TWO-WAY REAL-TIME SIGN LANGUAGE CONVERTER

by

Aksanur Konuk
Duru Baştunalı
Nidanur Demirhan

CSE4197 Engineering Project 1

Project Specification Document

Supervised by:
Assoc. Prof. Mustafa Ağaoğlu

Marmara University, Faculty of Engineering

Computer Engineering Department

15.11.2024

TABLE OF CONTENTS

1 PROBLEM STATEMENT

Hearing and hearing-impaired individuals face communication challenges between each other due to a lack of knowledge of sign language. With sign languages varying from country to country, there are currently no comprehensive solutions for Turkish hearing-impaired citizens to be a part of the society without relying on other people’s knowledge on Turkish Sign Language (TSL). This problem results in noncommunication between hearing impaired people and hearing people.

2 PROBLEM DESCRIPTION AND MOTIVATION

According to the World Report on Hearing by the World Health Organization, 5.50% of the global population, approximately 430 million, has moderate or higher levels of hearing loss [1]. In Turkey, the rate of individuals with hearing problems in 2022 is 3.40% [2]. This challenge can lead to social isolation, impacting individuals' professional, educational, and daily lives.

Sign language can be regarded as a solution for effective communication and is a tool that allows communication with hand, facial, and body expressions. More than 300 sign languages are used worldwide [3], and sign languages have their own grammatical structure. Moreover, sign languages may vary from country to country and sometimes even from region to region.

In Turkey, the current TSL Dictionary [4] consists of the 2,000 most used words. TSL is being progressively systematized. However, there are problems with the dissemination of sign language and raising awareness in society. The low number of individuals who know sign language is considered a factor that makes it difficult for hearing-impaired individuals to communicate effectively with the rest of society.

Current technologies offer various solutions to reduce the communication barriers between signers, individuals who communicate using sign language, and non-signers. A significant portion of these technologies focuses on converting sign language-to-text. However, these solutions are limited to one-way communication and far from providing a suitable environment for active communication. There is a lack of comprehensive technology that provides two-way real-time translation of speech and sign language.

This project aims to develop an effective application solution by eliminating the shortcomings of existing solutions and provides real-time conversion of sign-to-speech and vice versa. With two-way translation, we aim to create an effective communication environment among users of sign language and those who are unfamiliar with it. In this way, it will help reduce the obstacles to the socialization of hearing-impaired individuals.

In this project, techniques such as image processing, and deep learning will be utilized. We will detect the sign language/speech first, then the text conversion will be performed. Finally, the conversion process will be completed, resulting in the generation of appropriate output. During the detection process, in addition to the hand, body posture and facial expressions will be analyzed to increase accuracy.

3 MAIN GOAL AND OBJECTIVES

The main goal of the project is to create a real-time system where signers and non-signers can easily communicate without using any additional equipment.

Our objectives are as follows:

Objective 1: To recognize the sign language from the upper body movements and convert these signs into the corresponding Turkish expression.

Objective 2: To display the spoken sentence in sign language using animation.

Objective 3: To develop an Android mobile app for two-way translation.

4 RELATED WORK

4.1 Ankara University Turkish Sign Language Dataset (AUTSL)

In this study [5], a new large-scale, multimodal TSL dataset has been created. The dataset consists of 226 signs performed by 43 individuals and contains a total of 38,336 videos. Samples include various histories recorded both indoors and outdoors with different signer locations. These samples were recorded using Microsoft Kinect v2, which captures colour image, depth and skeletal data. Training and test sets have been prepared for user-independent evaluations of the models. Convolutional Neural Networks (CNNs), unidirectional and bidirectional Long-Short-Term Memory (LSTM) models were trained as basic models. The basic models were evaluated in both AUTSL and Montalbano [6] datasets. As a result, 95.95% accuracy and 96.11% accuracy are obtained in these datasets respectively. Although, this study provides a measurement by comparing dataset formation and base model but does not provide a complete translation system.

4.2 Bosphorus Sign 22k Dataset

This work [7] consists of a large database formed by the videos taken of six different individuals performing 744 TSL expressions, also known as glosses. These glosses are chosen from three different domains: health, finance and most used daily words. The videos are captured with 1080p video resolution at 30 frames per second as it is quite important to have a high resolution for better accuracy. For sign language recognition, 3D ResNet architecture with mixed 2D-3D convolutions is used. The key concept of the videos is that they are isolated: All individuals performing gloss expressions in front of a green screen, wearing a black shirt, and being native in TSL cause the lack of diversity compared with the AUTSL dataset [5]. However, considering the large scale of the glosses, Bosphorus Sign Dataset is one of the best datasets for TSL as it can understand more words compared to other works. Additionally, this work also has facial and upper body motion recognition as well as mouthing, which is another advantage. However, they lack text to sign language conversion and speech recognition.

4.3 Regional-CNN-Based Turkish Sign Language Recognition

This work [8] aims to recognize the letters of the TSL alphabet by using Region-based CNN. They emphasize on how different the letters are represented in TSL when compared to American Sign Language (ASL). In TSL, signers need to use both hands for letters. Moreover, having special characters such as “Ç, Ğ, İ, Ö, Ş, Ü” in the Turkish Alphabet is where the actual diversity lies. The dataset is formed by three individuals wearing a fixed black shirt in front of a green screen, which is similar to the BosphorusSign22 [7] dataset as both are isolated. In their case, the region of interest only consists of an individual’s hands. However, we aim to have our region of interest in a person's facial expression and body language to have a better translation of sign language to speech. Although we are going to have word recognition, there are points where our project should also detect the words letter by letter, such as spelling a person’s name. Unfortunately, they lack word, face, upper body motion and speech recognition as well as text to sign language conversion.

4.4 Sign Language Recognition with Optimized Deep Learning

This research [9] provides a deep learning model with CNNSa-LSTM for the problem of the sign language detection process. This model combines CNN, Self Attention, and LSTM methods. The spatial, geometric-based, and motion features are extracted using VGG16 and Optical Flow. Hyperparameters of the model are optimized with the Hippopotamus Optimization Algorithm and Pathfinder Algorithm. This research achieves 98.70% accuracy on the American Sign Language Digit Dataset. This project overlaps with some of our goals. While our project aims to provide real-time two-way translation, this project only provides one-way translation. In addition, while this project focuses only on hand movements in the recognition phase, our project aims to make a more comprehensive analysis including hand, upper body and face tracking.

4.5 Bi-directional Translation Methods of ASL and Speech/Text

In this study [10], an application that provides spoken language and ASL conversion with various input types has been created. This application provides the user with the ability to perform sign language using a camera or a sensorized glove. Then, it converts the received inputs into text and speech by using logistic regression or Inflated 3D (I3D) CNN methods according to their type. At the same time, it creates Unity Animation files to translate the input received from the user as text or speech into sign language animation after parsing with Natural Language Processing and Python spaCy Library. Thus, it offers users a two-way translation. Word-Level ASL consisting of 100 words was used for video classification. However, despite offering different options, this study achieved only 66.00% accuracy in word-level ASL to speech translation using camera.

Table 1 provides a summary of the analyses of related works.

Table 1 : Comparison of Related Work

Related Work / Parameter	AUTSL	Bosphorus	R-CNN Based TSL	Deep Learning Network	Bi-Directional ASL	Our Project
Sign Language to Text	Yes	Yes	Yes	Yes	Yes	Yes
Text to Sign Language	No	No	No	No	Yes	Yes
Objects Recognized	Word	Word	Letter	Digit	Word, Letter	Word, Letter, Digit
Face Recognition	No	Yes	No	No	No	Yes
Speech Recognition	No	No	No	No	Yes	Yes
Dataset Isolated	No	Yes	Yes	No	Yes	No
Dataset Size	226 Words	744 Words	29 Letters	10 Digit	100 Words	750 Words, 29 Letters, 10 Digits
Created Own Dataset	Yes	Yes	Yes	No	No	No
Sign Language	Turkish	Turkish	Turkish	Indian	American	Turkish
Technics Used	CNN, BLSTM	3D ResNet	R-CNN	CNN, LSTM	I3D CNN	CNN, LSTM-GRU
Mobile App	No	No	No	No	Yes	Yes

5 SCOPE

This project focuses on Turkish and Turkish Sign Language and has two main aspects for the communication process of signers and non-signers. Signers will be able to translate the sign language expressions into speech using our application. Our application will detect sign language expressions in real time via smartphone camera. The detected expressions will be converted into text. Then, this text will be converted into audio and vocalized by the application. In addition, we will display the text to user.

The second aspect of the project is the process of translating the speech into an equivalent sign language expression. Our application will recognize voice via microphone. This speech will first be converted into text. Then, the text will be translated into sign language expressions. This process will be achieved using the expressions created by 3D avatar model.

Dataset also plays a crucial role in addition to the techniques used. Our project aims to provide real-time translation at the daily conversation level. In this context, the AUTSL [5] dataset and BosphorusSign22 dataset [7] will be used as the dataset. This project will be developed as a mobile app that provides a real-time two-way sign language converter for users.

Assumptions:

- Translation between Turkish and TSL will be provided. Other spoken languages and sign languages will not be supported.
- The signer should hold the phone from the front view in such an angle where the upper body part is completely visible for sign-to-text/speech conversion.
- The camera of the android device should have at least 720p to recognize sign language accurately.
- The microphone of the android device should transmit audio at a minimum of 16 kHz sampling rate and 16-bit resolution to recognize speech.

Constraints:

- The project is only supported in Android mobile application platforms. No other platforms such as desktops, iOS are supported.

- The dataset does not include all words in the Turkish language; it only contains commonly used daily words provided by given datasets.

6 METHODOLOGY AND TECHNICAL APPROACH

6.1 Data Collection and Preprocessing

A dataset is needed to train the model for translation from sign language to spoken language. Compared to other data sets, we prefer to use the AUTSL dataset [5] and the BoshorusSign22 dataset [7] since they were created in TSL and the number of signs they contain. Additionally, their variety of people and environments offers a wider scope. Since these datasets are already normalized, we do not require additional normalization.

6.2 Feature Recognition

The key to sign language conversion relies on recognizing the body and the hand movements and the facial expressions of the signer. To achieve this, we are planning to use MediaPipe Framework, which is known for its abilities in processing real-time media such as videos, photographs, texts, and audio. Regarding our project, the MediaPipe will serve as a tool to recognize faces, hand gestures and body motions from videos using computer vision and machine learning pipelines [11]. For this recognition, we will use landmarking process to detect a person’s structure and motion. This process consists of landmarks, which are points of a human’s body mainly the joints and edges.

In hand marking, knuckles, fingertips and the point of wrist are detected to set the landmarks and provide the information of movement and position in real time. These landmarks consist of 21 different hand points.

The hand landmarking points are shown in Figure 1.

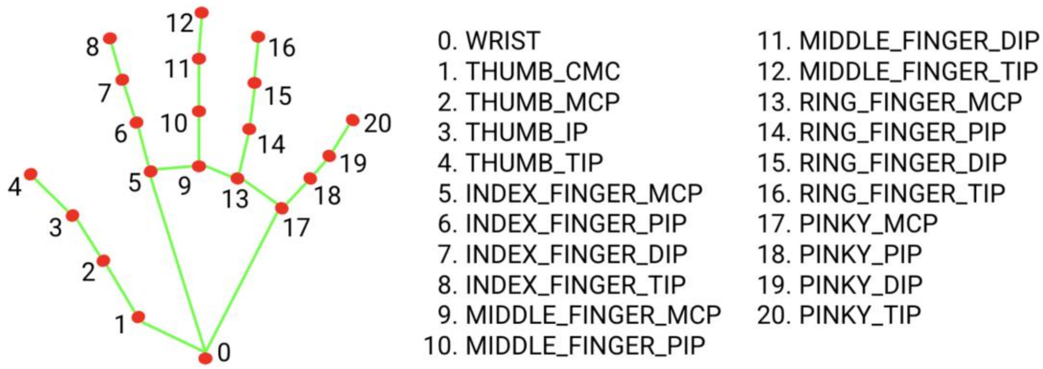

Figure 1 : Hand Landmarking [11]

To detect human body points, we will use Pose Landmarker to detect the joints and edges from real-time videos. Additionally, we are going to implement the Pose Detection Model to detect the presence of bodies and map the body in thirty-three points. The body landmarking points are shown in Figure 2.

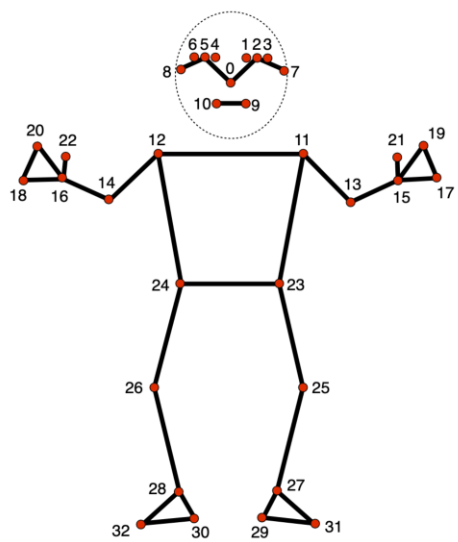

Figure 2 : Body Landmarking [11]

To detect human face points, we will use the Face Landmarker model, which is a combinational tool of various models. By using these different models, we are planning to achieve three different tasks in face landmarking: detecting faces with rough landmarking, completing the landmarks by mapping the face in more detail, and lastly receiving an output by predicting the facial expression. The landmarked output of a human face is shown in Figure 3.

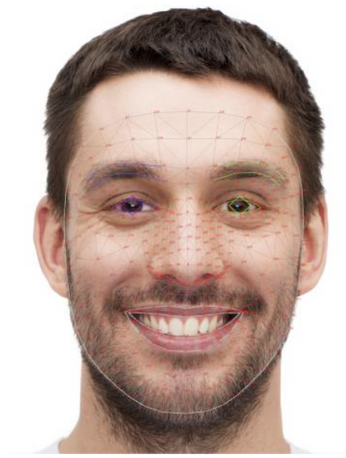

Figure 3 : Face Landmarking [11]

6.3 Model Architecture

We will translate sign-to-text by using a model consisting of various deep-learning techniques. This model comprises CNN, LSTM-Gated Recurrent Units (GRU), and Self-Attention mechanisms. Each algorithm in the model has a different purpose. Spatial features of hand, face, and upper body positions play a fundamental role in understanding sign language. We will use CNN to capture spatial feature of the frames in the video. These spatial features are transferred to the LSTM-GRU. We will use LSTM-GRU to handle the temporal dynamic of sign language, helping the model to understand how sign language expressions change over time. This method allows the model to understand the movement changes of sign language expressions. Additionally, we will use Self-Attention mechanism to identify which frame in the video is most relevant. This mechanism prevents the model from focusing on unnecessary frames and prioritizes frames that are appropriate for the context. The aim is to create a model that includes these functions to make a more accurate and reliable translation [9].

- **CNN:** CNN is widely recognized technique of deep learning for image recognition and processing. It is formed from multiple layers, convolutional layers, pooling layers, and fully connected layers. The patterns such as hand points and gestures are detected by applying filters in the convolutional layers. Pooling layers reduce the spatial data dimensions and preserve significant information. The fully connected layers make a prediction based on what was learned by the previous layers [12]. Consequently, CNN extracts spatial characteristics from each frame in the video. In this project, it is used to obtain hand, face, and upper body position information, and classify them.

- **LSTM-GRU:** Sign-to-text conversion has a dynamic structure rather than a static one. For that reason, CNN alone is not sufficient for that conversion. The LSTM and GRU are developed to solve vanishing descent problem in long-term dependencies by implementing gating mechanism. LSTM has three different gates: input gate, forget gate and output gate. The process such as added to, removed from, and output from the memory cell is determined by these gates. GRU is a simpler version of the LSTM. It has update and reset gates. While the update gate determines how much of new information is changed with previous information in the hidden state. It has been decided that the LSTM-GRU structure should be applied for efficiency based on the results of the research in [13].

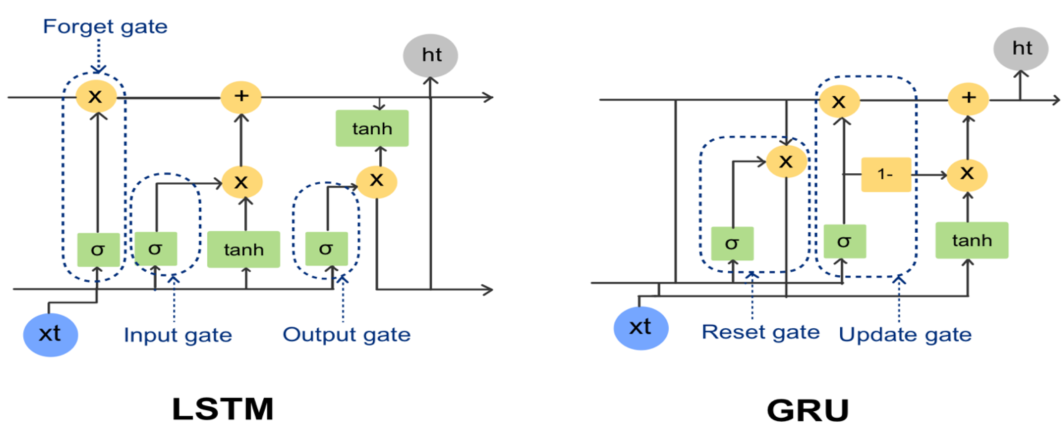

Figure 4 : Structure of LSTM and GRU

- **Self-Attention:** Self Attention determines the weights of the elements in the frame of the video dynamically and it makes it easier to match the related frames. Consequently, it specifies which part of the frame from the video should be focused. As a result, it increases the success of the project by achieving improvement in long-term dependencies.

6.4 Training Avatar

TSL expressions will be displayed with an avatar created in Unity 3D. We will train this avatar for simulating the expressions and integrate it into our application. In this process, the avatar will be provided with the proper physical gestures of TSL glosses. During this process, the avatar's ability to imitate movements will be evaluated and improvements will be made. Thus, we aim to provide more realistic simulation with this avatar model.

6.5 Development of Mobile App

Mobile app will be implemented using Java. The application will have access to the camera and the microphone on the device to capture real-time videos and speech. These videos will be categorized by the machine learning model and the extracted Turkish output will be vocalized through the application. This vocalization from text-to-speech will be done using Google API. Turkish speech will be extracted as text and the equivalent sign language expressions will be displayed on the application’s screen.

6.6 Performance Evaluation and Metrics

Various metrics will be used to evaluate the project. The accuracy, precision, recall, and F1-score metrics will be applied in the project. Using these metrics, we aim to identify the strengths and weaknesses of our project and make necessary improvements.

The processes of our methodology are summarized in Figure 5 and Figure 6.

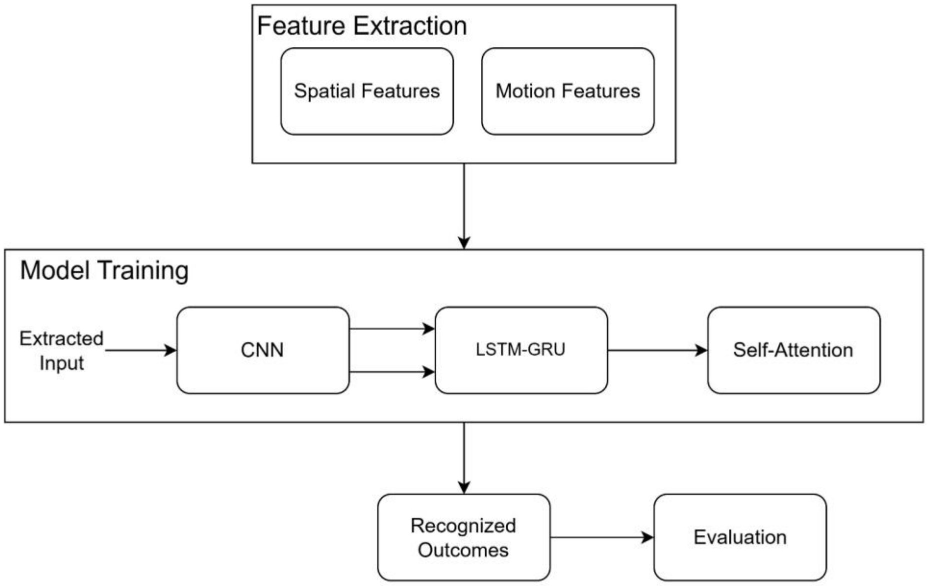

Figure 5 : Flow Chart of Sign-to-Speech Conversion

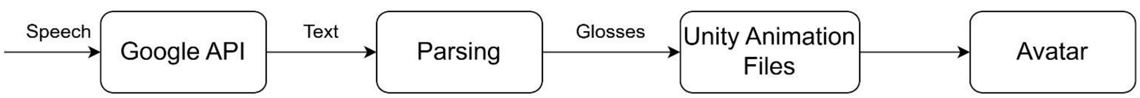

Figure 6 : Flow Chart of Speech-to-Sign Conversion

7 PROFESSIONAL CONSIDERATIONS

7.1 Methodological Considerations/Engineering Standards

To implement our machine learning model, we will use Sci-kit Learn, Matplotlib and NumPy libraries. To operate these models, we are going to use Python. For the speech recognition phase, we are planning to use Google API. To represent TSL expressions, we will create an avatar model with Unity 3D. For database, we will use MySQL to store our dataset.

The program will be implemented on the Android mobile platform using Java. For the Android application, we are going to apply Model-View-View Model (MVVM) architecture, to handle graphical user interface developments and data operations separately.

To systematically work on the project as a team, we are going to use Git and GitHub to version our source codes. To document our process, we are using Google Drive for file sharing. Additionally, we are using Google Docs to work on the same document together. On these documents, we are using Draw.io to add necessary graphics, charts and diagrams. Our communication as a team is handled on WhatsApp and Zoom platforms.

The methodological considerations are summarized in Table 2.

Table 2 : Table of Methodological Standards, Tools and Platforms

Methodological Standards, Tools and Platforms	
Machine Learning Model and Frameworks	• Python Programming Language • MediaPipe Framework • Sci-kit Learn, MatplotLib, NumPy • MySQL
Speech Recognition, Text-to-Sign Language Implementation	• Google API • Unity 3D
Android Mobile Application	• Java Programming Language • MVVM Architecture
Source Code Control	• Git, GitHub
Documentation and Preparation	• Google Drive, Google Docs • Microsoft: Word, Excel, PowerPoint • Draw.io • WhatsApp, Zoom

7.2 Realistic Constraints

7.2.1 Economical

For speech recognition and vocalization, the best tool is Google API, and it has a fixed pricing of $0,01 per minute. Considering that we are working for a good cause for and helping the hearing-impaired minority, it is possible for us to reach this tool free of charge if Google allows.

Moreover, the average pricing of similar products is around 149,90 Turkish Liras per month. To have a better understanding of the market, we might test these products to compare them with our project. On the other hand, since our main aim is to help the minority, we will charge no fees for our project.

7.2.2 Environmental

Our project has no negative issues as it does not affect the environment physically.

7.2.3 Ethical

It is a must for our project to have access to the camera and the microphone on the device, for analyzing personal data. To comply with the Personal Data Protection Law, none of these videos will be stored. Moreover, the project does not need to store the live videos, since the application runs in real time. The frameworks and libraries we use for processing the real time videos are open-source tools and they respect the privacy of their users by openly stating none of the input data is sent through servers.

7.2.4 Health and Safety

Our project has no potential issues that might affect human health and safety. In fact, our project directly helps people and affect their daily lives positively.

7.2.5 Sustainability

As long as there is a need for communication between singers and non-signers, our project will survive and continue to help people to understand each other. To achieve this, we will publish our mobile app on Google Play Store for people to download it.

7.2.6 Social

Our main objective is to help TSL users and non-signers to communicate easily. We are providing our project free of charge to ensure it reaches more people. Considering these facts, our project will have positive outcomes socially and be a part of hearing-impaired people’s daily life.

7.3 Legal Considerations

The frameworks we use are open-source models. While processing the live videos and audios, none of the input data will be stored into the system. Permissions are going to be asked for accessing the camera and the microphone on the device. Personal Data Protection Law (KVKK) will be applied.

8 MANAGEMENT PLAN

8.1 Description of Task Phases

8.1.1 Literature Survey: Literature survey of existing studies, and algorithms about sign languages.

1.1: Academic articles and market products will be researched and analyzed.

1.2: Deep learning algorithms such as CNN, RNN, LSTM, GRU, and tools for computer vision such as MediaPipe, YOLO will be researched and compared with each other.

1.3: Preparing project specification document (PSD).

8.1.2 Data Preparation: Research of open and closed source datasets for analysing hand, face and upper body, and creating our own dataset if these datasets are insufficient.

2.1: Open-source datasets will be collected.

2.2: Closed source will be collected, and we will contact Institutions to obtain necessary permits.

2.3: We will create our own datasets if the open-source dataset and closed-source dataset is not enough.

8.1.3 Tool Implementation and Integration: Implementation of landmark detection, speech recognition, and conversion from text-to-speech.

3.1: MediaPipe will be applied using collected datasets of hand, face, upper body. The results will be analyzed and tested for tracking of hand and facial gestures.

3.2: Google Cloud API will be used for the conversion of speech to text, and text to speech, and the necessary permissions will be obtained.

8.1.4 Avatar Training: Training an avatar in Unity 3D for TSL simulation.

4.1: The avatar is created and trained for TSL gestures.

4.2: The avatar is evaluated, and improvements are made.

8.1.5 Model Development: Implementation of model architecture consisting of CNN, Self Attention, LSTM and GRU.

5.1: Model architecture including CNN, Self Attention, LSTM-GRU will be built and trained for conversion of sign-to-text.

8.1.6 Mobile App Development: Development of Android mobile application.

6.1: Java will be used for the development process of the mobile application.

6.2: The mobile application will be tested for performance, functionality, usability.

8.1.7 Testing and Deployment: Integration and testing of the system components.

7.1: Testing of system integration between the components.

7.2: The model is tested and evaluated for performance on the testing data.

7.3: Perform functional testing to verify accuracy of text-to-speech, and speech-to-sign conversion.

8.2 Roles and Responsibilities Distribution

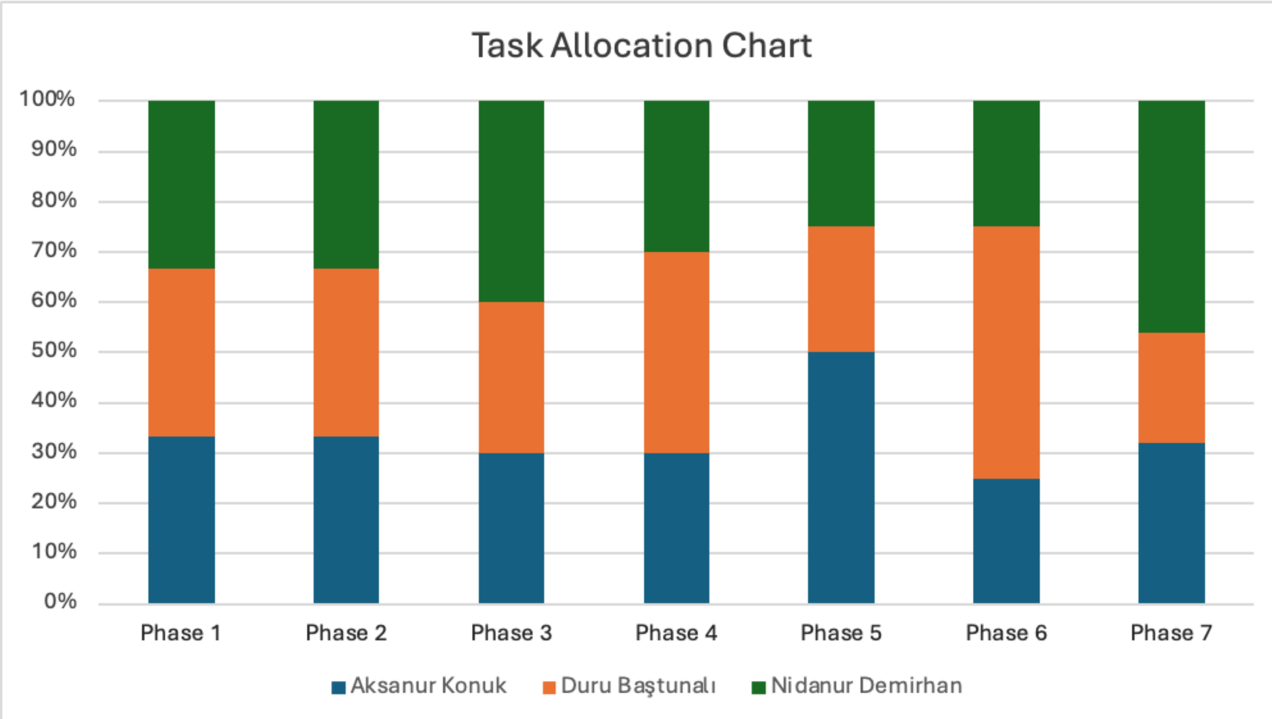

Figure 7 : Task Allocation of Team Members

8.3 Timeline

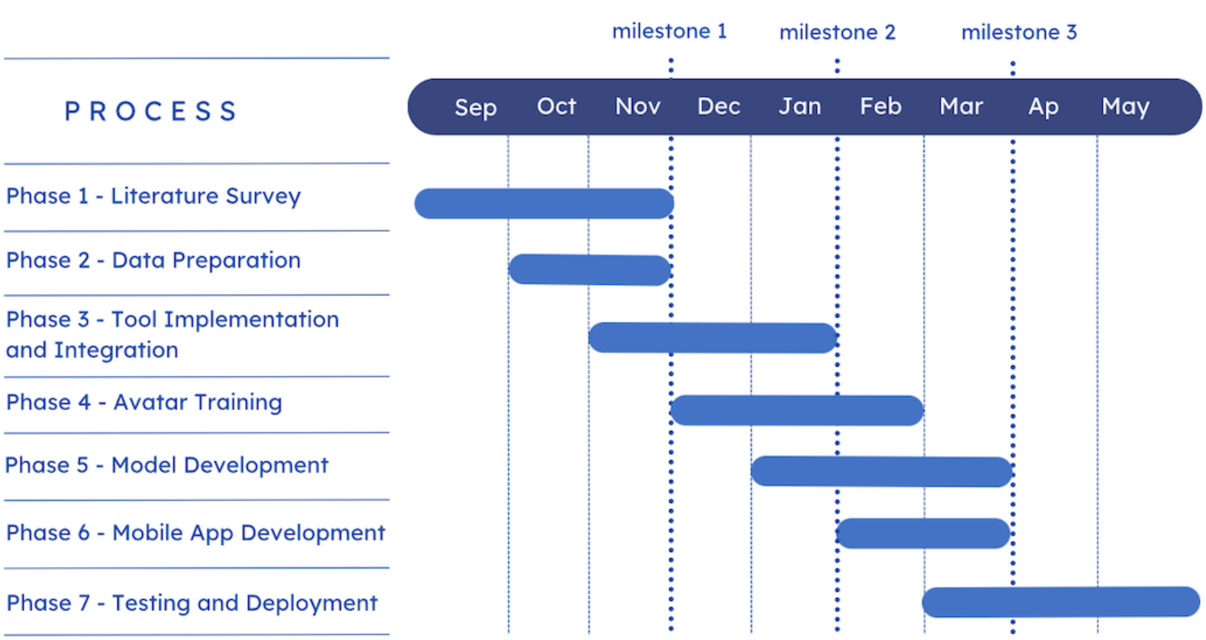

Figure 8 : Timeline of the Project

9 SUCCESS FACTORS AND RISK MANAGEMENT

9.1 Measurability/Measuring Success

The objectives are given in Section 3 and related success factors are listed below:

i. Objective 1: To recognize the sign language displayed by using upper body movements to the smartphone camera and convert these signs into the corresponding Turkish expression.

   Success Factor: Considering previous studies, achieving 90.00% accuracy in the 50-word word group of easy-to-recognize words, 95.00% accuracy at the letter level for proper names and 65.00% accuracy at the expression level.

ii. Objective 2: To display the spoken sentence in sign language using animation.

   Success Factor: To achieve 95.00% accuracy in sign language animation at the letter level corresponding to the spoken expression.

iii. Objective 3: To develop an Android mobile application for two-way translation to the user.

   Success Factor: To deploy the Android application to Google Play Store and to become user-friendly with at least 20 downloads.

9.2 Risk Management

i. Risk 1: The dataset decided to be used for model training may not be accessible.

   Resolution: We will attempt to obtain access to alternative datasets. In the condition that a dataset cannot be accessed, we will create our own dataset for model training within the scope of the project.

ii. Risk 2: The dataset used for the perception of some letters in the Turkish alphabet (ç, ğ etc.) may be insufficient.

Resolution: A new dataset containing all letters of the Turkish alphabet will be found and included for the model training process.

iii. Risk 3: The free access permission of Google Cloud API, which is intended to be used, may not be available.

Resolution: Alternative solutions will be searched. In the absence of a speech/text translation tool with Turkish support, the necessary budget for access to the required API will be supplied.

10 BENEFITS AND IMPACT OF THE PROJECT

The main benefit of this project is to facilitate communication between signers and non-signers by providing a two-way translation system.

i. *Scientific Impact*: The project offers a wider solution to a problem that has not been adequately solved despite previous studies. Since we aim to publish our project as a scientific paper, it will be a reference to future studies in this field.

ii. *Economic/Commercial/Social Impact:* The mobile app will provide an economic contribution to the user since it will be freely downloadable from Google Play Store as opposed to its paid alternatives.

iii. *Potential Impact on New Projects:* A more comprehensive dataset can be created to provide a wider range of use. In addition, it can be used as a base for studies on different sign languages used in other countries.

iv. *Impact on National Security:* Since the project will be domestic and national, it will be preferred against alternatives produced in different countries.

REFERENCES

[1] World Health Organization. 2021. World Report on Hearing. Geneva: World Health Organization.

[2] TUIK. 2022. "*Turkey Health Survey 2022: Distribution of individuals with hearing problems by gender and age group"*. Turkish Statistical Institute: https://data.tuik.gov.tr/Bulten/Index?p=Turkiye-Saglik-Arastirmasi-2022-49747. Date accessed: November 15, 2024.

[3] United Nations. "International Day of Sign Languages". United Nations: https://www.un.org/en/observances/sign-languages-day. Date accessed: 11 November 2024.

[4] Ministry of Family and Social Services. "About the TİD Dictionary Project". 1Date accessed: 11 November 2024.

[5] Sincan, O. M., Keles, H. Y. 2020. AUTSL: A Large Scale Multi-modal TSL Dataset and Baseline Methods. IEEE Access, vol. 8, pp. 181340-181355.

[6] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in European Conference on Computer Vision, pp. 459-473, Springer, 2014.

[7] Özdemir, O., Kındıroğlu, A. A., Camgöz, N. C., Akarun, L. 2020. Bosphorus Sign22k Sign Language Recognition Dataset. Proceedings of the 9th Workshop on the Representation and Processing of Sign Languages, 181–188.

[8] Yırtıcı, T., Yurtkan, K. 2022. Regional-CNN-Based Enhanced Turkish Sign Language Recognition. Signal, Image and Video Processing (2022) 16:1305–1311.

[9] Baihan, A., Alutaibi, A., & Sharma, S. 2024. "Sign Language Recognition using Modified Deep Learning Network and Hybrid Optimization: A Hybrid Optimizer (HO) based optimized CNN-Sa-LSTM Approach".

[10] Elnashar, A., Hamdan, K., Seiari, S. A., Shanableh, Y., Barlas, G. 2023. Bi-Directional Translition Methods of ASL and Speech/Text. Procedia Computer Science 239 (2024) 1879–1886.

[11] Google AI for Developers. "MediaPipe Solutions Guide" Google: https://ai.google.dev/edge/mediapipe/solutions/guide. Date accessed: 24 October 2024.

[12] GeeksforGeeks. "Convolutional Neural Network (CNN) in Machine Learning". GeeksforGeeks: https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/. Date accessed: 24 October 2024.

[13] Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil, A., Corchado Rodríguez, J. 2022. Deepsign: Sign Language Detection and Recognition Using Deep Learning. Electronics. 11. 1780. 10.3390/electronics11111780.