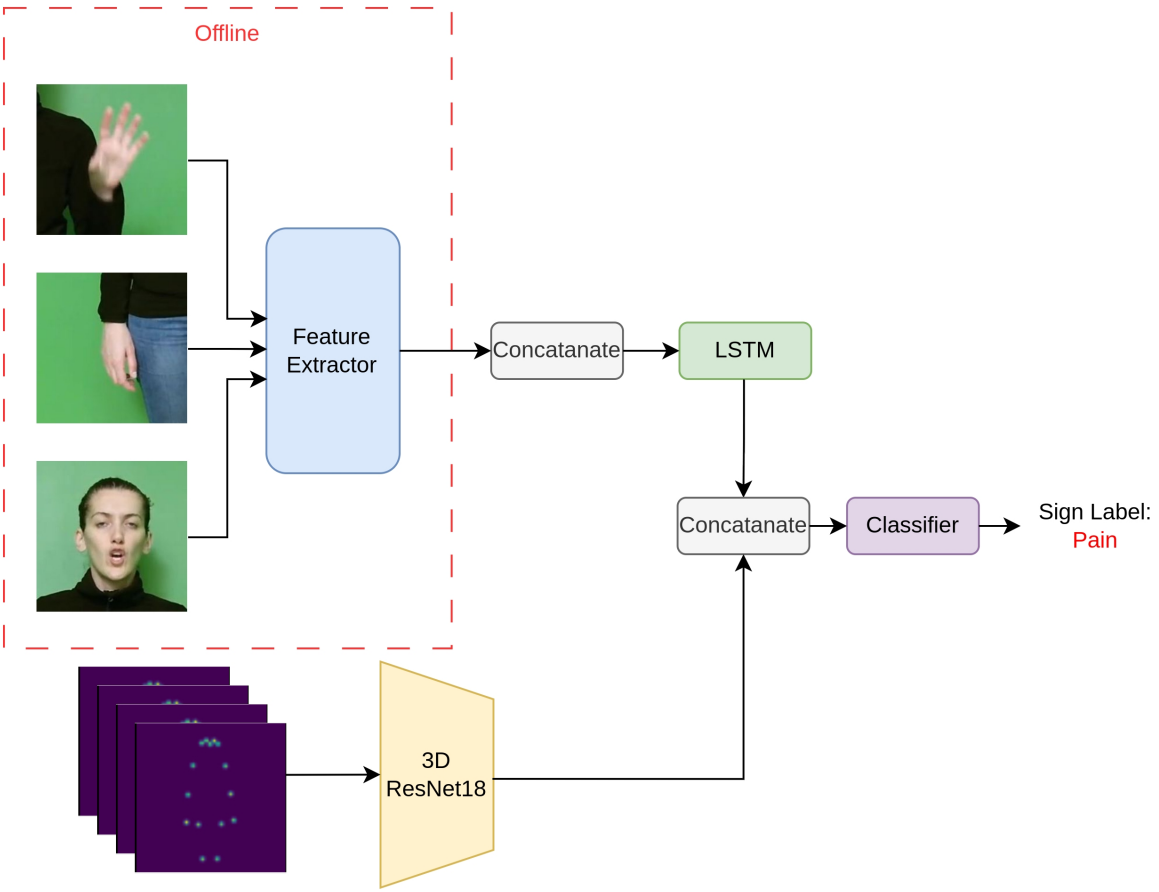

Figure 3.11. CueHeat3D: Mixed modeling with 3D CNN and LSTM.

4. EXPERIMENTS AND RESULTS

4.1. Datasets

In this study, we used the BosphorusSign22k [11] dataset, which is a large-scale Turkish Sign Language dataset designed for sign language recognition research. This dataset was built on top of the earlier BosphorusSign [63] dataset. The BosphorusSign22k dataset contains 22,542 video samples that cover 744 unique Turkish sign glosses. There are a total of six native signers, four female and two male. The signs can be split into three main categories: general, health, and finance [11].

The dataset consists of RGB videos with 1920×1080 resolution at 30 frames per second (fps). OpenPose and Kinect v2 skeleton information are provided with the dataset. The database contains approximately two million frames. There are 25 3D keypoints within Kinect v2 body pose information [11]. OpenPose has 25 2D keypoints for body pose, 21 keypoints for each hand, and 70 keypoints for facial landmarks. These additional annotations make the dataset useful for multi-modal learning.

To ensure robust evaluation, the dataset is separated in a signer-independent manner, where the test set consists of videos from a single signer, while the remaining five signers are included in the training set. The training subset contains 18,018 videos, whereas the test subset includes 4,524 videos. This setup ensures that the system learns to recognize signs independent of the individual performing them [11].

4.2. Evaluation Metrics

We evaluated the performance of our proposed architectures using Top-1 and Top-5 accuracy. Top-1 accuracy measures the percentage of correctly predicted classes, while Top-5 accuracy considers a prediction correct if the ground truth appears in the top five ranked outputs. These metrics provide insights into classification performance.

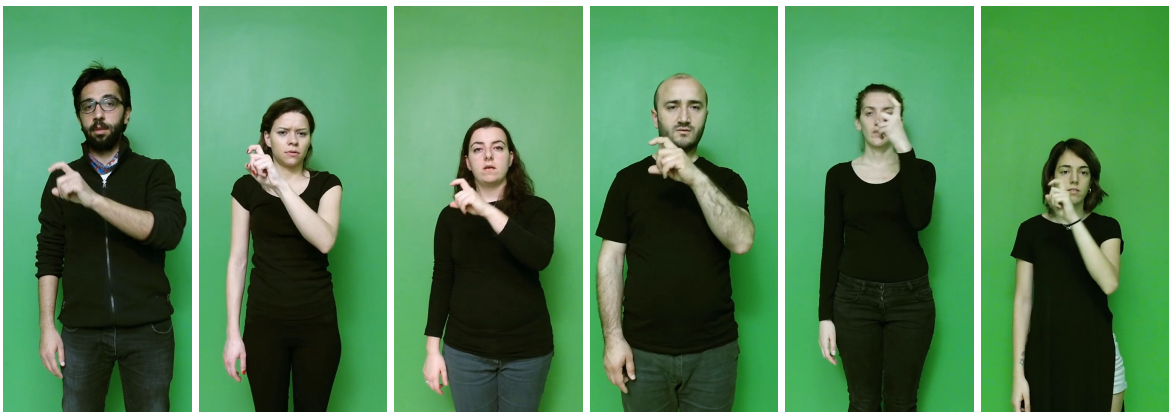

Figure 4.1. Signer from BosphorusSign22k for sign 'MONTHLY'.

The formula for Top-1 accuracy is expressed as follows:

$$\text{Top-1 accuracy} = \frac{\text{number of correct predictions}}{\text{total number of samples}}. \tag{4.1}$$

4.3. Implementation

For our experiments, we used a computer with an Intel i7-11800H processor (2.30Ghz) and an NVIDIA RTX 3060 graphics card (6 GB VRAM). The Ubuntu-Linux operating system was the Ubuntu-Linux for the setups. We used the AdamW optimizer with a learning rate (lr) of 0.00005 for the training phase. Learning rate scheduling was implemented with a step size of 5 and a gamma factor of 0.5. Additionally, a weight decay of 0.01 was introduced to regularize the model. With a batch size of 8, the training process ran for 20 epochs to get balanced computational efficiency with practical learning.

PyTorch library is used to construct models and conduct training for all architectures. Details of each component are listed below.

- Feature extraction: All features are extracted offline via DAN, DeepHand, and DINOv2 so that the training can be conducted more efficiently. While using extracted features, we sampled every two frames (the frame frequency as 2).

- LSTM backbone: LSTM networks within our models use a 2-layer LSTM architecture with 1024 hidden dimensions. As a dropout value, 0.1 Is used. After conducting some tests, we found that there is not much relation between bidirectional LSTM (BLSTM) and performance, so we decided to move on with unidirectional LSTM.
- 2D CNN Backbone: We used a pre-trained version of ResNet-18 for our 2D CNN backbone. However, to work with the heatmaps, we changed the first CNN layer for ResNet-18 to work with 1-dimensional images. To fuse with other modalities, we removed the last fully connected layer.
- 3D CNN Backbone: We used pre-trained version for 3D ResNet-18 for our 3D CNN backbone. However, the first layer of CNN has been changed so that it can be compatible with 1-dimensional heatmaps. In a similar way with 2D CNNs, we remove the last fully connected layer to fuse the 3 CNN model with other modalities. 3D ResNet-18 has a global average layer, so it could be used for different lengths in the temporal axis.
- Classifier: For the classification head, we used a Multi-Layer Perceptron (MLP) after dropout with a probability of 0.1

4.4. Results

We conducted multiple experiments to investigate the impact of each visual cue representation with different modalities. All experiments were tested on the BosphorusSign22k [11] dataset. To examine the contributions, we divided the experiments into various segments.

4.4.1. CueSeqNet

In the first set, we conducted an experiment with CueSeqNet, which is described in detail in Section 3.3.1 with Figure 3.10. Frame frequency, active frames, cue representation methods, pre-training, and intermediate layers of DINOv2 were investigated.

4.4.1.1. Frame Frequency. As a first experiment, we focused on experiments for sampling the frequency of frames in the videos. For this purpose, CueSeqNet model is tested with DeepHand and DINO left hand cue representations. Table 4.1 shows experiments with different frame frequencies. Even if there are no significant changes in the accuracy, sampling one out of every two frames gives better results with the DINO left hand cue representation but slightly less with the combination of DINO and DeepHand cue representations. We selected a sampling frequency of 2 for our proposed methods based on the experiment results.

Table 4.1. Accuracy (%) results for CueSeqNet with left hand cue representation and different frame sampling frequencies.

Left Hand Features	**Frequency**	**Training**	**Top-1**	**Top-5**
DINOv2	1	98.47	74.54	94.01
DINOv2	2	98.47	**80.57**	96.42
DINOv2	4	98.12	79.55	**96.64**
DINOv2	6	96.73	78.01	95.65
DeepHand and DINOv2	1	99.47	82.56	96.97
DeepHand and DINOv2	2	98.67	85.39	97.88
DeepHand and DINOv2	4	99.33	**85.48**	**98.17**
DeepHand and DINOv2	6	98.70	84.22	97.70

4.4.1.2. Active Frame. In the BosphorusSign22k dataset [11], there are videos of various lengths. The signer's actions generally do not start directly with the first frame of the video. An active frame extraction method is implemented to detect stationary frames (see Section 3.1). We tested our CueSeqNet model with left hand DINOv2 cue representation before and after active frame filtering to see the effect of stationary frames. As shown in Table 4.2, eliminating stationary frames and keeping only active frames increased performance by 27.81%.

Table 4.2. Accuracy (%) results for CueSeqNet with and without active frame filtering for left hand DINOv2 cue representation.

Filtering	**Training**	**Top-1**	**Top-5**
No Filtering	94.14	52.76	80.61
Active Frame	98.47	80.57	96.42

4.4.1.3. Cue Representations. We performed tests to see the effects of DINOv2, DeepHand, and DAN feature extraction methods with the combination of different body parts (left hand, right hand, and face). In order to see the contribution of each cue representation of body part and feature extraction method, CueSeqNet is used. The results of cue representation experiments are shown in Table 4.3. When we check the result for both DINOv2 and DeepHand features, the experiments with left hand features have much higher accuracy than with right hand features. The main reason is that for the videos within the BosphorusSign22k dataset [11], the left hand moves for all signs, while the right hand moves only for two-handed signs. For both left and right hand cue representation, DINOv2 outperforms the DeepHand model, which was trained over 1 million hand images [14]. In contrast, DINOv2 was trained for general purpose, not specifically for hands or faces.

Our experiments have shown that combining DINOv2 and DeepHand left hand cue representation gives much higher performance than individual features. It means that even though DINOv2 and DeepHand features are based on the same region (left hand), they focus on different types of cue representation. Furthermore, in terms of facial cues, even DINOv2 outperforms DAN; DAN performs better when integrated with other visual cues. Figure 3.4 shows that DINOv2 focuses on characteristics like hair and glasses rather than facial expressions. To get the best cue representations with CueSeqNet, DeepHand left hand, DINOv2 left and right hand, and DAN face representations are concatenated and end up with 88.04% Top-1 accuracy.

Table 4.3. Accuracy (%) results for CueSeqNet with different body part cue representations. L: Left hand and R: Right hand.

Features	**Dimension**	**Training**	**Top-1**	**Top-5**
DeepHand (L)	1024	98.80	76.57	94.14
DeepHand (R)	1024	67.37	45.49	63.70
DINOv2 Hand (L)	1024	98.47	80.57	96.42
DINOv2 Hand (R)	1024	67.21	51.11	67.75
DINOv2 Face	1024	72.05	23.92	45.65
DAN Face	512	67.61	17.26	33.21
DeepHand (L) and DINOv2 Hand (L)	2048	98.67	85.39	97.88
DINOv2 Hand (L + R) with DINOv2 Face	3072	99.62	78.47	96.09
DeepHand (L + R) with DAN Face	2560	99.96	80.59	95.78
DeepHand (L) and DINOv2 Hand (L + R) with DINOv2 Face	4096	99.46	86.89	98.50
DeepHand (L) and DINOv2 Hand (L + R) with DAN Face	3584	99.94	**88.04**	**98.89**

4.4.1.4. DINO Pre-trained on Body Parts. DINOv2 is trained on large amounts of data within LDM-162 for general purposes [12]. To increase the performance of DINOv2 for sign language recognition, we trained two different DINOv2 networks, which are DINOv2-H and DINOv2-F, as described in Section 3.1. Moreover, different setups are prepared with two different sizes of DINOv2, small and large. Based on the experiment in Table 4.4, continuing to train DINOv2 with hand images increases the accuracy of the CueSeqNet. However, when we check Table 4.5 for large DINOv2, this pre-training process affects left and right hand representation performance conversely. It decreased right hand and face accuracy. Creating DINOv2-H increased training accuracy drastically; however, Top-1 test accuracy decreased because when we checked the visualization of DINOv2 features in Figure 3.4, it focused on characteristics of the face rather than facial expressions.

Table 4.4. Accuracy (%) results for CueSeqNet with pre-trained small DINOv2 cue representation.

Small DINOv2 Features	**Dimension**	**Training**	**Top-1**	**Top-5**
Left hand	384	97.93	75.35	94.47
Left hand pre-trained	384	99.97	**81.65**	**96.86**
Right hand	384	66.19	46.04	63.88
Right hand pre-trained	384	78.84	**50.20**	**65.94**
Face	384	72.55	**22.28**	**44.47**
Face pre-trained	384	80.09	22.15	40.21

4.4.1.5. Intermediate Layers. The early layers of DINOv2 have more positional information and less semantic, whereas the last layers have more semantic information, as discussed in Section 3.2. We investigated the effect of semantic and positional information on sign language recognition using the feature representation from different layers of DINOv2. In these settings, the CueSeqNet model is used with small DINOv2 left hand cue representations. Small DINOv2, which is based on the ViT-S backbone, has

Table 4.5. Accuracy (%) results for CueSeqNet with pre-trained large DINOv2 cue representation.

Large DINOv2 Features	**Dimension**	**Training**	**Top-1**	**Top-5**
Left hand	1024	98.86	80.57	96.42
Left hand pre-trained	1024	99.97	**83.80**	**96.84**
Right hand	1024	67.21	**51.11**	**67.75**
Right hand pre-trained	1024	73.46	49.73	64.72
Face	1024	72.05	**23.92**	**45.65**
Face pre-trained	1024	97.63	21.33	39.63

a total of 12 layers. As can be seen in Table 4.6, the last layer (layer 12) has better results than the intermediate layers. Furthermore, the combinations of the intermediate and last layers were investigated to see the effect of including both positional and semantic information. However, there is an increase (0.53%) in accuracy when layer 9 and layer 12 are combined. We conclude that semantic information is more valuable than positional information in sign language recognition for a cropped hand region.

4.4.2. HeatPose2D and HeatPose3D

4.4.2.1. Heatmaps. In our proposed methods for heatmaps in Section 3.2, there are two kinds of heatmap extraction methods: limb and joint. To process these heatmaps, we introduced two different models, which are called HeatPose2D and HeatPose3D (see Section 3.3.2). Experiments in Table 4.3 show that, only with hand and face cues, it is possible to get high performance (88.04%). However, upper body pose is also one of the crucial parts of sign language recognition. Experiments in Table 4.7 have shown that joints give better results for both HeatPose2D and HeatPose3D modalities. Therefore, we chose joint heatmaps as upper body pose representation. This experiment also shows that HeatPose3D gives better results than HeatPose2D. In the HeatPose3D model, 3D CNN can capture spatial and temporal relations together, unlike HeatPose2D. In the HeatPose2D model, 2D CNN focuses on spatial features,

Table 4.6. Accuracy (%) results for CueSeqNet with small DINOv2 intermediate layer cue representation.

Layers	**Training**	**Top-1**	**Top-5**
Layer 5	49.80	19.87	46.77
Layer 7	71.72	35.12	66.22
Layer 9	80.41	53.18	82.23
Layer 10	84.95	58.42	85.72
Layer 11	84.26	54.69	83.27
Layer 12	97.93	75.35	94.47
Layer 9 and Layer 12	97.77	**75.88**	94.78
Layer 10 and Layer 12	97.99	75.81	94.94
Layer 11 and Layer 12	97.77	75.59	**95.11**
Layer 5, Layer 9 and Layer 12	78.84	41.58	74.51
Layer 7, Layer 9 and Layer 12	82.06	46.97	77.59

while LSTM focuses on temporal relations separately. It helps the HeatPose3D model to extract better upper body representation.

As an alternative to joint and limb heatmaps, DINOv2 upper body visual cues are also used in our tests. For this setting, the CueSeqNet model is used with a frame sampling frequency of 4. For all other test cases, the frame frequency is set as 2. Table 4.8 shows results for DINOv2 and joint heatmap for upper body pose.

Table 4.7. Accuracy (%) results for HeatPose2D and HeatPose3D models with limb and joint heatmaps.

Type	**Model**	**Training**	**Top-1**	**Top-5**
Joint	HeatPose2D	97.31	64.08	89.35
Limb	HeatPose2D	91.13	60.04	87.16
Joint	HeatPose3D	99.91	**79.09**	**95.62**
Limb	HeatPose3D	99.89	77.06	95.40

Table 4.8. Accuracy (%) results for HeatPose3D model with heatmaps and CueSeqNet model with DINOv2 upper body cue representation.

Cue Type	**Model**	**Training**	**Top-1**	**Top-5**
DINOv2	CueSeqNet	95.54	44.36	71.51
Heatmap	HeatPose3D	99.91	**79.09**	**95.62**

4.4.3. CueHeat2D and CueHeat3D

4.4.3.1. Mixed Modeling. As described in Section 3.3.3, we introduced two different models called CueHeat2D and CueHeat3D. We conduct more advanced experiments to get the best combination of cue representation and models. Based on the results from our experiments in Table 4.9, dense left hand representation with DeepHand and DINOv2 gives better cue representation. Furthermore, CueHeat3D outperforms CueHeat2D and CueSeqNet models because of the ability to learn better spatiotemporal features. We got the best accuracy when we combined DeepHand left hand, DINOv2 left hand, DINOv2 right hand, and DAN face cue representations with CueHeat3D modeling, which is a 91.71% Top-1 score.

Table 4.9. Accuracy (%) results for different mixed models with various visual cue representations. L: Left hand and R: Right hand.

Hand Features	**Face Feature**	**Heatmap Type**	**Model**	**Dimension**	**Train Acc**	**Top-1**	**Top-5**
DINOv2 (L + R)	DINOv2	-	CueSeqNet	1x3072	98.81	78.47	96.09
DINOv2 (L + R)	DINOv2	2D Heatmap	CueHeat2D	1x3072 + 1x64x64	99.10	77.21	95.65
DINOv2 (L + R)	DINOv2	3D Heatmap	CueHeat3D	1x3072 + Fx64x64	99.92	86.65	98.34
DeepHand (L) + DINOv2 (L + R)	DINOv2	-	CueSeqNet	1x4096	99.46	86.89	98.50
DeepHand (L) + DINOv2 (L + R)	DINOv2	2D Heatmap	CueHeat2D	1x4096 + 1x64x64	99.88	87.20	98.70
DeepHand (L) + DINOv2 (L + R)	DINOv2	3D Heatmap	CueHeat3D	1x4096 + Fx64x64	99.98	90.78	99.05
DeepHand (L) + DINOv2 (L + R)	DAN	-	CueSeqNet	1x3584	99.94	88.04	98.89
DeepHand (L) + DINOv2 (L + R)	DAN	2D Heatmap	CueHeat2D	1x3584 + 1x64x64	99.95	89.46	98.89
DeepHand (L) + DINOv2 (L + R)	DAN	3D Heatmap	CueHeat3D	1x3584 + Fx64x64	99.99	**91.71**	**99.43**

4.4.4. Discussion

The experimental results show that using a sampling frequency of 2 improves accuracy by 6.03% for the DINOv2 left hand representation. Moreover, eliminating stationary frames and keeping just active frames improved performance by 27.81%.

For hands cue representation, DINOv2 gives better performance than the DeepHand model. Combining DINOv2 and DeepHand for left-hand cue representation shows a significant accuracy increase of 4.82% compared to separate representation performance, showing that these models have a complementary effect on each other. Regarding facial cue representations, even though DINOv2 outperforms DAN with single cue representation, DAN shows higher accuracy when combined with other visual cues. CueSeqNet achieves a Top-1 accuracy of 88.04% using solely hand and face cue representation. It demonstrates the importance of hand and facial features in SLR.

Furthermore, pre-training the DINOv2 model separately for hand (DINOv2-H) and face (DINOv2-F) improves left hand recognition accuracy while decreasing performance for the right hand and face recognition.

The early layers of DINOv2 have more positional and less semantic information, while the last layers show more semantic information. The last layer (layer 12) shows better performance than the earlier layers. However, when layers 9 and 12 are concatenated, accuracy increases by 0.53%.

Joint heatmaps perform better than limb heatmaps in terms of recognition accuracy. Furthermore, 3D CNNs show better results than 2D CNNs with LSTM, showing that 3D CNNs capture better spatiotemporal relations in SLR.

The best recognition accuracy (91.71% Top-1) was obtained when DeepHand left hand, DINOv2 two hands, and DAN face cue representations were combined with 3D CNN heatmap modeling. Table 4.10 summarizes the most successful methods. The results of other studies on BosphorusSign22k dataset are shown in Table 4.11

Table 4.10. Accuracy (%) results for our successful methods.

DINO			DeepHand	DAN	Heatmap	Top-1	Top-5
Left Hand	Right Hand	Face	Left Hand	Face	Upper Body	(%)	(%)
✓	-	-	✓	-	-	85.39	97.88
✓	✓	✓	✓	-	2D	87.20	98.70
✓	✓	-	✓	✓	-	88.04	98.89
✓	✓	✓	✓	-	3D	90.78	99.05
✓	✓	-	✓	✓	3D	**91.71**	**99.43**

Table 4.11. Comparison with other studies on the BosphorusSign22k dataset.

Author	**Method**	**Top-1(%)**	**Top-5(%)**
Özdemir [11]	MC3	78.85	94.76
Kındıroğlu [64]	TAF	81.37	97.47
Gökçe [65]	MC3 + ST Sampling	86.91	98.17
Özdemir [11]	IDT	88.53	-
Çemkinli [66]	TSM	92.46	99.42
Özdemir [8]	ST-GCN + MC-LSTM	92.58	99.07
Sincan and Keleş [67]	I3D + RGB-MHI Fusion	94.83	-
Gökçe [65]	MC3 + ST Sampling + SF	**94.94**	**99.76**
Proposed Method	CueHeat3D	91.71	99.43

5. CONCLUSION

Deaf individuals use sign language as the primary communication method in their daily lives. Sign language mainly relies on hand gestures, upper body motions, and facial expressions to convey meaning. Therefore, SLR models should be able to learn the spatiotemporal dynamics of the hands, face, and upper body of signers.

In this thesis, we worked on self-supervised learning and heatmap methods to improve recognition accuracy for the Turkish isolated SLR problem. Our proposed methods focus on extracting various cue representations from hands, face, and upper body to create temporal modeling to predict glosses on the BosphorusSign22k dataset.

Various preprocessing methods, such as active frame extraction and frame sampling, were implemented for videos on BosphorusSign22k to improve performance. We adapted self-supervised learning DINO and heatmap techniques originally used in action recognition to SLR. The suggested approach uses DINO to concentrate on learning visual cues from hands and faces via self-supervised learning. At the same time, heatmap-based modeling is used to get visual cues of upper body posture. We used the LSTM layer with 2D CNN and 3D CNN separately in different setups to learn temporal information of visual cues. In addition to the DINO method, we investigate alternative techniques for extracting visual cues from hands and faces. For hand representation, we used DeepHand, while DAN was employed for facial feature extraction.

We analyzed various cue representations and temporal modeling to evaluate the impact of each component on the BosphorusSign22k SLR dataset. The results from our experiments enabled us to draw some conclusions. For both left and right hand cue representations, DINOv2 outperforms the DeepHand model. However, combining DINOv2 with DeepHand left hand cue representation results in better performance compared to individual features. Furthermore, for facial expression, even DINOv2 outperforms DAN; DAN works better when combined with other visual clues. Moreover, 3D CNN can learn better spatiotemporal relations than 2D CNN with LSTM. Ad-

ditionally, pre-training the DINO model for hands and faces separately improves left hand recognition performance while decreasing right hand and face performance.

Based on our experiments, combining DeepHand left hand, DINOv2 two hands, and DAN facial cue representations with CueHeat3D modeling resulted in the highest accuracy (91.71% Top-1) score. When it is compared to state-of-the-art methods, our proposed method gives competitive results in the BosphorusSign22k dataset.

As a future work, various enhancements can be applied to the model architecture. Firstly, upper body temporal dynamics can be learned using 3D CNN algorithms other than 3D ResNet-18, such as I3D, because upper body movement is one of the critical visual cues in SLR. Secondly, pre-training of DINO can be done more optimally because proper training of DINO specifically for hands and faces can increase recognition performance.

REFERENCES

1. Hu, L., L. Gao, Z. Liu and W. Feng, "Continuous Sign Language Recognition with Correlation Network", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2529–2539, Vancouver, BC, Canada, 2023.

2. Zuo, R., F. Wei and B. Mak, "Natural Language-assisted Sign Language Recognition", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14890–14900, Vancouver, BC, Canada, 2023.

3. Cheok, M. J., Z. Omar and M. H. Jaward, "A Review of Hand Gesture and Sign Language Recognition Techniques", *International Journal of Machine Learning and Cybernetics*, Vol. 10, No. 1, pp. 131–153, 2019.

4. Camgöz, N. C., A. A. Kındıroğlu and L. Akarun, "Sign Language Recognition for Assisting the Deaf in Hospitals", *Human Behavior Understanding: 7th International Workshop, HBU 2016*, pp. 89–101, Springer, Amsterdam, The Netherlands, 2016.

5. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale", *arXiv Preprint arXiv:2010.11929*, 2020.

6. Al-Qurishi, M., T. Khalid and R. Souissi, "Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues", *IEEE Access*, Vol. 9, pp. 126917–126951, 2021.

7. Li, R. and L. Meng, "Multi-View Spatial-Temporal Network for Continuous Sign Language Recognition", *arXiv Preprint arXiv:2204.08747*, 2022.

8. Özdemir, O., İ. M. Baytaş and L. Akarun, "Multi-Cue Temporal Modeling for

Skeleton-Based Sign Language Recognition", *Frontiers in Neuroscience*, Vol. 17, p. 1148191, 2023.

9. Caron, M., H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, Montreal, QC, Canada, 2021.

10. Duan, H., Y. Zhao, K. Chen, D. Lin and B. Dai, "Revisiting Skeleton-Based Action Recognition", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2969–2978, New Orleans, LA, USA, 2022.

11. Özdemir, O., A. A. Kındıroğlu, N. Camgoz and L. Akarun, "BosphorusSign22k Sign Language Recognition Dataset", *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, Marseille, France, 2020.

12. Oquab, M., T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "DinoV2: Learning Robust Visual Features without Supervision", *arXiv Preprint arXiv:2304.07193*, 2023.

13. Wen, Z., W. Lin, T. Wang and G. Xu, "Distract Your Attention: Multi-Head Cross Attention Network for Facial Expression Recognition", *Biomimetics*, Vol. 8, No. 2, p. 199, 2023.

14. Koller, O., H. Ney and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3793–3802, Las Vegas, NV, USA, 2016.

15. Doersch, C., A. Gupta and A. A. Efros, "Unsupervised Visual Representation Learning by Context Prediction", *Proceedings of the IEEE International Confer-*

ence on Computer Vision, pp. 1422–1430, Santiago, Chile, 2015.

16. Chen, T., S. Kornblith, M. Norouzi and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations", *International Conference on Machine Learning (ICML)*, pp. 1597–1607, Vienna, Austria, 2020.

17. Kim, S., S. Kang, F. Bu, S. Y. Lee, J. Yoo and K. Shin, "Hypeboy: Generative Self-Supervised Representation Learning on Hypergraphs", *arXiv Preprint arXiv:2404.00638*, 2024.

18. He, K., H. Fan, Y. Wu, S. Xie and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, Seattle, WA, USA, 2020.

19. Oord, A. v. d., Y. Li and O. Vinyals, "Representation Learning with Contrastive Predictive Coding", *arXiv Preprint arXiv:1807.03748*, 2018.

20. Zhang, R., P. Isola and A. A. Efros, "Colorful Image Colorization", *European Conference on Computer Vision (ECCV)*, pp. 649–666, Springer, Amsterdam, The Netherlands, 2016.

21. Gidaris, S., P. Singh and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations", *arXiv Preprint arXiv:1803.07728*, 2018.

22. He, K., X. Chen, S. Xie, Y. Li, P. Dollár and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, New Orleans, LA, USA, 2022.

23. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, MN, USA, 2019.

24. Feichtenhofer, C., Y. Li, K. He *et al.*, "Masked Autoencoders As Spatiotemporal Learners", *Advances in Neural Information Processing Systems*, Vol. 35, pp. 35946–35958, 2022.

25. Madan, N., N.-C. Ristea, K. Nasrollahi, T. B. Moeslund and R. T. Ionescu, "CL-MAE: Curriculum-Learned Masked Autoencoders", *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2492–2502, Waikoloa, Hawaii, USA, 2024.

26. Grill, J.-B., F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning", *Advances in Neural Information Processing Systems*, Vol. 33, pp. 21271–21284, 2020.

27. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

28. Hinton, G., O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network", *arXiv Preprint arXiv:1503.02531*, 2015.

29. Caron, M., I. Misra, J. Mairal, P. Goyal, P. Bojanowski and A. Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", *Advances in Neural Information Processing Systems*, Vol. 33, pp. 9912–9924, 2020.

30. Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, Vol. 115, No. 3, pp. 211–252, 2015.

31. Sablayrolles, A., M. Douze, C. Schmid and H. Jégou, "Spreading Vectors for Similarity Search", *arXiv Preprint arXiv:1806.03198*, 2018.

32. Kay, W., 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.

33. Soomro, K., A. R. Zamir and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild", *arXiv Preprint arXiv:1212.0402*, 2012.

34. Goyal, R., S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The "Something Something" Video Database for Learning and Evaluating Visual Common Sense", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5842–5850, Venice, Italy, 2017.

35. Choutas, V., P. Weinzaepfel, J. Revaud and C. Schmid, "PoTion: Pose Motion Representation for Action Recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7024–7033, Salt Lake City, UT, USA, 2018.

36. Yan, A., Y. Wang, Z. Li and Y. Qiao, "PA3D: Pose-Action 3D Machine for Video Recognition", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7922–7931, Long Beach, CA, USA, 2019.

37. Asghari-Esfeden, S., M. Sznaier and O. Camps, "Dynamic Motion Representation for Human Action Recognition", *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 557–566, Snowmass, CO, USA, 2020.

38. Sun, K., B. Xiao, D. Liu and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, Long Beach, CA, USA, 2019.

39. Wang, L., Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang and L. Van Gool, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–36, Springer, Amsterdam, The Netherlands, 2016.

40. Yan, S., Y. Xiong and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, New Orleans, USA, 2018.

41. Dalal, N. and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 886–893, IEEE, San Diego, CA, USA, 2005.

42. Laptev, I., M. Marszalek, C. Schmid and B. Rozenfeld, "Learning Realistic Human Actions from Movies", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, Anchorage, AK, USA, 2008.

43. Dalal, N., B. Triggs and C. Schmid, "Human Detection Using Oriented Histograms of Flow and Appearance", *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision*, pp. 428–441, Springer, Graz, Austria, 2006.

44. Wang, H. and C. Schmid, "Action Recognition with Improved Trajectories", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558, Sydney, Australia, 2013.

45. Sánchez, J., F. Perronnin, T. Mensink and J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice", *International Journal of Computer Vision*, Vol. 105, No. 3, pp. 222–245, 2013.

46. Sincan, O. M. and H. Y. Keles, "AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods", *IEEE Access*, Vol. 8, pp. 181340–181355, 2020.

47. Simonyan, K. and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv Preprint arXiv:1409.1556*, 2014.

48. Joze, H. R. V. and O. Koller, "MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language", *arXiv Preprint arXiv:1812.01053*, 2018.

49. Carreira, J. and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, Honolulu, HI, USA, 2017.

50. de Amorim, C. C., D. Macêdo and C. Zanchettin, "Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition", *International Conference on Artificial Neural Networks*, pp. 646–657, Springer, Munich, Germany, 2019.

51. Neidle, C., A. Thangali and S. Sclaroff, "Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus", *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, Language Resources and Evaluation Conference (LREC)*, Istanbul, Turkey, 2012.

52. Rajagopalan, S. S., L.-P. Morency, T. Baltrusaitis and R. Goecke, "Extending Long Short-Term Memory for Multi-View Structured Learning", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 338–353, Springer, Amsterdam, The Netherlands, 2016.

53. Adaloglou, N., T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakas, D. Papazachariou and P. Daras, "A Comprehensive Study on Deep Learning-Based Methods for Sign Language Recognition", *IEEE Transactions on Multimedia*, Vol. 24, pp. 1750–1762, 2021.

54. Forster, J., C. Schmidt, O. Koller, M. Bellgardt and H. Ney, "Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather",

LREC, pp. 1911–1916, Reykjavik, Iceland, 2014.

55. Cao, Z., G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 43, No. 1, pp. 172–186, 2019.

56. Örnek, E. P., Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster and T. Hodan, "FoundPose: Unseen Object Pose Estimation with Foundation Features", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 163–182, Springer, Milan, Italy, 2024.

57. Gueuwou, S., X. Du, G. Shakhnarovich and K. Livescu, "SignMusketeers: An Efficient Multi-Stream Approach for Sign Language Translation at Scale", *arXiv Preprint arXiv:2406.06907*, 2024.

58. Kristoffersen, J. H. and T. Troelsgård, "The Danish Sign Language Dictionary", *Proceedings of the 14th Euralex International Congress*, pp. 1549–1554, Leeuwarden, Netherlands, 2010.

59. McKee, R. L. and D. McKee, "Making an Online Dictionary of New Zealand Sign Language: Projects", *Lexikos*, Vol. 23, No. 1, pp. 500–531, 2013.

60. Dhall, A., R. Goecke, S. Lucey and T. Gedeon, "Collecting Large, Richly Annotated Facial-Expression Databases From Movies", *IEEE Multimedia*, Vol. 19, No. 3, pp. 34–41, 2012.

61. Li, S., W. Deng and J. Du, "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2852–2861, Honolulu, HI, USA, 2017.

62. Dhall, A., R. Goecke, S. Lucey and T. Gedeon, "Static Facial Expression Analysis in Tough Conditions: Data, Evaluation Protocol and Benchmark", *IEEE In-*

ternational Conference on Computer Vision Workshops (ICCV Workshops), pp. 2106–2112, IEEE, Barcelona, Spain, 2011.

63. Camgöz, N. C., A. A. Kındıroğlu, S. Karabüklü, M. Kelepir, A. S. Özsoy and L. Akarun, "BosphorusSign: A Turkish Sign Language Recognition Corpus in Health and Finance Domains", *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pp. 1383–1388, Portorož, Slovenia, 2016.

64. Kindiroglu, A. A., O. Ozdemir and L. Akarun, "Temporal Accumulative Features for Sign Language Recognition", *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1288–1297, IEEE Computer Society, Seoul, South Korea, 2019.

65. Gökçe, Ç., O. Özdemir, A. A. Kındıroğlu and L. Akarun, "Score-Level Multi Cue Fusion for Sign Language Recognition", *Computer Vision – ECCV Workshops*, pp. 294–309, Springer, Glasgow, United Kingdom, 2020.

66. Çelimli, A. F., O. Özdemir and L. Akarun, "Attention Modeling with Temporal Shift in Sign Language Recognition", *30th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, Malatya, Turkey, 2022.

67. Sincan, O. M. and H. Y. Keles, "Using Motion History Images with 3D Convolutional Networks in Isolated Sign Language Recognition", *IEEE Access*, Vol. 10, pp. 18608–18618, 2022.