Student Declaration of Authorship

Course code and name:	F20PA - Research Methods and Requirements Engineering - 2024-2025
Type of assessment:	**Individual**
Coursework Title:	Year 4 Dissertation
Student Name:	Shehryar Naeem
Student ID Number:	H00409539

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Shehryar Naeem*

Date: *27/03/2025*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Pakistani Word-level Sign Language Recognition Based on Deep Spatiotemporal Network

Author Shehryar Naeem

BSc (Hons.) Computer Science
Year 4 Dissertation

Supervised by Dr. Md Azher Uddin

Heriot-Watt University
School of Mathematical and Computer Sciences

March 2025

DECLARATION

I, Shehryar Naeem, confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Shehryar Naeem

Date: 27/03/2025

ABSTRACT

Sign language is a key factor in facilitating effective communication within deaf and hard-of-hearing groups, one of the primary sources of visual and gesture-based communication. Despite its significance, Pakistan Sign Language (PSL) research, particularly at the word level, has limited attention compared to well-studied sign languages such as American Sign Language (ASL) and Chinese Sign Language (CSL). The unavailability of standardized PSL datasets and robust spatiotemporal methods with the capability to accurately model the intricate motion patterns of signs has hindered progress in PSL recognition.

To address these research gaps, we propose a video-based deep spatiotemporal method tailored for PSL word recognition from dynamic video data. We initially employ a top-K key frame extraction method to choose the most informative frames. We then separately extract spatial and temporal features: we use ResNet-101 to extract spatial information, and temporal dynamics between frames are encoded by using an enhanced Motion Binary Pattern (MBP) handcrafted descriptor combined with Local Binary Pattern (LBP) histograms. The spatial and temporal features are combined and fed into a Transformer-based model with Convolutional Neural Network (CNN) layer and positional encoding for final gesture classification. Comprehensive experimental evaluations conducted on the challenging PkSLMNM dataset—a PSL video dataset with large variability among multiple participants—demonstrate that our framework achieves state-of-the-art results.

Keywords: Pakistan Sign Language (PSL), Pakistan Sign Language Recognition (PSLR), Word Level Sign Language, Spatial-Temporal Network, Feature Extraction, Global Average Pooling (GAP), ResNet-101, Motion Binary Pattern (MBP), Local Binary Pattern (LBP), Transformers, Convolutional Neural Network (CNN), Top-K Key Frame Extraction

ACKNOWLEDGEMENTS

I would like to thank from the bottom of my heart all those who were with me every step of the way. To my family, cats, and irreplaceable friends—thank you for your faith in me and being my constant support.

I would especially like to thank my supervisor, Dr. Azher, whose unwavering guidance was the cornerstone on which my dissertation experience turned. His relentless advice helped me overcome many hurdles and pushing through complex ideas with confidence and clarity. His mentorship extended beyond academics, subtly shaping how I approach problems in life, and for that, I am truly grateful.

Ami jan and Abu jan, thank you for keeping the importance on studies and creating the environment in which I was able to grow and concentrate without any distractions. Your love and encouragement have been the unperceivable power in all the achievements I have made and will make. To my lovely sisters, Aimen and Ayesha, thank you for never letting anyone disturb me and for loving me—your concern meant the whole world to me. And to the cats, my little pillows and shots of serotonin, thank you for always being there.

To all my friends in general and the Moogiwaras in particular, thanks for the laughter, love, loyalty, and giving me a place to look forward to in my time here. Special thanks to my closest and dearest friend Ishana—your friendship, presence in my life, and encouragement have all been nothing but a blessing.

Lastly, I would like to thank all the professors and university staff at Heriot-Watt for the opportunity to study, develop and exhibit my skills.

Table of Contents

List of Figures

List of Tables

ABBREVIATIONS

2D-CNN 2D-Convolutional Neural Networks. 5, 12

AP Average Precision. 26, 27
ArSL Arabic Sign Language. 3, 8, 12
ASL American Sign Language. iii, 3, 10, 12

Bi-GRU Bidirectional Gated Recurrent Unit. 28, 32, 34
Bi-LSTM Bidirectional Long Short-Term Memory. 6, 8, 11, 12, 28, 32, 34
BOVW Bag of Visual Words. 12
BOW Bag of Words. 4, 6

C3D Convolutional 3D. 5, 12
CNN Convolutional Neural Network. iii, 5, 9, 10, 12, 16, 32, 33
CPU Central Processing Unit. 23
CSL Chinese Sign Language. iii
CSOM Convolutional Self-Organising Map. 8, 12

DCT Discrete Cosine Transform. 8, 12
DDR4 Double Data Rate Fourth Generation. 23

Fps Frames Per Second. 24

GAP Global Average Pooling. iii, vii, ix, 17, 20, 29, 30, 32, 33, 37
GCN Graph Convolutional Network. 10, 12
GDPR General Data Protection Regulation. 48
GPU Graphics Processing Unit. 23, 28
GRU Gated Recurrent Unit. ix, 7, 12, 28, 29, 32, 34

HMM Hidden Markov Model. 8, 12
HOG Histogram of Oriented Gradients. 8, 12

I3D Two-Stream Inflated 3D ConvNet. 3, 5, 9, 11, 12, 35
IMUs Inertial Measurement Units. 38
ISL Indian Sign Language. 3, 5, 6, 8, 10, 12

KSL Korean Sign Language. 3, 10, 12

LBP Local Binary Pattern. iii, ix, 13–16, 29, 30, 32, 37

LBP-TOP Local Binary Patterns from Three Orthogonal Planes. 15

LSTM Long Short-Term Memory. ix, 5, 7–10, 12, 28, 32, 34

mAP Mean Average Precision. viii, 26, 27, 35, 37

MBP Motion Binary Pattern. iii, ix, 1, 11, 13–16, 18, 19, 22, 29, 30, 32, 35, 37, 43

PADEAF Pakistan Association of the Deaf. 1

PDPA Pakistan's Personal Data Protection Act. 48

PkSLMNM Pakistan Sign Language Manual and Non-Manual. iii, ix, xi, 2, 3, 10, 12, 13, 17, 22, 24, 25, 48

PLES Professional, Legal, Ethical, and Social. 2

PSL Pakistan Sign Language. iii, ix, 1, 3, 4, 10–13, 16, 18, 20, 23, 24, 26, 31, 33, 37, 38, 43, 48

PSLR Pakistan Sign Language Recognition. iii, 1, 26, 32, 34, 37, 38

RAM Random Access Memory. 23, 28

ReLU Rectified Linear Unit. 20, 28

RGB Red Blue Green. 17

RNN Recurrent Neural Network. 19

RPN Region Proposal Network. 3, 12

SIFT Scale-Invariant Feature Transform. 6, 12, 15

SSD Solid State Drive. 23

SURF Speeded-Up Robust Features. 4, 12

SVM Support Vector Machine. 4, 5, 12, 30, 34

TCN Temporal Convolution Networks. 8

TGCN Temporal Graph Convolutional Network. 9, 12

TSM Temporal Shift Module. 5, 12

VGG-19 Visual Geometry Group. 6, 12

VLBP Volume Local Binary Patterns. 15

VRAM Virtual Random Access Memory. 23

WHO World Health Organization. 1

WSL Windows Subsystem for Linux. 28

1 INTRODUCTION

Sign language is an essential form of communication for the deaf and hard-of-hearing communities. In 2021, World Health Organization (WHO) declared that around 430 million people suffer from moderate to severe hearing loss [Organization et al. 2021]. Sign language makes use of both manual and non-manual gestures where the former include hands and signs, the latter however uses the upper body and facial expressions [Sameena Javaid 2023]. Several different regions have their own sign languages, each with unique dialects and grammatical nuances [Kaur et al. 2023]. Sign language recognition has become an essential field of study to enhance communication accessibility.

In this field, there are two primary methods: sensor-based and vision-based. In sensor-based method sensors are physically attached to users to record position, motion as well as trajectories of fingers and hand data like seen done via an armband made by Shin et al. [2017]. In vision-based approach there have been research done with static signs which utilize images [Shah et al. 2023; Singla et al. 2024] and dynamic which uses video data as input. The key distinction between sensor-based and vision-based approaches is how data is gathered and preprocessed [Cheok et al. 2019].

According to Pakistan Association of the Deaf (PADEAF), there are around 250000 individuals in Pakistan who have hearing disabilities and their primary form of communication is Pakistan Sign Language (PSL) [PADEAF [n. d.]]. Not much extensive research is done on Pakistan Sign Language Recognition (PSLR) using both static and dynamic data. For this paper, We decided to only be doing a vision-based approach and more specifically using video data and in the next section will be exploring related works done. Current models often fall short in capturing both spatial and temporal features simultaneously, especially for PSL, where there is a lack of standardized datasets and complex sign variations across individuals. The proposed framework includes the use of the combination of pretrained ResNet-101 [He et al. 2016] and Motion Binary Pattern (MBP) [Baumann et al. 2014] as our feature extractors and for classification we used Transformer [Vaswani et al. 2017].

1.1 Aim and Objectives

The aim of this dissertation is to improve PSLR through a framework that combines spatial and temporal feature extraction methods, sequential classification models and appropriate evaluation metrics. The specific objectives are:

- Develop an efficient end-to-end framework for PSL recognition.
- Use Top K Key frame extraction to select key frames from videos.
- Use ResNet-101 to extract spatial features from key frames.
- Use MBP to capture temporal information.
- Implement Transformer model to predict PSL signs.

- Train the proposed framework using the PkSLMNM dataset.
- Evaluate the performance of the proposed framework based on empirical classification metrics.
- Optimize the model hyperparameters through rigorous testing on the PkSLMNM dataset.
- Compare the model's accuracy against state-of-the-art models.

1.2 Organisation

This dissertation follows an organized structure: In Section 2 we examined the existing research on sign language recognition approaches for video data. Then in Section 3 we describe our proposed framework in detail while in Section 4 we discuss the project's requirements analysis. Section 5 covers the dataset and the evaluation measures, whereas in Section 6 experimental setup and results are discussed in depth. The conclusion along with limitations and scope for future work can be viewed in Section 7. In the back matter Appendix A discusses project management, including risk analysis and Gantt charts for the project timeline. Appendix B discusses the Professional, Legal, Ethical, and Social (PLES) considerations relevant to this research.

2 LITERATURE REVIEW

Communication in sign language is both manual and non-manual. As briefly mentioned in Section 1, manual ones include signs which can be formed by the use of hands, whereas non-manuals include head movements, facial expressions, shoulder shrugs, and other forms of body language that add meaning to the context [Sameena Javaid 2023]. These can be captured via two main techniques: vision-based and sensor-based. In this review, we narrow our interest to the vision-based technique more precisely video-based. Researchers from different backgrounds have employed various strategies in overcoming this barrier. Our review discusses the previous work done on various methods done for various sign languages presenting architectures, datasets used, results found, merits, and demerits of each and every technique.

The review focuses on prior research in the following sequence:

- Pakistan Sign Language (PSL)
- Indian Sign Language (ISL)
- Arabic Sign Language (ArSL)
- American Sign Language (ASL)
- Korean Sign Language (KSL)

Let us take a look at PSL related works first. It is to be noted that each author has taken distinct approaches suitable to the datasets they chose to work upon.

2.1 Pakistan Sign Language (PSL)

Sameena Javaid [2023] introduced a novel framework for PSL recognition using the PkSLMNM dataset [Javaid 2022], consisting of dynamic video-based signs from 180 participants. As seen in Figure 1, I3D-ShuffleNet (inspired from [Carreira and Zisserman 2017]) was employed for feature extraction to capture spatiotemporal information from the videos, alongside data augmentation techniques such as flipping, cropping and contrast adjustment. Action Transformer with Region Proposal Network (RPN) was used for the sign classification with an attention mechanism for bounding box generation. The model achieved a testing accuracy of 82.66% and training accuracy of 86.12%. However, the small dataset size and issues like motion blur limit the model's effectiveness in real-world scenarios.

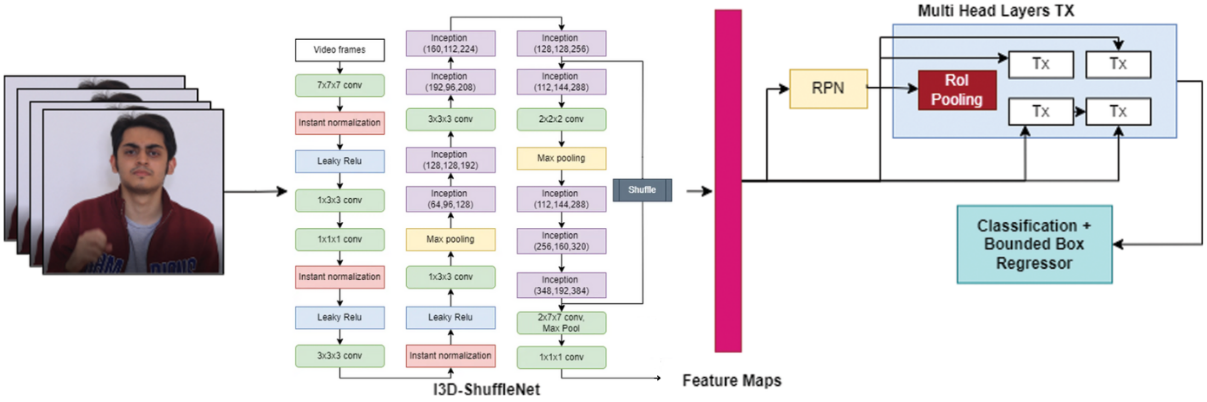


Fig. 1. System architecture of the proposed model by Sameena Javaid [2023]

Mirza et al. [2022] proposed a vision-based system using the self-collected dataset of 5120 static images and 353 dynamic videos of PSL signs from 10 native signers. As seen in Figure 2 the dataset was pre-processed first by resizing, then converting to grayscale images and later performing threshold-based segmentation. Then, features were extracted using Speeded-Up Robust Features (SURF) and clustered by K-means++ to form the BOW model. Classification is later done by the use of a Support Vector Machine (SVM). This in essence, allows the system to achieve an accuracy of 97.80% and 96.53% for static and dynamic signs, respectively.

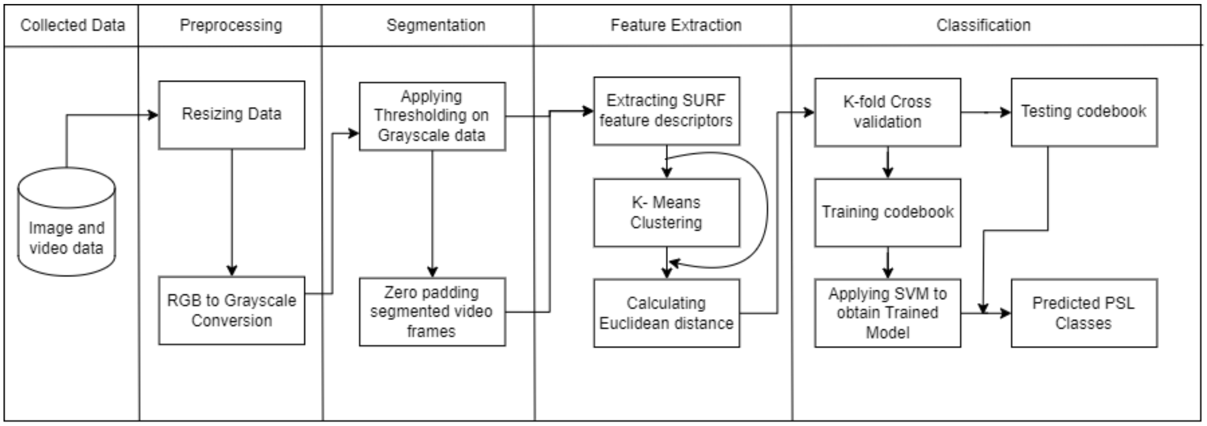

Collected Data
Preprocessing
Segmentation
Feature Extraction
Classification
Image and video data
Resizing Data
RGB to Grayscale Conversion
Applying Thresholding on Grayscale data
Zero padding segmented video frames
Extracting SURF feature descriptors
K- Means Clustering
Calculating Euclidean distance
K-fold Cross validation
Training codebook
Applying SVM to obtain Trained Model
Testing codebook
Predicted PSL Classes

Fig. 2. Proposed PSL recognition flowchart by Mirza et al. [2022]

Hamza and Wali [2023] addressed this challenge by the recognition of PSL on the limited dataset consisting of 80 signed words each having two samples per word. The authors have used data augmentation techniques to help improve model performance by adjusting brightness, rotation, scaling, and translation. They tested three different deep learning models, namely

Convolutional 3D (C3D), Two-Stream Inflated 3D ConvNet (I3D) [Carreira and Zisserman 2017], and a new approach was introduced via the Temporal Shift Module (TSM). Among them, C3D does the best with an accuracy of 93.33%, I3D reaches 87.50%, while TSM performs the poorest with only 35.83% accuracy.

2.2 Indian Sign Language (ISL)

Mittal et al. [2019] introduced a modified Long Short-Term Memory (LSTM) model for continuous sign language recognition using a Leap Motion sensor. The dataset was collected and included 942 sentences of ISL, capturing the 3D coordinates of fingertips. For feature extraction, 2D-Convolutional Neural Networks (2D-CNN) was employed. The modified 4 gated LSTM with 2D-CNN was proposed, featuring 3 layers and a reset gate, which achieved 72.3% accuracy for continuous sentences and 89.5% for isolated words.

Aparna and Geetha [2020] developed a dataset of six isolated words, containing 20 training videos and 10 testing videos for each word. The authors used Inception V3, a pre-trained Convolutional Neural Network (CNN) model for extracting features from video frames, converting video frames to feature vectors and further feeding the output to a stacked LSTM, recognizing temporal features. The model achieved a good accuracy of 94% on the training set.

Adithya and Rajesh [2020] developed a unique video dataset of hand signs focusing on emergency-related words. The dataset as shown in Figure 3, contains 824 videos of eight hand signs, such as "help" and "doctor," captured from 26 participants. They proposed two approaches for ISL recognition: Traditional feature based and deep learning . For feature extraction, 3D wavelet transform descriptors were used on key frames extracted through image entropy and clustering methods, classified using SVM and the deep learning model investigated using GoogleNet (pretrained CNN) and LSTM , achieving accuracy rates of 90% and 96.25%, respectively.

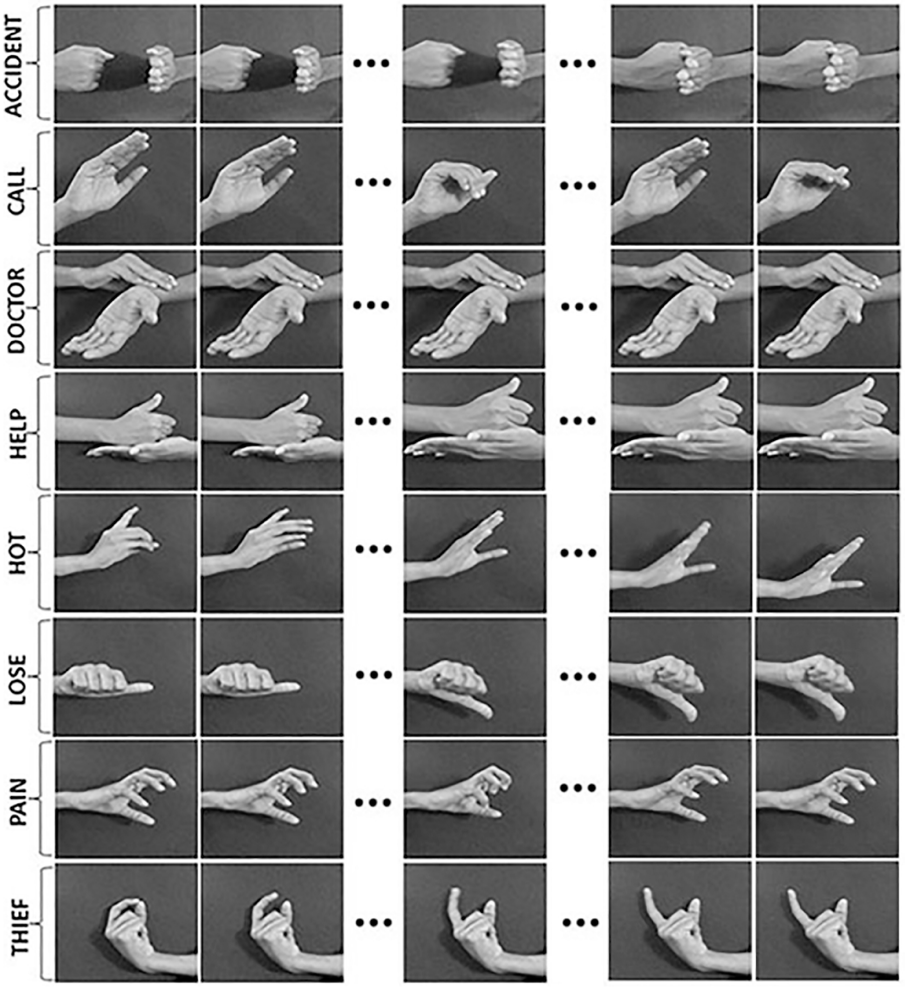


Fig. 3. Dataset created by Adithya and Rajesh [2020]

The work by Das et al. [2023] extends that of Adithya and Rajesh [2020] using the same dataset of 824 videos related to hand signs dealing with emergencies in ISL, proposing a hybrid approach that fuses Scale-Invariant Feature Transform (SIFT) and BOW together with Visual Geometry Group (VGG-19) for feature extraction. The classification model used was Bidirectional Long Short-Term Memory (Bi-LSTM). That gave an average accuracy of 94.42% using a Bi-LSTM network for classification.

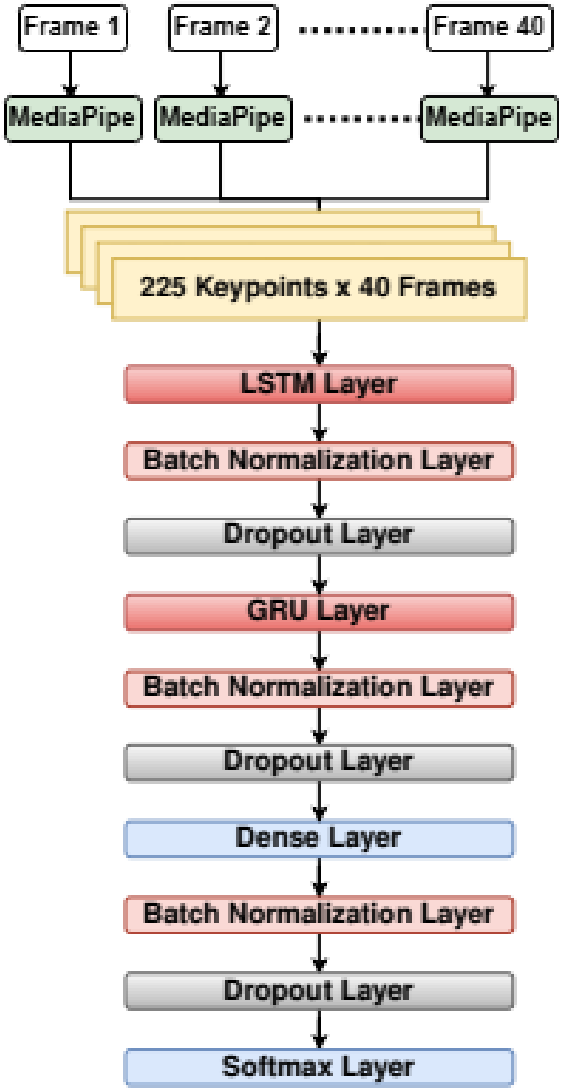


Fig. 4. Hybrid LSTM-GRU model architecture proposed by Navendu and Sahula [2024]

Navendu and Sahula [2024] proposed a hybrid LSTM-GRU network, leveraging the keypoints of video frames extracted through MediaPipe Hands and Pose for feature extraction. In line with that, 225 keypoints, describing hand and body landmarks, were extracted out of every frame to be processed. The hybrid model combines LSTM and Gated Recurrent Unit (GRU) layers as seen in Figure 4, to model the temporal dependencies that could exist within the signs. It was tested on the publicly available ISL dataset INCLUDE [Sridhar et al. 2020], and it reported 89.5% accuracy.

2.3 Arabic Sign Language (ArSL)

AL-Rousan et al. [2009] introduced a basic recognition system using Hidden Markov Model (HMM) and a self-collected dataset of 30 isolated words with 7,860 gestures recorded at a framerate of 25fps from 18 signers. Discrete Cosine Transform (DCT) was used for feature extraction and then Zonal coding was applied, followed by HMM classification, achieving 90.6% in online signer-independent mode and 97.4% accuracy in offline signer-dependent mode.

DeepArSLR is a framework introduced by Aly and Aly [2020]. As seen in Figure 5, DeepLabv3+ was used for precise hand segmentation which accurately extracts hand regions from video frames and Convolutional Self-Organising Map (CSOM) was implemented to capture detailed hand shape features. Temporal features were modeled using a three-layer Bi-LSTM network to learn the sequential nature of the gestures. DeepArSLR was tested and worked upon the ArSL database collected in [Shanableh et al. 2007a]. The framework achieved a respectable accuracy of 89.5%.

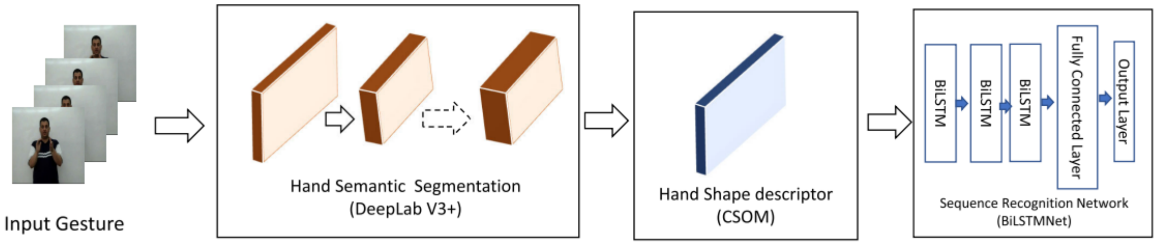


Fig. 5. Proposed DeepArSLR framework by Aly and Aly [2020]

Similar to AL-Rousan et al. [2009], Sidig et al. [2021]. used HMM for Classification. Histogram of Oriented Gradients (HOG) and skeleton joint coordinates that were obtained from the Kinect sensor were used to extract features. Sidig et al. [2021] presented a new large-scale KArSL dataset consisting of 502 signs repeated 50 times by 3 professional signers resulting in 75300 samples. The system achieved an overall accuracy of 89%. However, the accuracy dropped significantly for signer-independent scenarios.

Alyami et al. [2024] used a subset of 100 signs from Sidig et al. [2021]'s KArSL dataset to propose a transformer-based model using a combination of hand and face key points extracted with the MediaPipe pose estimator. Three models were explored LSTM, Temporal Convolution Networks (TCN), and Transformer where the latter showed the comparative best performance due to its self-attention mechanism that effectively captured the complex dependencies between the signs . The framework achieved a remarkable accuracy of 99.74% in signer-dependent mode and 68.2% in signer-independent mode outperforming other state-of-arts models on KArSL-100 dataset at the time.

2.4 American Sign Language (ASL)

Li et al. [2020] proposed a Pose-based Temporal Graph Convolutional Network (TGCN) by modeling both spatial as well as temporal dependencies within keypoint sequences. The model was trained on WLASL dataset introduced by authors themselves and comprising 21,083 videos of 119 individuals performing 2,000 signs. OpenPose was employed to capture 55 body and hand keypoints. The TGCN achieved 23.65% top-1 accuracy on the WLASL2000 subset, while a fine-tuned I3D model performed slightly better with 32.48% top-1 accuracy. Li et al. [2020] note that even though I3D is larger than the proposed Temporal Graph Convolutional Network (TGCN), pose-Temporal Graph Convolutional Network (TGCN) achieves comparable top-5 and top-10 accuracy to Two-Stream Inflated 3D ConvNet (I3D) on WLASL2000.

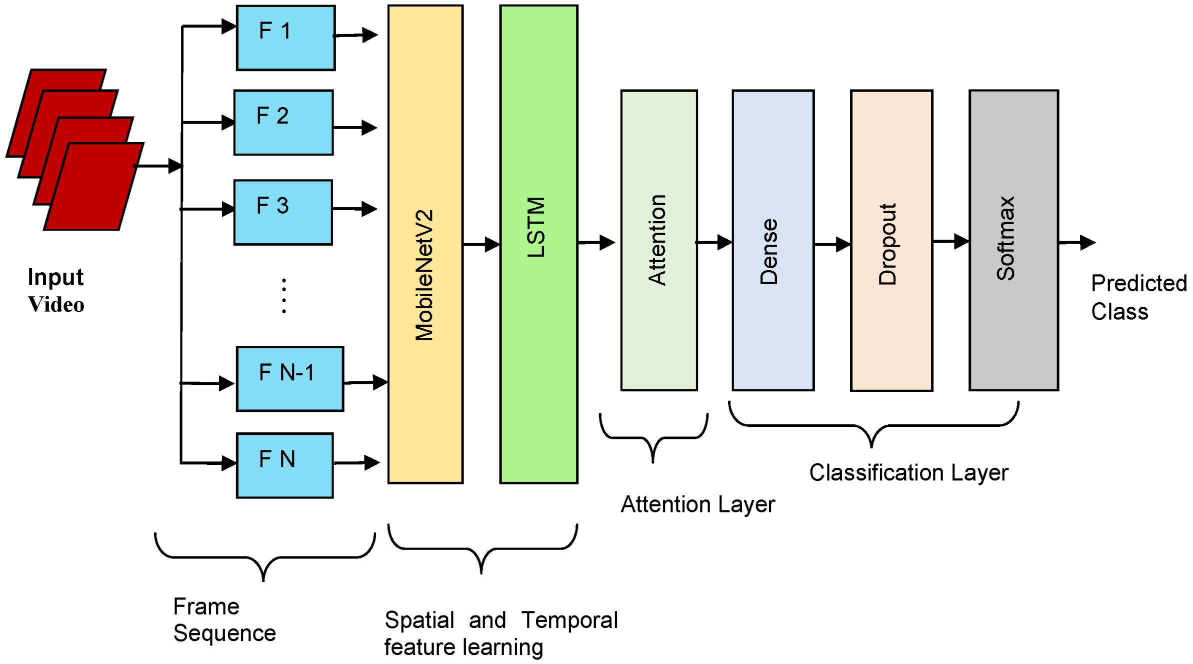


Fig. 6. Block diagram of the proposed approach by [Kumari and Anand 2024]

Building on the same dataset, Kumari and Anand [2024] proposed a hybrid CNN-LSTM framework integrated with an attention mechanism, as shown in Figure 6. This model was trained on a subset of the WLASL dataset [Li et al. 2020]. Kumari and Anand [2024] employed a pre-trained MobileNetV2 model to extract spatial features from video data. Afterwards, these features were subsequently processed by LSTM layers with an enhanced attention mechanism. This allows the model to pay attention to relevant information about hand signs throughout the time and yielded an accuracy of 84.65%.

2.5 Korean Sign Language (KSL)

Shin et al. [2023b] proposed a multi-branch architecture combining CNN and Transformer modules for KSL recognition. The reliability of this approach was demonstrated through experiments on the KSL-77 dataset [Yang et al. 2020] and their proposed dataset KSL-20. In this approach, grain architecture has been used to extract fine features from the beginning, followed by the parallel feature extraction through CNN for local features and transformer for capturing long-range dependencies and is finally classified through concatenation by means of a module that includes global average pooling and a fully connected layer. The model achieved a respectable accuracy of 89.00% for the KSL-77 dataset and a much higher impressive one, 98.30%, in the proposed dataset.

Building upon their previous work, Shin et al. [2023a] extended their research on KSL recognition using both KSL-77 and their self-collected dataset, KSL-20. Compared with their earlier work, While the used approach was based on a multi-branch CNN-Transformer in the previous paper, this advanced paper represents a two-stream deep learning net combined with Graph Convolutional Network (GCN) and attention-based neural networks. It proposes taking 47 pose landmarks from videos using MediaPipe and feeding them into the proposed model. In summary, one stream captures the spatial features of the appearance, while another stream focuses on joint motion. Proper refinement steps, including channel attention and a general CNN, will be done at both. The performance of the model has reached a remarkable accuracy of 99.87% for the KSL-77 dataset, as well as 100.00% on the KSL-20 dataset.

2.6 Critical Analysis

The most notable gap identified from our study is the lack of standardized and robust PSL datasets such as those available for ASL, by Li et al. [2020], ISL [Sridhar et al. 2020], which limits the generalizability and scalability of PSL-focused models. Research on PSL, including the work by Sameena Javaid [2023], often uses small, non-standardized datasets such as the PkSLMNM dataset [Javaid 2022], which limits the real-world performance of models.

The answer to this problem lies in large and standardized PSL datasets and efficient data augmentation techniques for improving model robustness with lesser overfitting under various user conditions. Additionally, ISL studies [Adithya and Rajesh 2020; Aparna and Geetha 2020; Das et al. 2023; Mittal et al. 2019; Navendu and Sahula 2024] and ASL research [Kumari and Anand 2024; Li et al. 2020] employ deep learning temporal-spatial models such as CNNs and LSTM. While these complex architectures are rather standard for dynamic gesture recognition, PSL studies tend to be carried out using simpler architectures that inadequately capture the subtle spatial and temporal aspects of PSL. For example, Although Sameena Javaid [2023] applies I3D-ShuffleNet for feature extraction, the approach lacks adaptive temporal modeling provided by Transformers or even Bi-LSTM networks observed in other research [Alyami et al. 2024; Shin et al. 2023a,b]. This difference in feature extraction and classification approaches suggest a limitation of the current PSL frameworks in relation to this issue. These include its ability to accurately capture gesture dynamics, emphasizing a potential for integrating hybrid architectures, a limitation addressed by Kumari and Anand [2024], where they combine spatiotemporal models to enhance the recognition accuracy.

Another critical gap includes the fact that dynamic texture descriptors have not been explored in PSL recognition. Dynamic texture descriptors are used for capturing motion patterns of video sequences but have not been widely applied in existing PSL studies. Also, the possible fusion of handcrafted descriptors with deep learning-based models has equally been overlooked. On one hand, handcrafted descriptors like Motion Binary Pattern (MBP) can give rich and complementary temporal information; on the other hand, deep learning models are very good at extracting high-level spatial features. In this framework, therefore, such an MBP is chosen to carry out the temporal feature extraction while ResNet-101 does the spatial features to allow us to perform a hybrid of both hand-crafted and deep learning-based approaches for enhanced PSL recognition. These features are then fed to a transformer model.

2.7 Comparison of Related Works

A comparison of all relevant studies was made. Table 1 shows an illustrative comparison for all the mentioned frameworks presented by authors, the dataset used, and type of data. It also shows which algorithm was used for feature extraction, as well as for classification. Finally, the accuracy is presented as a way to show how the frameworks performed after a test run using the dataset.

References	Domain	Dataset	Feature Extraction	Classification	Accuracy (%)
Sameena Javaid [2023]	PSL	PkSLMNM [Javaid 2022]	Spatiotemporal features with I3D-ShuffleNet	Action Transformer with RPN	82.66
Mirza et al. [2022]	PSL	Author collected data	SURF algorithm + Bag-of-Words Model	SVM	96.53
Hamza and Wali [2023]	PSL	Subset of PSL Dictionary [psl.org.pk 2020]	C3D, I3D, TSM	C3D	93.33
Mittal et al. [2019]	ISL	Author collected data	2D-CNN	Modified 4 Gated LSTM	89.50
Aparna and Geetha [2020]	ISL	Author collected data	Inception V3 (pretrained CNN)	Stacked LSTM	94.00
Adithya and Rajesh [2020]	ISL	Emergency Words [V and R 2021]	3D wavelet transform descriptors	GoogleNet + LSTM	96.25
Das et al. [2023]	ISL	Emergency Words [V and R 2021]	SIFT + BOVW + VGG-19	Bi-LSTM	94.42
Navendu and Sahula [2024]	ISL	INCLUDE [Sridhar et al. 2020]	Keypoints with MediaPipe Hands and Pose	Hybrid LSTM-GRU	89.50
AL-Rousan et al. [2009]	ArSL	Author collected data	DCT	HMM	97.40
Aly and Aly [2020]	ArSL	ArSL Database [Shanableh et al. 2007b]	DeepLab v3+ + CSOM	Bi-LSTM	89.50
Sidig et al. [2021]	ArSL	KArSL-100 [Sidig et al. 2021]	HOG + Skeleton joint coordinates	HMM	89.00
Alyami et al. [2024]	ArSL	KArSL-100 [Sidig et al. 2021]	2D pose landmarks of hands and face	Transformer	99.74
Li et al. [2020]	ASL	WLASL [Li et al. 2020]	OpenPose (55 keypoints)	Pose-based TGCN	23.65 (top-1)
Kumari and Anand [2024]	ASL	Subset of WLASL [Li et al. 2020]	MobileNetV2	LSTM with Attention	84.65
Shin et al. [2023b]	KSL	KSL-20 [Shin et al. 2023b]	Grain architecture with CNN and Transformer	Multi-branch CNN-Transformer	98.30
Shin et al. [2023a]	KSL	KSL-20 [Shin et al. 2023b]	MediaPipe Pose Landmarks + GCN	Two-stream GCN with Attention	100.00

Table 1. Comparison Table of Related Works

3 PROPOSED FRAMEWORK

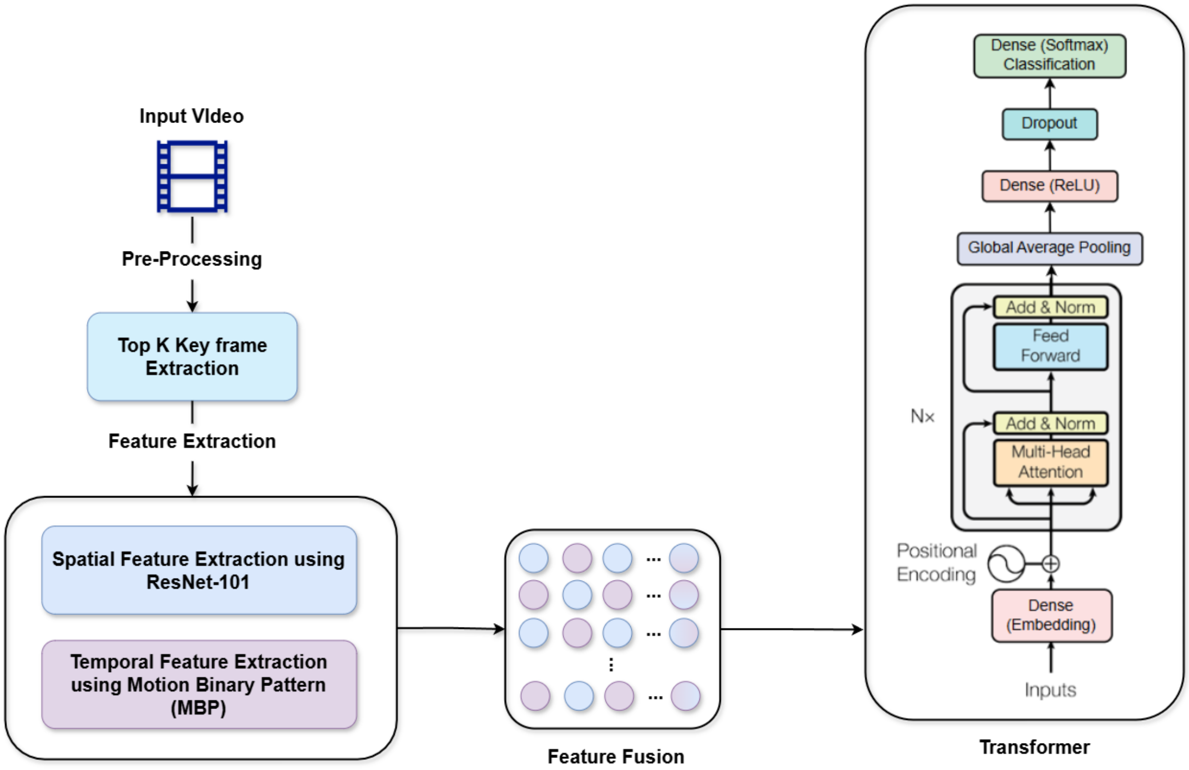


Fig. 7. Basic Architecture of the proposed method

3.1 Overview

As shown in Figure 7, The proposed framework here is focused on hand sign recognition from PSL signs using video data from the PkSLMNM dataset[Javaid 2022], which are pre-defined into categorized videos on specified signs. Our solution integrates advanced feature extraction with state-of-the-art deep learning architectures for effective spatial and temporal signs feature capture. Preprocessing for video frames is initially carried out with representative frames extracted through a Top-K Key Frame Extraction technique. Motion Binary Pattern (MBP) with Local Binary Pattern (LBP) are applied for motion dynamics and encoding of temporal patterns among successive frames. In parallel, spatial features are extracted from these crucial frames through the assistance of the ResNet-101 architecture, leveraging its residual network structure for effective capture of detailed visual features. These spatiotemporal features are fused and classified with a Transformer model, leveraging self-attention for modeling long-distance relationships and dependencies among sign sequences. In the following sections, a detailed explanation of each one of these phases is provided.

3.2 Pre-processing

Pre-processing is employed to ensure consistency, efficiency, and effectiveness in video data analysis by reducing redundancy and computational overhead. Key frame extraction using Top-K Key Frame Extraction is a crucial part of this stage proposed by Joolee et al. [2018]. As seen in Algorithm 1, this method drastically reduces redundancy by selecting representative frames based on comparisons of histograms calculated by Local Binary Pattern (LBP) [Ojala et al. 2002]. In contrast to threshold-based methods that compare consecutive frames in a video and compute histograms of consecutive frame differences, this method ranks these differences and picks top K maximum difference frames as key frames (say, K = 20 or 30). This ensures that selected frames capture distinctly significant dynamics and important motion patterns in video sequences with efficiency and retain maximum informative content while drastically cutting down on storage requirements and computational costs [Truong and Venkatesh 2007]. After extracting these key frames, we resize each frame to a standard dimension of 224x224 pixels, enabling efficient and uniform feature extraction in subsequent steps [He et al. 2016]. For this dissertation, the value of K is empirically selected to be set to 30.

Algorithm 1 Top-K Key Frame Extraction

```
procedure KEY-FRAME($video$)
    for $i \leftarrow 1$ to $NumberOfFrames - 1$ do
        $Frame_{\text{Curr}} \leftarrow ReadFrame(video, i)$
        $Frame_{\text{Next}} \leftarrow ReadFrame(video, i + 1)$
        $LBP_{\text{Curr}} \leftarrow Extract\_LBP\_Histogram(Frame_{\text{Curr}})$
        $LBP_{\text{Next}} \leftarrow Extract\_LBP\_Histogram(Frame_{\text{Next}})$
        $Distance \leftarrow Euclidean(LBP_{\text{Curr}}, LBP_{\text{Next}})$
        $X[i] \leftarrow Distance$
    end for
    $[sortedX, sortingIndices] \leftarrow sort(X,' descend')$
end procedure
```

3.3 Motion Binary Pattern Based Spatiotemporal Features Extraction

For extracting spatiotemporal features, we utilized a Motion Binary Pattern (MBP)-based descriptor to effectively extract these features from video sequences. Originally introduced by Baumann et al. [2014], MBP is capable of capturing temporal motion dynamics by exploring the pixel intensity variations between three consecutive frames—previous, current, and next frames. Unlike conventional texture-based descriptors such as Local Binary Pattern (LBP) [Ojala et al. 2002], which predominantly encode static spatial textures, MBP encodes temporal information explicitly and can therefore depict dynamic patterns in video data.

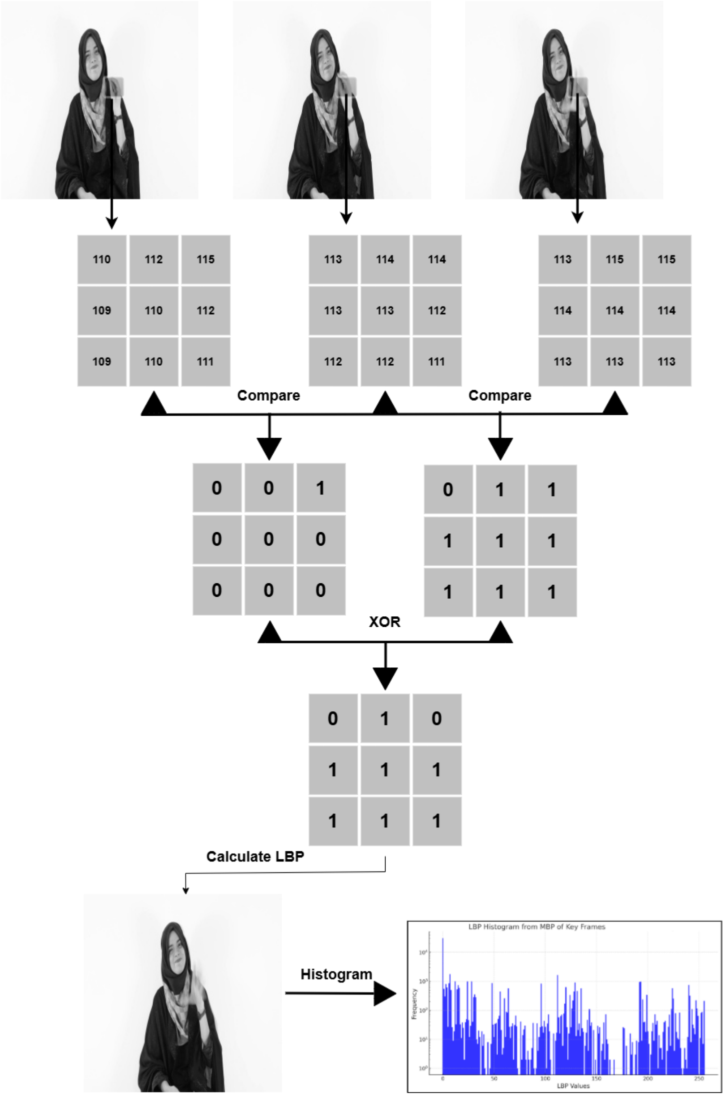


Fig. 8. Spatiotemporal feature extraction using MBP-LBP

Other descriptors, like Volume Local Binary Patterns (VLBP) [Zhao and Pietikainen 2007], 3D-Scale-Invariant Feature Transform (SIFT) [Scovanner et al. 2007] and Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) [Zhao and Pietikainen 2007], have also attempted to extend texture-based methods to the temporal domain. Such techniques, however, inherit the sensitivity to illumination changes and noise, impacting their robustness. MBP mitigates some of these limitations, due to its straightforward yet effective mechanism, by performing intensity comparisons between spatially segmented patches of consecutive frames.

In this algorithm, frames are divided into local spatial grids, and each corresponding grid in successive frames is considered. Differences in pixel intensity from the central reference frame generate binary patterns, which indicate directional motion. Next, binary patterns of successive frame-pairs are combined using an exclusive OR (XOR) operation, resulting in distinctive MBP matrices that denote significant temporal changes.

Whereas Baumann et al. [2014] relied on thresholding to emphasize strong motion areas, our implementation enhances the MBP descriptor by omitting fixed thresholds (see Algorithm 2). Instead, Local Binary Pattern (LBP) histograms are computed directly from the generated MBP matrices as shown in Figure 8. By doing this, the information captured is widened so that the histograms can successfully represent subtle in addition to strong motion-texture patterns.

Algorithm 2 Motion Binary Pattern (MBP)

```
procedure MBP_LBP($video, patch\_size$)
    for $i \leftarrow 2$ to $NumberOfFrames - 1$ do
        $Frame_{\text{Prev}} \leftarrow ReadFrame(video, i-1)$
        $Frame_{\text{Curr}} \leftarrow ReadFrame(video, i)$
        $Frame_{\text{Next}} \leftarrow ReadFrame(video, i+1)$
        $MBP \leftarrow Compute\_MBP(Frame_{\text{Prev}}, Frame_{\text{Curr}}, Frame_{\text{Next}}, patch\_size)$
        $LBP\_Hist \leftarrow Compute\_LBP\_Histogram(MBP)$
        Store $LBP\_Hist$
    end for
    return list of $LBP\_Hist$ histograms
end procedure
```

By integrating MBP's temporal encoding capability and explicit spatial texture analysis of LBP, our MBP-LBP descriptor inherits strong discriminative power and robustness, enhancing its ability for subtle motion recognition in Pakistan Sign Language.

3.4 Deep Spatial Feature Extraction Using ResNet-101

Spatial feature extraction was conducted using ResNet-101, a deep Convolutional Neural Network (CNN) architecture renowned for learning complex hierarchical visual representations, originally proposed by He et al. [2016]. ResNet-101 was chosen in this work because of its exceptional depth (101 layers), which was used for extracting very fine spatial features required for the accurate recognition of fine hand shapes, configurations, and orientations of hands in Pakistan Sign Language signs. One of the key advantages of employing ResNet-101 is that it prevents the vanishing gradients problem—a common problem of deep networks—by adding identity-based skip (residual) connections between layers as evident in Figure 9. These connections enable efficient gradient flow between layers, enhancing training stability and the learning of strong visual representations. [He et al. 2016]

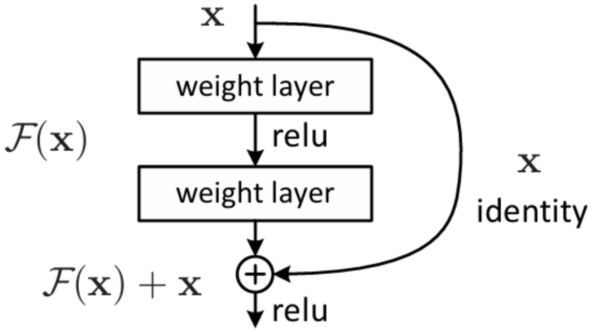

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 9. Residual learning: a building block [He et al. 2016]

In our implementation, video frames were resized to a standardized dimension of 224x224 pixels and color information (RGB) was preserved, as retaining rich color channels provides additional context valuable for deep spatial feature extraction [He et al. 2016]. Due to the limited size of the available PkSLMNM dataset, transfer learning was employed by using pre-trained weights from ResNet-101 trained on the large-scale ImageNet dataset [Russakovsky et al. 2015]. As per Figure 10, spatial features were specifically extracted from the Global Average Pooling (GAP) layer placed immediately after the final convolutional layer of ResNet-101. The GAP layer condenses rich convolutional feature maps into compact and discriminative feature embeddings, resulting in a compact vector of size 2048 per frame. Such embeddings nicely capture important visual details like hand shape changes, spatial location, and subtle appearance differences between gestures [Van Den Oord et al. 2016].

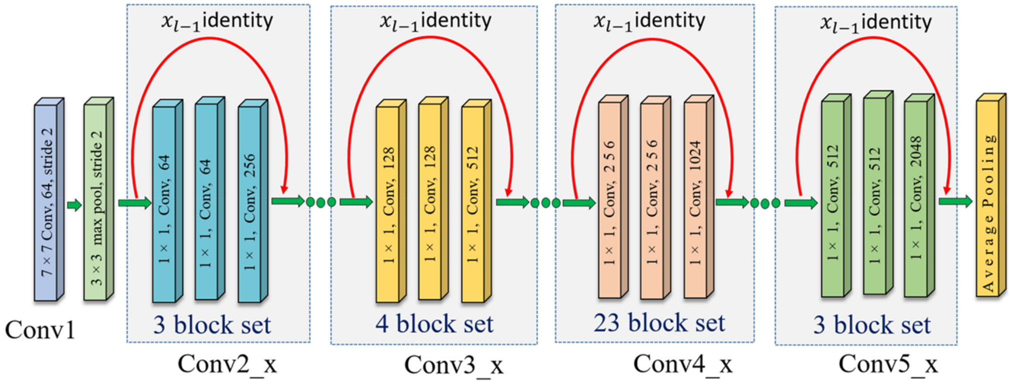


Fig. 10. Architecture of ResNet-101 GAP Layer [Tiwari et al. 2022]

Following the extraction of spatial embeddings using ResNet-101, we performed feature fusion by concatenating these spatial vectors (2048-dimensional) with temporal descriptors obtained from Motion Binary Pattern (MBP) (256-dimensional). The fusion process results in a comprehensive final feature representation of size 2304 features per frame, effectively encoding spatial and temporal gesture features necessary for PSL recognition [Zhou et al. 2020]. Therefore, for the top-k key frames of a video, we obtain a k × 2304 feature matrix. This feature matrix is given as input to the transformer model for classification.

3.5 Classification

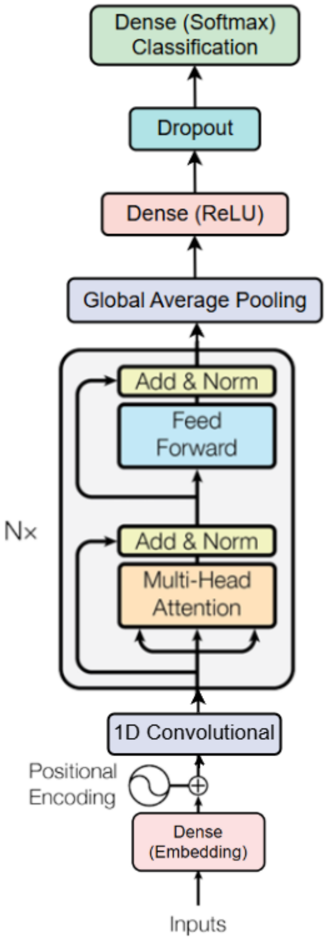


Fig. 11. Architecture of Transformer

For the classification stage a **Transformer-based model** is employed to capture long-term temporal dependencies within sequences of extracted visual features. While, Motion Binary Pattern (MBP) and ResNet-101 provide complementary spatial and temporal features for individual key frames, modeling the interactions across longer sequences is essential for effectively interpreting hand signs. The Transformer architecture, proposed initially by Vaswani et al. [2017], uses self-attention mechanisms that are able to model dependencies among frames irrespective of their position within a sequence, avoiding recency bias and vanishing gradients, which have been traditionally linked with Recurrent Neural Network (RNN). The architecture used is depicted in Figure 11, and the specifications for each layer are:

3.5.1 *Dense Layer (Projection to Embedding Dimension).* :

First, MBP and ResNet-101 visual features are projected into a common embedding space through a dense layer. Feature dimensionality is standardized through this process, facilitating effective learning and comparison in Transformer.[Dosovitskiy et al. 2020]

3.5.2 *Convolutional Embedding Layer.* :

To further capture local dependencies and subtle spatial-temporal patterns within embedded sequences, a one-dimensional convolutional layer (Conv1D) is introduced after the dense embedding. Given the embedded input features $X_{\text{dense}} \in \mathbb{R}^{k \times d_{\text{model}}}$, where k represents the sequence length (number of frames) and d_{model} is the dimension of the embedding, the convolutional embedding layer computes the enhanced embeddings X_{conv} as:

$$X_{\text{conv}} = \text{Conv1D}(X_{\text{dense}}) \tag{1}$$

This convolutional step helps to encapsulate short-range sequential patterns, providing more discriminative embeddings by integrating local spatial-temporal information before feeding into positional encoding and subsequent Transformer layers. [Wu et al. 2021]

3.5.3 *Positional Encoding Layer.* :

Because Transformers lack an inherent sequential structure (like RNNs), there is a dependence on positional encoding for maintaining information about temporal order [Vaswani et al. 2017]. Sinusoidal positional encodings are explicitly added to feature embeddings, providing explicit position information to the model, enabling effective interpretation over time for sign sequences. Specifically, Positional encoding is defined as:

$$\text{PosEnc}(p, 2i) = \sin\left(\frac{p}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{2}$$

$$\text{PosEnc}(p, 2i+1) = \cos\left(\frac{p}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{3}$$

where p is the position in the sequence, i is the dimension index, and d_{model} is the dimensionality of the model's embedding.

3.5.4 Transformer Block. :

The core of the model is a Transformer block, consisting of a number of sub-layers that allow the network to selectively focus on relevant temporal features within input sequences:

Multi-head attention allows the Transformer model to look at information from various subspaces of representation simultaneously. By computing attention weights across various heads, the model can capture various features of signs and hand movements distributed throughout the sequence [Vaswani et al. 2017]. This ability significantly enhances the detection of small differences among alike gestures significantly.

They are all connected with the initial input via **residual connections**, followed by layer normalization. The residual connections prevent vanishing gradients and stabilize training for deeper networks, and layer normalization speeds up training and convergence through normalization of intermediate values.

Subsequent to attention, a **position-wise feed-forward neural network** is used independently and individually on each position within a sequence. Comprising two linear transformations with Rectified Linear Unit (ReLU) activation, this layer further refines and adds features learned through attention, extracting complex nonlinear relationships within data.

Another **residual connection** after layer normalization supports efficient propagation of features learned and stabilizes training on deeper Transformer layers.

3.5.5 Global Average Pooling (GAP) Layer. :

Following Transformer blocks, Global Average Pooling is applied for reducing sequential output to a fixed representation. GAP is able to extract temporal information effectively, where one fixed-length vector is returned for every input sequence [Lin et al. 2013]. It makes downstream classification easier with less loss of valuable temporal information.

3.5.6 Fully Connected Layers. :

Then, the pooled features are passed on through fully connected layers, which are classifiers for decoding compressed representations. These successively convert the features into class-specific discriminative representations for classification.

3.5.7 SoftMax Output Layer. :

Finally, a SoftMax output function [Bishop and Nasrabadi 2006] produces probabilities for all PSL classes. The probabilistic output enables direct classification of input video sequences into their respective sign language gestures.

For efficiently training the Transformer model, Adam has been employed as our optimizer, one that learns adaptive learning rates throughout training for faster convergence as well as optimal model performance [Kingma and Ba 2014]. Gradient clipping has also been employed for stabilizing training as a technique for preventing exploding gradients through limiting gradients within a reasonable range [Pascanu et al. 2013]. A number of regularization techniques were experimented for further encouraging generalizability and preventing overfitting. L2 regularization was integrated within dense layers for preventing large weights and producing smoother decision boundaries [Ng 2004]. Additionally, dropout layers were inserted at strategic locations within Transformer blocks and fully connected layers for further preventing overfitting and ensuring stable model behavior.

The next section goes into details regarding the requirements necessary for the implementation of the proposed methodology.

4 REQUIREMENT ANALYSIS

In this section we see the main requirements for our model based on MoSCoW framework, the following requirements are classified as Must have, Should have, Could have, or Won't have. It is divided into three sections where functional, non-functional, and hardware requirements are shown in Table 2, Table 3, and Table 4 respectively.

4.1 Functional Requirements

ID	Requirement Description	Priority	Status
FR1	Extract key frames from input videos using Top K Key frame extraction before being fed to the model	Must	Completed
FR2	Extract spatial features using ResNet-101	Must	Completed
FR3	Extract temporal motion information using MBP technique	Must	Completed
FR4	Apply feature fusion to concatenate the extracted features.	Must	Completed
FR5	Implement baseline Deep Learning and Machine Learning models	Must	Completed
FR6	Split the PkSLMNM dataset into train, test and validation subsets	Must	Completed
FR7	Use the test subset to evaluate the performance of the models	Must	Completed
FR8	Perform hyperparameter optimization on the models	Must	Completed
FR9	Save the preprocessed data within the same directory for ease of use	Should	Completed
FR10	Save the extracted spatial and temporal features to a separate location locally for possible reusability and re-training	Should	Completed
FR11	Implement progress bars and debugging statements for readability	Could	Completed
FR12	Produce intermediate outputs for each model, such as attention weights or hidden states, to aid in interpretability	Could	Partial

Table 2. Functional Requirements Table

4.2 Non-Functional Requirements

ID	Requirement Description	Priority	Status
NFR1	The model must achieve a respectable 75% or more accuracy for PSL signs across the framework	Must	Completed
NFR2	Data must be stored locally and processed to ensure no potential data leak	Must	Completed
NFR3	The framework should perform consistently with minimal variation across the different signs	Should	Completed
NFR4	Access to the dataset, model configurations, and outputs should be restricted to the author and supervisor only	Should	Completed
NFR5	Model can be used to support additional PSL signs without significant restructuring	Could	Partial

Table 3. Non-Functional Requirements Table

4.3 Hardware & Software Requirements

Here are the Hardware & Software requirements needed for this project. Since we are working with video data, the computational performance will be demanding. Hence, to ensure efficient processing and model accuracy, the following hardware specifications are recommended.

ID	Component	Description
HR1	Processor (CPU)	AMD Ryzen 7, optimized for multi-threaded performance
HR2	Graphics Card (GPU)	CUDA-cores supported NVIDIA RTX 3050 GPU, 4 GB VRAM
HR3	Memory (RAM)	64 GB DDR4
HR4	Storage	2 TB SSD
HR5	Software Compatibility	Supports TensorFlow, CUDA and CUDnn back-end to leverage GPU acceleration for efficient model training

Table 4. Hardware & Software Requirements Table

In Section 5, we discuss the dataset that is used for our proposed approach, followed by the metrics that we used to evaluate the model.

5 EVALUATION

The following section talks about the dataset as well the metrics we use to evaluate our model on the dataset.

5.1 Dataset

For this research, we trained and evaluated the proposed method with a single dataset. By using the Pakistan Sign Language Manual and Non-Manual (PkSLMNM) dataset [Javaid 2022], which is specifically designed for Pakistan Sign Language (PSL) identification and includes both manual as well as non-manual signs. The dataset consists of 665 videos of 180 people, which include 70 females and 110 males, which vary in age from 20 to 50 years old as below in Figure 12.

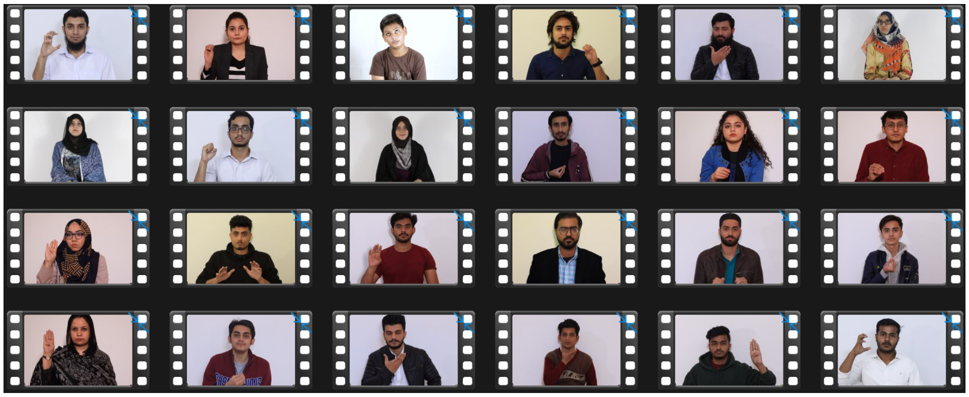


Fig. 12. Samples of the PkSLMNM dataset

The PkSLMNM dataset includes a range of PSL expressions represented by facial and hand signs covering seven emotional categories as seen in Table 5. Each video is recorded in HD at 1920x1080 resolution, lasting approximately 2 seconds per sample. The recordings were made at a frame rate of 25 Frames Per Second (Fps).

Sign	Number of Samples
Bad	97
Best	98
Glad	98
Sad	95
Scared	94
Stiff	85
Surprise	98

Table 5. Samples per sign in the PkSLMNM dataset

5.2 Evaluation Metrics

In this section, we will be exploring the following performance metrics that were used to evaluate our model:

(1) Accuracy;
(2) Mean Average Precision (mAP);

5.2.1 *Accuracy.* :

Accuracy is defined as the proportion of correctly predicted signs to the total number of predictions made[Foody 2023]. Usually, it would represent the performance of the model in the general success rate of the recognition of signs. However, this could not be useful as a metric when dealing with an imbalanced dataset. Mathematically, accuracy is represented as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

where

- **True Positives (TP):** The number of PSL signs model predicted accurately.
- **True Negatives (TN):** The number of non-PSL signs model predicted accurately.
- **False Positives (FP):** The number of non-PSL signs model predicted inaccurately as PSL signs.
- **False Negatives (FN):** The number of PSL signs model predicted inaccurately as non-PSL signs.

5.2.2 *Mean Average Precision.* :

Mean Average Precision (mAP) is a global performance measure that sums the overall performance of the model for all classes by both recall and precision[Foody 2023]. It is most applicable in multi-class classification and object detection where not just the detection of the correct instances but also the ranking thereof matter [Yue et al. 2007]. mAP calculates the mean of the Average Precision (AP) values for all classes that have been achieved. AP calculates the area under the precision-recall curve. Having a high mAP indicates the model is capable of detecting the true positives in all classes with minimal false positives and few false negatives. In the case of Pakistan Sign Language Recognition, having a high mAP score indicates the model is distinguishing between the signs well and hence improves the quality of communication and reduces misclassification errors. The Average Precision for each class is calculated using:

$$\text{AP} = \sum_{k=1}^{n} P(k) \cdot \Delta r(k) \tag{5}$$

where

- $P(k)$**:** The precision at cutoff rank k, i.e., the proportion of true positives among the top k retrieved results.
- $\Delta r(k)$**:** The change in recall at rank k, representing how much recall increases from the previous step.

Mean Average Precision (mAP) is then computed as:

$$\text{mAP} = \frac{1}{Q} \sum_{q=1}^{Q} \text{AP}(q) \tag{6}$$

where

- Q**:** The total number of classes or queries in the dataset.
- **AP**(q)**:** The Average Precision score calculated for the q^{th} class based on its precision-recall curve.

6 EXPERIMENTS

This section discusses about the setup of the experiments, the models used as well as other such implementation details.

6.1 Implementation Setup

The system used for training was the AMD Ryzen 7 5000u processor with 64 GB RAM and Nvidia RTX 3050 GPU. Python3 and TensorFlow were being used by the models on Ubuntu within Windows Subsystem for Linux (WSL). Rectified Linear Unit (ReLU) activation function was being used as the activation function throughout the whole Transformer model. In the sequential models such as Bi-LSTM, LSTM, GRU, and Bi-GRU, tanh and sigmoid were being used as the respective hidden and recurrent layer activation functions corresponding to TensorFlow's GPU-based implementations for optimal training.

The dataset underwent preprocessing to extract top 30 key frames. These frames were passed to the 2 stream feature extractors and once the spatial and temporal features were extracted and fused, the resulting post processed dataset was split into training and test subsets (80/20). Another 80/20 split on training subset for internal training and validation was performed for further robustness and data leakage avoidance. Optuna was employed for optimizing hyperparameters [Akiba et al. 2019] that ran 100 trials each with 200 epochs and Stratified 5-fold cross-validation, chosen empirically. Adam optimizer [Kingma and Ba 2014] with gradient clipping [Pascanu et al. 2013] was employed for training, with categorical cross-entropy as the loss function, and accuracy as the primary metric for model evaluation. This was identical for all the experiments.

Multiple iterations of cross-validation were used to tune the hyperparameters, resulting in an optimal configuration comprising an embedding dimension of 80, 6 attention heads, a feed-forward network dimension of 128, a dropout rate of 0.25, a learning rate of 9×10^{-4}, and a batch size of 16. The number of training epochs was fixed at 200 to strike a balance between model performance and computational efficiency.

6.2 Result Analysis

This section gives a detailed description of the results from experiments conducted during this research. The results are split into two subsections: ablation study and comparison with state-of-the-art in Section 6.2.1 and Section 6.2.2 respectively.

6.2.1 Ablation Study. :

In this ablation study, we test the contribution of each component in our proposed framework, experimenting with different feature extraction strategies and different Transformer configurations.

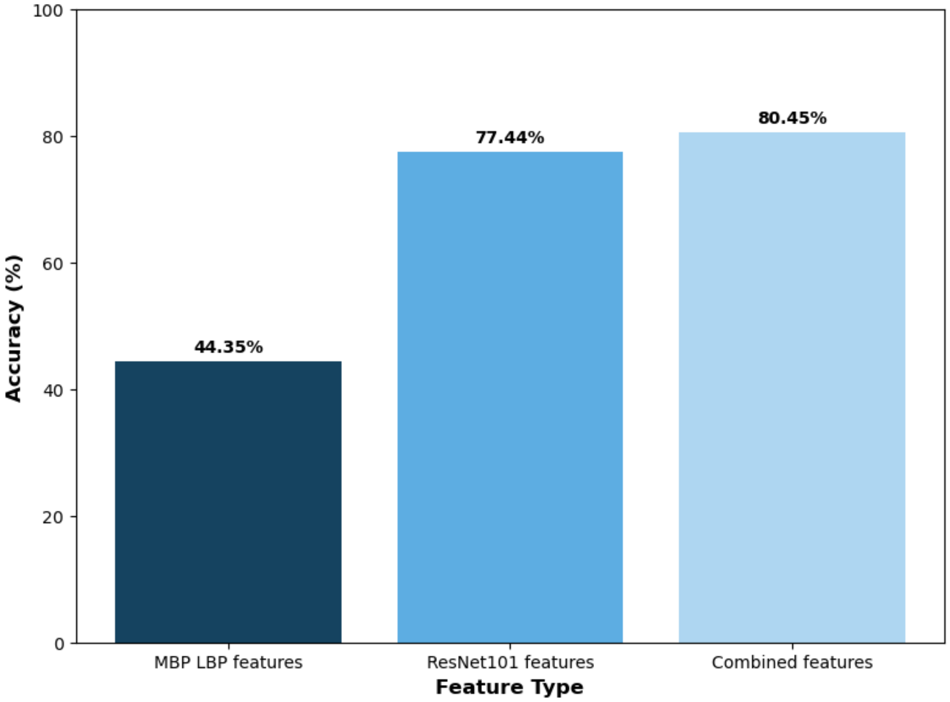


Fig. 13. Comparing performance of different components within proposed framework

From Figure 13, we can see employing MBP-LBP features with Gated Recurrent Unit (GRU) as a classifier achieves an accuracy rate of 44.35%, which shows limitations when only temporal features are employed. However, spatial features extracted with ResNet-101 GAP, along with our Transformer classifier, achieve much better accuracy at 77.44%, which indicates the superiority of deep spatial features. Most notably, a combination of both features achieves optimal performance with a maximum accuracy rate of 80.45%. The combination effectively utilizes each type's strengths, with a decent improvement in overall accuracy.

Figure 14 compares how different deep learning architectures and individual layer choices impact classification accuracy. Experiments comparing features extracted with GAP-layer and Top layers indicate that GAP-layer extracted features consistently have much better accuracies than features extracted with Top layers. More specifically, features extracted with GAP-layer from ResNet-101 offer a best accuracy rate of 77.44%, outperforming VGGNet GAP-layer features at 72.93%. Classification accuracy, however, drops drastically when top-layer features are employed, with VGGNet top-layer features yielding only 44.36%, and ResNet-101 top-layer features yielding even worse accuracy at 41.35%. These findings clearly indicate the superiority of GAP-layer extracted features thereby justifying their selection for our combined model.

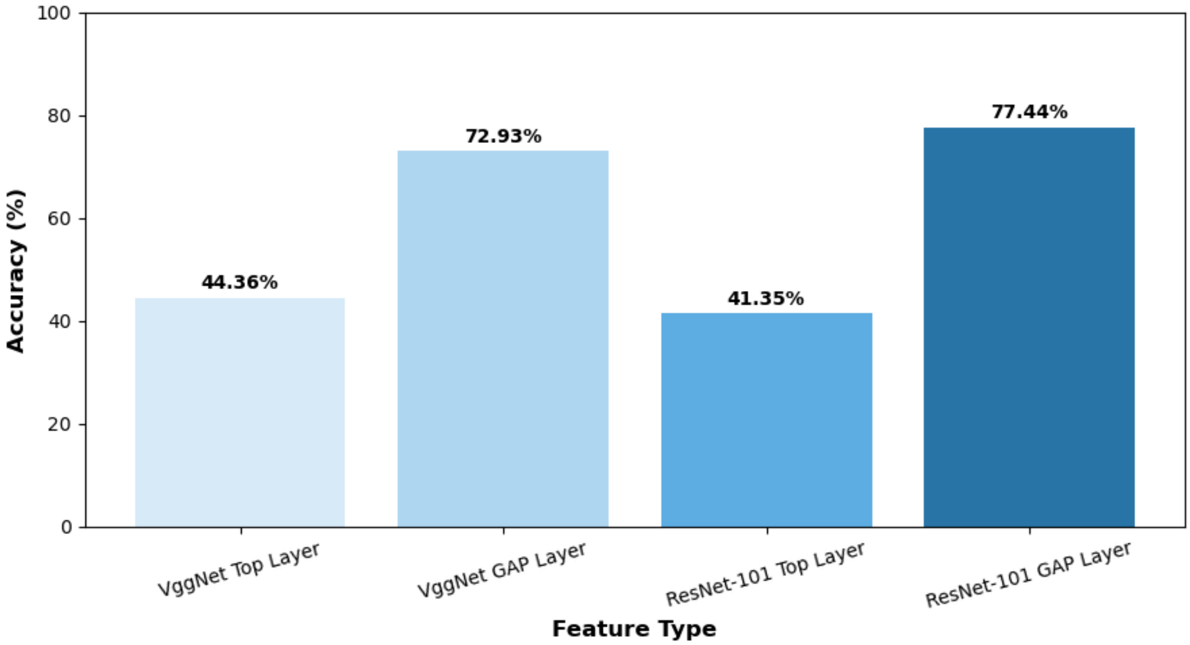


Fig. 14. Impact of individual layer and architecture choice for spatial feature extraction

To offer baseline performance, Figure 15 compares features extracted with traditional classifiers based on machine learning Decision Trees, Random Forest, and Support Vector Machine (SVM). SVM achieved the best with 63.91% accuracy, followed by Random Forest at 60.90%, with each performing best with spatial features extracted from GAP-layers of ResNet-101. Decision Trees worked much less well at 38.62%, the best accuracy coming from a combination of ResNet-101 GAP spatial features with MBP-LBP temporal features. These results demonstrate that, as much as traditional machine learning strategies may provide acceptable baseline accuracy, they are less effective than sequence models based on deep learning, pointing towards a need for further advanced deep learning methodologies for classifying PSL signs effectively.

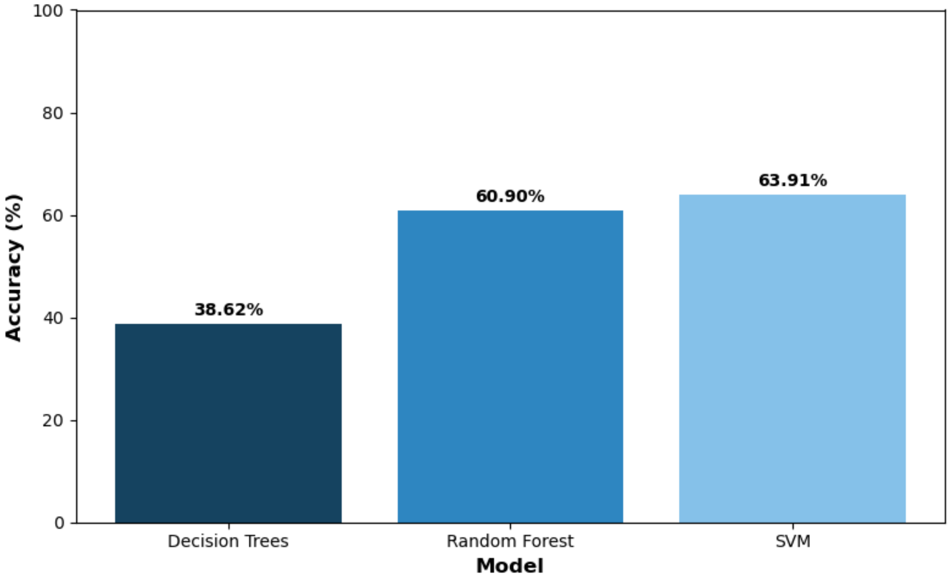


Fig. 15. Comparison of Machine Learning Models

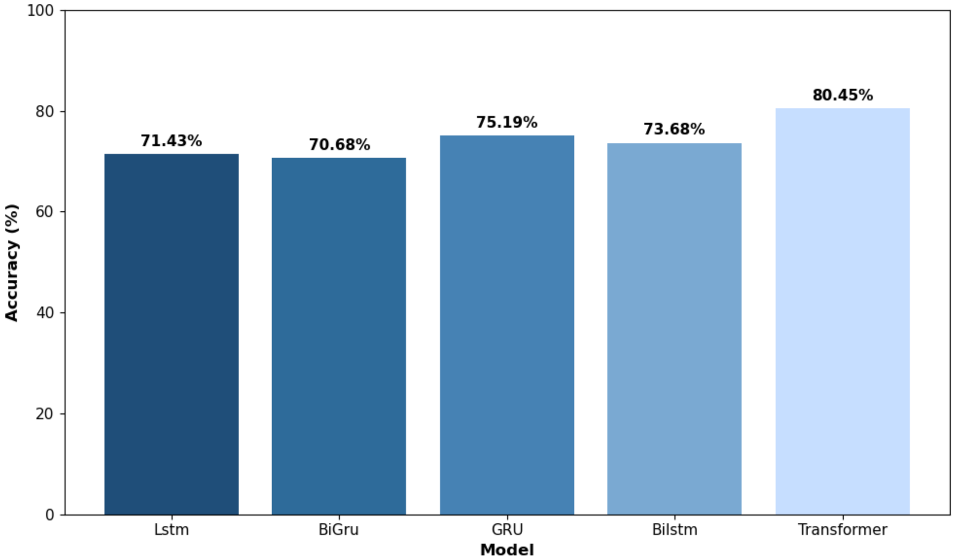


Fig. 16. Comparison of Deep Learning Models

Further experiments were conducted using deep learning sequence models as depicted in Figure 16. Highest accuracy was achieved using the Transformer model at 80.45% that outperforms all the competing models. Right after the Transformer are the Gated Recurrent Unit (GRU) (75.19%), Bidirectional Long Short-Term Memory (Bi-LSTM) (73.68%), Bidirectional Gated Recurrent Unit (Bi-GRU) (70.68%), and Long Short-Term Memory (LSTM) (71.43%). Notably, the Transformer, LSTM, and Bi-LSTM achieved their maximum accuracy using combined ResNet-101 GAP-layer and MBP-LBP feature types, further justifying the need for the use of spatiotemporal information in optimal classification. It is apparent that the results confirm the superiority of the Transformer model among the other sequence models in extracting complex long-term temporal dependencies that are essential for successful Pakistan Sign Language Recognition, justifying our selection in the use of Transformers as our primary classifier.

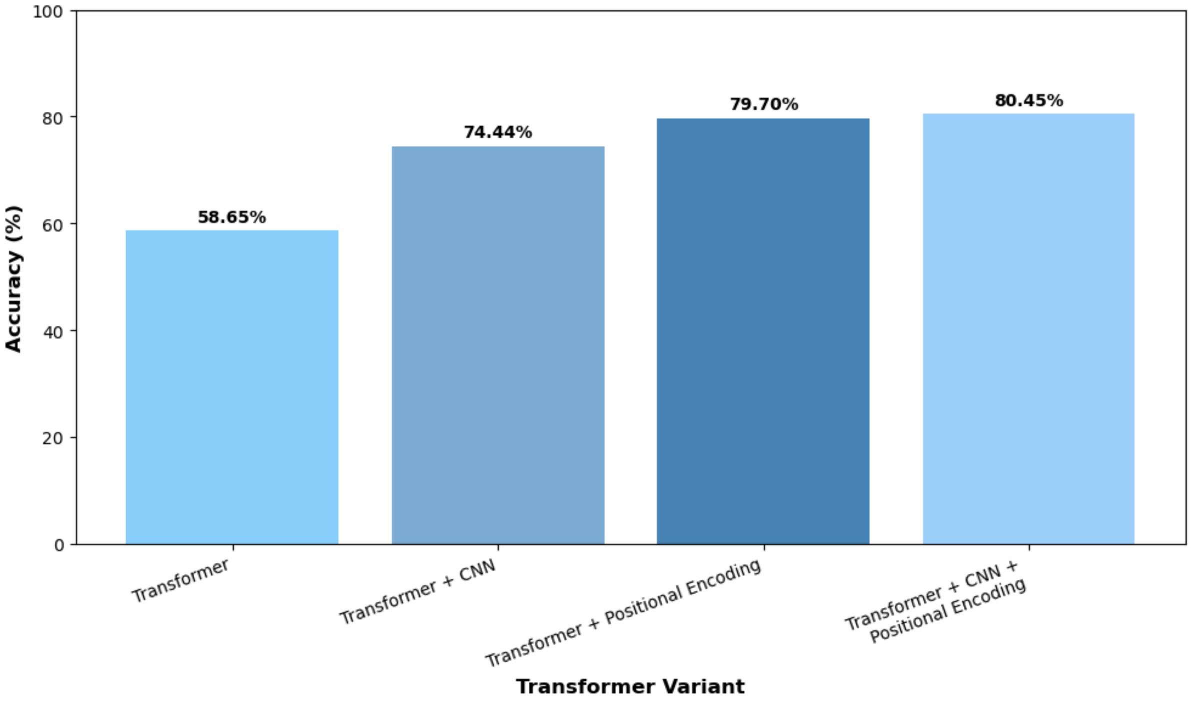


Fig. 17. Comparison with Transformer Layer Configurations

Then, to validate some Transformer design choices, we compared performance among four versions of the Transformer, as illustrated in Figure 17. The baseline Transformer model, which utilized combined ResNet-101 GAP spatial and MBP-LBP temporal features but without positional encoding and without CNN, resulted in the lowest accuracy of 58.65%, indicating the value of CNN and positional encoding in effectively encoding sequential information. Adding

Convolutional Neural Network (CNN) Layer significantly improved accuracy to 74.44%, which utilized spatial ResNet-101 GAP-layer features, indicating the value of convolutional spatial features. Most importantly, our optimized Transformer version, using positional encoding coupled with combined spatiotemporal features, achieved an accuracy of 79.70%. This was then further enhanced by adding a CNN layer which increased the accuracy to the highest at 80.45%. These comparisons demonstrate the critical contributions of positional encoding and comprehensive spatial-temporal feature addition of CNN in unlocking the full potential of the Transformer for accurate PSL recognition.

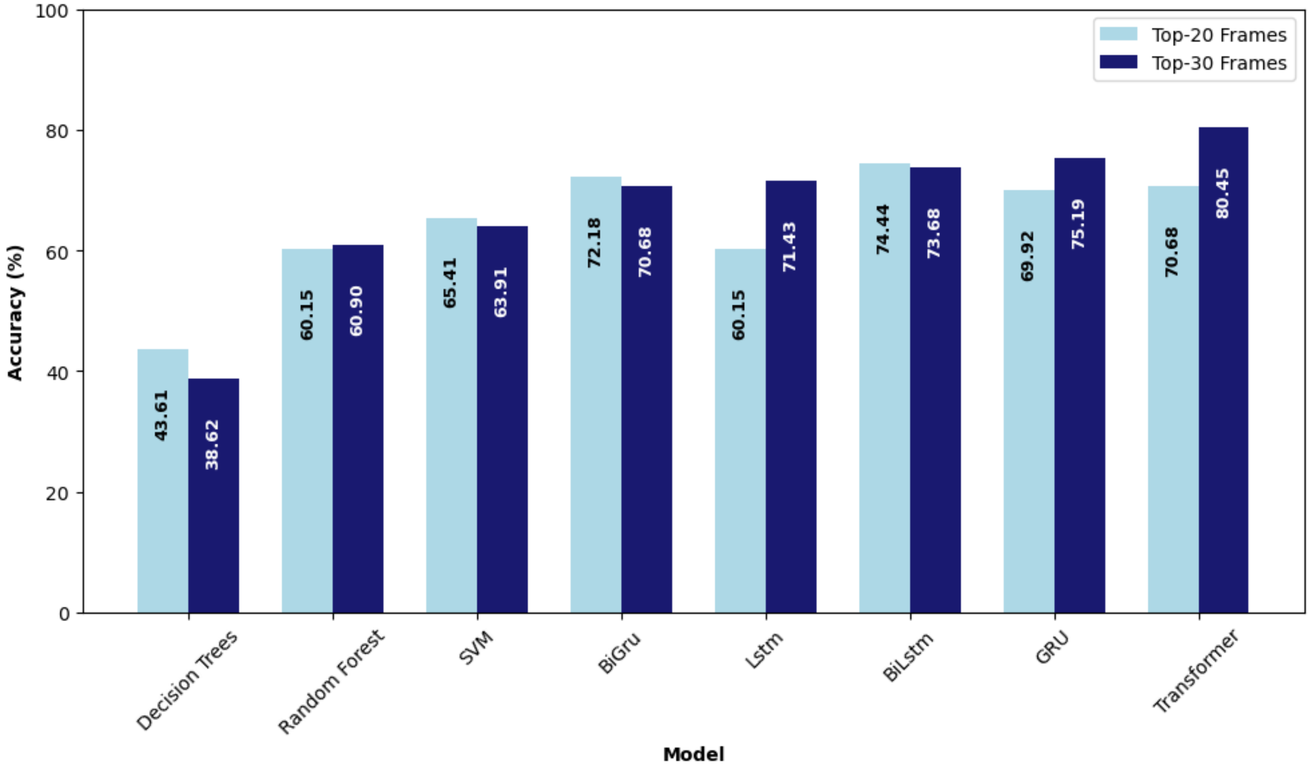


Fig. 18. Influence of Number of Key Frames on Model Performance

Finally, To test the influence of the number of key frames selected by the Top-K Key Frame method, we conducted comparative analysis using the 20-frame and the 30-frame subsets on all models, as illustrated in Figure 18. Results always verify that the application of 30 key frames tends to improve accuracy more than the application of 20 frames, implying that the use of more key frames improves performance by providing more contextual information. In particular, the Transformer model performed the highest overall, increasing from 70.68% accuracy using 20 frames up to 80.45% using 30 frames, representing an impressive improvement of 9.77%.

Out of the models that were tested, LSTM demonstrated the most significant improvement of 11.28%, followed by GRU (5.27%), which suggests that these temporal models benefit significantly from the additional temporal context offered by the extra key frames. Nevertheless, there were also several models that performed worse with extra frames, such as Bi-LSTM (-0.76%), Bi-GRU (-1.5%), SVM (-1.5%), and Decision Trees (-4.99%). These models might have been more sensitive to the higher likelihood of overfitting or noise introduced by the extra frames in the case where the extra frames offer redundancy rather than new motion information. [Goodfellow et al. 2016]

This performance trend validates our choice of the 30-frame setup providing the best compromise between computational expense and increased classification accuracy required for accurate Pakistan Sign Language Recognition.

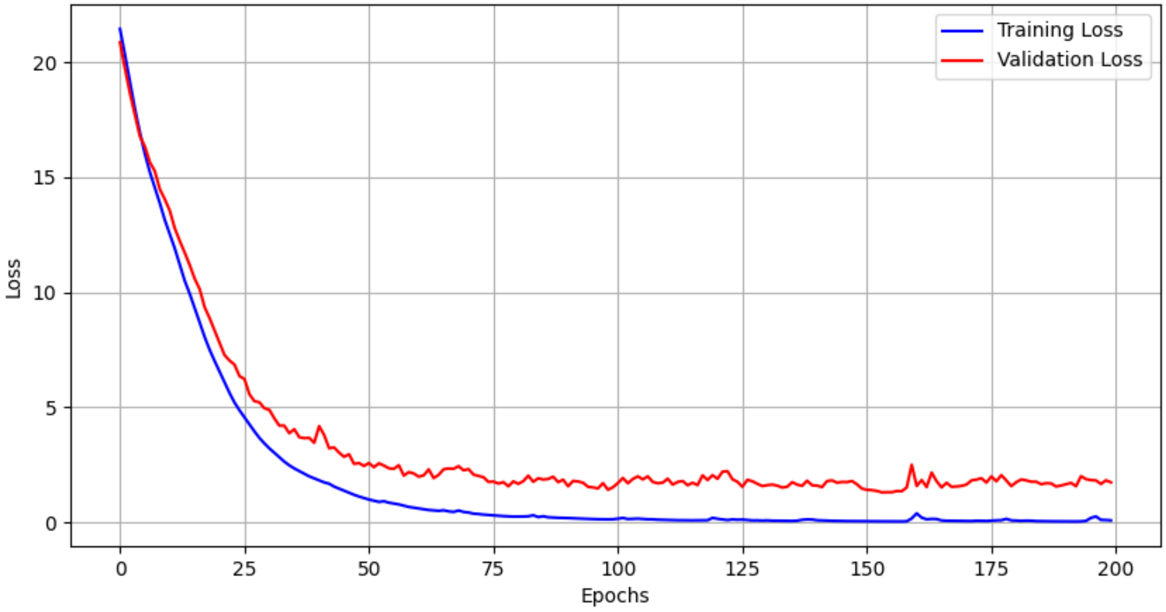


Fig. 19. Training vs Validation Loss Curve

Figure 19 graphs the training and validation loss curves for the optimal Transformer configuration obtained by Optuna optimization after 200 epochs [Akiba et al. 2019]. Both curves show consistent decrease, and the validation loss tightly follows the training loss. Their difference is small, indicating that the model learned with good generalization and little overfitting. The convergence is smooth during the training, and there is no instability, which indicates that the hyperparameters and learning strategy adopted were suitable. This stable convergence behavior is the mark of the effect of adequate regularization and good hyperparameter tuning, which are the essential features of a good deep learning model [Goodfellow et al. 2016].

6.2.2 Comparison with State-of-the-art. :

This section gives an extensive comparison of our proposed framework with existing state-of-the-art techniques, as presented in Table 6.

Different model	**mAP(%)**
I3D + super-events [Piergiovanni and Ryoo 2018]	19.41
ViVit [Arnab et al. 2021]	18.55
MViT (deep network) [Fan et al. 2021]	47.70
I3D + super-events + TGM [Piergiovanni and Ryoo 2019]	22.56
ViT-B-VTN [Neimark et al. 2021]	79.80
I3D + STGCN [Ghosh et al. 2020]	19.09
SLATN [Sameena Javaid 2023]	66.10
Proposed Framework	**87.78**

Table 6. Different models comparison

Our method achieved an **87.78%** Mean Average Precision (mAP), which is significantly better than comparable recent techniques. The proposed architecture brings substantial performance gain over the recent SLATN network of Sameena Javaid [2023], which achieved an mAP of 66.10%. This is a noteworthy gain of around **22%**, reflecting our model's enhanced capability to capture spatial-temporal dynamics from video data. Relative to the previously best-performing method, ViT-B-VTN proposed by Neimark et al. [2021], which had an mAP of 79.80%, our approach yields a considerable improvement of approximately **8%**. Furthermore, performance superiority is even more pronounced in comparison to other notable models such as MViT [Fan et al. 2021] and I3D combined with STGCN [Ghosh et al. 2020], which obtained considerably lower mAP scores of 47.70% and 19.09%, respectively.

These significant performance improvements can be substantially attributed to our effective key video frame selection through a Top-K key frame selection method, wherein we select only the most informative frames. Then we performed effective spatial-temporal feature extraction through ResNet 101 for deep spatial features, combined with Motion Binary Pattern (MBP) for optimally encoding temporal dynamics. Finally, our optimized Transformer architecture keeps strengthening the model in learning to effectively capture long-range temporal dependencies.

However, the slight overfitting observed, as shown by the validation curves presented above in Figure 19, highlights the need for further investigation into this discrepancy.

7 CONCLUSION

7.1 Summary

In conclusion, this dissertation aims to fill in the gaps in Pakistan Sign Language Recognition (PSLR) by offering a robust framework that leveraged advanced deep learning models and feature extraction techniques. As one of the main channels of communication for Pakistan's deaf and hard-of-hearing community, PSL has not been adequately researched because the standardized datasets and full methodologies are not readily available. An extensive review of related work (refer to Section 2) under PSL and related domains such as American, Indian, Arabic, and Korean Sign Language was done to find gaps in the literature.

Building upon these insights, we proposed a robust two-stream approach (refer to Section 3), uniquely integrating deep learning methods for spatial and handcrafted temporal feature extraction. In particular, we have utilized ResNet-101 with Global Average Pooling (GAP) in order to effectively extract sophisticated spatial appearance features of sign gestures. Moreover, To the best of our knowledge, this work represents the first attempt at using a dynamic texture descriptor Motion Binary Pattern (MBP) for recognition of Pakistan Sign Language. Performance was significantly improved by employing an optimized Transformer architecture capable of learning sophisticated long-distance dependencies in sequential data. Our experimental results and ablation study (see Section 5) proved that the fusion between spatial and temporal information significantly improved recognition accuracy, ultimately achieving the mAP score of 84.46%. Our result outperforms the current state-of-the-art methods by far, justifying our introduced framework as innovative and highly.

7.2 Main Limitations of Work

Although the progress made has been significant, our work faced several limitations that need to be recognized. First among these limitations was the small size and narrow scope of the current PSL datasets that limited large-scale validation on varied real-world situations. While strong feature extraction and cross-validation were utilized in attempts to mitigate these shortcomings, the dataset's small variability potentially has implications for generalization in highly unconstrained or dynamic environments.

Secondly, the MBP-LBP temporal feature extractor, while effective, had relatively poorer independent accuracy, indicating that the representation of temporal dynamics alone is insufficient for comprehensive recognition. This emphasizes the importance of further improvement in temporal feature extraction techniques.

Furthermore, computational complexity also remains a limitation as the application of deep neural networks and particularly Transformers brings about high computational cost. While our approach achieved high accuracy, the complexity of the model may become problematic in the case of deployment in resource-constrained environments or for real-time systems.

7.3 Scope for Future Work

Based on these limitations and the promising results obtained, there are several directions for further research. Among the most significant next steps is collecting and creating a larger and more diverse PSL dataset that has greater variability in terms of the demographics of the signers, the dynamics of the signs , the background environments, and the lighting. All this would significantly improve the robustness and generalizability of the models. [Koller et al. 2015]

Another direction that can be pursued is the optimization of the Transformer models for reduced computational complexity [Sanh et al. 2019]. Model distillation, pruning, and lightweight variants of the Transformer models can be investigated in order to provide high accuracy while significantly reducing computational needs for real-time recognition on mobile or embedded platforms. [Sun et al. 2020]

Last but not least, exploring multimodal systems incorporating extra modalities such as depth sensors [Pigou et al. 2015] or Inertial Measurement Units (IMUs) [Neverova et al. 2015] in addition to video information would most likely enhance recognition accuracy and robustness. Incorporating more modalities would provide additional contextual information that would better enable the model to accurately interpret sophisticated PSL signs. [Chen et al. 2024]

In summary, our method provides substantial improvement for Pakistan Sign Language Recognition, and these directions open up promising paths for extending the contribution of the work towards the accessibility, effectiveness, and robustness of PSLR in practice.

REFERENCES

V. Adithya and R. Rajesh. 2020. Hand gestures for emergency situations: A video dataset based on words from Indian sign language. *Data in Brief* 31 (Aug. 2020), 106016. https://doi.org/10.1016/j.dib.2020.106016

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

M. AL-Rousan, K. Assaleh, and A. Tala'a. 2009. Video-based signer-independent Arabic sign language recognition using hidden Markov models. *Applied Soft Computing* 9, 3 (June 2009), 990–999. https://doi.org/10.1016/j.asoc.2009.01.002

Saleh Aly and Walaa Aly. 2020. DeepArSLR: A Novel Signer-Independent Deep Learning Framework for Isolated Arabic Sign Language Gestures Recognition. *IEEE Access* 8 (2020), 83199–83212. https://doi.org/10.1109/ACCESS.2020.2990699

Sarah Alyami, Hamzah Luqman, and Mohammad Hammoudeh. 2024. Isolated Arabic Sign Language Recognition Using a Transformer-based Model and Landmark Keypoints. *ACM Transactions on Asian and Low-Resource Language Information Processing* 23, 1 (Jan. 2024), 1–19. https://doi.org/10.1145/3584984

C. Aparna and M. Geetha. 2020. CNN and Stacked LSTM Model for Indian Sign Language Recognition. In *Machine Learning and Metaheuristics Algorithms, and Applications*, Sabu M. Thampi, Ljiljana Trajkovic, Kuan-Ching Li, Swagatam Das, Michal Wozniak, and Stefano Berretti (Eds.). Springer, Singapore, 126–134. https://doi.org/10.1007/978-981-15-4301-2_10

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6836–6846.

Florian Baumann, Jie Lao, Arne Ehlers, and Bodo Rosenhahn. 2014. Motion Binary Patterns for Action Recognition. In *International conference on pattern recognition applications and methods*, Vol. 2. SCITEPRESS, 385–392. https://doi.org/10.5220/0004816903850392

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, action recognition? a new model and the Kinetics dataset. https://openaccess.thecvf.com/content_cvpr_2017/html/Carreira_Quo_Vadis_Action_CVPR_2017_paper.html

Hao Chen, Jiaze Wang, Ziyu Guo, Jinpeng Li, Donghao Zhou, Bian Wu, Chenyong Guan, Guangyong Chen, and Pheng-Ann Heng. 2024. Signvtcl: multi-modal continuous sign language recognition enhanced by visual-textual contrastive learning. *arXiv preprint arXiv:2401.11847* (2024).

Ming Jin Cheok, Zaid Omar, and Mohamed Hisham Jaward. 2019. A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics* 10, 1 (Jan. 2019), 131–153. https://doi.org/10.1007/s13042-017-0705-5

Soumen Das, Saroj Kr Biswas, and Biswajit Purkayastha. 2023. Automated Indian sign language recognition system by fusing deep and handcrafted feature. *Multimedia Tools and Applications* 82, 11 (May 2023), 16905–16927. https://doi.org/10.1007/s11042-022-14084-4

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6824–6835.

Giles M Foody. 2023. Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient. *Plos one* 18, 10 (2023), e0291908.

Pallabi Ghosh, Yi Yao, Larry Davis, and Ajay Divakaran. 2020. Stacked spatio-temporal graph convolutional networks for action segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 576–585.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

Hafiz Muhammad Hamza and Aamir Wali. 2023. Pakistan sign language recognition: leveraging deep learning models with limited dataset. *Machine Vision and Applications* 34, 5 (July 2023), 71. https://doi.org/10.1007/s00138-023-01429-8

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 770–778. https://doi.org/10.1109/CVPR.2016.90

Sameena Javaid. 2022. PkSLMNM: Pakistan sign language manual and non-manual gestures dataset. https://data.mendeley.com/datasets/m3m9924p3v/2

Joolekha Joolee, Md Uddin, Jawad Khan, Taeyeon Kim, and Young-Koo Lee. 2018. A Novel Lightweight Approach for Video Retrieval on Mobile Augmented Reality Environment. *Applied Sciences* 8, 10 (Oct. 2018), 1860. https://doi.org/10.3390/app8101860

Binwant Kaur, Aastha Chaudhary, Shahina Bano, Yashmita, S.R.N. Reddy, and Rishika Anand. 2023. Fostering inclusivity through effective communication: Real-time sign language to speech conversion system for the deaf and hard-of-hearing community. *Multimedia Tools and Applications* 83, 15 (Oct. 2023), 45859–45880. https://doi.org/10.1007/s11042-023-17372-9

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Oscar Koller, Jens Forster, and Hermann Ney. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding* 141 (2015), 108–125.

Diksha Kumari and Radhey Shyam Anand. 2024. Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism. *Electronics* 13, 77 (March 2024), 1229. https://doi.org/10.3390/electronics13071229

Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, and Hongdong Li. 2020. Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Snowmass Village, CO, USA, 1448–1458. https://doi.org/10.1109/WACV45572.2020.9093512

Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

Muhammad Shaheer Mirza, Sheikh Muhammad Munaf, Fahad Azim, Shahid Ali, and Saad Jawaid Khan. 2022. Vision-based Pakistani sign language recognition using bag-of-words and support vector machines. *Scientific Reports* 12, 1 (Dec. 2022), 21325. https://doi.org/10.1038/s41598-022-15864-6

Anshul Mittal, Pradeep Kumar, Partha Pratim Roy, Raman Balasubramanian, and Bidyut B. Chaudhuri. 2019. A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion. *IEEE Sensors Journal* 19, 16 (Aug. 2019), 7056–7063. https://doi.org/10.1109/JSEN.2019.2909837

Kumar Navendu and Vineet Sahula. 2024. Word Level Sign Language Recognition using MediaPipe and LSTM-GRU Network. *Authorea Preprints* (July 2024). https://doi.org/10.36227/techrxiv.172054945.57389794/v1

Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. 2021. Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3163–3172.

Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. 2015. Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 8 (2015), 1692–1706.

Andrew Y Ng. 2004. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*. 78.

Timo Ojala, Matti Pietikainen, and Topi Maenpaa. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence* 24, 7 (2002), 971–987.

World Health Organization et al. 2021. *World report on hearing* (1st ed ed.). World Health Organization, Geneva.

PADEAF. [n. d.]. Deaf Statistic | PADEAF. https://www.padeaf.org/quick-links/deaf-statistics

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*. Pmlr, 1310–1318.

AJ Piergiovanni and Michael Ryoo. 2019. Temporal gaussian mixture layer for videos. In *International Conference on Machine learning*. PMLR, 5152–5161.

AJ Piergiovanni and Michael S Ryoo. 2018. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5304–5313.

Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. 2015. Sign language recognition using convolutional neural networks. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*. Springer, 572–578.

psl.org.pk. 2020. https://www.psl.org.pk

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (Dec. 2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Safdar Rizvi Sameena Javaid. 2023. A Novel Action Transformer Network for Hybrid Multimodal Sign Language Recognition. *Computers, Materials & Continua* 74, 1 (2023), 523–537. https://doi.org/10.32604/cmc.2023.031924

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

Paul Scovanner, Saad Ali, and Mubarak Shah. 2007. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*. 357–360.

Syed Muhammad Saqlain Shah, Javed I. Khan, Syed Husnain Abbas, and Anwar Ghani. 2023. Symmetric mean binary pattern-based Pakistan sign language recognition using multiclass support vector machines. *Neural Computing and Applications* 35, 1 (Jan. 2023), 949–972. https://doi.org/10.1007/s00521-022-07804-2

Tamer Shanableh, Khaled Assaleh, and Mohammad Al-Rousan. 2007a. Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37, 3 (2007), 641–650.

Tamer Shanableh, Khaled Assaleh, and M. Al-Rousan. 2007b. Spatio-Temporal Feature-Extraction Techniques for Isolated Gesture Recognition in Arabic Sign Language. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37, 3 (June 2007), 641–650. https://doi.org/10.1109/TSMCB.2006.889630

Jungpil Shin, Abu Saleh Musa Miah, Kota Suzuki, Koki Hirooka, and Md. Al Mehedi Hasan. 2023a. Dynamic Korean Sign Language Recognition Using Pose Estimation Based and Attention-Based Neural Network. *IEEE Access* 11 (2023), 143501–143513. https://doi.org/10.1109/ACCESS.2023.3343404

Jungpil Shin, Abu Saleh Musa Miah, Md. Al Mehedi Hasan, Koki Hirooka, Kota Suzuki, Hyoun-Sup Lee, and Si-Woong Jang. 2023b. Korean Sign Language Recognition Using Transformer-Based Deep Neural Network. *Applied Sciences* 13, 55 (Feb. 2023), 3029. https://doi.org/10.3390/app13053029

Seongjoo Shin, Youngmi Baek, Jinhee Lee, Yongsoon Eun, and Sang Hyuk Son. 2017. Korean sign language recognition using EMG and IMU sensors based on group-dependent NN models. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1–7. https://doi.org/10.1109/SSCI.2017.8280908

Ala Addin I. Sidig, Hamzah Luqman, Sabri Mahmoud, and Mohamed Mohandes. 2021. KArSL: Arabic Sign Language Database. *ACM Transactions on Asian and Low-Resource Language Information Processing* 20, 1 (Jan. 2021), 14:1–14:19. https://doi.org/10.1145/3423420

Venus Singla, Seema Bawa, and Jasmeet Singh. 2024. Enhancing Indian sign language recognition through data augmentation and visual transformer. *Neural Computing and Applications* 36, 24 (Aug. 2024), 15103–15116. https://doi.org/10.1007/s00521-024-09845-1

Advaith Sridhar, Rohith Gandhi Ganesan, Pratyush Kumar, and Mitesh Khapra. 2020. INCLUDE: A Large Scale Dataset for Indian Sign Language Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*. Association for Computing Machinery, New York, NY, USA, 1366–1375. https://doi.org/10.1145/3394171.3413528

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984* (2020).

Vaibhav Tiwari, Rakesh Chandra Joshi, and Malay Kishore Dutta. 2022. Deep neural network for multi-class classification of medicinal plant leaves. *Expert Systems* 39, 8 (2022), e13041. https://doi.org/10.1111/exsy.13041

Ba Tu Truong and Svetha Venkatesh. 2007. Video abstraction: A systematic review and classification. *ACM transactions on multimedia computing, communications, and applications (TOMM)* 3, 1 (2007), 3–es.

Adithya V and Rajesh R. 2021. A Video Dataset of the Hand Gestures of Indian Sign Language Words used in Emergency Situations. *Data in Brief* 1 (Aug. 2021). https://doi.org/10.17632/2vfdm42337.1

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. In *International conference on machine learning*. PMLR, 1747–1756.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need, In Advances in Neural Information Processing Systems. *Advances in Neural Information Processing Systems* 30. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 22–31.

Seunghan Yang, Seungjun Jung, Heekwang Kang, and Changick Kim. 2020. The Korean Sign Language Dataset for Action Recognition. In *MultiMedia Modeling*, Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve (Eds.). Springer International Publishing, Cham, 532–542. https://doi.org/10.1007/978-3-030-37731-1_43

Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 271–278.

Guoying Zhao and Matti Pietikainen. 2007. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence* 29, 6 (2007), 915–928.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2020. Spatial-temporal multi-cue network for continuous sign language recognition. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 13009–13016.

A PROJECT MANAGEMENT

This section highlights our project plan toward this study which includes the setting of milestones, and monitoring progress towards the attainment of objectives. We have organized the plan around key deliverables associated with timelines for timely completion in the form of gantt charts. In addition, we have identified some of the risks that could be associated with this project and proactively suggest mitigation strategies as a table. The comprehensive approach shall guide the project in its various phases while focusing on quality and objective accomplishment.

A.1 Project Scope

The scope of this project goes toward the development and proposal of a reliable framework for Pakistan Sign Language (PSL) recognition by addressing significant gaps in PSL research, for example the limited use of hybrid methodologies and algorithms that have not been used in the realm of sign language recognition, particularly PSL. This architecture design emphasizes the recognition of PSL signs with high accuracy from the video data, taking a pre-processing step through Top-K Key Frame Extraction, followed by spatial feature extraction with ResNet-101, while temporal analysis is achieved via MBP. Sequential classification is realized through transformer. This research aims to contribute to the advancement of recognizing PSL and help in promoting communication within the deaf and hard-of-hearing community in Pakistan.

A.2 Project Deliverables

This project is divided into a set of deliverables, which are to be delivered at their respective deadlines. All said tasks will be completed in a timely and organized manner for the completion of the project. The deliverables are listed below:

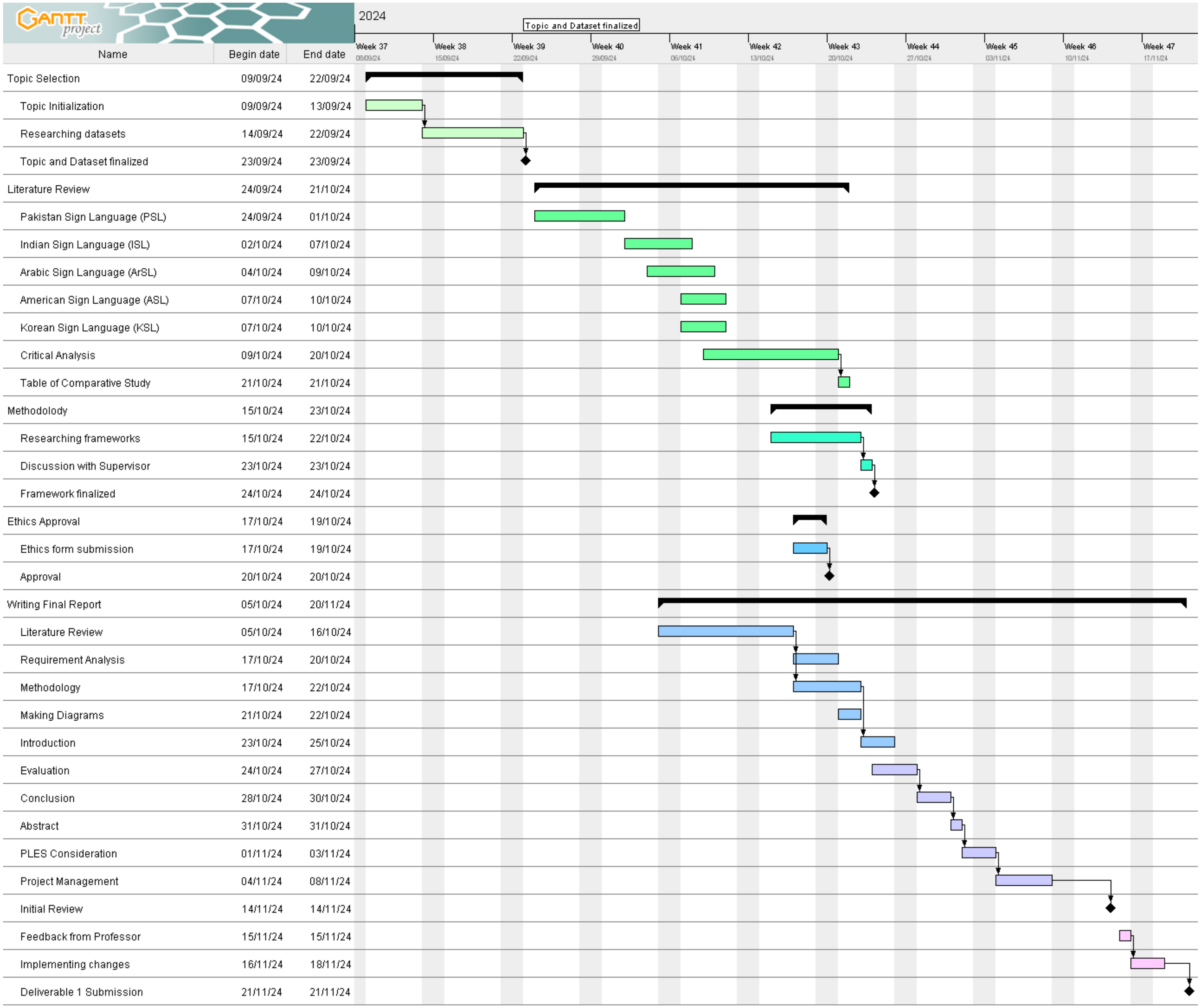
- Research Report (Semester 1): The first deliverable is the Research Report containing the fundamental layout at the start of the project. This report shall contain the aims and objectives of the project, literature review of related works, detailed requirement analysis, overview of the methodology, lays down an approach toward evaluation, and gives a preliminary timeline for the project in order to set directions for subsequent work. This report also addresses professional, legal, ethical and social issues.

- Dissertation Report (Semester 2): The second deliverable is a comprehensive dissertation that documents the entire lifecycle of the project. It shall contain an in-depth description of the methodology used, the evaluation results, and a discussion of the findings in light of related work while also highlighting the project's contributions and areas for future research.

- Viva/Poster Presentation: The last deliverables of the project are the poster and mini-viva. The poster is a simple overview of the project, a description of the main objectives, methodologies, results, and conclusions. The mini-viva extends this by allowing an in-depth discussion and explanation of the work to be presented, questions to be answered, and the significance of the work to be outlined. Each contributes to another in presenting the work both effectively and easily understood.

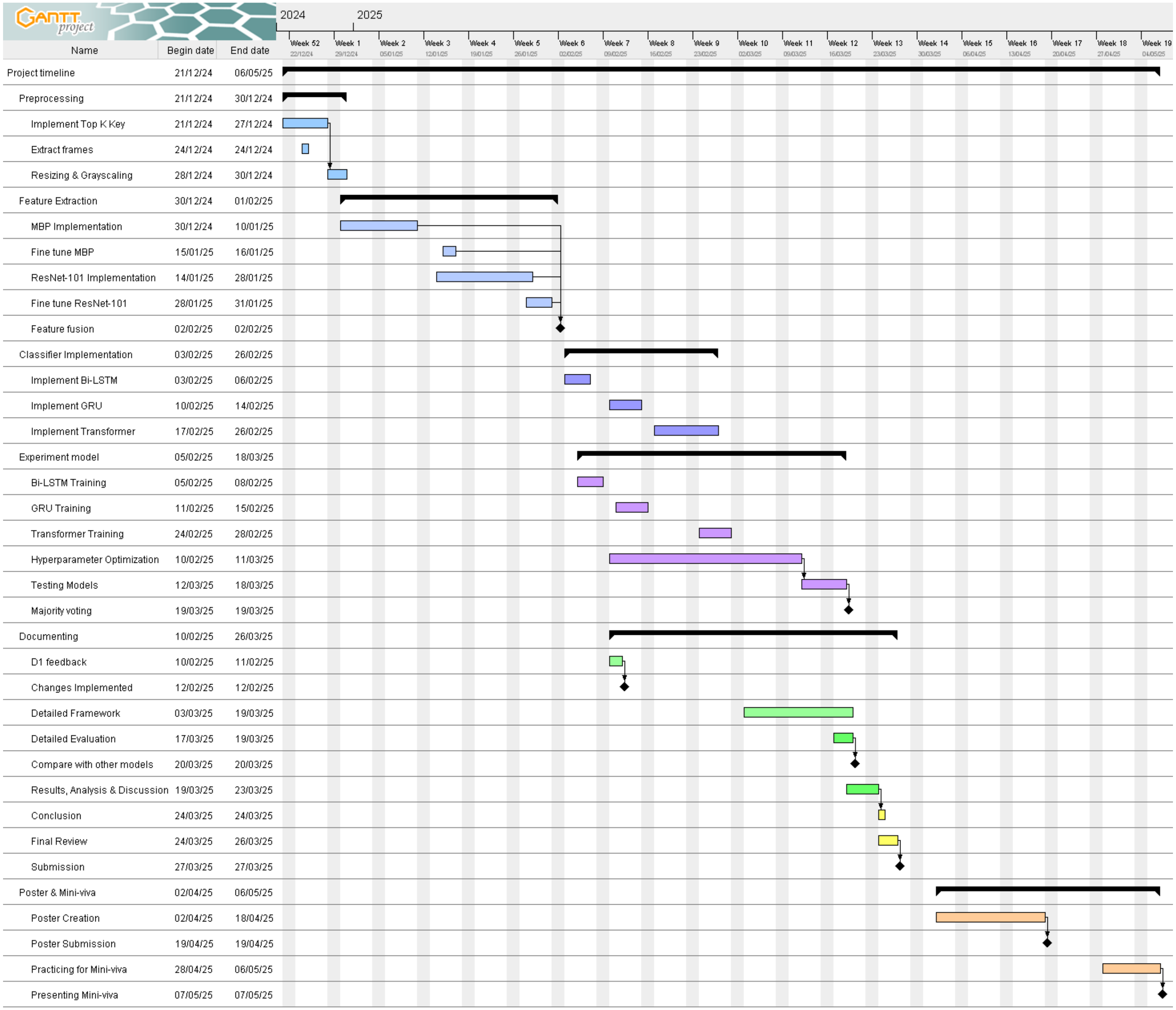
A.3 Project Plan

The project plan accounts for the main tasks, deadlines, and also milestones. This is shown via gantt charts in Figure 20 and Figure 21 which depicts them in phases, starting with the initialization of the project right through to the end submission for both deliverables. The first chart covers the first deliverable, and all the tasks that were covered and will be covered during that period will be listed accordingly for easy view. The second chart will be an approximation of how things are meant to come along according to set time period for each task for the next semester.



Name	Begin date	End date
Topic Selection	09/09/24	22/09/24
Topic Initialization	09/09/24	13/09/24
Researching datasets	14/09/24	22/09/24
Topic and Dataset finalized	23/09/24	23/09/24
Literature Review	24/09/24	21/10/24
Pakistan Sign Language (PSL)	24/09/24	01/10/24
Indian Sign Language (ISL)	02/10/24	07/10/24
Arabic Sign Language (ArSL)	04/10/24	09/10/24
American Sign Language (ASL)	07/10/24	10/10/24
Korean Sign Language (KSL)	07/10/24	10/10/24
Critical Analysis	09/10/24	20/10/24
Table of Comparative Study	21/10/24	21/10/24
Methodolody	15/10/24	23/10/24
Researching frameworks	15/10/24	22/10/24
Discussion with Supervisor	23/10/24	23/10/24
Framework finalized	24/10/24	24/10/24
Ethics Approval	17/10/24	19/10/24
Ethics form submission	17/10/24	19/10/24
Approval	20/10/24	20/10/24
Writing Final Report	05/10/24	20/11/24
Literature Review	05/10/24	16/10/24
Requirement Analysis	17/10/24	20/10/24
Methodology	17/10/24	22/10/24
Making Diagrams	21/10/24	22/10/24
Introduction	23/10/24	25/10/24
Evaluation	24/10/24	27/10/24
Conclusion	28/10/24	30/10/24
Abstract	31/10/24	31/10/24
PLES Consideration	01/11/24	03/11/24
Project Management	04/11/24	08/11/24
Initial Review	14/11/24	14/11/24
Feedback from Professor	15/11/24	15/11/24
Implementing changes	16/11/24	18/11/24
Deliverable 1 Submission	21/11/24	21/11/24

Fig. 20. Gantt Chart for Semester 1



Name	Begin date	End date
Project timeline	21/12/24	06/05/25
Preprocessing	21/12/24	30/12/24
Implement Top K Key	21/12/24	27/12/24
Extract frames	24/12/24	24/12/24
Resizing & Grayscaling	28/12/24	30/12/24
Feature Extraction	30/12/24	01/02/25
MBP Implementation	30/12/24	10/01/25
Fine tune MBP	15/01/25	16/01/25
ResNet-101 Implementation	14/01/25	28/01/25
Fine tune ResNet-101	28/01/25	31/01/25
Feature fusion	02/02/25	02/02/25
Classifier Implementation	03/02/25	26/02/25
Implement Bi-LSTM	03/02/25	06/02/25
Implement GRU	10/02/25	14/02/25
Implement Transformer	17/02/25	26/02/25
Experiment model	05/02/25	18/03/25
Bi-LSTM Training	05/02/25	08/02/25
GRU Training	11/02/25	15/02/25
Transformer Training	24/02/25	28/02/25
Hyperparameter Optimization	10/02/25	11/03/25
Testing Models	12/03/25	18/03/25
Majority voting	19/03/25	19/03/25
Documenting	10/02/25	26/03/25
D1 feedback	10/02/25	11/02/25
Changes Implemented	12/02/25	12/02/25
Detailed Framework	03/03/25	19/03/25
Detailed Evaluation	17/03/25	19/03/25
Compare with other models	20/03/25	20/03/25
Results, Analysis & Discussion	19/03/25	23/03/25
Conclusion	24/03/25	24/03/25
Final Review	24/03/25	26/03/25
Submission	27/03/25	27/03/25
Poster & Mini-viva	02/04/25	06/05/25
Poster Creation	02/04/25	18/04/25
Poster Submission	19/04/25	19/04/25
Practicing for Mini-viva	28/04/25	06/05/25
Presenting Mini-viva	07/05/25	07/05/25

Fig. 21. Gantt Chart for Semester 2

A.4 Risk Analysis

Risk analysis is a vital component in the management of a project, as it locates those potential risks that might impede the successful completion of the project. For each risk identified, two main dimensions have been considered: the likelihood, relating to the opportunity that the risk will occur, and the impact, which describes the magnitude of the damage if the risk actually happens. These can be combined to provide a means of categorizing risks so that mitigation efforts can be targeted effectively. For instance, highly likely risks with a high impact-threats such as overfitting due to limited data require an immediate and intense mitigation strategy. A low-likelihood risk that does have a high impact, such as the failure of a system, requires the implementation of precautionary measures beforehand, for instance through periodic backups, to limit damage. Table 7 below summarizes key identified risks for this project, their likelihood, impact, and corresponding mitigation strategies.

Risk	**Likelihood**	**Impact**	**Mitigation Strategy**
Main system(s) stops working	Unlikely	Very High	Regularly back up work to multiple locations (external devices or cloud storage).
Data loss or corruption	Unlikely	Very High	Keep track of progress by using version control.
Overfitting due to limited data	Likely	High	Implement data augmentation techniques and cross-validation.
Incompatibility between software tools	Possible	Medium	Test the software during initial setup extensively and use proper versions.
Ethics form gets disapproved	Unlikely	High	Communicate with supervisor to ensure validity of the ethics form.
Low performance of the proposed framework	Likely	High	Fine-tune the hyperparameters or increase the training set size.

Table 7. Risk Analysis Table

B PROFESSIONAL, LEGAL, ETHICAL AND SOCIAL ISSUES

B.1 Professional Issues

The proposed framework is implemented using Python and by using only licensed open-source software and libraries some of which include TensorFlow, Pandas and Keras. Any models or libraries used within the research is correctly referenced with regard to their terms and conditions. This project follows the code of conduct as specified by institutions such as the British Computing Society.

B.2 Legal Issues

This project does not store any personal information or any user data. It strictly follows data protection laws and regulations of the United Arab Emirates, Pakistan's Personal Data Protection Act (PDPA) and relevant General Data Protection Regulation (GDPR) guidelines. The data is stored locally and under the rules and guidelines of the university and is not misused in any way. The relevant files have not been distributed to any system other than the author's system(s). The proposed framework does not and will not violate any laws.

B.3 Ethical Issues

In this research based study we only used the publicly available PkSLMNM dataset introduced by Sameena Javaid [2023] and does not have any human subjects. We have not involved any external human user-based interaction or feedback during this project. The performance of the framework has been only validated on empirical data alone. Each sign category is treated equally during training to avoid bias. The study ensure that no ethical policies and principles are violated in any way shape or form.

B.4 Social Issues

The purpose of this study is to develop a sign language recognition system specifically for PSL. Since this project is strictly technical, it does not introduce any sensitive or controversial social issues. The project aims to enhance PSL recognition to support the deaf and hard-of-hearing community without any potential for misuse or ethical conflicts. It is important to note that this research does not address, nor does it introduce, any broader social or moral issues.