Arabic Saudi sign language Translation

1. Preface

In the following I will present the details of my work in bidirectional translation of Arabic Sign Language (Ar-Sl) and Saudi Sign Language (SSL). As the director of Center for Smart Robotics Research (CS2R) we were conducting research in a system for translation between the deaf and normal people. We were able to build such system. The system consisted of speech to sign system and sign to speech systems. For better user interaction we put the system on a robot that we designed and built specifically for the system. This work on the Arabic Sign Language uses latest techniques in artificial intelligence, machine learning, and deep learning. It is directly related to computerization of Arabic language and services not only the Arabic speaking community but contributes to Sign Language recognition in general.

Due to our expertise in the speech processing group, which became part of CS2R, the speech to sign system was not a big challenge, the sign to speech was more challenging. To conduct research on sign recognition we built a sign database that contained 80 signs performed by 40 signers in 5 repetitions. The signs were recorded in an unconstrained environment which makes recognizing the sign difficult, nonetheless we conducted research using it and got excellent results that we published in ISI journals.

Due to our expertise in SSL translation we were able to secure fund for a project titled “Saudi sign language Translation Companion System”, where I was the PI of the project. This project ended two months ago. The details of our work and accomplishments in this project is included the project report at the end of in this report

As a further work in the area I am supervising a PhD thesis with the title “Sign Language Recognition using Transformer Technique with Multi Modalities Input” with my colleague in the center Prof. Abdulwadood as a co-supervisor.

I am also doing research in an essential point for practical sign translator. The Arabic sign language structure is different from the Arabic language; hence this need a translator between them. Currently I am supervising a Master thesis with the title “A Deep Neural Machine Translation System from Arabic Text to Sign Language” with my colleague in the center Dr. Yousef Alohali. The Arabic sign language is different from the Arabic language and has its own structures and rules. It is not possible to directly translate word by word from an Arabic text into

a sign. Rather, the sentences and phrases in the Arabic texts must be converted into their equivalent in sign language, text which is signed text and is known as Gloss. In this research, the student built a database of children's stories containing the Arabic text and the corresponding sign text, which was translated by certified sign language translators. The student is now applying the latest machine translation techniques, which rely on deep learning methods, and the experiments have given good initial results and are working to improve them.

In the following I will present a list of my publications in the area of Arabic sign language translation, the letter from National Science Technology and Innovation Plan (NSTIP) for funding the project then an abstract of the project, next I will end this report by a detailed report of our work in the project.

List of my publications in Arabic sign language translation

1- Al-Hammadi, Muneer, Mohamed A. Bencherif, Mansour Alsulaiman, Ghulam Muhammad, Mohamed A. Mekhtiche, Wadood Abdul, Yousef A. Alohali, Tareq S. Alrayes, Hassan Mathkour, Mohammed Faisal, Mohammed Algabri, Hamdi Altaheri, Taha Alfakih, and Hamid Ghaleb. 2022. "Spatial Attention-Based 3D Graph Convolutional Neural Network for Sign Language Recognition" Sensors 22, no. 12: 4558. https://doi.org/10.3390/s22124558.

2- Bencherif, Mohamed A., Mohammed Algabri, Mohamed A. Mekhtiche, Mohammed Faisal, Mansour Alsulaiman, Hassan Mathkour, Muneer Al-Hammadi, and Hamid Ghaleb. "Arabic sign language recognition system using 2D hands and body skeleton data." IEEE Access 9 (2021): 59612-59627.

3- Abdul, Wadood, Mansour Alsulaiman, Syed Umar Amin, Mohammed Faisal, Ghulam Muhammad, Fahad R. Albogamy, Mohamed A. Bencherif, and Hamid Ghaleb. "Intelligent real-time Arabic sign language classification using attention-based inception and BiLSTM." Computers and Electrical Engineering 95 (2021): 107395.

4- Al-Hammadi, Muneer, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, Mohammed A. Bencherif, Tareq S. Alrayes, Hassan Mathkour, and Mohamed Amine Mekhtiche. "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation." IEEE Access 8 (2020): 192527-192542.

5- Muneer Al-Hammadi, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, Mohamed A. Bencherif, and Mohamed Amine Mekhtiche, “Hand Gesture Recognition for Sign Language Using 3DCNN,” IEEE Access, vol. 8, no. 1, pp. 79491-79509, December 2020. DOI: 10.1109/ACCESS.2020.2990434

6- Al-Hammadi, Muneer, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, and M. Shamim Hossain. "Hand gesture recognition using 3D-CNN model." *IEEE Consumer Electronics Magazine* 9, no. 1 (2019): 95-101.

7- Hamdi Altaheri, Mohamed A. Bencherif, Mansour Alsulaiman, et al. “KSU-ArSL: Arabic sign language dataset and validation using the latest deep convolutional neural networks”, Heliyon. (Under revision, submitted August 23, 2022, revised 31/5/2023)

8- Mansour Alsulaiman, et al. “Facilitating the Communication with Deaf People: Building a Largest Saudi Sign Language Dataset”, Journal of King Saud University - Computer and Information Sciences. (Under review, submitted February 28, 2023, minor revision received at 5/6/2023).

9- Mohamed Mekhtiche, et al. “Speech/Text to Avatar translator for Saudi Sign Language”, 9th IEEE International Conference on Applied System Innovation 2023 (IEEE ICASI 2023), Japan, 21-25 April 2023. (Accepted and presented)

Letter of National Science Technology and Innovation Plan (NSTIP) about funding the project

جامعة الملك سعود (034)
هاتف +966 11 4694843
فاكس +966 11 4693877

المملكة العربية السعودية
ص.ب 2454 الرياض 11451
www.ksu.edu.sa

جامعة الملك سعود
King Saud University

وكالة الجامعة للدراسات العليا والبحث العلمي
برنامج الخطة الوطنية للعلوم والتقنية والابتكار

إفادة

تفيد الخطة الوطنية للعلوم والتقنية والابتكار بجامعة الملك سعود بان سعادة الدكتور / منصور بن محمد السليمان الأستاذ بكلية علوم الحاسب والمعلومات، رقم وظيفي (١١٩٦٢٣)، هو الباحث الرئيس للمشاريع المبينة بالجدول ادناه، وهذه المشاريع ممولة من برنامج التقنيات الاستراتيجية بالخطة الوطنية للعلوم والتقنية والابتكار:

م	رقم المشروع	اسم المشروع	المدة
١	08-INF167-02-	التعرف على المتحدث العربي ARABIC SPEAKER RECOGNITION	٢٠١٠-٢٠١٢
٢	١٢-MED2474-02	تقييم الامراض الصوتية بالحاسب Automatic Voice Pathology Assessment	٢٠١٣ - ٢٠١٥
٣	3-17-09-001-0003	نظام حاسوبي لتعليم اللغة العربية لغير الناطقين بها Computer-Aided Pronunciation Training System for Non-native Learners of the Arabic Language	٢٠٢٠-٢٠٢٢
٤	5-18-02-001-0003	نظام ترجمة محمول للغة الإشارة السعودية Saudi Sign Language Translation Companion System	٢٠٢٠-٢٠٢٢

وقد اعطيت له هذه الإفادة لسعدته بناء على طلبه لتقديمها الى من يهمه الامر ودون أدنى مسؤولية على الوحده،

مدير وحدة العلوم والتقنية والابتكار

ا. د. أحمد بن عبد الله الخازم

رقم الملف ١/٣/٢٨٦٧٣٤
التاريخ ١٤٤٣/١١/١٣
المرفقات

Abstract of the project "Saudi sign language Translation Companion System"

Recently modern societies are trying hard to be inclusive of disabled individuals by ensuring equal opportunities for the disabled through ease of access to social services and daily human needs. This project aims to enable two-way communication of deaf individuals with the rest of society, thus enabling their migration from marginal elements of society to mainstream contributing elements.

The communication from the deaf to the normal person is done firstly through a computer vision sign recognition module. The output of this module is given to the text generation and speech synthesis modules, producing text and speech representation of the sign. The communication from the normal person to the deaf will be processed firstly by the speech recognition module. The output of this module is passed to the text generation and the Avatar generation modules, producing text and sign language representation of the speech of the normal person.

The project is a collaboration between the Center of Smart Robotics Research and specialists from the Higher Education Program for Deaf and Hard-of-Hearing at King Saud University, hence the project has an excellent experienced multidisciplinary team in the project needed areas.

In the first year, we got promising results from our computer vision-based module that we published in IEEE access journal. We partially built a sign database that was recorded in a novel way that allowed us to get excellent results from it compared to a previous database that we developed before the project. We built a speech recognition system with excellent accuracy. We also built an Avatar representation of the selected signs following the Saudi sign dictionary.

In the second year, we completed recording the sign database of the project and used it to build an excellent sign recognition module. We published the details of our new sign recognition module in Sensors journal. We submitted a paper about the project database, (KSU-SSL), to Journal of King Saud university (Q1 journal) and received a minor revision decision few days ago. We also submitted a paper about our first sign database, (KSU-ArSL), to an ISI journal and it passed first revision.

We are investigating using transformers for sign recognition and are getting encouraging results.

We integrated the project modules into an easy non-invasive solution that we tested successfully. We will continue publishing our results and findings.

Transmittal Letter

Date: 12th/March/2023

Researcher name: Mansour Alsulaiman

College: Computer and Information Sciences

Department: Computer Engineering

Address: P.O. Box 51178, Riyadh 11543

E-mail: msuliman@ksu.edu.sa

Dear Prof. Ahmed Alkhazim

We are submitting to you the final report of our project. The report is entitled technical report for the first year of the project "Saudi sign language Translation Companion System". The purpose of the report is to inform you of our work in the project. The content of this report concentrates on the results that we got and published for the developed system. This report also discusses the Saudi sign language database that we recorded in order to develop a machine learning model to recognize signs in real time scenarios. If you should have any questions concerning our project, please feel free to contact Mansour Alsulaiman at 0503255927 or msuliman@ksu.edu.sa.

Sincerely,
Professor
Mansour Alsulaiman (PI)

Affiliation: College of Computer and Information Sciences, King Saud University.

Title Page

Submitted for:

National Science, Technology and Innovation Plan (NSTIP)

King Saud University

Project title

Saudi sign language Translation Companion System

Project number

5-18-03-001-0003

Project Investigator

Mansour Alsulaiman

Year

2023

Abstract

Recently modern societies are trying hard to be inclusive of disabled individuals by ensuring equal opportunities for the disabled through ease of access to social services and daily human needs. This project aims to enable two-way communication of deaf individuals with the rest of society, thus enabling their migration from marginal elements of society to mainstream contributing elements.

The communication from the deaf to the normal person is done firstly through a computer vision sign recognition module. The output of this module is given to the text generation and speech synthesis modules, producing text and speech representation of the sign. The communication from the normal person to the deaf will be processed firstly by the speech recognition module. The output of this module is passed to the text generation and the Avatar generation modules, producing text and sign language representation of the speech of the normal person.

The project is a collaboration between the Center of Smart Robotics Research and specialists from the Higher Education Program for Deaf and Hard-of-Hearing at King Saud University, hence the project has an excellent experienced multidisciplinary team in the project needed areas.

In the first year, we got promising results from our computer vision-based module that we published in IEEE access journal. We partially built a sign database that was recorded in a novel way that allowed us to get excellent results from it compared to a previous database that we developed before the project. We built a speech recognition system with excellent accuracy. We also built an Avatar representation of the selected signs following the Saudi sign dictionary.

In the second year, we completed recording the sign database of the project and used it to build an excellent sign recognition module. We published the details of our new sign recognition module in Sensors journal. We submitted a paper about the project database to Journal of King Saud university (Q1 journal). We are investigating using transformers for sign recognition and are getting encouraging results.

We integrated the project modules into an easy non-invasive solution that we tested successfully. We will continue publishing our results and findings.

Acknowledgments

This work is supported by National Science Technology and Innovation Plan (NSTIP) in King Saud University under grant number 5-18-03-001-0003. The authors are grateful for this support.

Table of Contents

Contents
Transmittal Letter 1
Title Page 2
Abstract 3
Acknowledgments 4
Table of Contents 5
List of Tables 9
List of Figures 11
Report Body 14
1. Introduction 14
2. Literature review 19
2.1. The need for Saudi sign language translation companion system from a sign language specialist perspective 19
2.2. Literature review of Databases of Arabic signs 23
2.3. Literature review of research on sign Recognition 27
2.4. Literature review of research on Arabic speech recognition 31
2.4.1. Databases for AASR 31
2.4.2. Research on AASR 31
2.5. Literature review of use of Avatar for displaying Arabic signs 32
3. Objectives 34
3.1. Design and development of a Saudi sign language database (KSU-SSL) 34
3.2. Design and development of a sign recognition module. 35
3.3. Design and development of the speech recognition and speech synthesis modules 35
3.4. Design and building of the Avatar module 36
3.5. Integrate the developed system 36

3.6. Establishment of a multidisciplinary research group through the collaboration of the CCIS-team and the HEPD-team 36
3.7. Dissemination of the results and conclusions at conferences and in journals 36
4. Design and development of an Arabic sign language database (KSU-ArSL) 37
4.1. Signs Selection 37
4.2. Participants 38
4.3. Recording Devices and Configuration 39
4.4. Data Verification and Post Processing 41
5. Design and development of a Saudi sign language database. 43
5.1. Selection of the signs 43
5.1.1. Types of signs 46
5.2. Design of the Recording System of KSU-SSL database 48
5.2.1. Preparation and testing the recording environment 48
5.2.2. Building the recording system 52
5.2.3. Archiving and saving protocol 53
5.3. Recording of the KSU-SSL database 55
5.3.1. Selection of Volunteers 55
5.3.2. Number of repetitions of the signs 56
5.3.3. Reference signs by experts 56
5.3.4. Estimation of the recording time 56
5.3.5. Recording Mechanism 57
5.3.6. Recording with painted hands 57
5.3.7. Recording some signs without the presence of sign specialists 59
5.4. Recording verification 60
5.5. Database labeling and segmentation 60

5.6. Comparison with the KSU-ArSL 60
5.7. Quality of the recording in both KSU-ASL and KSU-SSL 62
5.8. Hands and fingers detection of samples from the KSU-ArSL 69
5.9. Hands and fingers detection in the newly recorded KSU-SSL dataset. 73
6. Design and development of a sign recognition module 76
6.1. Spatial multi-branch 3D CNN fused with MLP and autoencoder 76
6.1.1. Development of the sign recognition system 77
6.1.1.1. Input Preprocessing 77
a- Signer body normalization 78
b- Hand cropping and normalization 79
6.1.2. Feature learning 81
6.1.3. Feature fusion and classification 81
6.1.4. Experimental results and discussion 82
6.1.5. C3D knowledge transfer optimization 82
6.1.6. Results of MLP Fusion 83
a- Signer independent mode 83
b- Signer dependent mode 86
6.1.7. Results of the autoencoder fusion 87
a- Signer independent mode 87
b- Signer dependent mode 89
6.1.8. Discussion 90
6.2. Spatial Attention Based 3D Graph Convolutional Neural Network 92
6.2.1. Datasets 92
6.2.2. Methodology 93
6.2.3. Results 99

6.3. Space-Time Transformer 106
6.3.1. Methodology 107
6.3.2. Results 109
6.4. Conclusion 111
7. Design and development of speech recognition and speech synthesis modules 113
7.1. Database Selection and labeling 113
7.2. Overview of building an Arabic ASR system using KALDI 114
7.2.1. Data preparation 115
7.2.2. Training phase 115
7.3. Initial investigation using 416 hours of GALE 117
7.4. Building the Speech Recognition Module 118
7.5. Testing the AASR system on the speech corresponding to the KSU-SSL 119
7.6. Building the speech synthesis module 121
8. Design and building of the Avatar module 124
8.1. Design of the Avatar 124
8.2. Improvement of the Avatar 125
8.3. Design of the Avatar External Components 125
8.4. Recording the signs 125
9. Integration between different modules 130
9.1. Integration between speech recognition module and avatar module 130
9.2. Integration between sign recognition module and speech synthesis module 134
10. Future work 137
11. References 138
12. Publications/Presentations 149
13. Appendices 151

List of Tables

TABLE 1, IMPORTANT ARABIC SIGN LANGUAGE DATASETS 24
TABLE 2, LIST OF THE RECORDED ARABIC SIGNS 38
TABLE 3, RECORDING DEVICES USED IN THE KSU-ARSL DATASET AND THEIR CONFIGURATIONS 39
TABLE 4, LIST OF THE 293 SELECTED SIGNS 44
TABLE 5, MINUTES OF ONE OF THE MEETINGS FOR THE KSU-SSL RECORDING PREPARATION AND OTHER POINTS 47
TABLE 6, the most important choices in developing the KSU-SSL dataset 47
TABLE 7, RECORDING STUDIO COMPONENTS 49
TABLE 8, PROS AND CONS OF THE KSU-ARSL 61
TABLE 9, IMPROVEMENTS MADE IN THE KSU-SSL 62
TABLE 10. ACCURACY (%) ACHIEVED BY MLP, AND AUTOENCODER FUSION IN DIFFERENT MODES 91
TABLE 11, STATISTICS OF DIFFERENT DATASETS USED IN THIS STUDY 95
TABLE 12, RESULTS (% ACCURACIES) OF THE HYPERPARAMETERS' OPTIMIZATION 101
TABLE 13, PERFORMANCE COMPARISON ON THE AUTSL DATASET. 105
TABLE 14, PERFORMANCE COMPARISON ON THE ASLLVD DATASET. 106
TABLE 15, ARCHIVED PERFORMANCE USING THE SSL DATABASE 111
TABLE 16, INITIAL GALE ARABIC TOTAL HOURS 113
TABLE 17, GALE ARABIC BROADCAST CONVERSATIONAL SPEECH 114
TABLE 18, GALE ARABIC BROADCAST NEWS 114
TABLE 19: SUMMARY OF THE "RUN.SH" SCRIPT. 115
TABLE 20. GENERATED ACOUSTIC MODELS LIST 116
TABLE 21. WER OF AN ARABIC ASR USING GALE ARABIC SPEECH DATABASE. 117
TABLE 22, DETAILS PERFORMANCE OF THE CHAIN MODEL 120
TABLE 23, WORD LEVEL SPEECH SAMPLE RECOGNITION 121
TABLE 24: EXAMPLES OF THE GENERATED SPECTROGRAMS. 122
TABLE 25, SELECTED RECORDED SIGNS PER DOMAIN 127

TABLE 26. THE RESULTS OF ONLINE REAL TIME VALIDATION OF THE SIGN RECOGNITION SYSTEM. 135

List of Figures

FIGURE 1, SAUDI POPULATION CAUSES OF DISABILITY 14
FIGURE 2, PERCENTAGE DISTRIBUTION OF SAUDI POPULATION (10 YEARS AND OVER) WITH DISABILITY BY EDUCATIONAL STATUS – ONE DIFFICULTY 15
FIGURE 3, PERCENTAGE DISTRIBUTION OF SAUDI POPULATION (10 YEARS AND OVER) WITH DISABILITY BY EDUCATIONAL STATUS – MULTI DIFFICULTIES 15
FIGURE 4, PERCENTAGE PER SEX OF THE SAUDI POPULATION USING SIGN LANGUAGE 16
FIGURE 5, PROPOSED SIGN LANGUAGE RECOGNITION SYSTEM 17
FIGURE 6, PROPOSED AVATAR SIGN DISPLAYING SYSTEM 17
FIGURE 7 RECORDING SETUP USED IN THE KSU-ARSL DATASET 40
FIGURE 8, RECORDING ENVIRONMENT FOR THE KSU-ARSL DATASET 40
FIGURE 9, SAMPLE FRAMES FROM THE KSU-ARSL DATASET 41
FIGURE 10, EXAMPLE OF THE SIGN LANGUAGE GESTURE –WEEK 46
FIGURE 11, RECORDING STUDIO DESIGNED BY OUR EXPERT 48
FIGURE 12, SUPPORT FOR THE CAMERAS AND THE PHONE 51
FIGURE 13, RECORDING SOFTWARE INTERFACE (SAMPLE SIGN ON THE RIGHT) 53
FIGURE 14, DAILY FOLLOW-UP FORM. 55
FIGURE 15, COLORED GLOVES IN SIGN RECORDING 58
FIGURE 16, COLORED HANDS SIGN RECORDING 59
FIGURE 17, ONE HAND SAMPLE FROM KSU-ASL (BLURRED HAND IN MANY FRAMES OF THE VIDEO) 63
FIGURE 18, TWO HANDS SAMPLE FROM KSU-ASL (BLURRED HANDS IN MANY FRAMES OF THE VIDEO) 64
FIGURE 19,DETAILED SAMPLE OF A BLURRED HAND IN KSU-ASL 65
FIGURE 20, DETAILED SAMPLE OF TWO BLURRED HAND IN KSU-ASL 65
FIGURE 21, SAMPLE FRAMES OF THE VIDEOS OF THE NEW KSU-SSL 66
FIGURE 22, SAMPLE FRAMES OF THE VIDEOS OF THE NEW KSU-SSL 67
FIGURE 23, SAMPLE FRAMES OF THE VIDEOS OF THE NEW KSU-SSL 68
FIGURE 24, OPENPOSE HAND DETECTION OF A FIRST SAMPLE FROM THE KSU-ASL 69
FIGURE 25, GOOGLE MEDIA PIPE HAND DETECTION OF A FIRST SAMPLE FROM THE KSU-ARSL 70
FIGURE 26, OPENPOSE HAND DETECTION OF A SECOND SAMPLE FROM THE KSU-ARSL 71

FIGURE 27, GOOGLE MEDIA PIPE HAND DETECTION OF A SECOND SAMPLE FROM THE KSU-ARSL 72
FIGURE 28, TWO HANDS + FINGER JOINTS DETECTION USING MEDIA PIPE LIBRARY IN KSU-SSL (SAMPLE 1) 73
FIGURE 29, TWO HANDS + FINGER JOINTS DETECTION USING MEDIA PIPE LIBRARY IN KSU-SSL (SAMPLE 2) 74
FIGURE 30, TWO HANDS + FINGER JOINTS DETECTION USING MEDIA PIPE LIBRARY IN KSU-SSL (SAMPLE 3) 75
FIGURE 31. PROPOSED SYSTEM FOR HAND GESTURE RECOGNITION USING LOCAL AND GLOBAL CONFIGURATION FEATURES. 77
FIGURE 32. SPATIAL NORMALIZATION OF THE INPUT FRAMES. 78
FIGURE 33. THE UPPER BODY OPENPOSE KEY POINTS. 79
FIGURE 34. ESTIMATED HAND DIRECTIONS 80
FIGURE 35. HAND DIRECTION ESTIMATION. 80
FIGURE 36. THE EFFECT OF THE NUMBER OF TRAINABLE LAYERS ON THE EFFICIENCY OF KNOWLEDGE TRANSFER. 83
FIGURE 37. HEAT MAP SHOWS THE SYSTEM ACCURACY IN THE SEARCHED SPACE OF THE MLP HYPERPARAMETERS. 84
FIGURE 38, THE AVERAGE ACCURACY OF ALL ARCHITECTURES WITH DIFFERENT INITIAL LEARNING RATES 85
FIGURE 39. CONFUSION MATRIX OF MLP TWO STREAMS FUSION IN SIGNER-INDEPENDENT MODE 86
FIGURE 40. CONFUSION MATRIX OF MLP TWO STREAMS FUSION IN SIGNER-DEPENDENT MODE 87
FIGURE 41. A HEAT MAP SHOWS THE SYSTEM ACCURACY IN THE SEARCHED SPACE OF THE AUTO-ENCODER HYPER-PARAMETERS 88
FIGURE 42. THE AVERAGE ACCURACY OF ALL ARCHITECTURES WITH DIFFERENT INITIAL LEARNING RATES 88
FIGURE 43. CONFUSION MATRIX OF AUTOENCODER TWO STREAMS FUSION IN SIGNER-INDEPENDENT MODE 89
FIGURE 44. CONFUSION MATRIX OF AUTOENCODER TWO STREAMS FUSION IN SIGNER-INDEPENDENT MODE 90

FIGURE 45, SAMPLE FRAMES FROM THE KSU-SSL DATASET 93
FIGURE 46, AVERAGE VIDEO LENGTH IN DIFFERENT DATASETS 94
FIGURE 47, MEDIAPIPE LANDMARKS ESTIMATION SAMPLE 96
FIGURE 48, MEDIAPIPE HAND LANDMARKS [46]. 97
FIGURE 49, THE PROPOSED 3DGCN ARCHITECTURE. 98
FIGURE 50, BASIC ARCHITECTURE ACCURACY ON DIFFERENT DATASETS 101
FIGURE 51, THE BEHAVIOR OF THE OPTIMIZED ARCHITECTURE ON THE KSU-ARSL DATASET. 102
FIGURE 52, ENHANCED ARCHITECTURE ACCURACY ON DIFFERENT DATASETS. 102
FIGURE 53, VISION TRANSFORMER (VIT) ARCHETECTURE 108
FIGURE 54, THE VIDEO SELF-ATTENTION BLOCK THAT WE INVESTIGATE IN THIS WORK. 109
FIGURE 55: TRAINING AND VALIDATION LOSS OF CHAIN MODEL USING GALE CORPUS 118
FIGURE 56: DURATION IN SECONDS FOR EACH OF TRAINING ITERATIONS OF THE CHAIN MODEL. 118
FIGURE 57: PERFORMANCE OF THE TRAINED MODEL USING THE TEST FILES OF THE GALE CORPUS. 119
FIGURE *58*, INTERFACE OF THE RECOGNITION PROCESS WITH SOME RECOGNIZED SPEECH SAMPLES 120
FIGURE 59, LOCAL SIGN LANGUAGE CHARACTERS 124
FIGURE 60, SAMPLE OF THE DEVELOPED EXTERNAL COMPONENTS – CLASSROOM 125
FIGURE 61, SIGN CHECKING IN THE NEWLY DEVELOPED MOBILE APPLICATION 128
FIGURE 62, AVATAR SAMPLE SIGNS 129
FIGURE 63 ROS STRUCTURE 130
FIGURE 64 COMMUNICATION BETWEEN SPEECH RECOGNITION MODULE AND AVATAR MODULE 131
FIGURE 65 START SPEECH RECOGNITION BUTTON 132
FIGURE 66 AVATAR PERFORMING THE NUMBER 2023 IN ARABIC SIGN LANGUAGE 133
FIGURE 67 MOTION OF THE SIGN "SALAM ALAIKUM" 134
FIGURE 68 MOTION OF THE SIGN "KING SAUD UNIVERSITY" 134
FIGURE 69: FLOW DIAGRAM OF INTEGRATION TTS WITH THE SIGN LANGUAGE MODEL. 135

Report Body

1. Introduction

Over one billion people, globally, experience disability according to the WHO (World Health Organization). This accounts for one of seven people suffers from disability. The Demographic Survey 2017 [1], issued by the General Authority for Statistics – KSA, states that 1810358 Saudi residents suffer from disabilities, i.e. 7.1 % of the population are disabled. FIGURE 1 shows the Saudi Population's cause of disability and type of difficulty [1]. Saudi Vision 2030 [2] emphasize on providing all the facilities and tools required to put the disabled people on the path to receive the education and job opportunities that will ensure their independence and integration as effective members of society[3][4][5].

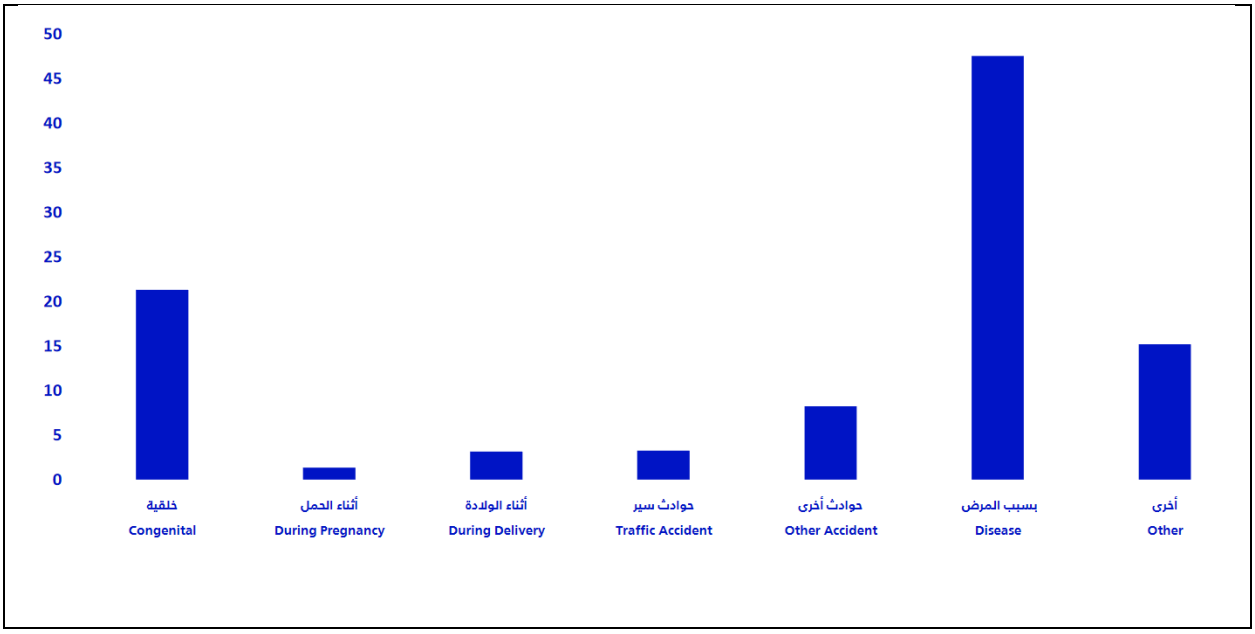


FIGURE 1, SAUDI POPULATION CAUSES OF DISABILITY

FIGURE 2 and FIGURE 3 present the Saudi Population with disability by age and educational status for one and multi-difficulties[1], From both figures, we can see that: 33% are illiterate, 12% can read/write, and only 11 % are university and higher. These statistics indicate that there is a correlation between disability and education status.

360 million people over the world have a hearing disability, 9 % of them are children, as stated by the WHO [6]. In KSA, the number of deaf is about 720,000 [3]. Deaf persons have great

difficulty in communicating with other people in the society. Only a small number of them know and use sign language to communicate with others.

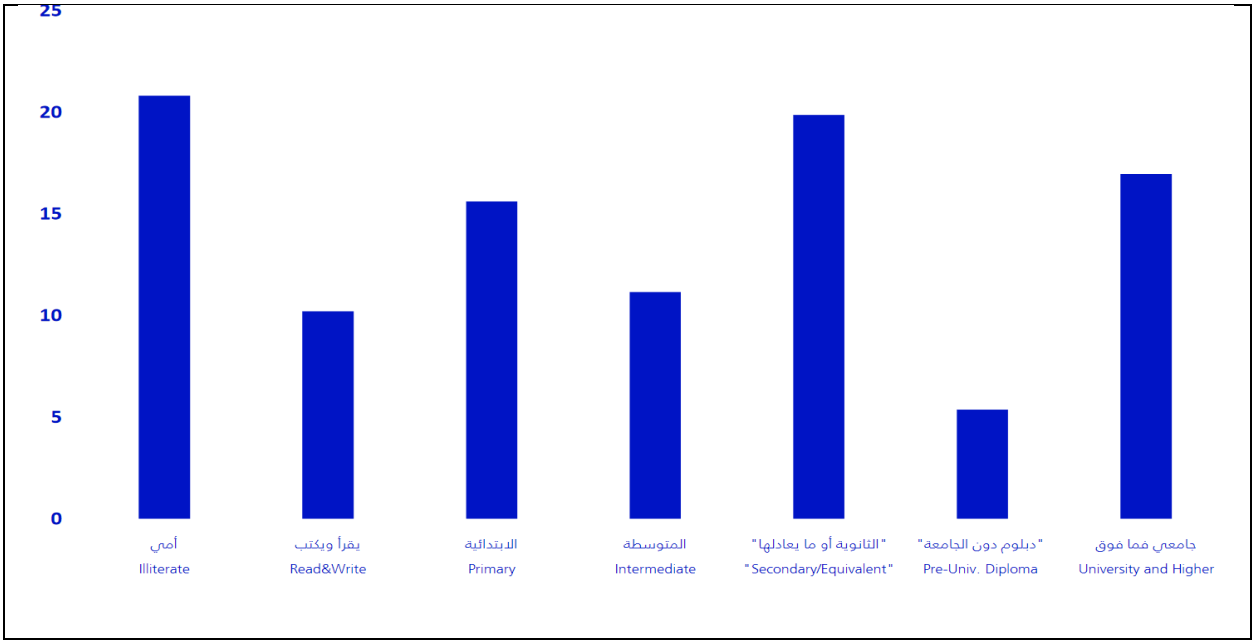


FIGURE 2, PERCENTAGE DISTRIBUTION OF SAUDI POPULATION (10 YEARS AND OVER) WITH DISABILITY BY EDUCATIONAL STATUS – ONE DIFFICULTY

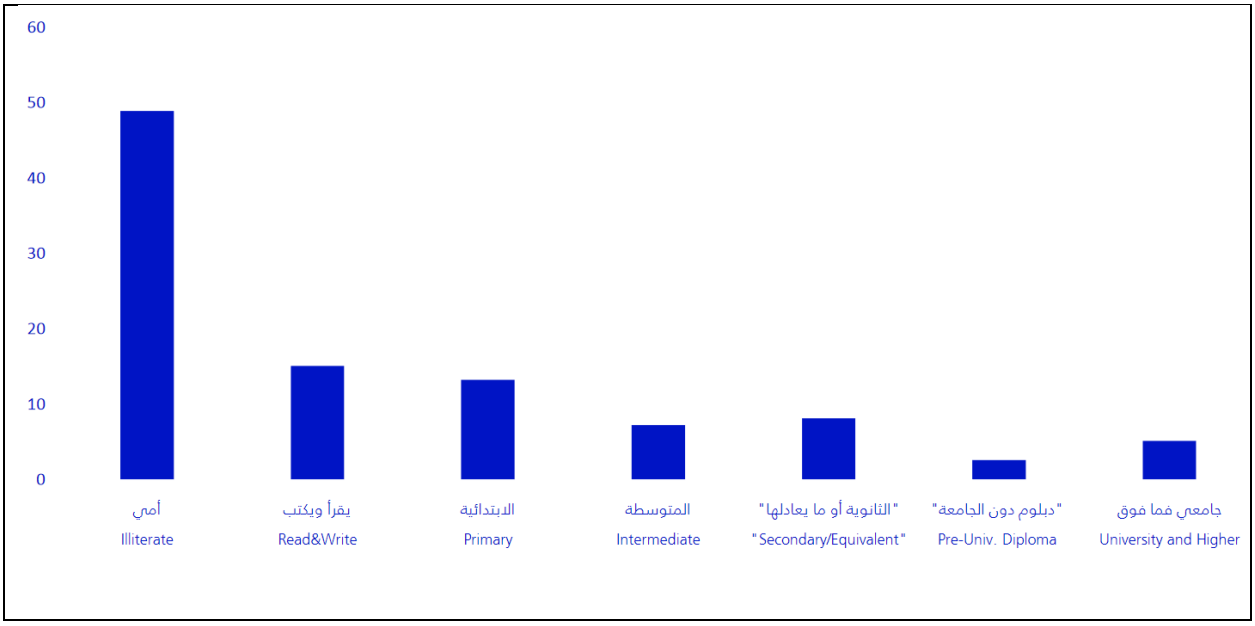


FIGURE 3, PERCENTAGE DISTRIBUTION OF SAUDI POPULATION (10 YEARS AND OVER) WITH DISABILITY BY EDUCATIONAL STATUS – MULTI DIFFICULTIES

FIGURE 4 shows the percentage of Saudi males and females using sign language.

The lack of sign language interpreters [7][8][9] amplifies the difficulty of the deaf in communicating with the rest of society, especially in the government services, specifically in healthcare as stated by the study in [10].

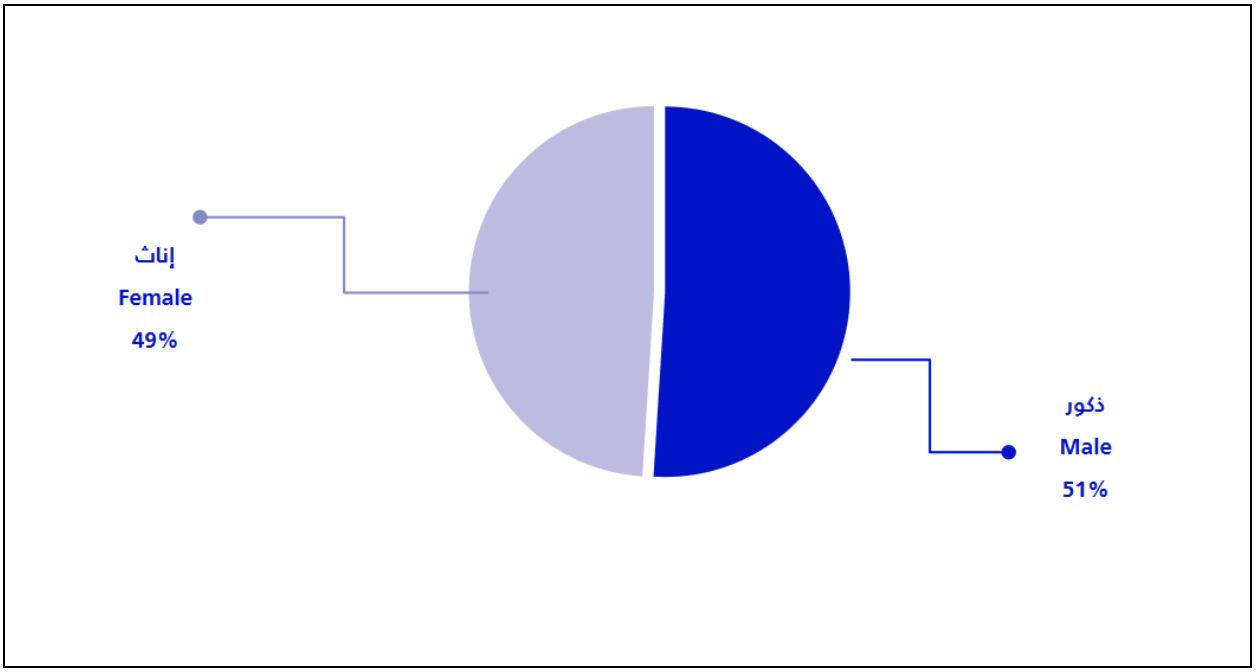


FIGURE 4, PERCENTAGE PER SEX OF THE SAUDI POPULATION USING SIGN LANGUAGE

As a contribution to help the deaf people get involved in the society and integrate with it, we aim to develop a system for a two-way translation of Saudi sign language based on Avatar. This system can be implemented in a carry-on electronic device (laptop, tablet, or mobile) and integrates four basic functions:

1. Recognizes the speech of the normal person and produce the corresponding text.
2. Converts the recognized text of the normal person to sign language and performs this sign by the Avatar. This allows the deaf person to understand the speech of the normal person.
3. Recognizes the sign performed by the deaf person and produces the corresponding text.
4. Converts the text of the recognized sign that the deaf person performed into speech. This allows the normal person to understand the sign of the deaf person.

This system will help the deaf in two ways. First, they can carry it with them anywhere they go and use it to communicate with the rest of society. Second, it can be used to teach the deaf the sign language, especially the children.

FIGURE 5 and FIGURE 6 show the proposed two subsystems, the sign language recognition system and the Avatar sign language displaying system respectively.

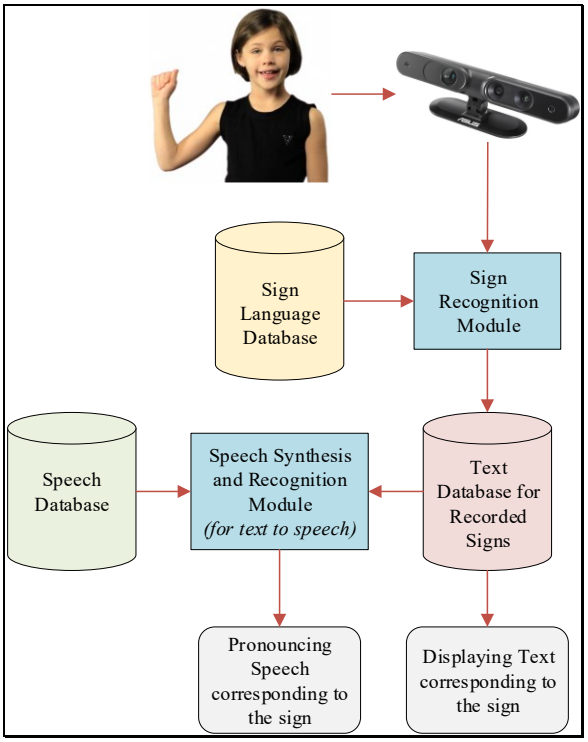


FIGURE 5, PROPOSED SIGN LANGUAGE RECOGNITION SYSTEM

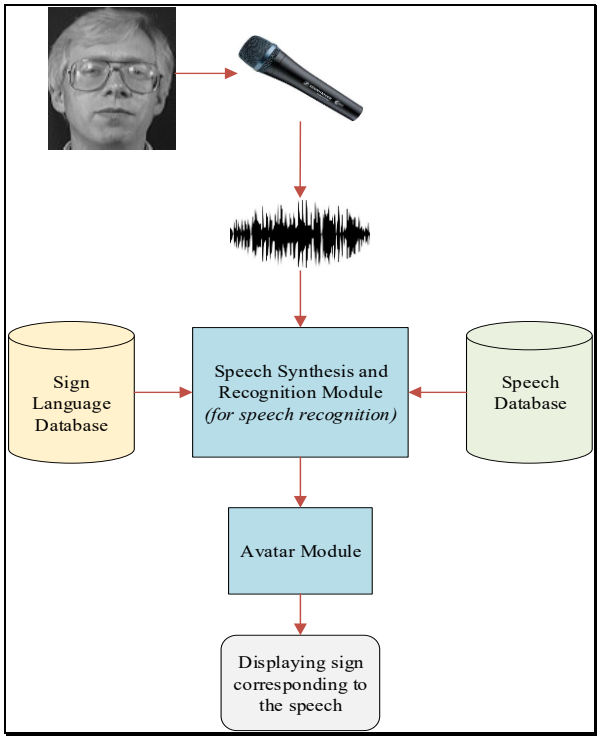


FIGURE 6, PROPOSED AVATAR SIGN DISPLAYING SYSTEM

To build the desired system the project has the following technical objectives:

- Design and development of a Saudi sign language database (SSL).
- Design and development of sign recognition module.
- Design and development of speech recognition and speech synthesis modules
- Design and development of module.

- Design and build of the Avatar module.
- Integrate the developed modules.

In this project we accomplished the following:

Designed and developed a Saudi sign language database (SSL). The database contains videos of 293 selected signs from the Saudi Language dictionary performed by 32 signers, each signer performed the signers 5 times (one with the hand painted).

Designed and developed a high-performance sign recognition module for the 293 signs.

Designed and developed a speech synthesis module for any text.

Designed and developed an unlimited vocabulary speech recognition module.

Designed and developed the Avatar module for the 293 signs and more.

Integrated the different module into a complete system,

Published two papers for the sign recognition module in ISI journals.

Submitted two papers about the sign database.

Organized a workshop about the project that was attended by 245 persons.

In the following sections we will present literature review of the state-of-the-art research on the different parts of the system, followed by the project objectives. In section 4 we will present the design and development of an Arabic language sign language database that was used in our first paper, followed we will present in section 5 the design and development of the project Saudi sign language database. In section 6 we will present the design and development of the sign recognition module, where we proposed different high-performance recognition systems. Next, we will present the design and development of the speech recognition and speech synthesis modules in section 7. The design and building of the Avatar module will be presented in section 8, followed by the system integration in section 9. In section 10 we present future work.

2. Literature review

In the following sections, we will review the literature for the different areas of the project. In section 2.1 we will review the literature for work to help in translating between the deaf and the non-deaf people in the society. This survey will mostly be from a sign language perspective by a sign specialist to show the need for the solution that will come out of the project. In section 2.2 we will review the databases used by researchers conducting research in recognition of the signs as they are performed by the subjects. In section 2.3 we will review the literature for research on sign recognition. In section 2.4 we will review the literature for research on Arabic speech recognition. Lastly, in section 2.5 we will review some of the works on representing the Arabic signs by Avatar.

2.1. The need for Saudi sign language translation companion system from a sign language specialist perspective

Ref. [11] stated, "language is culture, a product and manifestation of culture". (p. 118) Like any other community, Deaf (with capital D to represent Deaf culture) people have their own culture, and sign language is the essence of their culture. sign language is different from one country to another, just like spoken languages, even if the commonly spoken language in two countries is the same [12]. For example, although the commonly spoken language in the United States and the United Kingdom is English, the American sign language and British sign language have significant visual-manual modality differences as each is based on the culture in each country. This remark applies also to Arabian countries.

Sign language is the first and native language of Deaf people, while the spoken language in whatever country they live in is their second language (L2). As a matter of fact, Bilingual/

Bicultural philosophy emerged from the previous point. The bilingual-bicultural approach is based on the premise that sign language is the first and natural language of a Deaf child [13].

The idea is that the L2 is taught through sign language and not through the language of the community in which they live. Ultimately, this requires linkage of language with culture and introducing Deaf children to hearing culture through sign language, allowing them to better integrate and learn in that society.

However, most deaf people do not have the opportunity to learn sign language at home. To elaborate, research has proved that approximately 90% of Deaf children are born to hearing parents who are not familiar with Deaf culture and do not know how to use sign language [14], [15]. Therefore, Deaf children learn their first language at the school level [16]. Conversely, the scenario becomes worst at the school level for these children.

[17] demonstrate that Deaf students stay half of their day with teachers who are considered their fundamental language paradigm. Also, [17] (P. 38) asked a significant question related to this matter “How can we ensure that Deaf children will succeed in learning a second language if we cannot guarantee that they have a strong first language?” Unfortunately, this is rarely the case, the majority of the teachers of Deaf students are hearing teachers, and their sign language level is not sufficient to cover all aspects of teaching Deaf students what they need to learn. In the reality of Deaf peoples’ education, the number of Deaf teachers of Deaf students is severely underrepresented compared with hearing teachers of deaf students. According to[18], in the U.S, more than 75% of the total teachers of Deaf students are hearing teachers. Correspondingly, in the entire United Kingdom, the number of Deaf instructors of Deaf students is negligible. Furthermore, in other countries like Mexico, hearing teachers are the only teachers for Deaf students. While in

KSA, in the whole education system (primary, secondary and postsecondary schools and universities), there are only two Deaf teachers (an elementary school teacher and a teacher-assistant at King Saud University).

The issue for Deaf students becomes evident at universities because they encounter social and academic barriers that work against them and limit their ability to continue, no less complete their university studies [19],[20]. These two barriers stem mainly from how to communicate and educate Deaf students. Research has shown that the instructors of Deaf students do not know how to communicate with their Deaf students and the instructors depend mainly on the interpreters [19], [21][22][23][24][25][26]. Further, the absence of existing interpreters leads to the social barrier between the Deaf and their hearing counterparts, making the first students suffer from social isolation [20], which is considered one of the most critical reasons that affect Deaf students' progress at the university [27]. The fact becomes more apparent based on what [28] said, that is, without the requirement that an interpreter is present in every class that has a Deaf student taught in spoken language, Deaf student is likely to fail. In fact, that is the primary reason why Deaf students do not finish their postsecondary schooling. Almost 75 percent of students with hearing loss withdraw from the university before they obtain their degrees [29]. This proves that the Deaf students depend mainly on interpreters to communicate and educate as being the only facilitating method available they must receive the knowledge.

The question is that are these interpreters qualified enough to cover the pivotal role they have in Deaf students' life, whether in academic or any other context? Several studies revealed that there are different mistakes interpreters make/ fall in when they translate to Deaf people that affect negatively Deaf individuals' education and communication[30][31][32][33][34] and that includes:

- Failure to commit to professional ethics.
- Deaf culture: either they do not consider it or are unfamiliar with it.
- Inaccurate translation with major mistakes.
- There is a negative correlation between the time of speech and immediate translation in which less time given to the interpreter leads to more mistakes in the translation quality.
- They do not adhere to the translation accurately by concentrating only on the general idea and skipping the details.
- Face expression (whether they do not use it, or it does not match the sign).
- Using some signs that the Deaf community is not familiar with.
- Using wrong sign that has a different meaning.
- Fast translation.

A doctoral dissertation conducted by [35] revealed different issues related to the previous problems in sign language interpretation in Saudi Arabia. Even though there are several official bodies that grant licenses to practice the profession of interpretation in sign language, there are no high and consistent standards among these bodies. This issue created a huge gap between licensed interpreters, and this is what [36] dissertation confirmed. The licensed interpreters were able to translate only 28 percent of the television news accurately. In other words, the group of arbitrators, which included two Deaf people and two licensed interpreters, understood that simple part of the news.

One effective way to avoid all these mistakes that interpreters fall into and enhance the communication and education of Deaf people is through using technology such as automatic translation programs from speech to sign language and vice versa. [37] found that the program they designed, which recognizes Arabic speech of Alphabets and some common words, then translate it to Arabic sign language using Avatar, helped Deaf people to communicate with Hearing society. In the other direction of communication,[38] presented a program for recognizing the signs of Arabic Alphabets using the deep CNN technique and producing the corresponding text or speech. [39] indicated that a program to recognize the signs would not only improve the communication between Deaf and hearing people, but also it will make their education easier and more effective. In the same way,[40] underscored that a text-to-sign translation application for Iraqi sign language would enhance the communication of Deaf people with the hearing community and their learning reading and writing. Moreover, [41] in his doctoral dissertation showed that the usage of text-to-sign programs would improve the reading, writing, and math education of Deaf students.

All of the above programs were for text-to-sign translation, only[38] was for sign-to-speech but only for the Alphabet. This highlights the need for a two-way translation system, which translates signs to text or speech and translates speech to signs with high performance.

In the light of the above facts, the need for this technology in Saudi Arabia is very high, especially that in the lack of professional interpreters and educational programs to prepare them, also it will improve the quality of life for Deaf people.

2.2. Literature review of Databases of Arabic signs

Due to the lack of a public and rich Arabic sign language dataset, many research groups developed their local recordings. These local datasets might contain tens or hundreds of signs,

nevertheless, they are domain-specific or contain few signs and variations or performed by few signers. In general, the recorded signs do not exceed hundreds and have few signers.

TABLE 1, IMPORTANT ARABIC SIGN LANGUAGE DATASETS

Ref	# signers	# signs	Type of sign	Devices	Modality	Remarks	Year	Publicly Available	Country
[42]	3	23	Words Phrases	Video camcorder	Upper body	50 repetitions and 3450 samples	2007	No	UAE
[43]	40	32	Alphabet	Camera Gray Images	Hands	54049 samples	2018	Yes	Saudi Arabia
[44]	10	500	Alphabet Words Sentences	Canon Power Shot A490 Camera	Hands +Upper body +Facial expression + Lips motion	signs World Atlas	2014	No	Egypt
[45]	4	1216	Words Sentences	Leap Motion KinectV2, Digital Cam	Upper body+ Facial expression	4 Rec. angles and 19456 samples in total	2015	Yes	Egypt
[46]	1	300	Words Phrases	Sony DVD Handy Camera	Upper body	15 repetitions and 4500 samples With Gloves	2007	No	Saudi Arabia
[47]	2	20	Words	Leap Motion +2 Cameras	Facial expression + Body +Lips motion	Small dataset and the number of samples differs in a different module.	2015	No	-
[48] [49]	40	80	Alphabet Numbers Words phrases	-Kinect V1 -Kinect V2 -Sony Handy Cam	Full Body + Skeleton	5 repetitions and 16000 samples in total	2020	No	Saudi Arabia
Recorded Dataset for the current project KSU-SSL	32 signers	293	Alphabet Numbers Words or phrases for daily life and medical field	-RGB color camera - Infrared camera - Mobile	Upper body	4 repetitions without painting and one with painted hands - Daily control of the signs recording by a deaf translator	2022	Not yet	Saudi Arabia

In [49] we surveyed the databases used in research in recognizing Arabic signs and summarized their finding in a table. We noticed three points: Firstly, the number of the signers does not generally exceed a set of tens of signers and some are for less than ten signers. In fact, the number of recorded signs varies inversely with the number of signers, teams were obliged either to increase signers or signs. The second point is the variety of recording devices, which starts from simple cameras such as webcams to RGB + depth cameras to cameras using infrared. The third point is

that most datasets include words and short sentences, as a sequence or single images cropped around the hands [49].

In TABLE 1, we reproduce the comparison table of our work in [49] and include in it the database that we developed in this project which we named KSU-SSL. Among the databases in table 1, other than our project database, the database of [48] and [49] is the most suitable for building practical sign recognition module, because it the best balanced on number of signs and number of signs. In the paragraphs below, we will highlight the main points of the databases surveyed in Table 1.

The ArSL dataset reported in [42] is an isolated words dataset. It was created by the College of Engineering at the American University of Sharjah. It contains 23 gesture classes performed by three participants. Each participant was asked to repeat the gestures 50 times. Therefore, there are a total number of 3450 samples in that dataset, distributed evenly among the 23 classes such that each class has 150 samples. An analog camcorder was used to record that dataset. Another dataset is the ArSL2018 dataset reported in [43], which is an image-based dataset for 32 Arabic Alphabets (the basic 28 Arabic alphabets and the extended four alphabets "لا، أل، ة، ئ"). The ArSL2018 consists of 54,049 images in gray scale with 64×64 dimension. The images were collected by recording 40 participants in different lighting conditions and with different backgrounds. This dataset was made publicly available.

The sign World ArSL dataset reported in [44] is an image and video dataset. It was developed by some researchers in Egypt to evaluate their methods for real-time ArSL gesture and posture recognition. The dataset contained:

1- Handshapes in isolation and in single signs.
2- Arabic alphabets.
3- The numbers.
4- Movement in single signs.
5- Movement in continuous sentences.
6- Lip movement in Arabic sentences.
7- Facial expressions.

The dataset was performed by 10 participants under controlled lighting conditions. It was acquired by a Canon Power Shot A490 digital camera with an image quality of 1024×768 pixels and video samples of 10 MB each.

Furthermore, the ArSL dataset reported in [45] was established based on an Arabic sign language dictionary approved by the League of Arab states. The total number of sign classes in this dataset is 1216 signs including the Arabic alphabets and numbers. The entire dataset was recorded by four sign language experts and reviewed and validated by the other two experts. The four experts who recorded the dataset are mixed of right-handed and left-handed persons. Three different sensors were used to collect the dataset, ordinary HD camera, Kinect 2, and leap motion. The dataset videos were captured from four different viewing angles (0, 270 then 315, 225), with four different frame rates: 5, 10, 30, and 50 frames per second, hence the database size is 4× 1216 × 4 = 19,456 samples. The samples of this dataset were recorded in different lighting conditions and involving facial expressions in most of the cases. The samples were presented in the form of RGB videos, Infrared and depth maps as well as the skeleton indexes produced by the lip motion sensor.

On the other hand, the dataset reported in [46] consists of 300 classes selected from the Arabic sign language dictionary (it is not mentioned which exact dictionary). This dataset was recorded by a single hard of hearing fluent signer, where each sign was repeated 15 times, to produce a dataset that contains 4500 samples in total. The video recording camera (Sony DVD Handy cam, model no. DCR-DVD 200E) was placed right in front of the standing signer. A frame rate of 25 frames per second was maintained throughout the recording. Moreover, a high-intensity halogen light source was used to avoid any shadow effects in the environment. A whitewashed wall background was also set to match the clothes of the signer, hence this made it easy to isolate the hand of the signer from the background based on the hand color. An Arabic sign language expert was also involved to help in dataset recording.

The King Saud University Arabic Sign Language (KSU-ArSL) dataset reported in [48] and [49] consists of 80 gesture classes. It was created by the Center of Smart Robotics Research with collaboration from the Higher Education Program for the Deaf and Hard of Hearing at King Saud University. The dataset comprises selected gestures from common Arabic sign language words and expressions. These expressions contain single-handed actions as well as two-handed actions.

40 non-deaf subjects were recorded, where the recordings were attended by a knowledgeable in sign language who made sure the signs were performed correctly. Each subject was asked to perform the gestures five times. Different devices such as RGB cameras and Microsoft Kinect were used for recording this dataset. The dataset recording sessions were performed without restrictions, in an uncontrolled environment. There were no constraints on the clothing of the participants, the lighting conditions, or the background color. There was also some variation in the distances between the camcorder device and the signers. Because of this restriction-free recording, KSU-ArSL is a challenging dataset. In some cases, the signer's hands are blurred and difficult to detect and track. From table 1 we can notice the richness of the KSU-ArSL database. For the number of signers only [43] has 40 signers similar to KSU-ArSL but with 32 signs (alphabet only) while KSU-ArSL has 80 signs (alphabets, numbers, and common words or phrases).

2.3. Literature review of research on sign Recognition

As a form of human-computer interaction, hand gesture recognition has attracted the attention of many researchers since the end of the last century. Plenty of research works have been conducted to tackle this problem. Most research works followed two approaches, namely, vision-based approach and non-vision-based approach. In the non-vision-based approach, hand gesture data is collected via interfacing devices like data gloves, motion sensors, and position trackers [50][51][52]. The hardware setup of this approach is costly and inconvenient as it restricts the signer movement. The vision-based approach on the other hand avoids these downsides. It collects the data via cameras and imaging sensors. However, research works in this approach encounter many challenges that degrade the performance of existing systems. Lighting inconsistency, motion blur, background clutter, and hands occlusion are examples of these challenges. Moreover, the studies in this approach can be classified into two categories, conventional techniques as in [53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64], and [65], and deep learning-based techniques as in [66], [67], [68],[69],[24],[70],[71], and [72].

The paper by Murakami et.al is one of the earliest papers in the field [58]. In that paper, they utilized Artificial neural networks (ANN) to recognize 42 alphabets of Japanese sign language. Another robust method based on ANN classifier and skin color segmentation was also presented for Thai alphabets recognition [53]. Histogram of Oriented Gradient (HOG) is used in this approach to represent the segmented hand shape.

HMMs, on the other hand, were extensively utilized for hand gesture recognition starting from the 90s of the last century. For instance, HMMs based method utilized different combinations of principal component analysis, kurtosis position, and motion chain code descriptors [60]. The best accuracy was achieved on the RWTH-BOSTON-50 database by combining the three descriptors. Killy et al. in [54] used a single HMM for each hand with colored gloves for hand segmentation and tracking. A small dataset of eight gestures was used to evaluate the proposed method. Pu et al. also utilized HMMs to model the segmented trajectory of the hand gesture for 100 Chinese sign words[73][61]. The trajectory segments were represented as histograms of shape context.

In another work proposed by Li et al. an entropy-based K-means technique was used to evaluate the number of states in each HMM model [62]. A combination of the Baum-Welch algorithm with the artificial bee colony algorithm was used to determine and learn the structure of HMM. Recently, Yang et al. classified the hand gesture trajectory of ASL in a hierarchical way to generate a sequence of observations [59]. HMMs are then applied to model and classify these sequences.

An SVM classifier was utilized for recognizing Irish sign language [55] and ASL [63]. A skin color model was used in [55] for hand segmentation and a combination of weight Eigenspace size function and Hu moments to represent the hand shape. On the other hand, the fingertips coordinates collected by Leap Motion and Intel RealSense 3D cameras were used in [63]. In other work, Aly et al. utilized SVM to recognize 23 Arabic sign language words [64]. The authors proposed a local binary pattern in three orthogonal planes to represent the appearance and motion features of signs. The proposed method in [56] used particle filtering for hand tracking. Feature covariance matrix and the minimum Riemann distance metric were then used on the detected hand for representation and classification. Lim et al. utilized Sparse observations from a video of RGB-D frames [57], where the skin color and depth maps were used for hand segmentation and HOG for posture representation. The similarity between postures of different samples was then measured. Abid et al. also utilized bag-of-visual words with the local part model approach to recognize simple six dynamic gestures [65].

Recently, deep neural network architectures such as CNN, LSTM became predominant for hand gesture recognition. For instance, Huang et al. utilized CNN and ANN for the representation and classification of 20 Italian gestures [74]. To perform well, this method requires a multimodality

input, which includes the RGB frames, the depth maps, and the skeleton joints. Similarly, Lionel et al investigated temporal convolutions with bidirectional recurrence for gesture recognition in the Montalbano dataset [69].

Another deep learning architecture proposed for ASL hand posture recognition [24]. The depth data is used to segment the hand region and deep belief neural network and CNN for feature learning and classification. Another recent approach proposed by Okan et al. involved the fusion of the optical flow and RGB frames to adapt the pre-trained inception model for hand gesture recognition [70].

Another CNN-based architecture was also proposed in [66] for static hand gesture recognition. The input to this architecture was a small image of size 32 × 32 containing only the hand region. CNN and an LSTM were also combined for temporal 3D pose gesture recognition [71], where the input frames of this approach contain the 3D joints of the human body. Furthermore, in [72], two streams of 3DCNN were presented for gesture recognition. The two streams' inputs were interleaved volumes of depth maps and preprocessed Sobel gradient with different resolutions. The ResNet architecture was utilized in another work by Chen et al. to encode the features of frames' sequence in a single 2D matrix [75]. Then, another CNN was utilized to catch the evolution of the spatio-temporal features for classification. Recently, Hu et al. utilized the skeletal data of hand gestures to design a deep learning-based controlling system for unmanned aerial vehicles [67]. Both CNN and different MLP architectures were investigated for feature learning. Another recent work for Arabic sign language recognition utilized semantic segmentation to detect the hand [76]. Unsupervised learning via a convolutional self-organizing map is then applied for feature extraction and bi-directional LSTM for sequence modeling.

The system we propose in this study is based on a single modality input (RGB video). It does not require other modalities such the depth maps or skeleton joints. It combines both the local and global configurations of hand gestures and performs well on dynamic sign language gestures.

In a previous hand gesture recognition work [77], we implemented a variation of C3D architecture [78] [78] and utilized knowledge transfer from human action recognition to hand gesture recognition. The mentioned C3D architecture composed eight convolutional layers, five pooling layers, and two fully connected (FC) layers.

Even though we got encouraging results in[77], we noticed that the direct application of 3DCNN for hand gesture modeling has two main drawbacks. The first one is that the 3DCNN modeling is

not robust enough to capture the long-term temporal dependency of the hand gesture signal. The second one is that modeling the hand gesture signal in a video should be slightly different than other video-based analyses for human activity recognition or event recognition in general. For the second drawback, the whole scene and maybe multiple interacting objects in the frame are involved discriminative descriptors for the overall recognition. Contrary, the discriminative features, in the hand gesture recognition, are located mainly in the fingers' configuration, the hand's orientation and the hand's relative position to the body. In other words, most of the frame area contains non-relevant features that increase the misclassification ratio. In [79] we addressed the first mentioned drawback of modeling the long-term temporal dependency, where we utilized independent instances of 3DCNN to model the local spatio-temporal features of different temporal segments. We also explored different techniques to globalize the local representations.

Experimental results showed outperforming performance by this temporal modeling enhancement. Currently, we address the second drawback by utilizing both the local and global configurations of the hand gesture while giving more attention to the fingers' configuration and eliminating most non-relevant features. In another direction of investigation we achieved excellent accuracy on the same dataset based on using the concatenation of a 3D CNN skeleton network and a 2D point convolution network [49].

2.4. Literature review of research on Arabic speech recognition

A module of the sign translator system, hence one objective of the project, performs automatic Arabic speech recognition (AASR). In this section, we review the state-of-the-artwork in the AASR field. The review is divided into two parts: the speech corpora and the techniques used in the recognition engine.

2.4.1. Databases for AASR

From the database point of view, we can notice the lack of availability of large-scale Arabic speech corpora compared to other languages. Most of the existing databases were recorded from TV broadcasts. One of the largest Arabic corpora is GALE Arabic, which was developed by the linguistic data consortium (LDC). There are many versions of GALE corpus. The corpus consists of a recording of the broadcast conversation and broadcast news from Arabic TV channels [80]. Another important database is the KSU speech database, which was produced by the speech processing group at King Saud University [81]. This corpus was designed to fulfill the requirement of Arabic speaker/speech recognition systems. It contains the recording of a large number of speakers, about 257 in 3 sessions, from different nationalities (Saudi, Arab, Non-Arab) [81]. MGB-2, is another important database and stands for Multi-Genre Broadcast and consists of 1200 hours of recording from TV programs [82].

2.4.2. Research on AASR

From the recognition techniques point of view, deep learning has become the dominant technique in automatic speech recognition (ASR), hence we focus herein only on the latest research on AASR based on deep-learning.[80]developed initially a broadcast news system based on 200 hours from GALE Phase 2. They used a conventional acoustic model and a deep neural network acoustic model. The best results were achieved by the DNN-MPE model with recorded 15.8% 32.21%, and 26.95% word error rate (WER), for reports, conversational, combined sets, respectively [80]. Long short-term memory (LSTM) and gated recurrent unit (GRU) are used for Arabic speech recognition in [83]. They tested the proposed system for 10 spoken Arabic digit recognition task and 10 spoken command TV tasks [83]. Deep auto-encoder was used for speech enhancement in [84], for remote Arabic speech recognition system. The authors used isolated words Arabic speech database for their experiments, where the database contained only a recording of 20 words.

In terms of available toolboxes for ASR, with the era of deep learning, the availability of a huge amount of training data, and the fast growth in computation devices resulted in the release of a lot of ASR toolkits such as Baidu's Deep Speech from Mozilla [85], wav2letter from Facebook [86], PyTorch-Kaldi [87], openseq2seq from Nvidia [88], and ESPnet [89]. This is in addition to the old ASR toolkits such as HTK and Sphinx. Each of these surveyed toolkits has advantages and disadvantages. In our project, we are using Kaldi because it supports an available recipe of the Gale database, which we are using in the project.

In a conclusion, we noticed that AASR needs more study and enhancement. Moreover, applying End-to-End (E2E) ASR for the Arabic speech recognition system still needs more investigation.

2.5. Literature review of use of Avatar for displaying Arabic signs

In [90] a translation system was constructed for Arabic sign language, the words are converted into Hamburg notation system where the hand shape, hand orientation, hand location, and hand movement are transferred to manual parameters and facial expression, shoulder raising, mouthing gesture, hand tilling and body movement are converted to non-manual parameters. Then the signs are converted into the sign gesture markup language file where the 3d avatar will perform them. [91] present an avatar-based translation system for Arabic sign language, where an Arabic sign language 3D motion database was recorded using data gloves, then an avatar will animate the signs. In [92] a mobile-based communication framework for Arabic sign language was presented. In this system a 3D motion database of 588 signs was created using synthetic animation where the user can exploit a sign-editor to create the video representation of the signs. In [93] the authors present a machine translation system based on rule-based interlingua and example-based approaches. The system uses SAFAR platform [94] and ALKHALIL morpho system [95] to extract the morphological properties of the words, then it generate a video sequence to represent the word in Arabic sign language. In [96] the authors develop an application that translate Arabic text into Arabic sign language and vice versa. If the user is non deaf he can write a text then the application will translate it to sign language using 3D avatar, and if the user is deaf he can select a sign from the database and the application will translate it to a text. [41] present a virtual learning environment for deaf and hard hearing (DHH) students where the DHH students can learn computer programing. This virtual environment is based on 3D avatar where the avatar was created

using: first Adobe Fuse CC to model 3D character and set the shape, skin, hair and clothes the 3d character, second Mixamo is used to rig the created 3D character in step 1 so it can be animated in next step where Unity 3d Engine was used to record the movements of the avatar.

3. Objectives

As we briefly presented in the introduction that the project has five technical objectives, namely:

- Design and development of a Saudi sign language database (SSL).
- Design and development of a sign recognition module.
- Design and development of the speech recognition and speech synthesis modules.
- Design and build the Avatar module
- Integrate the developed system

In addition to these technical objectives, the project has two objectives namely: establishment of a multidisciplinary research group through the collaboration of the Center of smart robotics research (CS2R) and the Higher Education Program for Deaf and Hard-of-Hearing Students (HEPD), King Saud University for a Saudi sign recognition system. As well as the dissemination of the results and conclusions at conferences and in journals.

In the following section, we will briefly present our accomplishments in each of these objectives.

3.1. Design and development of a Saudi sign language database (KSU-SSL)

We designed developed a very useful database of Saudi sign language. Great efforts were put in making this database. We can summarize the main characteristics of the database in the following:

- The signs adhered strictly to the Saudi sign language dictionary.
- The selected signs cover most of the signs used in daily life and a selected application field to show the usefulness of the system.
- The selected application field is the medical field for its importance and need by the deaf.
- Each sign was recorded in 4 repetitions plus one with painted hands. We started with wearing gloves in both hands then switched to painted hands and fingers.

- Constructed suitable studio with 3 cameras: High speed RGB camera with high fps, IR, and mobile.
- We Recorded deaf, hard-of-hearing, experts, and non-deaf.
- Recording approved by expert attending in person in almost all cases while in few cases the recording of some part of the signs were attended remotely by zoom and similar applications.

3.2. Design and development of a sign recognition module.

We developed three systems for dynamic hand gesture recognition via multiple deep learning techniques. The first system uses a fusion of two 3DCNN deep learning networks to represent the hand gesture as well as the global body configuration features. MLP and autoencoders were used to aggregate and globalize the extracted features. The proposed architectures were evaluated on the KSU-ArSL dataset (a sign database that we developed before the project) and excellent results have been obtained. The second system proposed a lightweight 3D Graph Convolutional Neural Network (3DGCN) for sign language recognition. The proposed architecture utilizes a few 3DGCN layers to avoid the common over-smoothing effect in deep GCN architectures, which results from the high repetitions of messages passing between the graph nodes. The third system used the vision transformer approach (TimeSformer) for sign language recognition. Both 3DGCN and TimeSformer architectures were evaluated on the project KSU-SSL dataset and outstanding results have been obtained. We published the results and the findings of the first two systems in Q1 and Q2 journals (IEEE access was a Q1 journal when we submitted the paper and at time of publication of the paper).

3.3. Design and development of the speech recognition and speech synthesis modules

In this objective, we have developed an Arabic speech to text engine (STT) that is able to convert speech to text. We got excellent results with part and the complete Gale, which is a large speech database containing approximately 1000 hours of recording of Arabic speech. The STT has been tested on the pronounced speech for the signs selected for the project and gave very accurate recognition rate. We also tested the STT model using continuous speech from streaming videos and we got excellent results. For the speech synthesis module (i.e., TTS), which aims to translate

the text of recognized signs to an audio, we used a recent neural TTS model called FastSpeech2 [97], because it can work in real time and support multi-speaker embedding. We trained the FastSpeech2 on producing Arabic speech. We got good quality natural like speech. We tested it for vocalizing the selected signs and it gave excellent results.

3.4. Design and building of the Avatar module

In this objective, we developed a functional mobile app Avatar for all the project signs., Two virtual characters (man and woman) with the Arabic cloth were designed.

3.5. Integrate the developed system

We integrated the four previous modules in the desired system. The system accepted the speech of the non-deaf, recognized the speech, then displayed the corresponding sign using the Avatar. In the other direction, the system recognized the sign from the video, produced the corresponding text, then vocalized the text to the non-deaf.

3.6. Establishment of a multidisciplinary research group through the collaboration of the CCIS-team and the HEPD-team

The CCIS-team and HEPD –team worked together and established a research group in Saudi signs that have accomplished: Literature review on the need of the project main output which is the sign translator and literature review on the different modules of the project. The two teams selected the signs of the project and the intended application field. The two team cooperated in developing the videos of signs dataset. The CCIS team developed the sign Avatar with cooperation and consultation from the HEPD team.

3.7. Dissemination of the results and conclusions at conferences and in journals

We published two papers in Q2 journals on the two systems for recognizing dynamic signs. A paper about some of our work on the avatar was accepted by an IEEE conference. We submitted two other papers to two reputed journals. We organized an online workshop about the system and its different parts. The workshop was conducted online for the benefit of researches from all the world and was attended by 245 participants.

4. Design and development of an Arabic sign language database (KSU-ArSL)

The Saudi database of the signs of the project was not developed in the early stage of the project, and we did not want to delay our investigation of building the sign recognition module until we develop the database, hence we used the KSU-ArSL database which is a database that we have developed before and used in a publication of this project and in previous publications and research. Moreover, in building the project database, King Saud University Saudi Sign Language Database (KSU-SSL), we benefited from our experience in developing KSU-ArSL, hence it will be beneficial to have this section about KSU-ArSL database.

In this section, we present the detailed methodology for developing the KSU-ArSL dataset, including signs and subjects' selection, recording procedures, and verification approaches. The methodology we benefited from our previous work on the KSU Arabic speech dataset [81], publicly available on the Linguistic Data Consortium website [98], the KSU speech voice pathology dataset [99], and the date fruit dataset [100], publicly available on the IEEE DataPort platform [101].

The process of creating the KSU-ArSL dataset began with the examination and study of a variety of sign language datasets, including the RWTH-Boston Dataset [102] and the American SL Dataset [103], which are both well-known public SL datasets. Then, based on our resources, we determined the global parameters required for the recording, e.g., number of signers, dynamic and static signs, signer speed, recording devices, storage extension, etc., and designed a protocol for the complete recording process.

4.1. Signs Selection

The sign selection team, professional sign language experts from the Department of Deaf and Hard of Hearing (HoH) at King Saud University (KSU), has created a list of 80 signs consisting of commonly used signs in daily life and signs of the Arabic Alphabet letters and numbers. In selecting the signs, the the experts consulted some deaf people in their sign selection The 80 signs were divided into two groups: static and dynamic signs. Static signs consist of a single movement and require the subject to keep his hand still, while dynamic signs, known as moving signs, consist of

a sequence of consecutive movements. Numbers and letters are static signs and the rest are dynamic signs.

TABLE 2 shows the recorded Arabic signs. These signs were selected and performed based on the ArSL dictionary [104].

TABLE 2, LIST OF THE RECORDED ARABIC SIGNS

Numbers (11 signs)										
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Arabic alphabets (28 signs)										
أ 'Alif'	ب 'Ba'	ت 'Ta'	ث 'Tha'	ج 'Gim'	ح 'Haa'	خ 'Kha'	د 'Dal'	ذ 'Thal'	ر 'Ra'	ز 'Zai'
س 'Sin'	ش 'Shin'	ص 'Sad'	ض 'Thad'	ط 'Taah'	ظ 'Daah'	ع 'Ain'	غ 'Gin'	ف 'Fa'	ق 'Qaf'	ك 'Kaf'
ل 'Lam'	م 'Mem'	ن 'Non'	هـ 'Ha'	و 'Waw'	ي 'Ya'					
Common words/phrases/sentences (41 signs)										
أب Father	أين؟ Where?	مستشفى Hospital	ملك King	مدير Manager	اجتماع Meeting	مسجد Mosque	الملك سعود King Saud	صلاة Prayer	آسف Sorry	شكراً Thanks
كيف حالك How Are You?	عملية جراحية Surgery	لغة الإشارة Sign Language	أمير الرياض Riyadh Governor	اللغة العربية Arabic Language	اللغة الانجليزية English Language	تفضل Come In	بماذا تشعر؟ What are you feeling?	السلام عليكم Peace be upon you	حبوب الدواء Pills	أصم Deaf
وفاة Death	دكتور Doctor	مساء Evening	أسرة Family	ملف File	حار Hot	أخت Sister	أخصائي نفسي Psychologist	صباح Morning	أم Mother	عمل Job
اسم Name	ألم Pain	جامعة University	السبب Reason	بارد Cold	أخ Brother	متعب Tired	إجازة Vacation			

4.2. Participants

Our aim in this dataset was to record the dataset by deaf subjects. We coordinated with the Deaf and HoH department at KSU to record the signs from paid deaf and hard-of-hearing volunteer students under the supervision of an ArSL translator. The translator's job was to ensure that the recorded signs follow the ArSL dictionary [104] and are not specific to the subject. However, the recording did not go as intended, and we encountered many challenges. Deaf subjects were unwilling to complete all five recording sessions and tended to refuse to repeat the sign and to

correct it according to the dictation of the ArSL translator. As a result, we enrolled non-deaf trained participants. The 40 participants were selected from King Saud University, including researchers and undergraduate and postgraduate students. The participants' ages range from 20 to 48 years old, with heights ranging from 1.5 to 1.76 meters. Subjects were trained on the selected 80 signs during training sessions that lasted a few weeks and allowed the subjects to perform the signs with confidence.

4.3. Recording Devices and Configuration

The dataset was recorded using one RGB camera (Sony, HDR-CX405, Tokyo, Japan) and two Microsoft Kinect cameras (Microsoft, versions 1 and 2, Washington, United States). Kinetic cameras recorded separate videos for each sign using a graphical user interface. This helps to check each sign individually during recording and to repeat incorrectly performed signs. Videos of both Kinect cameras were captured using the same software, providing identical start and stop times. This helps synchronize the data of both cameras and allows the frame content of both videos to be analyzed using the same index. The RGB camera was used to record continuous video of the entire recording session. Continuous videos can be used for a real-world validation scenario where the sign recognition system should detect signs from a continuous video and recognize them in real-time. Configurations of cameras and recording modalities are detailed in *TABLE 3*.

TABLE 3, RECORDING DEVICES USED IN THE KSU-ARSL DATASET AND THEIR CONFIGURATIONS

Device	Modality	Image Size	Field of View	Depth Distance	Bits	Additional Info
Kinect V1	RGB	640 x 480 x 3	58.5° x 46.6°	_	24 bits	Up to 30 fps
	Depth image	320 x 240 pixels		0.4 to 8m	11 bits	
Kinect V2	RGB	1280 x 920 x 3	70.6º x 60°	_		Up to 30 fps
	Depth image	512 x 424 pixels		0.5 to 4.5m	13 bits	
	Skeleton Data	2D points		_		
Sony HandyCam	Continuous recording	1920 x 1080 HD	_	_	_	24p/ 60i/ 60p

The recording process began with some test samples to estimate the recording duration and set up the cameras. The cameras were placed in front of the subject at a height of 1.3 meters from the ground, as shown in FIGURE 7 and FIGURE 8. The distance between the subjects and recording devices varied between 1 and 1.2 meters to reflect real-life scenarios. On average, each session lasted over

20 minutes followed by a rest period of 5 to 15 minutes (depending on the subject's endurance). Each subject took around three hours to complete the five recording sessions. Two to three subjects were recorded per day.

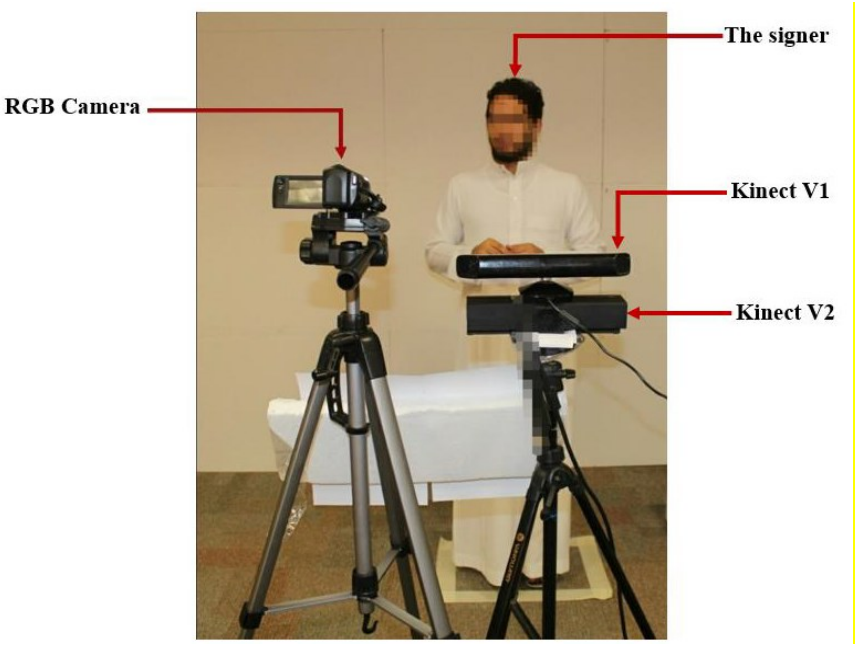


FIGURE 7 RECORDING SETUP USED IN THE KSU-ARSL DATASET

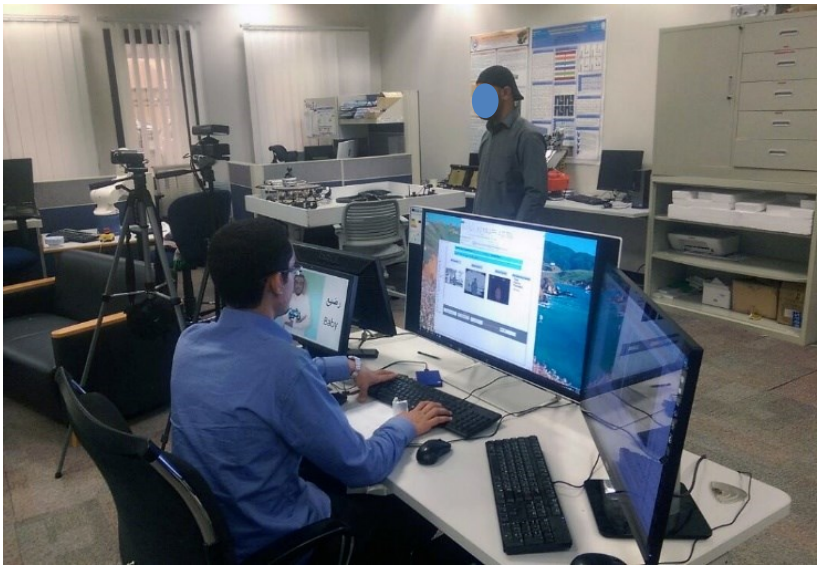


FIGURE 8, RECORDING ENVIRONMENT FOR THE KSU-ARSL DATASET

In contrast to many other sign language datasets, the KSU-ArSL dataset was recorded in a completely uncontrolled environment to reflect real-world conditions. No restrictions were imposed on lighting conditions, background color, and subjects' clothing, which makes the KSU-

ArSL dataset a very challenging dataset. Each video was captured in a room with a light gray background and daylight lamps. Subjects were allowed to wear any cloth to reflect real-life wearing conditions. Sample frames from the KSU-ArSL dataset are shown in FIGURE 9.

FIGURE 9, SAMPLE FRAMES FROM THE KSU-ARSL DATASET

The total size of the dataset, which included the Kinect v1 (RGB + Depth), Kinect v2 (RGB + Depth + Skeleton), and the Sony camera, was about 450 GB. Each subject recorded 80 videos per session (per camera), and each sign was captured five times by 40 sub-jects, for a total of 16,000 videos per camera. Each video was captured at a frame rate of 30 frames per second.

4.4. Data Verification and Post Processing

The recording team consisted of two people: a recorder, to control the recording process, and an ArSL expert, to check the quality and correctness of each sign. During recording, the ArSL expert checked and registered the following issues in the verification form: incorrect sign,

incorrectly performed sign (i.e., wrong hand used, wrong movement order, missing sub-movement of sign), performing extra movement not related to the sign (e.g. touch nose), and a recording error in one of the cameras. Signs that had one of the previous issues were re-recorded. At the end of the day, another verification team checked the videos of the three cameras and backed up the data.

Videos of the two Kinect cameras were manually scanned to remove extra waiting frames at the beginning and end of each video. These frames varied from 60 to 120 frames (two to four seconds).

5. Design and development of a Saudi sign language database.

To achieve this objective, the project team, members from CS2R and HEPD, designed a protocol for the sign language database. The protocol defined the recording environment (light, background, distance between signer and recorder), equipment of recording (RGB cameras, depth cameras, desktops) signs to be recorded, deaf/non-deaf participation, and attendance of sign specialist in the recording. Details of the protocol with justification will be presented in the next sections. We will also present details of the database and its recording.

5.1. Selection of the signs

Since our system is for translating Saudi signs hence we adhered to the Saudi sign dictionary produced by Saudi Association for Hearing Impairment [105]. There are other variants of the signs used in KSA and sometimes the deaf do not follow the dictionary, but it is extremely difficult to include all the variants and we have to follow the most accepted and agreed upon standard which is the Saudi sign dictionary. The dictionary is subdivided into sections of varying importance, ranging from the name of towns to daily usual words, words from the medical domain, etc…

The sign selection is an important step in order to produce a pilot system that will convey the importance and usefulness of the project and be appealing to the deaf and non-deaf community. Moreover, the Saudi sign dictionary is more than 3000 signs and it will be hard to cover them in the project. Hence, we decided it will be better to limit the sign to a certain important domain. The project team discussed this in many meetings and considered many domains such as Abshir (the KSA government e-service), children stories, courts, and other law institutions, universities…. etc. After many meetings, the team decided that the medical field is a very important area where many deaf have difficulties in communicating with the medical staff, such as explaining their illness and/or pain, and our system can be used immediately and be beneficial. In addition to selecting signs from the medical field, we selected other daily life signs that are needed to build a system in the medical field and also needed in daily life communications. The team has 4 sign specialists with good contact with the deaf and the deaf society, hence these members suggested the initial draft list of the signs. This list was discussed and refined many times in many meetings (4 or 5) until we narrowed it to 293 signs, i.e. around 300 signs. We originally planned for around 200

signs based on our estimation of the time needed to record the videos of the signs, but we had to come close to 300 signs in order to have signs that are needed to build a useful sign translator that can be used in the medical field and very common daily life communication. The selection of the signs concentrated mostly on the signs that will most probably be used in the daily medical life of any dialogue between a deaf or hard of hearing and a nurse or doctor.

This selection took a long time of reflection and consultation as the medical signs within the dictionary are about 150 signs, while practically the more there are signs to be video recorded in many sessions, the more the time for each signer increases, and this might be a burden for some signers and need time longer than can be in the project time frame. In this context, we selected 293 signs, as presented in Table 4.

TABLE 4, LIST OF THE 293 SELECTED SIGNS

ة t	ب b	ال al	آ	ا	ئ e	إ i	ؤ o	أ a	ء hamza
س s	ز z	ر r	ذ dh	د d	خ kh	ح h	ج dj	ث th	ت t
ك k	ق q	ف f	غ gh	ع e	ظ dh	ط t	ض dh	ص s	ش ch
3	2	1	ي y	ى y	و w	ه h	ن n	م m	ل l
			10	9	8	7	6	5	4
الجمعة Friday	الخميس Thursday	الأربعاء Wednesday	الثلاثاء Tuesday	الاثنين Monday	الاحد Sunday	السبت Saturday	يوم Day	سنة Year	أسبوع Week
أسفل Down	زواج marriage	أسرة family	ابنة daughter	ابن son	أخت sister	أخ brother	أم Mother	أب Father	أمس yesterday
خلف behind	ثم Then	تحت Under	بعد after	بدون without	أنت You	أنا Me	أمام ahead	إلى to	أعلى upper
يختفي To disappear	يجلس To sit	يأكل To eat	منذ since	مع With	قبل before	في in	فوق above	على upon	دائما Always
يصعد To ascend	يشم To smell	يشرب To drink	يسمع To hear	يستيقظ To Wake-up	يستطيع Can	يستحم To shower	يساعد To help	يدخل To enter	يخرج To go out
السلام Peace	الخير Goodness	اسم Noun	ينزل To descend	ينام To sleep	يمشي To walk	يقف To stand up	يفتح To open	يغلق To close	يظهر To appear
مرات Many times	بيت House	ثلاث thrice	بوابة Gate	بسبب Because	أين where	وسهلا welcome	أهلا Hello	النور light	عليكم on you
غدا	عفوا	صباح	شكرا	ساعة	ربع ساعة	رئيس	رئيس قسم	دورة مياه	حسنا

tomorrow	Excuse me	morning	Thanks	hour	quarter an hour	President	Head of Department	W.C	Okay
مرة Once	متى when	ماذا what	لو سمحت Excuse me	لماذا Why	لا No	كيف How	كم How many?	قسم Department	في الخارج out
آل سعود Al (family)	يمين right	يسار left	وعليكم السلام May peace be upon you	هل Does?	هذه This (female)	هذا This (male)	نهار day	مساء evening	مرتين twice
الدمام Dammam	أبها Abha	الملك فيصل King Faisal	الملك فهد King Fahd	الملك عبدالله King Abdullah	الملك عبدالعزيز King Abdulaziz	الملك سلمان King Salman	الملك سعود King Saud	الملك خالد King Khaled	الأمير محمد بن سلمان Prince Mohammed bin Salman
أزمة قلبية Heart attack	إزالة Removal	أرق insomnia	إدمان addiction	إجهاض abortion	مكة المكرمة Makkah	جدة Jeddah	جازان Jazan	المدينة المنورة Medina	الرياض Riyadh
اعتذر I apologize	إعاقة Obstruction	إعاقة سمعية Impaired hearing	إصابة infection	أشعة ليزر laser beams	إشعاع radiation	إسهال Diarrhea	أسنان teeth	إسعاف Ambulance	استخدام Usage
بخير Fine	انتشار Spread	إمساك constipation	النفس self	ألم pain	الطب النفسي Psychiatry	الصحة Health	الخدمة In service	التهاب inflammation	اكتئاب Depression
تنفس Breathing	تقرير report	تغيير To change	تعب fatigue	تدليك massage	تحليل دم blood analysis	تأمين insurance	بهاق vitiligo	بنج Anesthesia	بطن Belly
رئتان lungs	ذراع arm	حساسية sensitive	حروق burns	حادث مروري Traffic accident	جهاز قياس الضغط Blood pressure device	جهاز قياس الحرارة Temperature measuring device	جفن العين Lid	جرح injury	تورم swelling
سماعة أذن headphone	سم poison	دورة شهرية Menstrual period	دوار Vertigo	دواء Medicine	دم blood	حيض menstruation	حمى Fever	حمل Pregnancy	حقنة Injection
طبيب Doctor	ضماد طبي medical bandage	سكر sugar	سكتة قلبية Heart failure	سرير bed	سرطان cancer	زكام common cold	رقبة neck	رضيع infant	رحم womb
غدة gland	عينة مختبر lab sample	ضغط الدم blood pressure	صيدلية pharmacy	صيدلي pharmacist	صورة أشعة X-ray	صداع headache	شلل نصفي hemiplegia	شعور Feeling	شرايين arteries
قدم Foot	فيروس virus	عناية Attention	عمود فقري Back bone	عملية جراحية Surgery	عظم bone	عزوبة celibacy	عدوى infection	عدم nil	ظهر Back
كوع elbow	كعب القدم Heel	فيروس كورونا Corona Virus	فك untie	فقدان المناعة Immunodeficiency	فشل كلوي Renal failure	فرشاة أسنان Toothbrush	فحص examination	فحص سريري Clinical examination	فحص النظر Eye examination
يدخن smokes	مدير director	كرسي a chair	كبد liver	قياس السمع audiometry	قوقعة cochlea	قلب heart	قفص صدري Chest	قطرة Drop	قصبة هوائية Trachea
مناعة immunity	ممرضة nurse	مخدرات drugs	مختبر laboratory	ليل night	لم أفهم I do not understand	لحظة One Moment	لاصق جروح Adhesive wounds	لا بأس Fine	لا يستطيع can't
ولادة Birth	وصفة طبية Prescription	ملح salt	مغص colic	معجون أسنان toothpaste	مصعد elevator	مستشفى hospital	مريض Sick	مرهم ointment	مرض السكر Diabetes
		وراثة heredity	وجه Face	هيكل عظمي Skeleton	هاتف phone	نزيف bleeding	نتيجة result	نبضات القلب heart beats	ميزان حرارة Thermometer

5.1.1. Types of signs

Some of the signs are just a simple fixed hand position of one or two hands, hence are called static signs, while the rest of the signs use a moving action with one or two hands, hence are called dynamic signs. In all cases, any sign can be viewed as dynamic if the pre-action (hands up) and post-action (hands down) are included in the whole sign video. Since we hope that our system will allow the deaf to use it live with minimal restrictions, hence all signs will be considered dynamic, where we consider that the middle frames of interest of each sign, are preceded and followed by some transition frames. A sample sign language performed by the hands is shown in FIGURE 10.

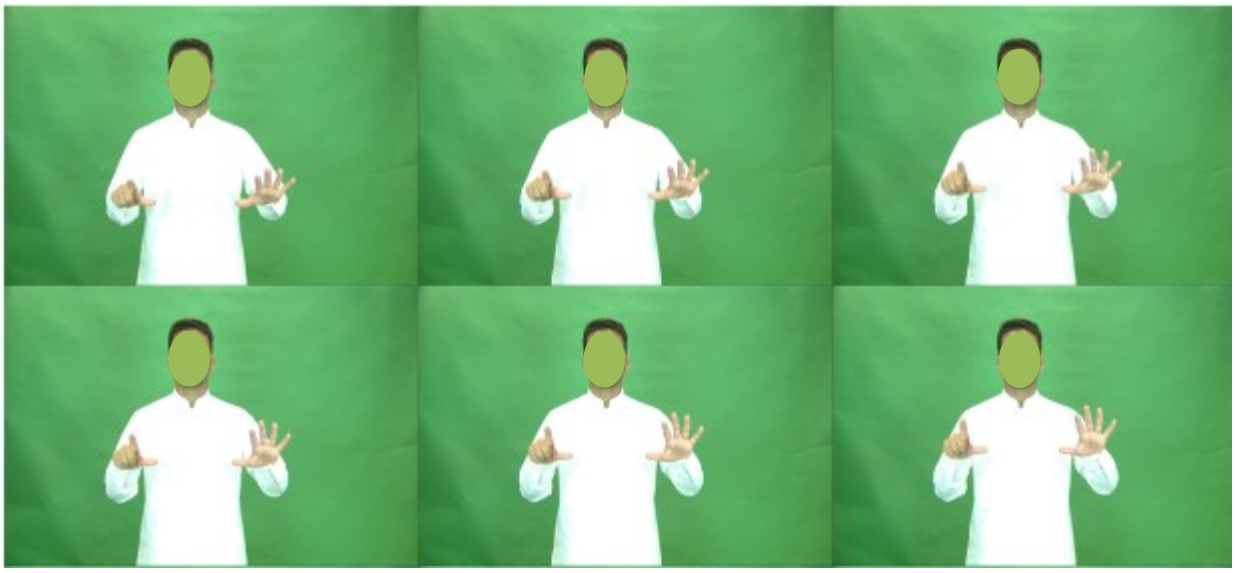


FIGURE 10, EXAMPLE OF THE SIGN LANGUAGE GESTURE –WEEK

It is to be noted that defining the number of signs to be recorded was only one of the numerous characteristics of the database that were discussed and refined in many meetings of the project team. The characteristics were interrelated, as deciding on one characteristic may depend on another characteristic and affect other characteristics, hence arriving at the final decision on the characteristics took a long time, many meetings and discussions, many consultations with outside experts, and many testings.

As an example of the discussions of the project team, Table 5, shows the minutes of one of the meetings needed to arrive to the final decision. Further details of the characteristics of the database and the recording studio will be presented in the next sections.

TABLE 5, MINUTES OF ONE OF THE MEETINGS FOR THE KSU-SSL RECORDING PREPARATION AND OTHER POINTS

	Conditions for the sign recordings (Previously agreed points)
1	Fast Camera (120 FPS). Responsible member: Dr. Bencherif + Eng. Mekhtiche
2	Distance from Camera? (Depends on the Camera Lens) Responsible member: Eng. Mekhtiche
3	Camera calibration? Responsible member: Eng. Mekhtiche
4	Gloves versions. Responsible member: Dr. Bencherif + Eng. Mekhtiche
5	Nine repetitions (3 without gloves without background color, 3 without gloves with background color, and 3 with gloves without background color). All with a background box.
6	Verification of performance of the signs. (During the performance by a human)
7	Calibrate the background box for each signer.
8	Waiting in the last position after performing each sign. (3 seconds for each sign)
9	Pausing between signs.
10	For static signs, no need to return the hand to the initial position. (only the original sign)
11	For dynamic signs, the hands should be returned to the starting position.
12	Speed of signs for static and dynamic. (same speed as performed by Muneer)
13	Five signers + 9 repetitions + 80 signs. (Still under discussion and we did not agree)

The main characteristics of the database and the recording studio were gathered from the many meetings, hence we list some of the most important choices in TABLE 6.

TABLE 6, the most important choices in developing the KSU-SSL dataset

Database Characteristics	
Signs to record	Both static and dynamic signs
Percentage of dynamic signs	Use of gloves
Number of repetitions of each sign	Number of signers

Participation of deaf and non-deaf as volunteers	Supervision by a deaf specialist
Maximum time for daily record	Specification of the recording studio
Use of high-speed camera	Use of camera of Mobile
Use of additional Lightening	Distance between the volunteer and cameras
Height of Cameras	Type and color of the background
Use of a standing mark up in the ground	Types of cameras
Number of Cameras	Use of IR camera

5.2. Design of the Recording System of KSU-SSL database

5.2.1. Preparation and testing the recording environment

Recording videos of high quality requires very experienced people and material, as well as a very stable recording machine and excellent programming skills. In this context, our specialist in charge managed to create a video recording studio as shown in Figure 11.

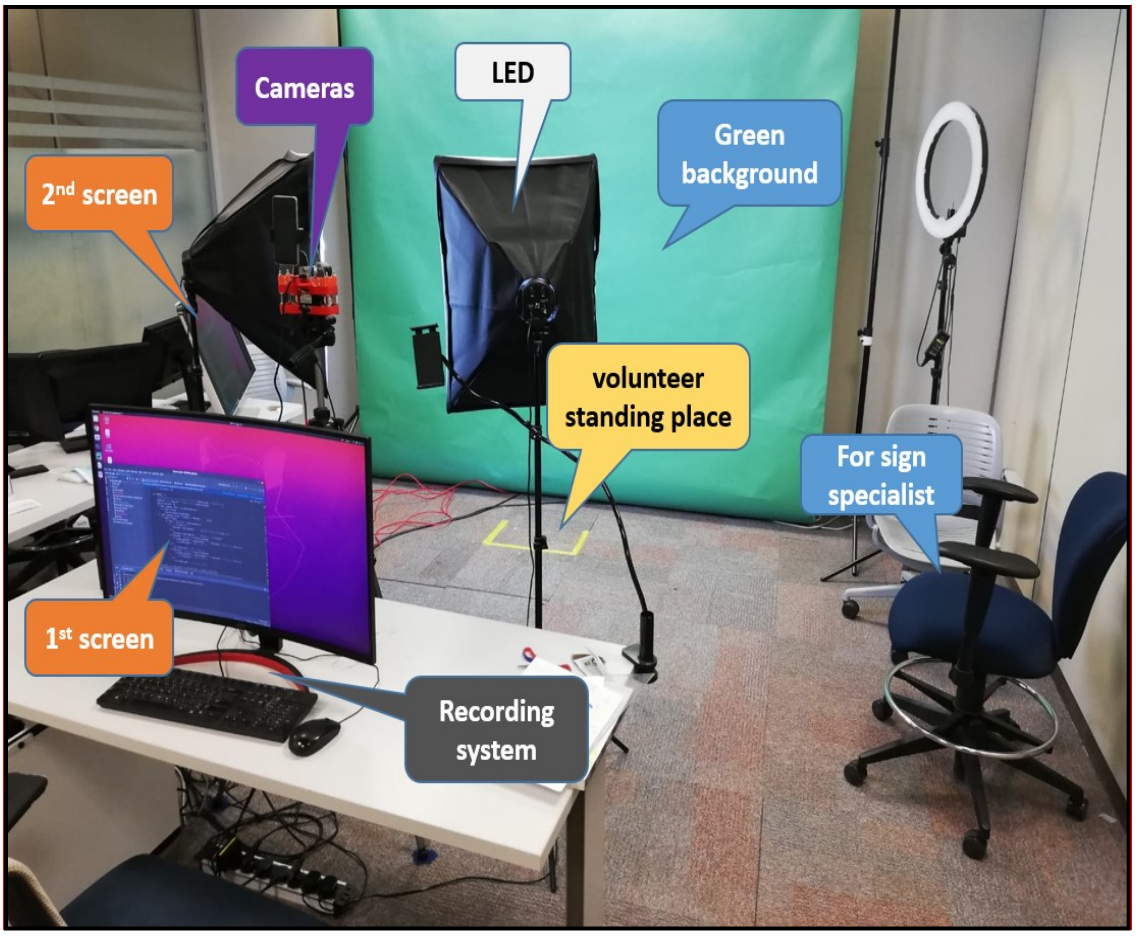


FIGURE 11, RECORDING STUDIO DESIGNED BY OUR EXPERT

The designed studio is comprised of the material listed in TABLE 7. We opted for this design after many trials and recording tests, as well as changing many lightings and camera positions. Different setups and configurations were tried for the recording studio until we reached an optimum setup and configuration.

TABLE 7, RECORDING STUDIO COMPONENTS

Item	Description	Remarks
acA1920-150uc - Basler ace	RGB color camera	High speed frame rate up to 150fps
ELP-USBFHD05MT-KL36IR	Infrared camera	used to record IR videos
Huawei P20 pro	Mobile	To record with a phone camera
LED	Light Stand	Used for better light conditions
First Screen	32” screen	For recording team member
Second Screen	32” screen	For the volunteer to see reference videos.
Camera Holder	Holds RGB and Infrared cameras and mobile	Made by the team
chair	Blue chair	For sign specialist
Marked space in the floor	Marked space where the volunteer will stand	To make sure all volunteers will be facing the camera at the same distance
Paper background	The paper background behind the volunteer	to facilitate the video processing after recording

- The recording of the signs was done inside the arena of the Center for Smart Robotics Research at the College of Computer and Information at King Saud University, where a special computer with high specifications was prepared so that it could record high-definition videos at high speed. The recording set up contained the computer with a recording system, two screens, two cameras, a mobile, and the cameras stand. One of the two screens is facing the team member who is working on the recording system, and the other screen is towards the volunteer who performs the signs to display the reference videos to the volunteer.

- The recording is done using two cameras and a mobile. The main camera is a high speed colored camera and the other is an infrared camera. The two cameras are on a stand. The volunteer stands in front of the cameras at a distance of 2 meters so the upper body of the volunteer will occupy most of the field of view of the camera. The expert sits on a chair from an angle that allows him to watch the volunteer performs the sign to be recorded and corrects him when needed.

- The two cameras were connected to the recording computer, where the recording program records videos for each repetition of each sign from the two cameras at the same time. As for the mobile, the recording takes place throughout the session, then later the signs can be cut manually.

- Color camera: In order to avoid the blurring effect generated by low fps cameras, which we encountered in our previous database KSU-ArSL, where some fast actions were not well recorded and appeared as shadowed actions. Our experts investigated the use of a very advanced camera, with high fps and choose an industrial camera (acA1920-150uc - Basler ace) with a high frame rate. The camera originally has a 120fps, but during testing, we noticed that 60fps were very satisfactory frames per second, and the actions or recorded signs could be tracked easily with all the images of the video. This camera can record videos with 1920*1200px at 150fps, but at high FPS the images will be a little bit dark because the sensor will not have enough time to receive the light from the lens. To solve this issue, first, we changed the lens of the camera with a new lens that has a big aperture, then in the second step we reduced the frame rate to 60fps where we get the best recording videos with our light conditions. The second issue was that the recording PC was taking too much time to save those videos to the drive and the videos were occupying a large space in memory, so we decided to change the camera setting to keep capture videos at 60fps but it will send only 30fps to the recording PC.

- Infrared camera: It is a special camera that captures the reflection of infrared rays on objects. This camera was chosen in order to avoid the problem of lighting, as these cameras depend on IR LEDs that are installed around the lens. To further improve performance, the technical team added an additional IR LED to the camera.

- The signs Recording Studio contained a place marked with a yellow box, where the volunteer stands to record, and to his right a screen was placed showing a video of the sign to be recorded so that it was easier for the volunteer to perform the signs correctly, as shown in FIGURE 11.

- A white LED lighting has also been added to the studio to improve the quality of the captured videos, and a green background has been added behind the recording location to facilitate the video processing after recording.

- As our videos are recorded from three sources: a color camera, an infrared camera, and a mobile phone camera, and in order for all the videos to be captured from the same location, the technical team designed a special stand as shown in Figure 12 to hold the two cameras and the phone, so all recordings will be from the same direction at the same distance.

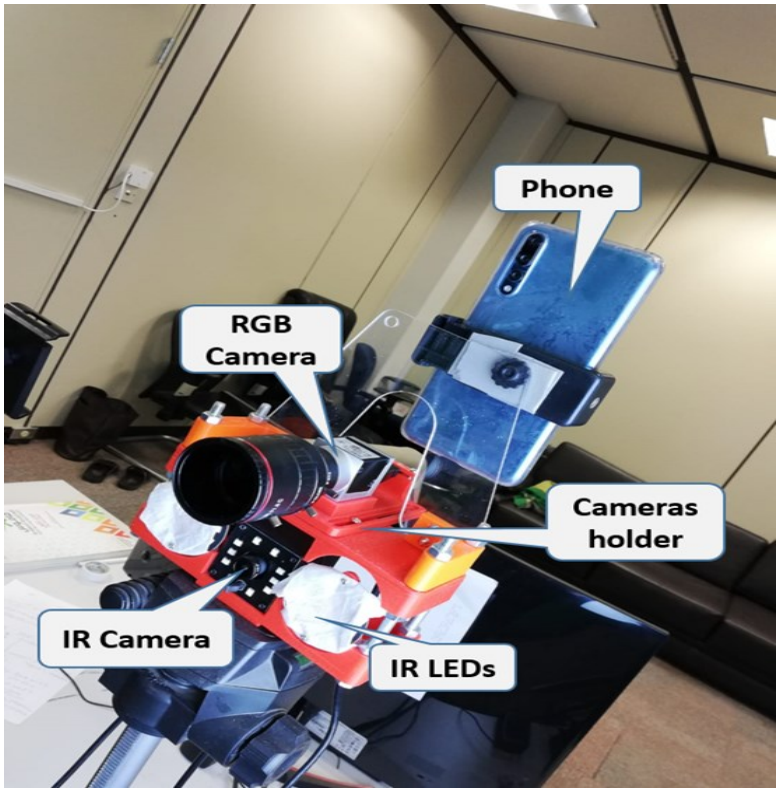


FIGURE 12, SUPPORT FOR THE CAMERAS AND THE PHONE

5.2.2. Building the recording system

The project's technical team developed a system to record the videos of the required signs. The system contained reference video clips of all the signs to be recorded. The reference video clip for each sign was shown to the volunteer before recording the sign, to help him remember the correct way to perform the sign.

The recording system has been developed taking into consideration the following points:

- The program contains a "**signer#**" field to specify the volunteer's number.
- When the recording member presses the "Open" button, as shown in FIGURE 13, the program opens the sub-folder of the selected signs and creates folders for saving those signs automatically, in order to avoid errors and standardize the pattern of saving in the database.
- The program records the signs from the two cameras (color and infrared) connected to the device at the same time as shown in FIGURE 13.
- The recording member can choose the number of sign repetitions to record by entering the number of repetitions in the field "Max_rep" as shown in FIGURE 13.
- The program also contains a "Color" check button which can be specified by the recording member to record signals with colored hands (originally it was with gloves)
- When recording begins, the program displays a video illustrating the way the sign is performed, as shown in the right display part of FIGURE 13.
- The program records each sign several times in succession, according to the number of repetitions entered, then moves to the next sign automatically and displays the reference video for it.
- The recording system contains a list of the signs to be recorded. It also contains a box for the sign number "**Current Rep**" where the recording coordinator can choose any sign and specify the repetition number to be recorded in case the coordinator wants to re-record a specific sign.

- The program, as shown in FIGURE 13, contains a "**Start REC**" button to start and stop recording, and an "**Ignore**" button to cancel the recording in case something went wrong.

- At the end of recording each session, the coordinator presses the "**Check Files**" button so that the program automatically checks that all the signs of the selected section are recorded and that the recordings are intact.

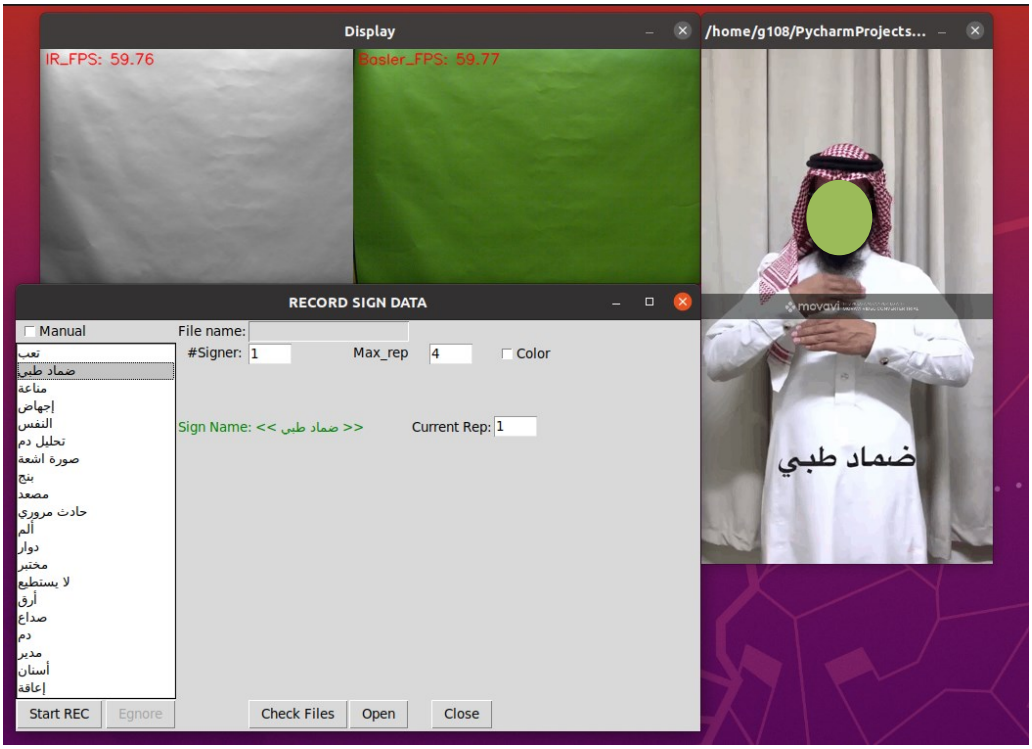


FIGURE 13, RECORDING SOFTWARE INTERFACE (SAMPLE SIGN ON THE RIGHT)

5.2.3. Archiving and saving protocol

With the beginning of the recording of each volunteer, the recording system creates a folder called "**signer_X**" where "**X**" stands for the volunteer's number, then under this folder, the program creates two folders "**group_1**" and "**group_2**" and within each of these two folders sub-folders are created for the signs sections according to following:

- The folder "**group_1**" contains the following subfolders:
    - Alphabet
    - Common

  - Days
  - Family
  - Numbers
  - Pronouns and adverbs
  - Verbs

- The folder "**group_2**" contains the following subfolders:
  - Hospital and common
  - Kings
  - Regions

Two copies of the recordings are made on different hard drives. In addition, after the daily session, two copies are uploaded into cloud storage: one copy to the **IDRIVE** cloud storage site and another copy to the **Google Drive** cloud storage site.

A follow-up form was created using Excel, which includes the list of participants, to follow up on the volunteers recording, the specialists review of the recorded signs if the signs were recorded in their absence. This form is shown in FIGURE 14 below.

		group 1						group 2		
Number of signs		37	39	11	8	11	18	20	133	9	7
gner numb	signer name	Alphabet	common	days	family	numbers	ouns and adv	verbs	hospital and common	kings	regions
Signer 1		370	390	110	80	110	180	200	1330	90	70
Signer 2		370	390	110	80	110	180	200	1330	90	70
Signer 3		370	390	110	80	110	180	200	1330	90	70
Signer 4		370	390	110	80	110	180	200	1330	90	70
Signer 5		370	390	110	80	110	180	200	1330	90	70
Signer 6		370	390	110	80	110	180	200	1330	90	70
Signer 7		370	390	110	80	110	180	200	1330	90	70
Signer 8		370	390	110	80	110	180	200	1330	90	70
Signer 9		370	390	110	80	110	180	200	1330	90	70
Signer 10		370	390	110	80	110	180	200	1330	90	70
Signer 11		370	390	110	80	110	180	200	1330	90	70
Signer 12		370	390	110	80	110	180	200	1330	90	70
Signer 13		370	390	110	80	110	180	200	1330	90	70
Signer 14		296	312	88	64	88	144	160	stoped at "examination"	72	56
Signer 15		296	312	88	64	88	144	160	stoped at "hearing test"	72	56
Signer 16		370	390	110	80	110	180	200	1330	90	70
Signer 17											
Signer 18											
Signer 19											
Signer 20											
Signer 21											

record with painting	
record 4 repetitions	
all recorded	

FIGURE 14, DAILY FOLLOW-UP FORM.

5.3. Recording of the KSU-SSL database

5.3.1. Selection of Volunteers

The project aimed to record the signs of 30 volunteers who may be deaf, hard of hearing and hearing. The initial volunteers were selected from non-deaf students from King Saud University, and they were contacted to explain the project and its goal and the need for a sign database for the project. Having a short time of recording is very important, hence at the initial stages of the DB recording and before accepting the first volunteers we sent each of them a representative sample of the signs (10%) about 30-35 signs to practice. When the volunteer is ready we record his signs, then we send their videos of performing the signs to the specialist who will choose those with near-perfect performance. This step is necessary because it will save us time when recording all signs of the database of the project. After accepting the volunteers, we sent them pre-recorded reference videos of the project signs so that the volunteer can train on them for a period ranging from two to five days before recording his signs. After recoding about 7 initial volunteers and being satisfied with recording process, we stopped the initial checking of the volunteers by sending them about 10% of the signs to practice.

We have completed the recordings of 32 volunteer signers (9 deaf, 3 hard hearing, 3 sign experts and 17 non-deaf). Each signer performed 4 sessions (repetitions) without wearing gloves or colored hands, and one session with gloves or colored hands.

5.3.2. Number of repetitions of the signs

In our previous database, KSU-Asrl, we recorded five repetitions of each sign by each volunteer. While surveying the literature for work on sign recognition we found that some databases were recorded with the volunteers wearing gloves, hence we thought it will be beneficial and interesting to include at least one recording of each sign with the volunteer wearing gloves. To keep the recording time reasonable, we recorded four repetitions without gloves and one with gloves. As will be explained in section 5.3.6 we later switched to recording with the hands and the fingers painted in color instead of wearing gloves in the hands.

5.3.3. Reference signs by experts

As we discussed in section 5.2.1 the recording setup included having a screen in front of the volunteer where we display a reference video of performing the desired sign. Our sign experts could not record the reference videos in the recording studio, hence they recorded it on their mobile and sent it to us. We wanted the reference videos to be in the same setup as the actual recording, hence selected experienced members of the project recorded the signs in the same setup as the actual recording based on the videos sent by the sign experts. Then we sent these new videos to the sign specialists to verify the correctness or ask to repeat wrong signs. When all signs were accepted by the sign specialists, we started recording the volunteers.

5.3.4. Estimation of the recording time

Knowing the time needed to record each volunteer is important because is is needed for many points of the execution of the project: we will know the total time needed to record all volunteers; hence we may keep or reduce the number of signs to be able to finish within the time allocated in the plan, and to help in deciding the honorarium to give to each volunteer. Hence, we choose two of the best volunteers and started with them and recorded their signs. This gave us a good approximation of the time needed for recording and based on it we planned the time for the recording operator, the specialist, and the volunteers. The recording took approximately 4 to 5 hours daily for three days divided into two sessions with a recess in between. This time also

conformed with our previous time estimation on which we limited the number of signs to almost 300 signs.

5.3.5. Recording Mechanism

At the beginning of the recording, several recording mechanisms were tried in order to determine the appropriate mechanism that achieves the highest quality with the least effort. It was required to record each sign 5 times, including 1 time with gloves (later with painted hands). The recording was 5 days a week, from Sunday to Thursday under the supervision of the team specialists in sign language.

The signs that had common denominators were collected and placed in one folder. The signs of the first section, which consists of 144 signs, were divided into 7 folders as follows (letters, numbers, days, family, pronouns and adverbs, verbs, common). For the second section, which consists of 149 signs, the signs were distributed into 3 folders (Saudi Kings, regions, medical, and common). One specialist attended the first session, and another specialist attended the second session. All the signs were recorded in reference videos as described in section 5.3.3, then the videos were sent to the volunteers to train on them before coming to the studio for recording. Each volunteer was recorded in five sessions with an average of 15 hours divided over 3 days.

5.3.6. Recording with painted hands

After recording some initial volunteers and checking the videos with the volunteers wearing gloves, we found that there were problems with recording with gloves, sometimes there will be light reflection, and hand fingers are not clear, as the light is reflected from the gloves, as shown in FIGURE 15.

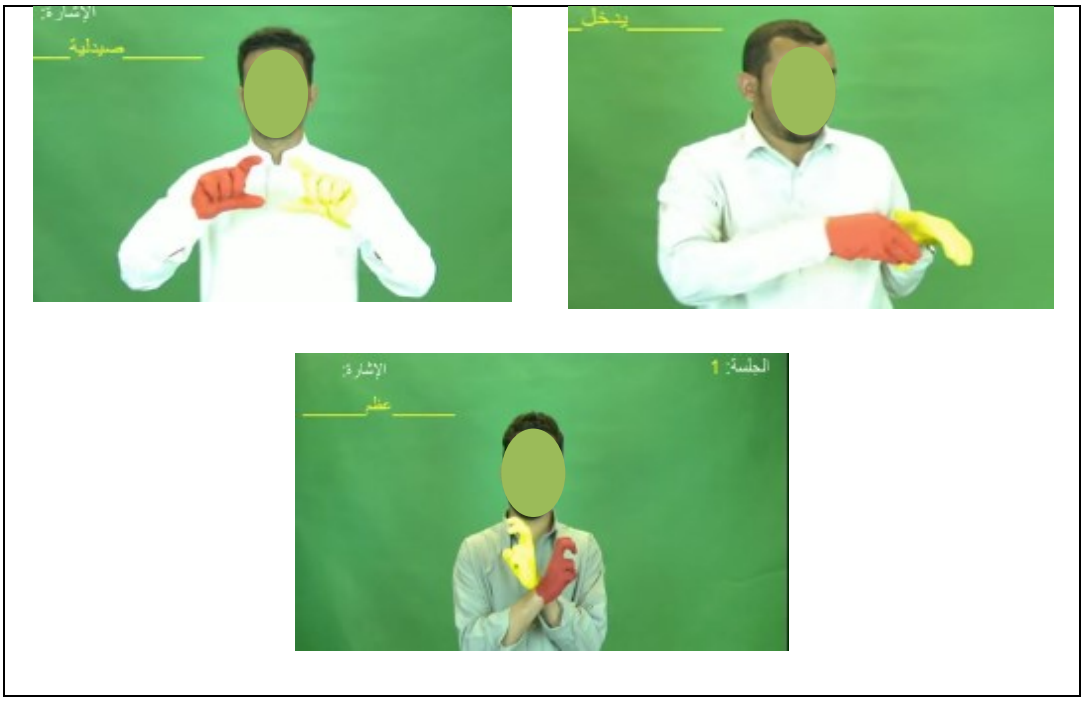


FIGURE 15, COLORED GLOVES IN SIGN RECORDING

Hence, we substituted recording with gloves to recording with the hands painted in colors. We choose blue for the right hand and red for the left hand because these two colors are dark colors and are the best light-absorbing colors, hence the light will not reflect and affect the camera. The green color, which is also dark color and light-absorbing, was used as the background, hence we avoided using it to paint the fingers. We tested our choice of blue and red and it was a good decision, as shown in FIGURE 16.

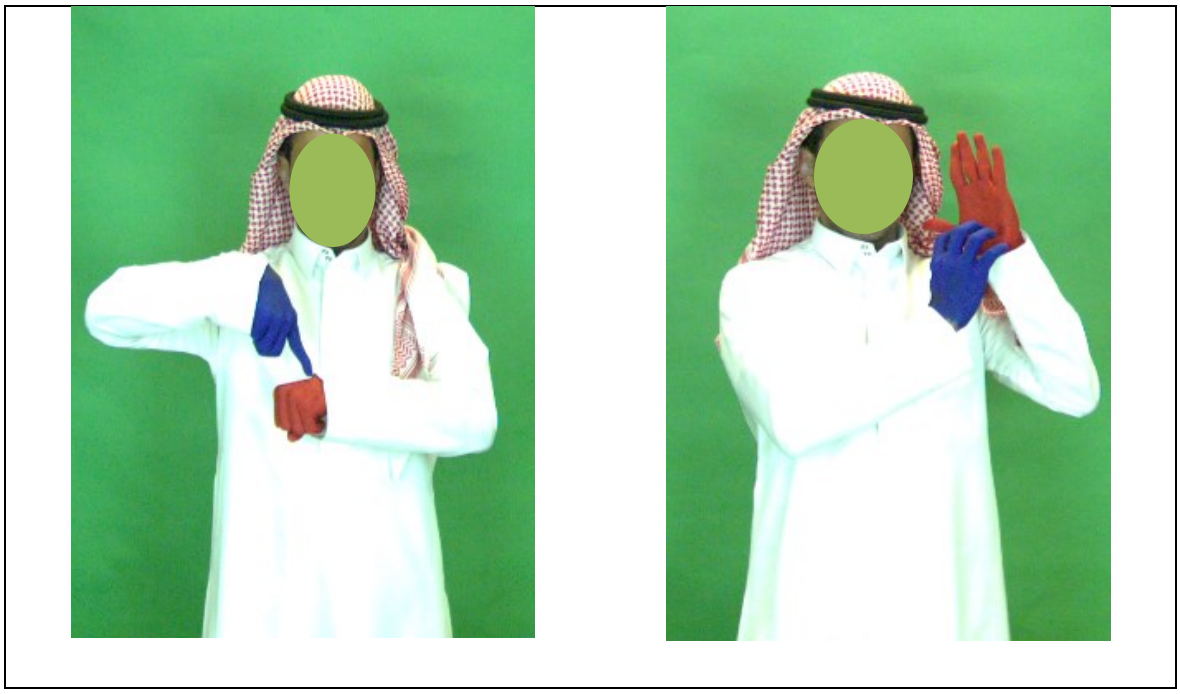


FIGURE 16, COLORED HANDS SIGN RECORDING

We recorded first the four repetitions of all the signs without painting (or gloves for the first 6 volunteers), usually in two days per signer, next we recorded the signs with the hands painted (or with gloves) in the last recording session, usually at the 3rd day.

5.3.7. Recording some signs without the presence of sign specialistS

With the start of the registration of the seventh volunteer, the sign specialists could not attend the recording sessions due to the corona effect, hence we video recorded the next four volunteers without the supervision of the specialists, then the recordings were sent to the specialists to check the correctness of the signs and to identify the signs that need to be repeated. The signs in error were recorded again with the presence of the specialists.

By analyzing the signs in error, we find that they were 23, 10, 10, and 10 for the four volunteers respectively. This is almost 4.5% as a total, while for 3 of them the percentage in error is 3.4%. These volunteers were non-deaf and performed the signs without the supervision of the sign specialist, but were only supervised by our recording team member, hence this low error is very acceptable. From this low error we can see that our team members making the recording were experienced enough to correct supervise the volunteers. Moreover, to the team members who are not sign specialists, we consider most of the errors, in performing the signs, as pointed out by the

sign specialists, as truly acceptable variations of performing the signs, and the faulted signs can be part of the database.

5.4. Recording verification

This phase is used to ensure that the signs have been performed in a correct manner. To ensure that we applied two steps of verification:

- During the sign recoding step: the recording was under the supervision of the team specialists in sign language. It means, that verification is done during the recording session with the attendance of sign specialist.
- The post recording step: the post recording verification step was performed after the completing of all recording session per volunteer. In case of any error or mismatching of the performed sign, the specialist re-train the volunteer and re perform the sign.

5.5. Database labeling and segmentation

In recording our previous database (KSU-Asrl) and to shorten the recording time of the volunteers, who were supposed to be from the deaf community and cannot afford or withstand long recording time, hence we made the video recording continuous for all signs and all repetitions. After finishing the recording, the database team segmented the signs and labeled them. Based on this previous experience we included database labeling and segmentation in the proposal for the project. After the start of the project, we had a choice either to follow the previous method and have a short time at the expense of adding time for labeling and segmentation, which is also not error-proof. The other choice was to record each performance of each sign and save it in a pre-named file then record the next repetition. We went with the second choice although it will take a longer time to build the database and will take a long time for our specialist within their crowded daily schedule at the university, but it will be error-proof.

5.6. Comparison with the KSU-ArSL

One important objective and output of the project is a sign database for the Saudi signs. We are calling this database the King Saud University-Saudi Sign Language (KSU-SSL). KSU-SSL contains 293 recorded signs by 32 Signers. Aiming to accomplish a better dataset, we used our experience from developing KSU-ArSL. Many problems in the quality of the KSU-ArSL

recordings impacted the results of the recognition [48] [49]. In Table 8, we present the pros and cons of the KSU-ArSL dataset.

TABLE 8, PROS AND CONS OF THE KSU-ARSL

Pros	Cons
40 speakers with 5 repetitions	Frame rate of 30fps
Dynamic and Static signs	Blurring in many videos
Multi cameras (KinectV1, KinectV2, Handy-Cam) : multi- channels	Uncontrolled Lighting (ceiling neon bulbs)
Recording of the Skeletons of Kinect V2	Signs included post and pre-recording time; this induced a complementary step for splitting the signs from the video files
	Uncompressed format while recording. (Huge video files, problems in storage and copy)
	Multi-Cameras incurred data bottleneck on the recording station
	Angle of recording not identical for all speakers
	Framing not identical for all signers
	Varied camera setup for all signers (due to some slight position changes of the cameras).
	Background contained some distractive objects
	Skeletons of Kinect V2 delayed sometimes the recording procedure, and brought some bottleneck in the recording station
	Local backup only (delays in storage as files were very huge)

These cons were the apparent deficiencies that have been corrected in the new KSU-SSL. The most important recording deficiencies have been corrected as shown in Table 9.

TABLE 9, IMPROVEMENTS MADE IN THE KSU-SSL

Item	Designation
Frame rate	Increased to 60fps
Cameras	➢ Professional Basler RGB camera ➢ Infra-Red Camera ➢ Mobile phone recording ➢ 3D printed stable Holder for the cameras + Mobile phone
Lighting	Controlled LED lighting with professional materials: ➢ Reflector ➢ Circular LED lamps
Recording	Each sign is recorded in its own video
Number of signs	Increased to 293 (Alphabet, numbers, daily and medical signs)
Framing	Improvement of the signer's position control
Background	Uniform Green Background for future professional video editing
Backups	Daily online backups
Ongoing Hand tracking methods using :	➢ Openpose ➢ Google media pipe ➢ C3D

5.7. Quality of the recording in both KSU-ASL and KSU-SSL

In order to show that the KSU-SSL has a better frame resolution and a better quality of the recording, we present in Figure 17 to Figure 20 some samples of the hands and the fingers at the frame level, for the KSU-ArSL, while in to Figure 21 to Figure 23, we present some samples from KSU-SSL. We could visually notice that the KSU-SSL dataset recordings are clearly non-blurred and most of the frames can be hopefully used for hand and/or finger tracking and sign recognition.

FIGURE 17, ONE HAND SAMPLE FROM KSU-ASL (BLURRED HAND IN MANY FRAMES OF THE VIDEO)

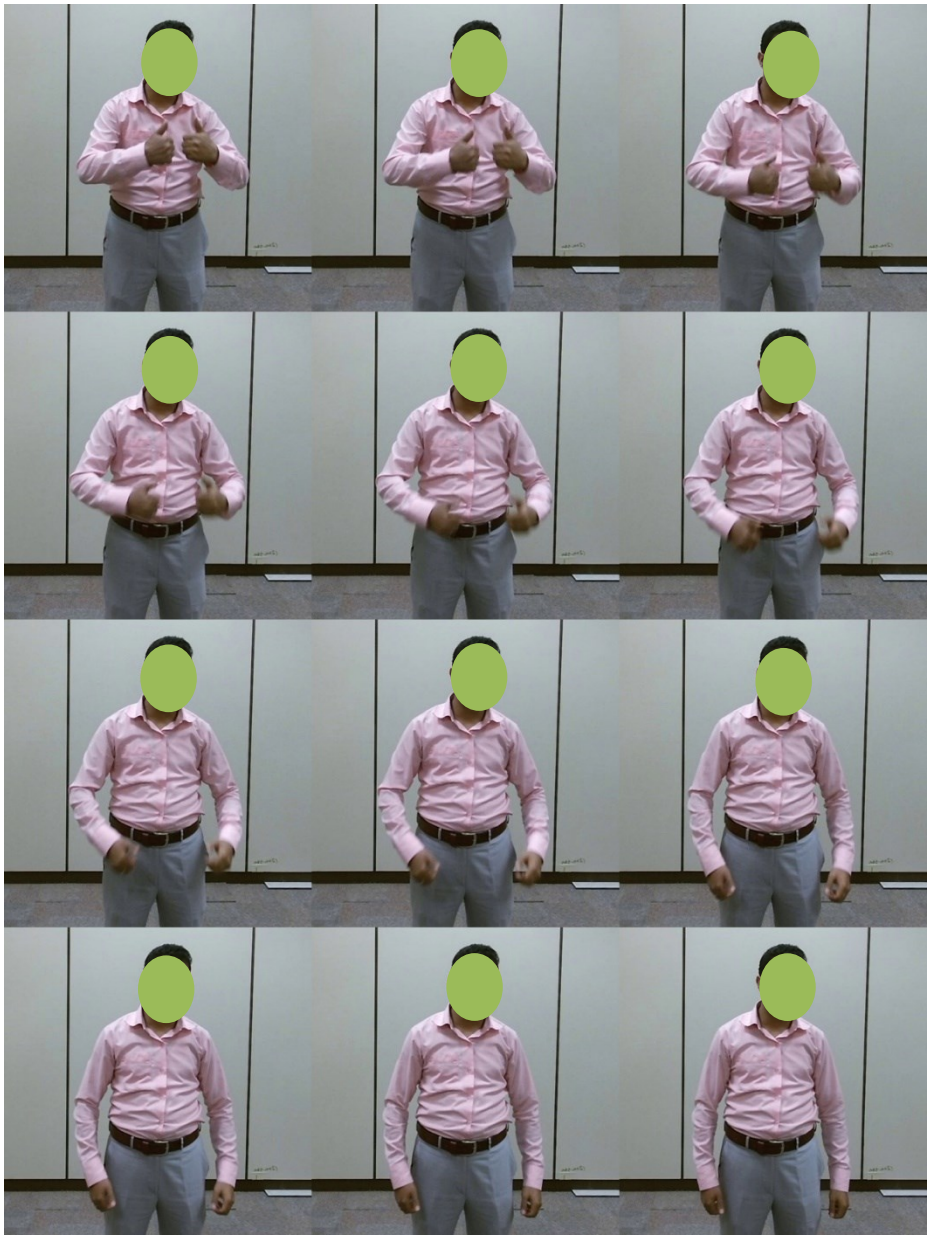


FIGURE 18, TWO HANDS SAMPLE FROM KSU-ASL (BLURRED HANDS IN MANY FRAMES OF THE VIDEO)

FIGURE 19,DETAILED SAMPLE OF A BLURRED HAND IN KSU-ASL

FIGURE 20, DETAILED SAMPLE OF TWO BLURRED HAND IN KSU-ASL

In Figure 21, to Figure 23, we show some samples of the newly KSU-SSL recorded dataset, where the hand is clearly appearing at all frames.

FIGURE 21, SAMPLE FRAMES OF THE VIDEOS OF THE NEW KSU-SSL

FIGURE 22, SAMPLE FRAMES OF THE VIDEOS OF THE NEW KSU-SSL

FIGURE 23, SAMPLE FRAMES OF THE VIDEOS OF THE NEW KSU-SSL

In inspecting the many samples of the KSU-ArSL, we observed that hands or fingers were very blurred at many frames of the videos. This blurring problem occurred essentially, when the signer accelerates the position of the hand, in either the change of position of the hand from mid-screen to rest position (hands down) or rest to up position, as shown in Figure 19 for a one-hand sample and in Figure 20 for a two hands sample. In contrast Samples of the newly KSU-SSL recorded dataset in Figure 21 to Figure 23, show that the hand is clearly appearing at all frames.

One can notice that in KSU-SSL the background is green similar to the case in movies recording, where some special effects need to be added later, after the initial shoots or recordings. In our case we anticipated changing the background to diverse places such as an office, a hospital,

a court, so when we will start training the deep learning models, the view augmentation can be very beneficial for diversity and generalization.

5.8. Hands and fingers detection of samples from the KSU-ArSL

In Figure 24 to Figure 27, we show the Open-pose skeleton results, for our KSU-ArSL, where in many frames the hands were not well detected. This was very cumbersome while training our deep learning networks, as many frames contained valuable information but could not be useful while training the sign recognition system.

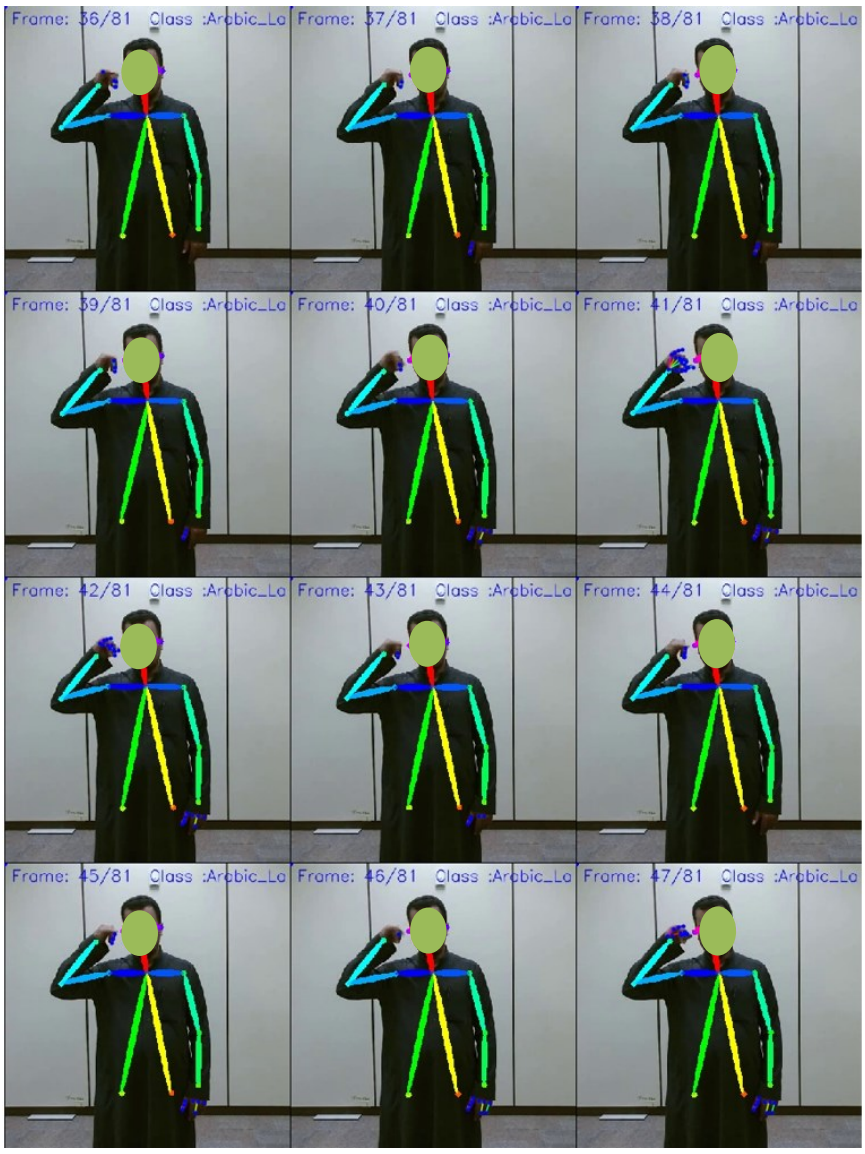


FIGURE 24, OPENPOSE HAND DETECTION OF A FIRST SAMPLE FROM THE KSU-ASL

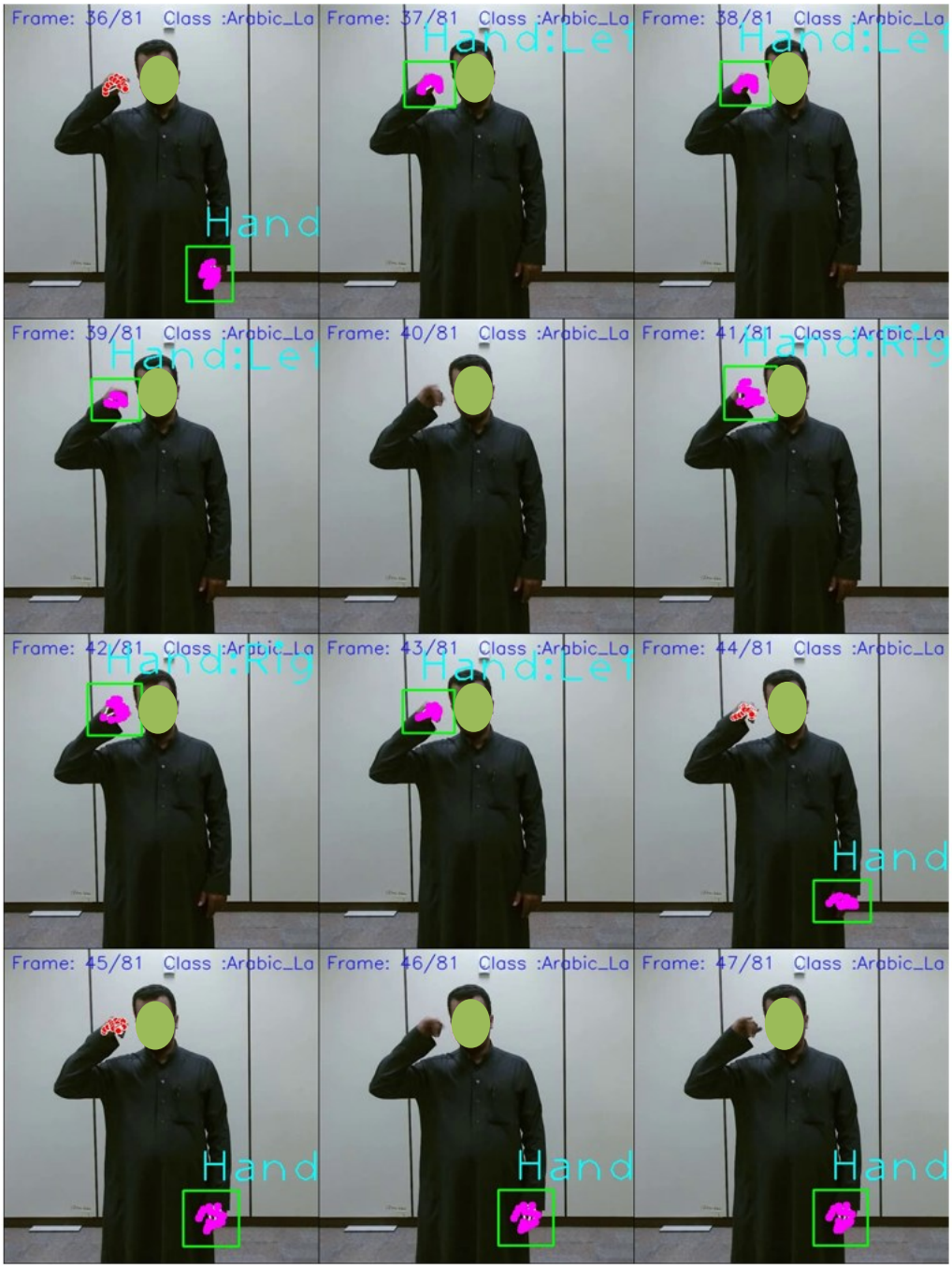


FIGURE 25, GOOGLE MEDIA PIPE HAND DETECTION OF A FIRST SAMPLE FROM THE KSU-ARSL

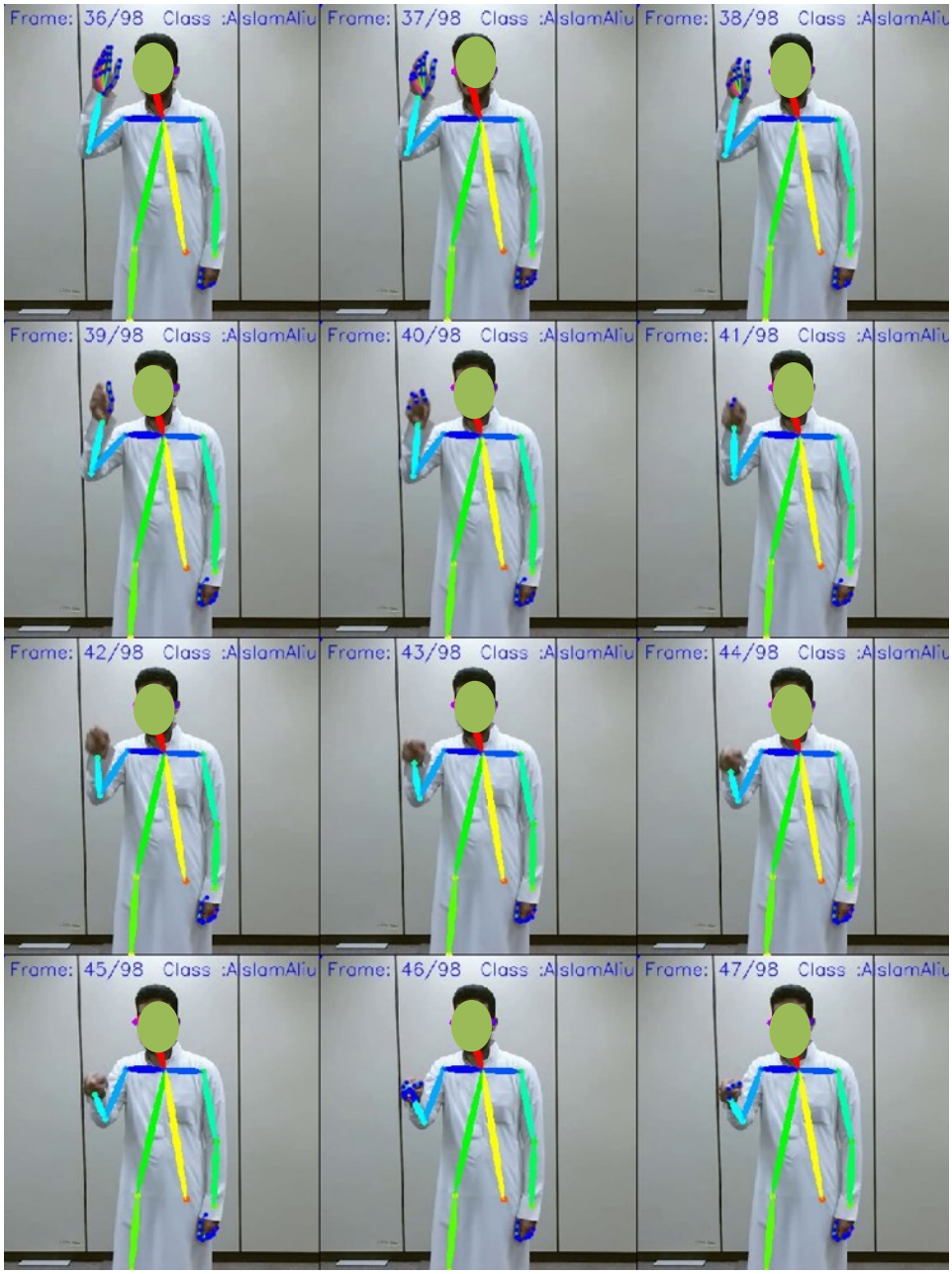


FIGURE 26, OPENPOSE HAND DETECTION OF A SECOND SAMPLE FROM THE KSU-ARSL

FIGURE 27, GOOGLE MEDIA PIPE HAND DETECTION OF A SECOND SAMPLE FROM THE KSU-ARSL

We have shown two different videos to Open-pose and Google media pipe libraries, of the KSU-ArSL and we can clearly notice from the frames that hands were not clearly detected, that is why we increased the fps to 60 fps in the recording of the KSU-SSL to avoid hand blurring and track both hands and fingers in all successive frames. In addition, we used a fixed green background to manipulate the videos for data augmentation when we will train our deep learning models.

5.9. Hands and fingers detection in the newly recorded KSU-SSL dataset.

As a preliminary investigation, we present in Figure 28 to Figure 30, some video frame samples of the KSU-SSL that were processed by Media pipe of Google and Open-pose respectively, where we notice the detection of the hands and fingers, in nearly all the frames.

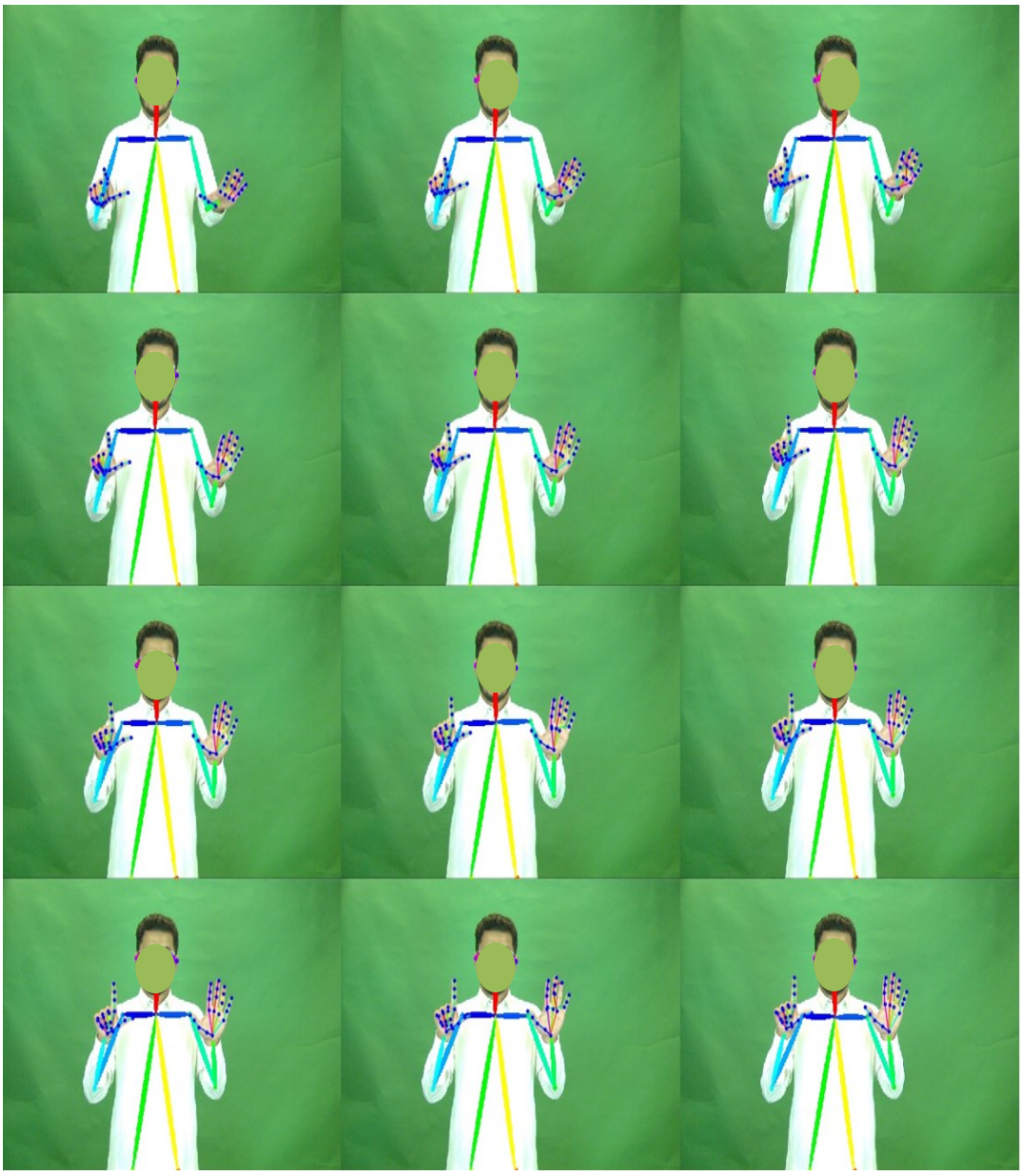


FIGURE 28, TWO HANDS + FINGER JOINTS DETECTION USING MEDIA PIPE LIBRARY IN KSU-SSL (SAMPLE 1)

FIGURE 29, TWO HANDS + FINGER JOINTS DETECTION USING MEDIA PIPE LIBRARY IN KSU-SSL (SAMPLE 2)

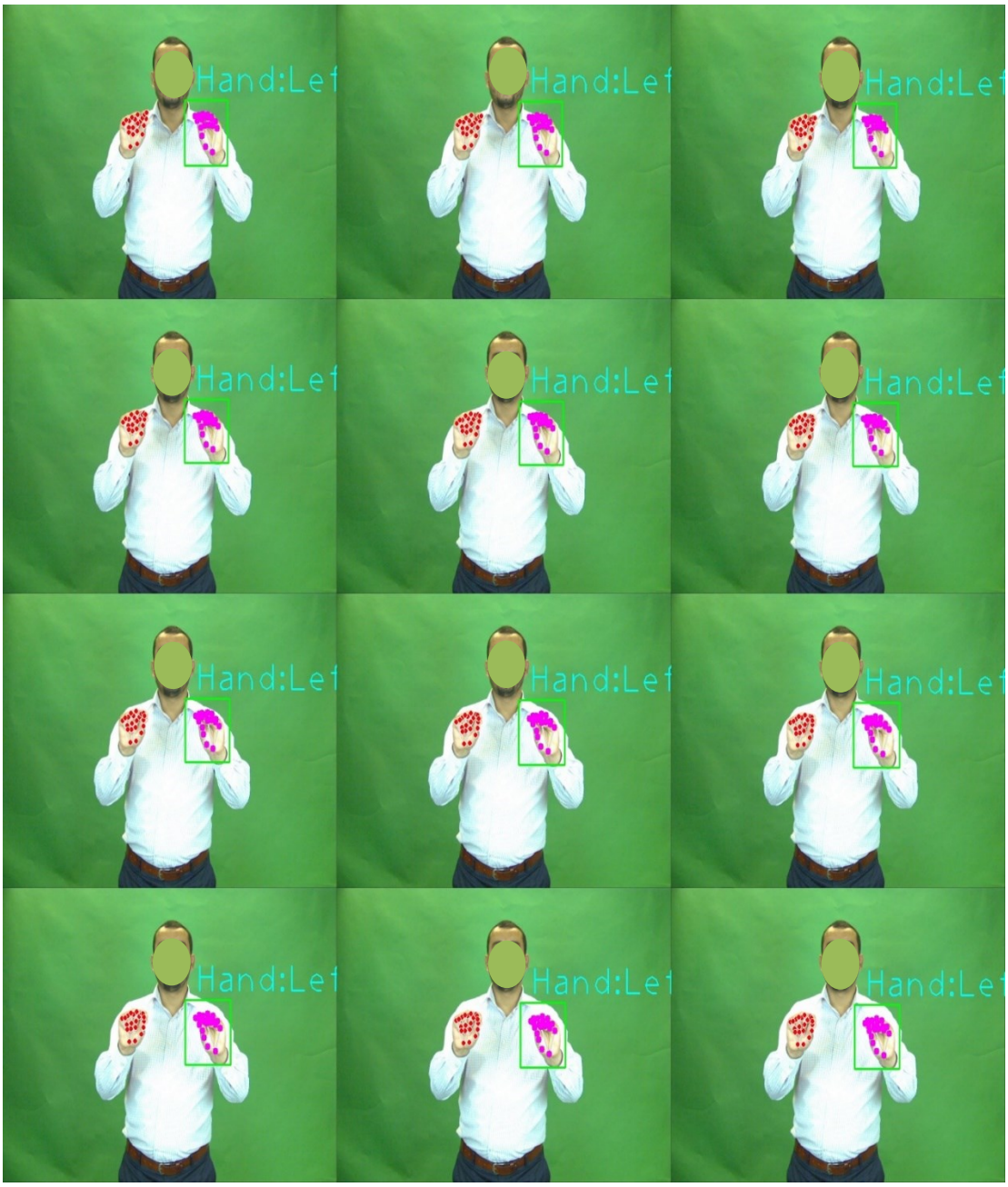


FIGURE 30, TWO HANDS + FINGER JOINTS DETECTION USING MEDIA PIPE LIBRARY IN KSU-SSL (SAMPLE 3)

6. Design and development of a sign recognition module

The second objective of the proposal is to develop a sign recognition language module that will be integrated into the proposed two-way sign translator. This module will process the video of sign language, recognize the signs, and produce the corresponding text. This core objective is still a challenging point of research. The signers, recording environment, illumination, and ways of expressing can be diverse. The signs themselves can be static or dynamic. All these variabilities plus some hidden variables contribute to the complexity of the module. We need to design a module that can reduce the effects of these variables and achieve a high recognition accuracy. In particular, we are interested in developing an online SSL translation system.

Although KSU-SSL was not ready in the first year of the project, but this did not stop us from conducting research and investigation on sign recognition systems. We used our KSU-ArSL database to conduct the investigations. Our investigation gave excellent results using deep neural networks and other methods. We published our findings in a Q1 journal [106] (The IEEE access was a Q1 journals at the time of sending it and when the paper was published, while now it became a Q2 journal). When KSU-SSL was ready we used it to investigate an efficient architecture for sign language recognition based on a graph convolutional neural network (GCN). We published our findings in a Q2 journal [107].

We are also investigating using transformers for sign recognition and it is giving promising results.

In the following subsections we will present the details our work and investigation.

6.1. Spatial multi-branch 3D CNN fused with MLP and autoencoder

In this section we will present the details of our work and research on [98]. In spite of the fact that the sign language formation involved multiple modules such as hands' gestures, lips and head movements, but hands' gestures encode most of the sign language message information. As with all time-varying signals, hand gestures cannot be directly compared in Euclidean space because of their temporal dependency. This dependency indicates important discriminative features. Temporal misalignment, in addition to massive irrelevant regions in every frame, makes it very hard to extract representative hand-engineered features for hand gestures. For conventional classifiers to perform well, the extracted features should implicate vigorous descriptors. These

descriptors code enough information for the inter-frames temporal dependency, as well as the hand position, shape, and orientation in each frame. The computed features should be able to minimize the effect of different circumstances like background clutter and occlusions. Therefore, we employed deep learning in this work as a promising solution.

In recent years, many researchers have efficiently exploited convolutional neural networks (CNNs) deep architectures for feature engineering. CNNs have shown excellent performance in fields such as object and speech recognition, image classification, and human activity recognition [108][109][110].

6.1.1. Development of the sign recognition system

In this work, we utilized a 3DCNN architecture for spatiotemporal feature learning with a focus on enhancing the learned features from the hand region as the most important articulate organ. FIGURE 31 illustrates the proposed system. It consists of three main phases: input preprocessing, feature learning and feature fusion, and classification. In the next subsections, we detail these different phases.

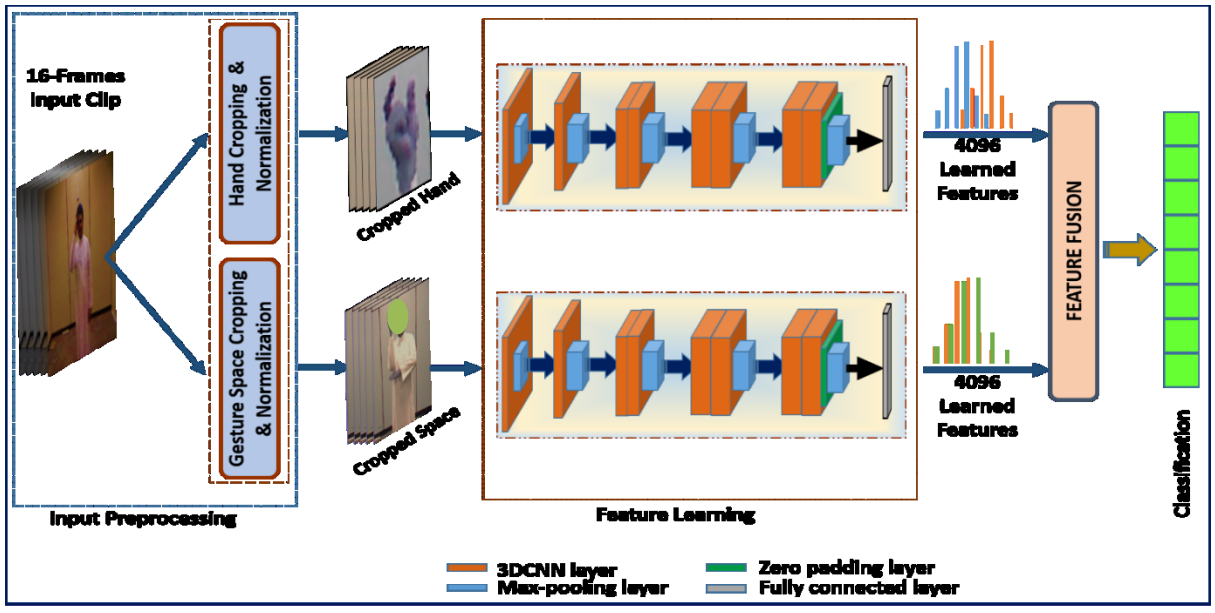


FIGURE 31. PROPOSED SYSTEM FOR HAND GESTURE RECOGNITION USING LOCAL AND GLOBAL CONFIGURATION FEATURES.

6.1.1.1. Input Preprocessing

The input videos are converted into sequences of RGB frames of different lengths. Then, a linear sampling is utilized for temporal dimension normalization, where 16 frames are linearly selected from each video sequence. The selected frames' indices are calculated as in Eq. 1.

$$index_i = \text{round}\left(\frac{\text{len}(input)}{16} \times i\right), i \in \{1,16\} \quad (1)$$

where len(input) is the length of the input sequence.

This temporal normalization step for the input can be achieved by different techniques such as the bag-of-visual words. These techniques are very efficient when the sequence order is of low importance for discrimination as in the video event and human action recognition. For hand gesture recognition, the sequence order should be preserved as it encodes highly discriminative features and linear sampling is the preferred technique for that. Two cropping and normalization methods are also performed simultaneously on the selected frames, signer body normalization, and hand cropping and normalization.

a- Signer body normalization

The first method locates the signer face using Viola-Jones algorithm [111]. Then, the gesture space is estimated and cropped in each frame based on the detected facial length and body parts ratios information. Then each frame is resized to a fixed size of 112×112 pixels while preserving the aspect ratio. The normalized gesture space is illustrated in FIGURE 32. This method outputs a sequence $X_B \in \mathcal{R}^{112\times112\times3\times16}$ of 16 frames. Each frame includes the entire gesture space. In addition, to avoid the effects of the variations of the signers' height and distance from the camera, this spatial normalization and cropping reduce the effect of non-relevant features in each frame.

The second method is devoted to crop and normalize the hand region to give more focus to the fingers' configuration.

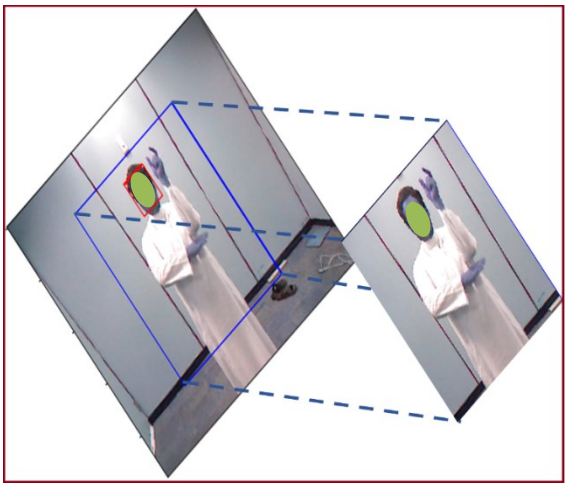


FIGURE 32. SPATIAL NORMALIZATION OF THE INPUT FRAMES.

b- Hand cropping and normalization

This method utilizes an open-source real-time human pose estimation called Open-pose. It is a deep learning-based framework to detect the 2D key points of each human individual in an image. This framework is aimed to improve the machine understanding of human activity in image or video sequence. It takes as an input an RGB image and returns as an output a list of (x, y) coordinates for all human body key points. *FIGURE 33* illustrates the upper body Open-pose key points generated by Open-pose.

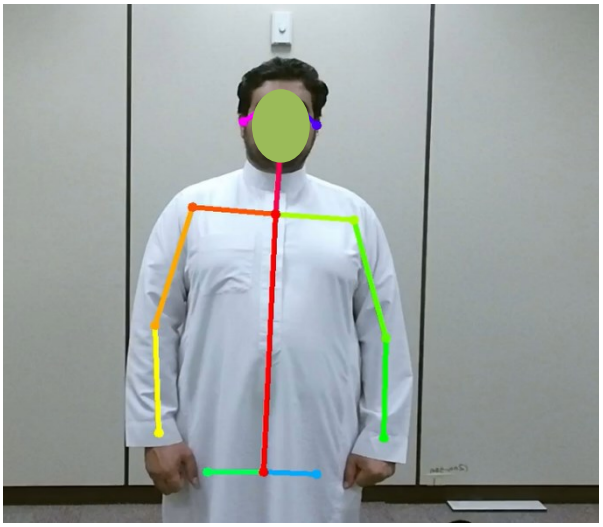


FIGURE 33. THE UPPER BODY OPENPOSE KEY POINTS.

From the whole list of the returned key points, only the wrist and elbow joints are utilized to perform the hand region cropping. For instance, the vector from the elbow joint (x_e, y_e) to the wrist joint (x_w, y_w) indicates the arm axis. Based on the arm axis direction, we propose an efficient method to estimate a small square region around the hand to be cropped. The length of this square region is equal to the absolute value of the distance between the wrist and the elbow joints as in Eq. 2.

$$length = \sqrt{(x_w - x_e)^2 + (y_w - y_e)^2} \tag{2}$$

The proposed method estimates the hand orientation to one of the nine basic directions illustrated in FIGURE 34. These directions are:

1- The hand axis is perpendicular to the frame plane pointing at the camera.
2- The hand axis is vertical pointing up.
3- The hand axis is diagonal pointing to the top right.

4- The hand axis is horizontal pointing to the right.
5- The hand axis is diagonal pointing to the bottom right.
6- The hand axis is vertical pointing down.
7- The hand axis is diagonal pointing to the bottom left.
8- The hand axis is horizontal pointing to the left.
9- The hand axis is diagonal pointing to the top left.

The calculation steps of estimating the top left (x_B, y_B) point and bottom right (x_E, y_E) point of the square-region of cropping are summarized in algorithm 1 in appendix A. The cropped hand region is then resized to 112×112 pixels. The horizontal and vertical distances between the wrist and the elbow joints (X difference and Y difference) are illustrated in FIGURE 35. (x_{mid}, y_{mid}) is the middle point between the elbow and the wrist. The threshold value α was empirically set to 40-pixels.

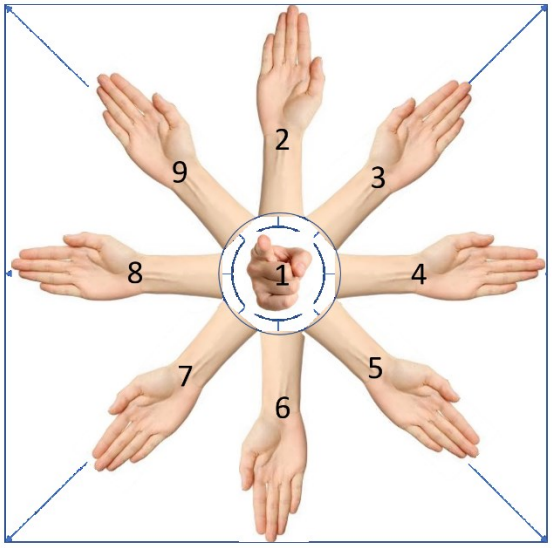


FIGURE 34. ESTIMATED HAND DIRECTIONS

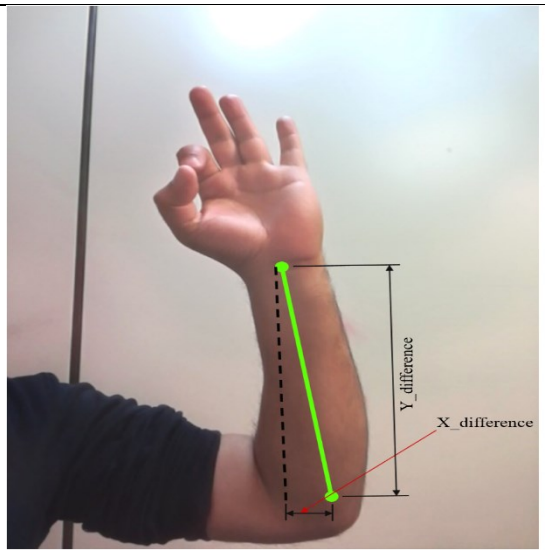


FIGURE 35. HAND DIRECTION ESTIMATION.

Please refer to Appendix A, for the details of the algorithm.

The preprocessing phase outputs two tensors per sample each of size 112 × 112 × 3 × 16. These two volumes are delivered to the feature learning phase where one of them represents the entire gesture space and the other is dedicated to the hand region.

6.1.2. Feature learning

We start with the pre-trained C3D architecture which composes eight convolutional layers, five pooling layers, and two FC layers [78]. This model is already trained on the large-scale Sport-1M human action recognition dataset [109]. We know that in domain adaptation learning the transferred knowledge has less impact as we move toward the layers at the top of the model, especially when the source and target domains are far away from each other. For that reason, we replace the last block which contains two FC layers, each has 4096 neurons, with a new FC layer of 4096 neurons to reduce the training cost of those two FC layers with an expansive number of parameters.

Then, we optimize the level of knowledge transfer from the source domain to the target domain. This optimization step is detailed in the experimental results and discussion section.

Two instances of the optimized C3D architecture are utilized after that to learn the spatio-temporal features in different levels of the video sequence (the hand region and the entire gesture space region).

The output of the 3DCNN kernel at any level can be calculated as in Eq. 3.

$$V_{LK}^{xyz} = ReLU\left(b_{LK} + \sum_{m}\sum_{p=0}^{P_L-1}\sum_{q=0}^{Q_L-1}\sum_{r=0}^{R_L-1} W_{LKm}^{pqr}\, v_{(L-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (3)$$

where w_{LKm}^{pqr} is the (p,q,r)th value of the filter connected to the mth feature map in the previous layer, R_L is the temporal dimension, and P_L, and Q_L, are the two spatial dimensions.

The first C3D instance is devoted to learning the fine spatio-temporal features of the hand configuration. The hand has been made dominant in each input frame of this instance. The second C3D instance, on the other hand, learns the coarse spatio-temporal features of the whole-body configuration. This phase produces as an output two feature vectors each has a dimension of 4096.

6.1.3. Feature fusion and classification

Two different techniques, namely, MLP and autoencoder are then investigated to fuse the two feature vectors before feeding them to the classifier. In contrast to the system proposed in [64], we avoid the use of LSTM with this system, as the two streams are not temporal segments of the gesture. We perform end-to-end training for the fusion architecture with the classifier. The classification layer is activated by a SoftMax function.

6.1.4. Experimental results and discussion

To evaluate the proposed system, we conducted extensive experiments on 40 dynamic gestures from the KSU-ArSL dataset, where the total number of samples is 8000, performed by 40 participants. Each gesture was repeated 200 times in total. The list of gestures was illustrated in **Error! Reference source not found.** in section 4.1 where we presented the details of the KSU-ArSL dataset. Our experiments were conducted in two scenarios as follows:

- signer dependent mode: In this scenario, the samples were randomly shuffled and split into two subsets for training and evaluation. In other words, we divided the samples of each signer into training and evaluation with a random ratio. The total number of samples in the training set represents 80% of the dataset and the remaining 20% are in the validation set.
- signer independent mode: In this scenario, the signers were divided into two mutually exclusive sets. The first one contains 32 signers while the second one contains 8 signers. All the samples performed by the first set of signers were used for training, while all the samples performed by the signers of the second set were used for validation. The ratios of training and validation samples are still 80% and 20% respectively

6.1.5. C3D knowledge transfer optimization

Typically, when utilizing transfer learning some of the architecture layers are iteratively fine-tuned on the target domain data to adapt their parameters for the target domain. Other layers are frozen to keep the original values of their parameters. In this experiment, we investigated how the performance of the C3D architecture was affected by changing the number of trainable layers to find the optimal case. This optimization step was performed in the signer independent mode. All the samples of the KSU-ArSL dataset, which were performed by the first 32 signers (80% of the samples) were used for training the architecture. The remaining 1600 samples which were performed by the other eight signers (20% of the samples) were used for evaluation.

We linearly sampled 16 frames from each sequence while each frame contained the entire gesture space. Then, an end-to-end training was conducted for the C3D architecture after replacing the last two FC layers and the classification layer. Mini-batch gradient descent with a learning rate of 10^{-4}, a weight decay of 10^{-6}, and a momentum of 0.9 were used to fit the entire model over 100 iterations, with a batch size of 16 samples. We repeated this experiment by changing the number of trainable and frozen layers each time to find the optimal level for knowledge transfer. We started by training only the last 3DCNN layer with the FC layer and the classification layer while the

remaining layers were frozen. Then, in each repetition, we incremented the number of trainable layers by activating the next nearest layer to the previously activated ones. FIGURE 36 illustrates the results of this experiment in terms of evaluation loss and recognition accuracy. It shows that the performance of the model is enhanced as we increase the number of trainable layers as long as the first layer is frozen. In other words, the best performance was achieved by fine-tuning all the layers except the first one. This result supports the intuition that the first layer learns the common preliminary motifs in both the source and target domains. As a result, the parameters of this layer were optimized well on the source data and there is no need to distort them by a small and maybe noisy data of the target domain.

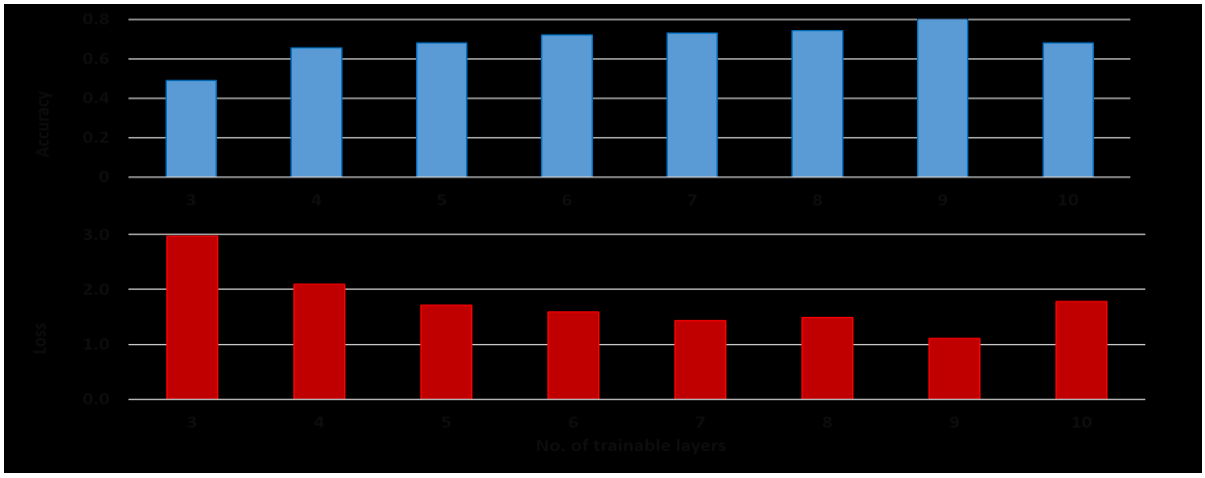


FIGURE 36. THE EFFECT OF THE NUMBER OF TRAINABLE LAYERS ON THE EFFICIENCY OF KNOWLEDGE TRANSFER.

This optimal case of knowledge transfer was used in the rest of our experiments for feature representation by taking the output of the FC layer as a feature vector for the architecture of fusion and classification.

6.1.6. Results of MLP Fusion

a- Signer independent mode

In this part, we investigated the MLP network for feature fusion. We studied the effect of the number of layers of the MLP (the depth), and the number of neurons per layer on the performance. The mini-batch gradient descent optimizer was used with an initial learning rate of 10^{-4}, a decay of $10^{-6,}$ and a momentum of 0.9. We conducted an extensive grid search to optimize the architecture and the initial learning rate as they are the most important hyperparameters for the MLP fusion network. The search space was defined as follows:

- The architecture depth in terms of number of layers $d \in \{1,2,3\}$.
- The number of neurons in any layer $N \in \{256, 512, 1024, 2048, 4096, 8192\}$ with a constraint that the number of neurons in any layer should be less than or equals to the number of neurons in the previous layer.
- The initial learning rate $lr = x \times 10^n : x \in \{1,5\}\ and\ n \in \{-3, -4, -5, -6, -7\}$

The result of this grid search step is illustrated as a heat map in FIGURE 37. The average accuracy of all architectures with different learning rates is illustrated in FIGURE 38. From the heat map and the average accuracy figures, we can notice that the learning rates between 10^{-4} and 10^{-5} achieved competing accuracies for all architectures.

	256	256-256	256-256-256	512	512-256	512-512	512-512-256	512-512-512	1024	1024-256	1024-512	1024-1024	1024-1024-256	1024-1024-512	1024-1024-1024	2048	2048-256	2048-512	2048-1024	2048-2048	2048-2048-256	2048-2048-512	2048-2048-1024	2048-2048-2048	4096	4096-256	4096-512	4096-1024	4096-2048	4096-4096	4096-4096-256	4096-4096-512	4096-4096-1024	4096-4096-2048	4096-4096-4096	8192	8192-256	8192-512	8192-1024	8192-2048	8192-4096	8192-8192	8192-8192-256	8192-8192-512	8192-8192-1024	8192-8192-2048	8192-8192-4096	8192-8192-8192
5.E-03	3	3	3	3	3	3	3	5	3	3	3	3	3	7	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	17	3	3	3	3	3	2	2	3	3	3	3	9	26
1.E-03	35	13	8	82	43	66	58	43	82	83	81	83	81	79	82	82	83	82	82	83	83	82	83	84	82	83	83	83	81	83	82	82	82	83	83	82	83	83	82	82	84	83	84	83	83	83	81	81
5.E-04	83	62	40	83	82	82	75	71	83	84	84	83	82	82	84	84	84	83	83	83	83	82	84	83	84	83	83	84	84	84	82	83	82	83	83	83	83	83	83	83	83	83	83	84	83	83	83
1.E-04	85	86	78	85	86	85	84	85	85	85	85	85	84	84	85	85	85	85	85	84	85	85	85	84	84	86	85	85	85	84	85	84	86	84	85	85	86	85	85	85	85	84	85	85	85	85	85	85
5.E-05	85	85	79	85	85	86	85	84	86	87	86	86	85	86	86	86	88	87	86	86	85	86	86	86	86	86	87	86	86	85	85	86	86	86	86	85	87	86	87	86	86	86	86	86	86	86	86	86
1.E-05	85	83	63	85	83	85	82	82	85	84	85	84	83	83	84	85	84	85	85	85	85	84	84	85	85	85	85	85	85	86	84	85	85	84	86	86	85	85	85	86	86	86	84	85	84	85	85	86
5.E-06	84	79	42	83	80	82	73	76	83	82	83	83	80	81	83	83	83	83	83	83	82	82	82	82	83	83	84	85	85	82	82	83	83	83	83	83	83	84	84	84	84	84	83	84	83	85	84	84
1.E-06	75	47	19	80	68	67	40	42	81	72	72	74	63	62	62	81	77	79	81	80	68	71	75	75	82	79	79	81	82	82	67	76	76	79	79	84	78	80	83	83	82	82	71	80	80	82	83	83
5.E-07	65	33	16	73	46	50	28	24	77	60	65	68	45	49	49	79	66	72	76	76	50	57	61	64	82	63	74	76	78	80	44	67	71	74	73	82	57	73	78	81	80	80	52	69	74	76	80	79
1.E-07	25	11	4	42	17	17	8	5	48	26	29	27	11	14	17	59	24	34	43	36	16	21	21	22	67	17	37	48	51	51	15	19	24	32	31	70	16	44	55	49	55	58	19	24	30	37	41	43

FIGURE 37. HEAT MAP SHOWS THE SYSTEM ACCURACY IN THE SEARCHED SPACE OF THE MLP HYPERPARAMETERS.

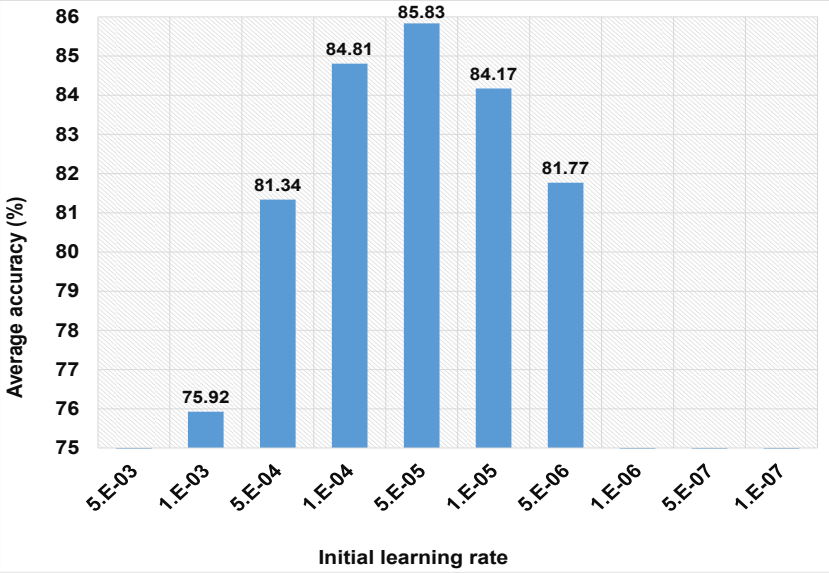


FIGURE 38, THE AVERAGE ACCURACY OF ALL ARCHITECTURES WITH DIFFERENT INITIAL LEARNING RATES

The highest recognition accuracy of 87.69 % was achieved by the two layers' architecture, the first one of 2048 neurons and the second one of 256 neurons with an initial learning rate of 5×10^{-5}. We can notice that there is no clear trend for the performance concerning the architecture. The performance of the trained system on the evaluation dataset is detailed in the confusion matrix in Figure 39.

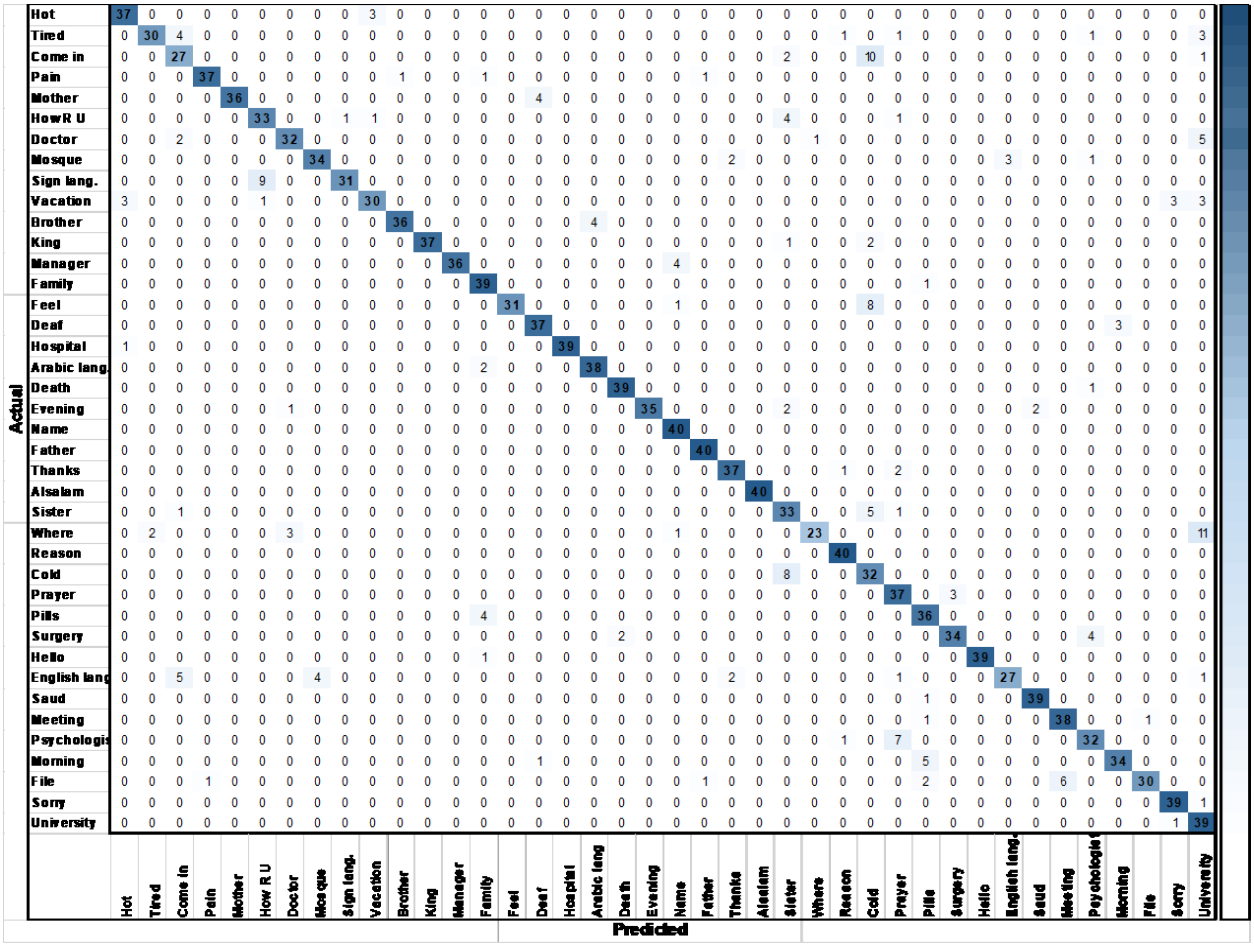


FIGURE 39. CONFUSION MATRIX OF MLP TWO STREAMS FUSION IN SIGNER-INDEPENDENT MODE

b- Signer dependent mode

The optimal hyperparameters obtained in the signer independent scenario were used to evaluate the performance of the system in the signer dependent scenario. In this scenario, some of the signer's samples were used for training the model while the rest of the samples performed by the same signer were used for evaluation. The ratios of the two sets of samples were random and variated from one signer to another. The evaluation results are illustrated in the confusion matrix in Figure 40. An accuracy of 98.62% was achieved in this scenario.

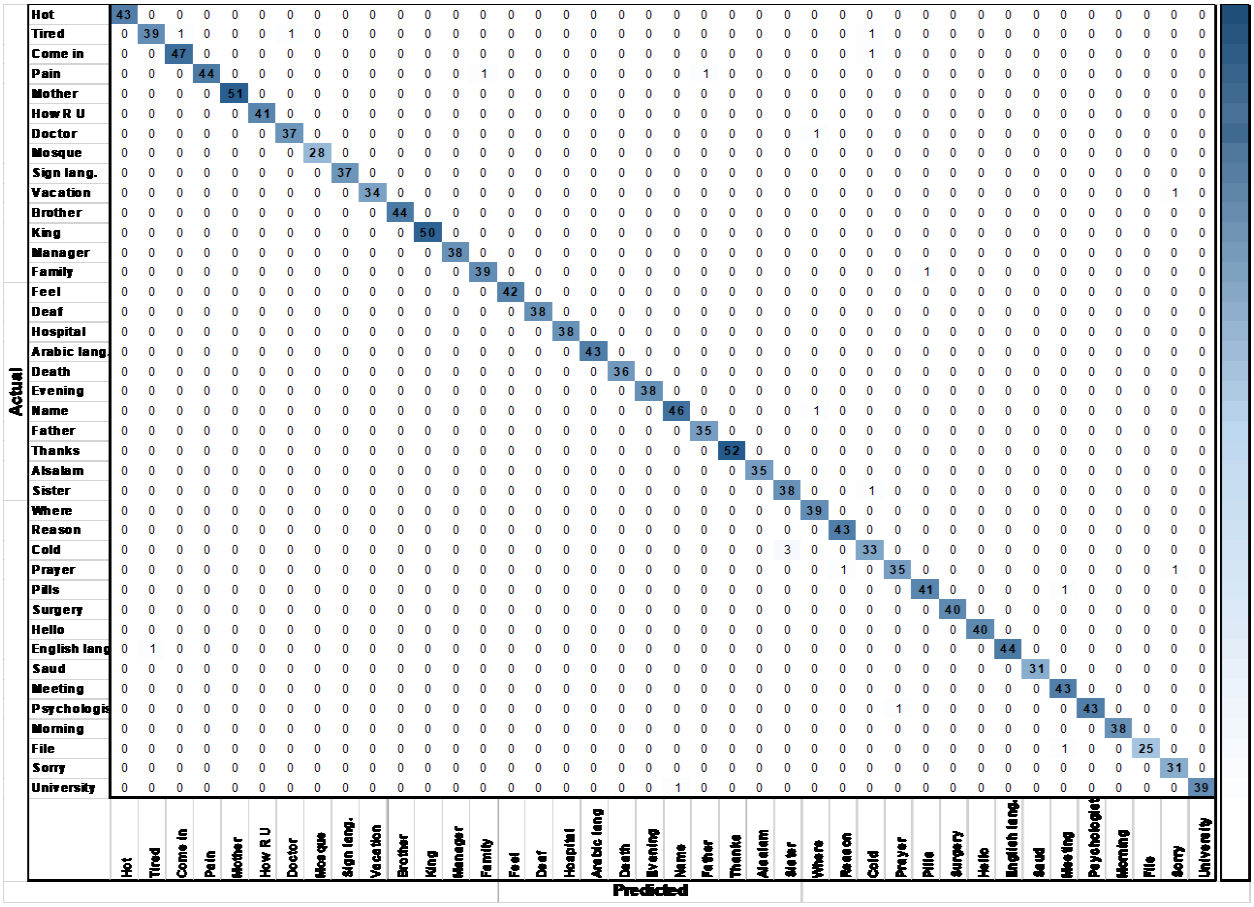


FIGURE 40. CONFUSION MATRIX OF MLP TWO STREAMS FUSION IN SIGNER-DEPENDENT MODE

6.1.7. Results of the autoencoder fusion

a- Signer independent mode

In this part, we investigated an autoencoder network for feature fusion. We also investigated the effect of the autoencoder depth (number of hidden layers), and the number of neurons in each layer on the performance of the system.

The mini-batch gradient descent optimizer was used in this part with the same parameter setup that was used in the MLP fusion. We conducted an intensive grid search to optimize the architecture of the autoencoder and the initial learning rate.

The search space was defined as follows:

- The encoder architecture depth in terms of number of layers $d \in \{1,2,3\}$.

- The number of neurons in any layer $N \in \{256, 512, 1024, 2048, 4096, 8192\}$, with a constraint that the number of neurons in the layers is reduced as we move from the input layer toward the latent layer.

- The initial learning rate $lr = x \times 10^n : x \in \{1,5\}\ and\ n \in \{-3, -4, -5, -6, -7\}$

The result of this grid search step is illustrated as a heat map in FIGURE 41. The average accuracy of all architectures with different learning rates is illustrated in FIGURE 42. From the heat map and the average accuracy figures, we can notice that the system of one pair of hidden layers performed better than that of two pairs of hidden layers.

We can also notice that the maximum accuracy and the best average accuracy were achieved by an initial learning rate of 10^{-5}.

	512-256-512	1024-256-1024	2048-256-2048	4096-256-4096	8192-256-8192	1024-512-1024	2048-512-2048	4096-512-4096	8192-512-8192	2048-1024-2048	4096-1024-4096	8192-1024-8192	4096-2048-4096	8192-2048-8192	8192-4096-8192	1024-512-256-512-1024	2048-512-256-512-2048	4096-512-256-512-4096	8192-512-256-512-8192	2048-1024-256-1024-2048	4096-1024-256-1024-4096	8192-1024-256-1024-8192	4096-2048-256-2048-4096	8192-2048-256-2048-8192	8192-4096-256-4096-8192	2048-1024-512-1024-2048	4096-1024-512-1024-4096	8192-1024-512-1024-8192	4096-2048-512-2048-4096	8192-2048-512-2048-8192	8192-4096-512-4096-8192	4096-2048-1024-2048-4096	8192-2048-1024-2048-8192	8192-4096-1024-4096-8192	8192-4096-2048-4096-8192
5.E-03	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	6	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
1.E-03	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	10	3.5	3.5	5	15	4	6	5	4	3.5	3.5	3.5	3.5	3.5	3.5	30	3.5	3.5	24	3.5
5.E-04	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	15	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	5	3.5	3.5	13	3.5
1.E-04	3.5	3.5	3.5	3.4	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.4	3.4	3.5	3.4	3.5	3.5	3.5	19	3.5	3.5	3.5	3.5	3.5	3.6	3.4	3.5	3.5	3.5	3.5	3.4	3.4	3.5	3.4	3.5
5.E-05	3.5	3.5	48	53	59	69	79	79	49	75	83	83	81	82	81	3.5	3.5	3.5	3.6	3.5	3.5	3.4	3.5	3.5	3.4	3.5	3.5	3.5	3.5	3.5	3.2	3.4	3.5	3.7	3.4
1.E-05	77	80	82	81	82	82	83	82	82	84	83	83	83	85	84	79	80	79	79	80	79	81	80	82	82	81	81	83	83	82	82	82	83	82	84
5.E-06	76	77	78	80	82	80	81	82	84	82	83	83	84	83	84	67	76	79	77	72	78	78	78	75	81	80	80	81	81	81	81	81	82	82	84
1.E-06	50	56	69	72	76	67	71	74	76	78	78	80	80	82	82	35	27	45	50	43	48	56	56	57	67	56	63	66	65	69	71	66	76	73	76
5.E-07	32	44	63	61	68	59	63	64	73	69	74	76	76	80	79	18	33	27	39	35	38	52	41	47	49	38	39	58	47	59	61	55	65	64	69
1.E-07	14	16	22	17	24	18	22	27	36	27	34	44	38	47	52	6.5	9.9	15	20	9.9	16	16	11	20	17	14	17	18	17	25	20	21	24	32	40

FIGURE 41. A HEAT MAP SHOWS THE SYSTEM ACCURACY IN THE SEARCHED SPACE OF THE AUTO-ENCODER HYPER-PARAMETERS

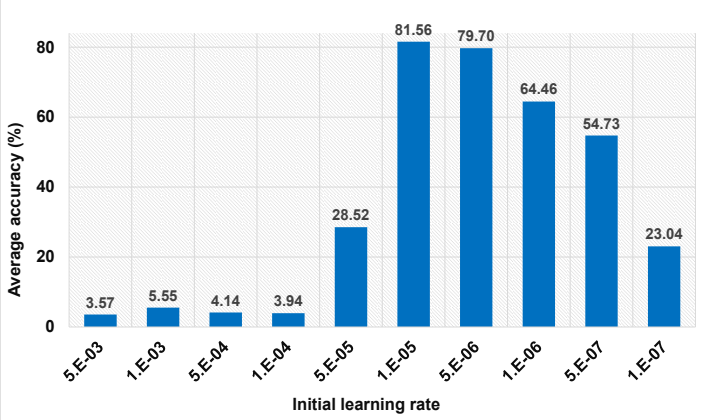


FIGURE 42. THE AVERAGE ACCURACY OF ALL ARCHITECTURES WITH DIFFERENT INITIAL LEARNING RATES

The highest recognition accuracy of 84.89 % was achieved by an architecture of 2048 neurons in the latent layer and a single pair of hidden layers of 8192 neurons each. The recognition rate of the trained system on the evaluation dataset is detailed in the confusion matrix in FIGURE 43 .

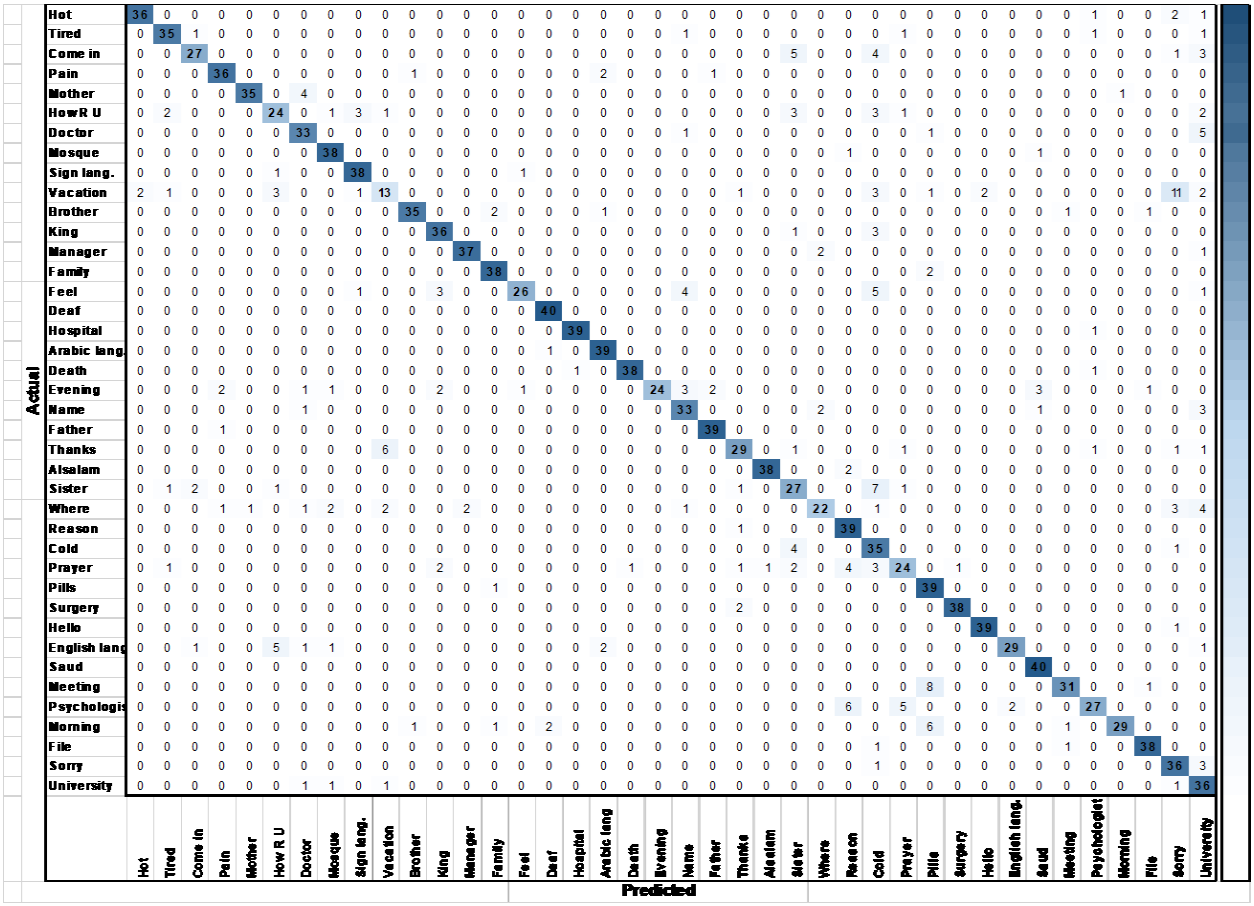


FIGURE 43. CONFUSION MATRIX OF AUTOENCODER TWO STREAMS FUSION IN SIGNER-INDEPENDENT MODE

b- Signer dependent mode

The optimal hyper-parameters obtained in the signer independent scenario were used to evaluate the system performance in the signer dependent scenario. The evaluation results are illustrated in the confusion matrix in FIGURE 44. A recognition accuracy of 98.75% was achieved in this scenario.

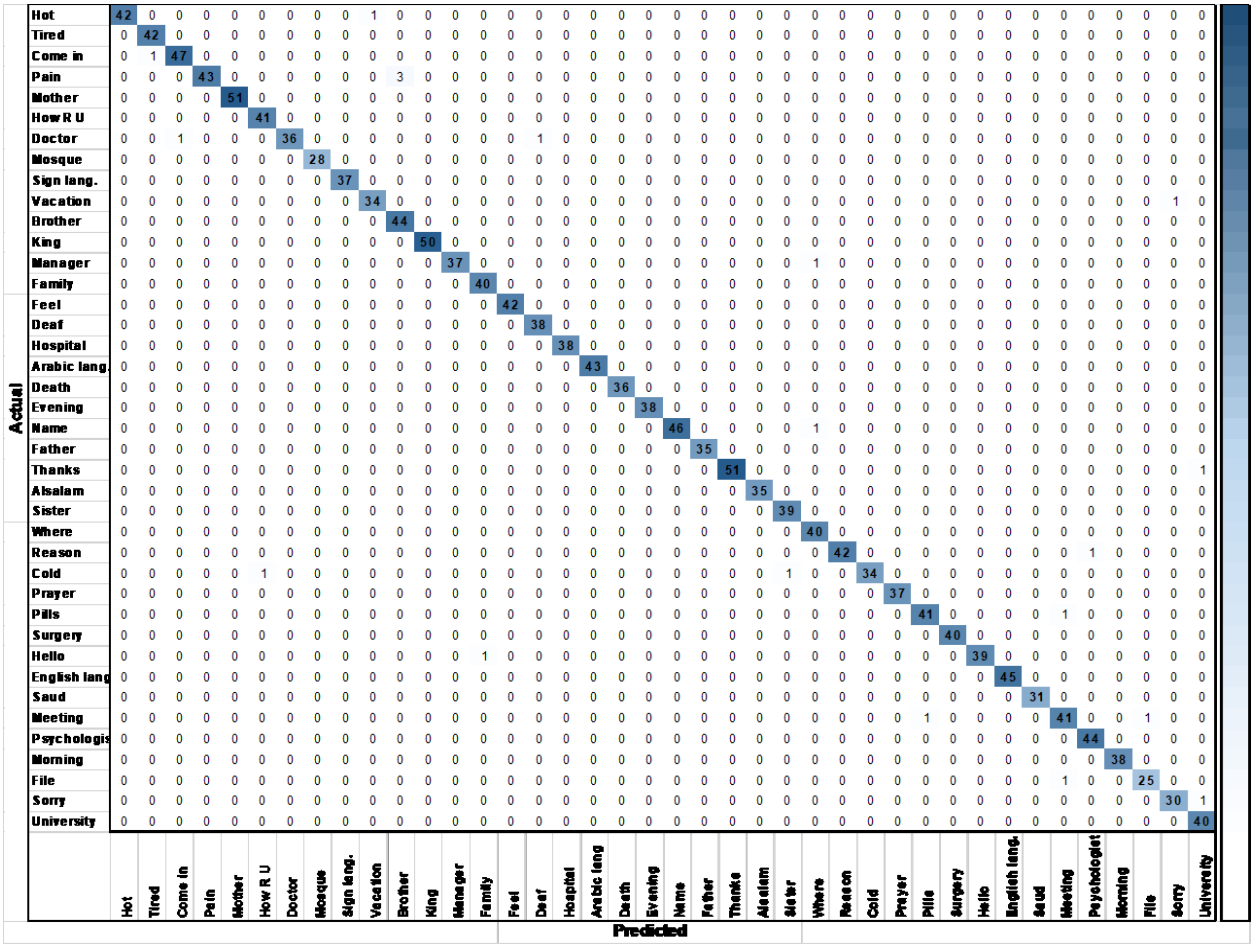


FIGURE 44. CONFUSION MATRIX OF AUTOENCODER TWO STREAMS FUSION IN SIGNER-INDEPENDENT MODE

6.1.8. Discussion

TABLE 10 summarizes the MLP and the autoencoder accuracies achieved with different batch sizes. We notice that both architectures obtained comparable performance in the signer-dependent mode, while the performance of the MLP in the signer independent mode is much better than that of the autoencoder. From the optimization heatmaps of both MLP and autoencoder systems, we can notice that:

- In MLP there is no clear trend change in the accuracy with the change in the depth of the architectures.
- The performance of the autoencoder was slightly enhanced by increasing the number of neurons while fixing the depth of the architecture.
- The performance of the autoencoder, on the other hand, is degraded by increasing the depth of the architecture.

- The smallest batch size achieved the highest accuracy for both architectures. This might be attributed to the fact that minimizing the batch size leads to updating the model weights more frequently. Even though, such updates using few noisy samples involve a regularizing effect, which reduces generalization error.

- Moreover, in the confusion matrices, it is noticed that the system performance in the signer-independent mode is weaker than that in the signer-dependent mode. As the gestures in the KSU-ArSL dataset are performed by a large number of participants, the dataset samples exhibit significant variations. When the training and evaluation samples are performed by two mutually exclusive sets of signers (the signer independent scenario), the intra-class variation is very high and as a result, the recognition accuracy is low.

The proposed system gave more consideration to the hand region by dedicating a separate stream to learn the hand configuration features. This consideration led to excellent enhancement in the system performance. Compared to the results achieved by the base C3D architecture in the first experiment and those achieved by the temporally enhanced system in [32], this system achieved the best recognition rate with both MLP and autoencoders in all scenarios.

TABLE 10. ACCURACY (%) ACHIEVED BY MLP, AND AUTOENCODER FUSION IN DIFFERENT MODES

	Batch size	MLP	Autoencoder
signer dependent	16	98.62	98.75
	32	97.31	97.70
	64	97.00	97.12
signer independent	16	87.69	84.89
	32	84.17	81.87
	64	83.56	81.44

6.2. Spatial Attention Based 3D Graph Convolutional Neural Network

This section presents an efficient architecture for sign language recognition based on a convolutional graph neural network. The presented architecture consists of a few separable 3DGCN layers, which are enhanced by a spatial attention mechanism. The limited number of layers in the proposed architecture enables it to avoid the common over-smoothing problem in deep graph neural networks. Furthermore, the attention mechanism enhances the spatial context representation of the gestures. The proposed architecture is evaluated on different datasets and shows outstanding results.

6.2.1. Datasets

The proposed architecture is evaluated on the project KSU-SSL dataset and four other benchmark datasets for sign language recognition. These datasets vary in terms of scene complexity, number of classes, number of samples per class, and the average length of videos.

3.1. King Saud University Saudi Sign Language (KSU-SSL) dataset

KSU-SSD was collected by the Center of smart robotic research at King Saud University. It consists of 293 classes from the daily life sign language gestures in the Kingdom of Saudi Arabia (KSA). Most of the gestures were selected from the medical field for their importance and high demand by deaf people. The recording studio setup utilizes three frontal imaging devices: a high-speed RGB camera, an IR camera, and a mobile camera. The signs were performed by 32 participants, each performing all the signs 5 times (one of the 5 is with painted hand and fingers). In this study, only the RGB camera videos without colored hands are involved. Sample frames from the KSU-SSL dataset are illustrated in *Figure 45*. More details of KSU-SSD were presented in section 5.

The other datasets used for evaluation are the AUTSL dataset (Sincan & Keles, 2020), the Argentinean sign language dataset (LSA64) (Ronchetti & et al., 2016), the American sign language lexicon video dataset (ASLLVD) (Neidle, Thangali, & Sclaroff, 2012), and Jester (Materzynska & et al., 2019). Some explanatory statistics of these datasets are summarized in

TABLE *11*. *FIGURE 46* also illustrates how the average number of frames per video varies in different datasets.

6.2.2. Methodology

To build an efficient graph-based sign language recognition system, on one hand, we proposed a lightweight 3DGCN with a low number of trainable parameters for representation learning. On the other hand, we utilized MediaPipe, which is an efficient human landmarks estimator to extract the required graph nodes for recognition. Furthermore, other techniques such as multi-head self-attention and frame nodes' partitioning were also utilized to boost the learning efficiency.

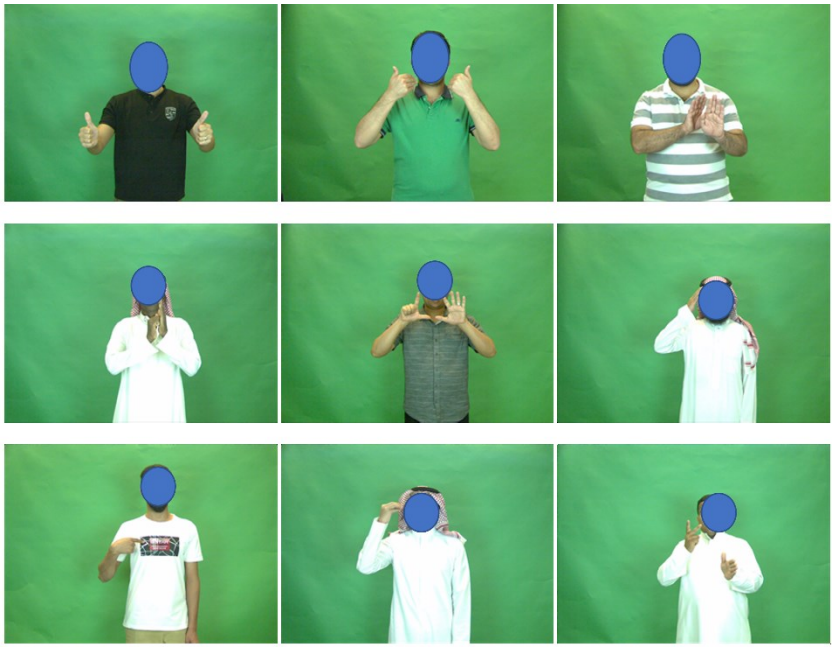


FIGURE 45, SAMPLE FRAMES FROM THE KSU-SSL DATASET

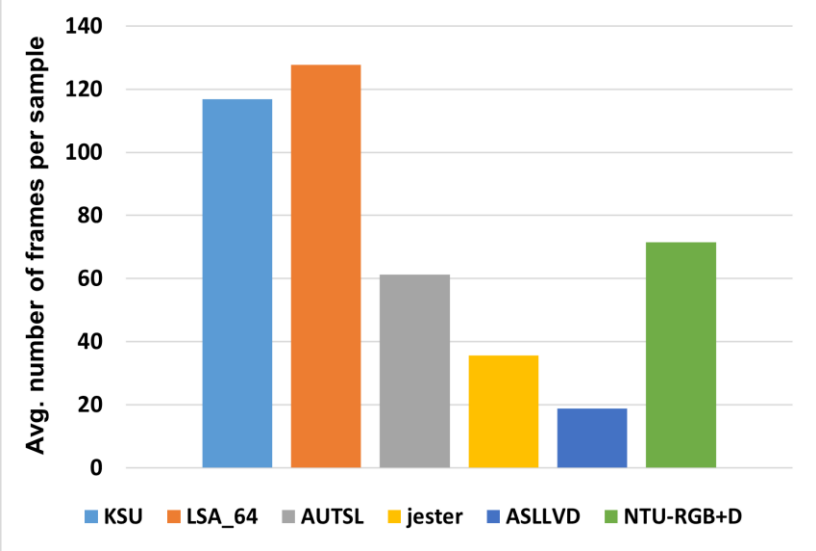


FIGURE 46, AVERAGE VIDEO LENGTH IN DIFFERENT DATASETS

TABLE 11, STATISTICS OF DIFFERENT DATASETS USED IN THIS STUDY.

Dataset	Num. of classes	Num. of training samples	Num. of validation samples
KSU-ArSL	293	28021	5860
AUTSL	226	28142	4418
ASLLVD-20	20	85	42
ASLLVD	2745	7798	1950
SLA-64	64	2560	640
Jester	27	118558	14786

MediaPipe-based Graph Construction

MediaPipe is a recent framework presented by Google. It offers cross-platform, machine learning solutions for streaming media. It enables the live perception of human pose, hand tracking, and face landmarks on mobile devices and in real time (Lugaresi & et. al, 2019). Each solution enables a wide range of modern life applications such as augmented reality, fitness, and sports analysis. MediaPipe can detect and track 33 pose landmarks, 21 landmarks per hand and 468 face landmarks in the three mentioned solutions, respectively. Furthermore, MediaPipe provides a holistic solution, which integrates models for the pose, the hands, and the face components. The holistic solution can accurately estimate and track 543 landmarks in total. Each estimated landmark is represented in x, y, and z coordinates. *FIGURE 47* depicts some of the estimated landmarks in a sample frame from the KSU-SSL dataset.

To build the sign graph, we only selected the most relevant 25 landmarks so that we can obtain an excellent recognition rate while maintaining a minimum computation complexity. From the 21 landmarks illustrated in *FIGURE 48*, for each hand, we selected ten landmarks (0, 4, 5, 8, 9, 12, 13, 16, 17, and 20). The other five selected landmarks represent the nose, the shoulders, and the elbows. The nose landmark was used as a reference to normalize the landmarks in each frame individually.

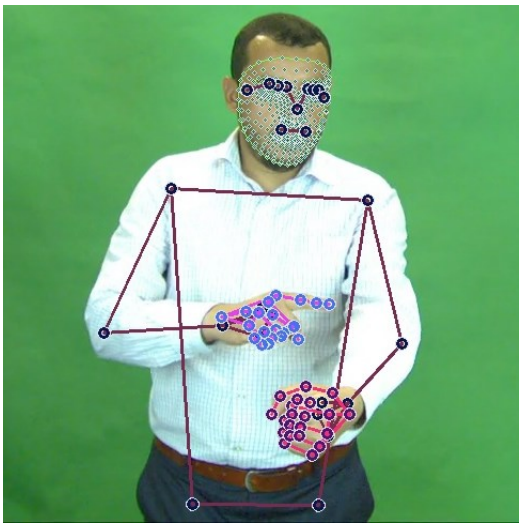


FIGURE 47, MEDIAPIPE LANDMARKS ESTIMATION SAMPLE

The output of this step is an undirected spatial–temporal graph $G = (V, E)$ with 25 nodes and T frames featuring both intra-body and inter-frame connections. The nodes set in this graph $V = \{v_{ti} | t = 1, \dots, T, i\ 1, \dots, 25\}$ includes the selected 25 landmarks in each frame through the sequence. The nodes in one frame are connected by edges according to the connectivity of the human body structure, and each node is connected with itself in consecutive frames. Consequently, the edge set E consists of two types of edges, the intra-frame edges at each frame and the inter-frame edges, which connect the nodes in consecutive frames.

Moreover, the number of selected frames from each video sample was controlled by a predefined window size. For simplicity, the value of the window size was set based on the average number of frames per video in the dataset.

Graph Representation Learning

The proposed architecture for representation learning consists of five consecutive layers of separable 3DGCN. As illustrated in *FIGURE 49*, the spatial and temporal convolution operations were separated by a spatial multi-head self-attention layer. The input of the layer was also passed to the output through a residual connection. For the spatial convolution, this work assumes that the spatial neighborhood of any node consists of all the nodes within a single step distance from that node.

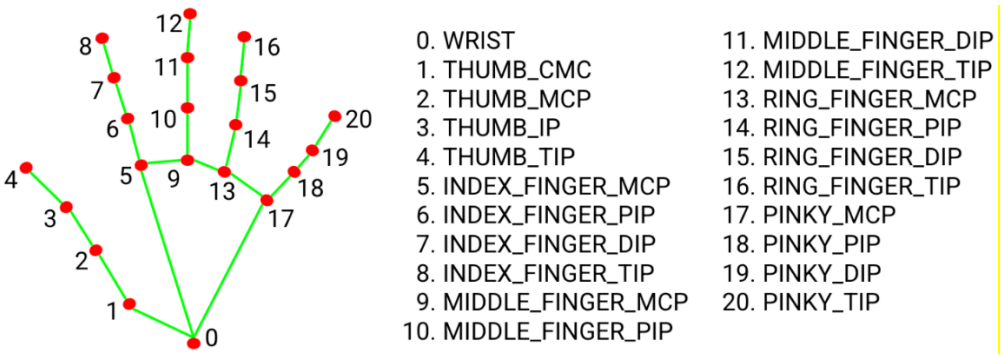


FIGURE 48, MEDIAPIPE HAND LANDMARKS [46].

Basic separable 3DGCN implementation

The implementation of the convolution operation on graph data is not straightforward, as it is implemented on image data, which is in the form of a regular grid. Similar to most of the work in the literature, we investigate an implementation such as the one presented in (Kipf & Welling, 2016). Within a single frame, the nodes' connections are represented by an adjacency matrix A. The self-connections of nodes are also represented by an identity matrix I; hence, the spatial convolution was implemented as in Eq. 4.

$$f_{out} = \Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}} f_{in} W \quad (4)$$

where f_{out} is the output embedding of nodes, $\Lambda^{ii} = \sum_j (A^{ij} + I^{ij})$, and the shared weight matrix for node-wise feature transformation W is formed by stacking the weight vectors of multiple kernels. The input nodes' embedding f_{in} is also represented by a tensor of the form (C, V, T), where C is the number of channels, V is the number of nodes per frame, and T is the number of frames. The convolution operation in Eq. 4 performs a standard 2D convolution on each frame separately and then multiplies the resulted tensors with the normalized adjacency matrix $\Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}}$. Another 2D convolution was then performed along the temporal dimension of the output tensors to model the dependency between the consecutive frames. The output layer utilized the categorical cross-entropy function for loss optimization

Enhanced separable 3DGCN implementation

A multi-head self-attention layer was added after each spatial convolution operation to enhance the nodes' context representation in each frame. To achieve that, the adjacency matrix was scaled by a matrix of normalized attention scores of the same size. Each element in the attention matrix

encoded a pair-wise normalized attention score between two neighbors. Accordingly, the new spatial embedding $h^{(l+1)}$ of the nodes was computed as follows:

1. The node features were transformed through a 2D spatial convolution as in Eq. 5.

$$z^{(l)} = W^{(l)}h^{(l)} \tag{5}$$

where $W^{(l)}$ is the convolution kernel and $h^{(l)}$ is the input embedding of nodes.

2. Unnormalized attention scores were computed between each pair of neighboring nodes as in Eq. 6.

$$e_{ij}^{(l)} = LeakyReLU(\vec{a}^{(l)^T}\left(z_{i\_batch}^{(l)} || z_{j\_batch}^{(l)}\right)) \tag{6}$$

where $e_{ij}^{(l)}$ is the unnormalized attention score between nodes I and j, $\vec{a}^{(l)}$ is a learnable weight vector, and $z_{i\_batch}^{(l)}$ and $z_{j\_batch}^{(l)}$ are the feature means of nodes i and j on the entire batch.

3. The attention scores were then normalized via SoftMax as in Eq. 7. The normalized attention matrix was then used to scale the adjacency matrix.

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k\in N(i)} \exp(e_{ik}^{(l)})} \tag{7}$$

4. Finally, the output of the convolution was multiplied by the scaled adjacency matrix to form the new embedding of the nodes as in Eq. 8.

$$h^{(l+1)} = \Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}}z^{(l)} \tag{8}$$

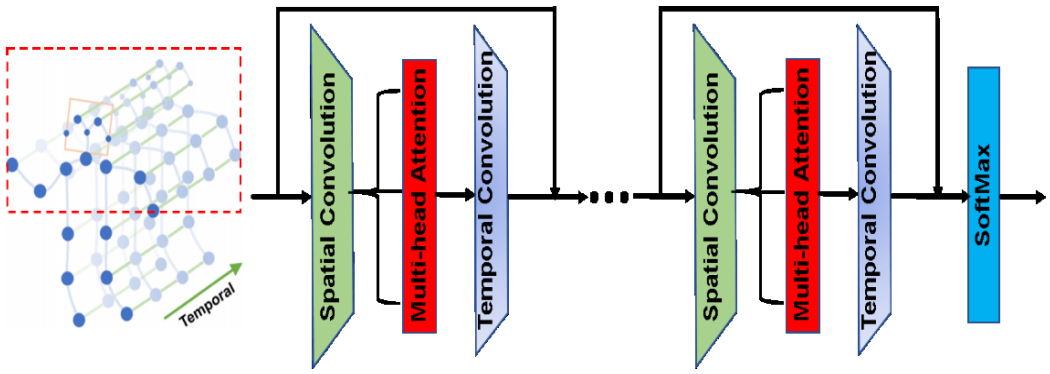


FIGURE 49, THE PROPOSED 3DGCN ARCHITECTURE.

As in the basic approach, the temporal dependence between consecutive frames was modeled by another 2D convolution operation along the temporal dimension of the output tensors. The output layer utilized the categorical cross-entropy function for loss optimization.

Furthermore, we investigated three partitioning strategies suggested by [112] for more enhancement of spatial representation.

- The uni-labeling partitioning: All the neighboring nodes of a corresponding root node are considered as a single set, including the root node itself. The representations of all nodes are transformed by a single learnable kernel.
- The distance partitioning: The nodes are partitioned into two subsets based on their distance from the root node. The first subset includes the root node with a distance d = 0, and the second subset includes the remaining nodes with d = 1. Two different learnable kernels are utilized to transform the nodes' representation in the two subsets.
- The spatial partitioning: The neighboring nodes are partitioned into three subsets based on their distance from the root node and the gravity center of the whole skeleton as follows: (1) the root node itself; (2) the centripetal: the set of nodes that are closer to the gravity center of the skeleton than the root node; and (3) the centrifugal: the set of nodes that are closer to the root node than the gravity center of the skeleton. In this work, the nose point was set as a reference point instead of the center of gravity.

6.2.3. Results

The proposed architecture in this work was implemented using Pytorch and the training was conducted on NVIDIA RTX 3090 24 GB GPU. The recognition accuracy, which is the percentage of recognition rate, is used as an evaluation metric in our experiments. It is defined as:

$$Acc = \frac{The\ numer\ of\ correctly\ recognized\ samples}{Total\ number\ of\ samples\ used\ for\ evaluation} \times 100\ \% \qquad (9)$$

Evaluation of The Basic 3DGCN-Based Architecture

To demonstrate the performance of the lightweight basic 3DGCN architecture, the evaluation was conducted on the KSU-SSL dataset and the other four datasets. Most of these datasets are publicly available with training and validation splits but not the test split. Hence, the reported results in this part are in terms of recognition accuracy on the validation data. The number of samples in both training and validation splits used in this work is summarized in

TABLE *11*. ASLLVD-20 is a partial dataset of 20 classes from the entire dataset ASLLVD, which contains 2745 classes. In this partitioning, we followed the same criteria in (Amorim, Macêdo, & Zanchettin, 2019) to create a more balanced dataset with a small number of classes.

A fixed configuration was used to conduct the training, where mini-batch gradient descent was utilized with a batch size of 32 samples and an adapted learning rate. The initial value of the learning rate was set to 0.1 with an updating step size of 40 epochs. The learning rate was decayed after each step size which enables a smoother fine tuning for the trainable parameters with the advancement in training time.

$$Lr_{new} = Lr_{current} \times \gamma \tag{10}$$

where γ was set to 0.5.

In each experiment, the architecture was trained for 200 epochs. After each epoch, if the architecture achieved better performance, the parameter values were saved so that the final model is the one that achieved the best performance regardless of the number of training epochs. The performance of the architecture on different datasets is illustrated in *FIGURE 50*.

Even though the basic architecture is very light and has a small number of trainable parameters, *FIGURE 50* illustrates that the architecture had generalized well and achieved encouraging performance on most of the datasets. The highest number of trainable parameters was $\approx 0.3\ M$ (in the case of the KSU-SSL dataset). Furthermore, the architecture was able to achieve such performance despite it being trained from scratch on the original datasets without any augmentation. This encouraging performance demonstrates that graph neural networks might perform better when protected from over smoothing via reducing the repetitions of messages passing.

Evaluation of The Enhanced 3DGCN-Based Architecture

This experiment was started by optimizing the most important hyperparameters of the partitioning strategy and the number of self-attention heads on the KSU-SSL dataset. We conducted a grid search to optimize these two hyperparameters. The search space was defined as follows:

- The partitioning strategy $PS \in \{unilabling, distance,\ spatial\}$, where these strategies are defined in Section 4.2.2.
- The number of self-attention heads $h \in \{1,2,3,4\}$.

The result of this grid search step in terms of recognition accuracy (%) is illustrated in *TABLE 12*.

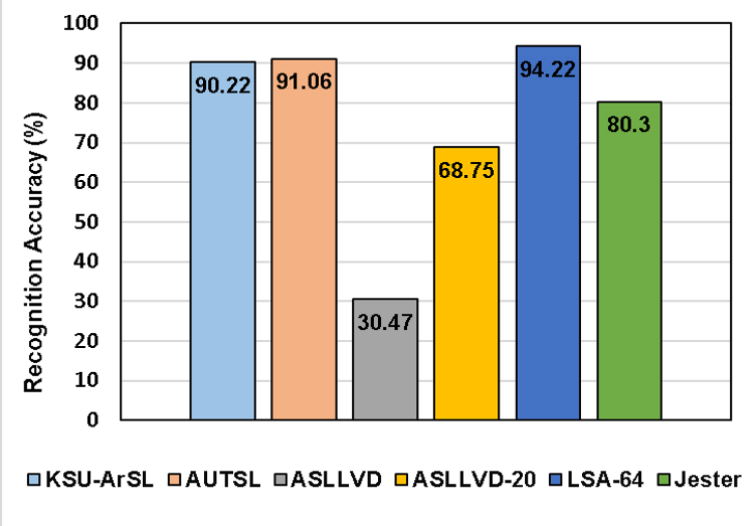


FIGURE 50, BASIC ARCHITECTURE ACCURACY ON DIFFERENT DATASETS

TABLE 12, RESULTS (% ACCURACIES) OF THE HYPERPARAMETERS' OPTIMIZATION.

		Partitioning		
		Uni-labeling	*Distance*	*Spatial*
No. of heads	1	96.62	96.77	96.79
	2	96.7	96.82	96.52
	3	97.08	96.93	96.86
	4	96.57	97.03	97.25

FIGURE 51 illustrates the convergence of the architecture with the optimal hyperparameters over training time on the KSU-SSL dataset. It also demonstrates how the architecture is smoothly tuned on the dataset with the regular decay in the learning rate.

After that, the optimized architecture was evaluated on the other datasets. The results are illustrated in *FIGURE 52*. It is clear from these results that the performance of the proposed architecture was significantly enhanced by the multi-head attention layer on the KSU-SSL, AUTSL, and Jester datasets. These three datasets are comprehensive in terms of the number of classes and the number of samples in each training and validation split, as shown in

TABLE *11*. The comprehensiveness of these datasets benefited from the increased number of parameters in the multi-head attention layer. The multi-head attention parameters generalized the architecture and enabled it to reflect the large variety in sign configuration at the frame level.

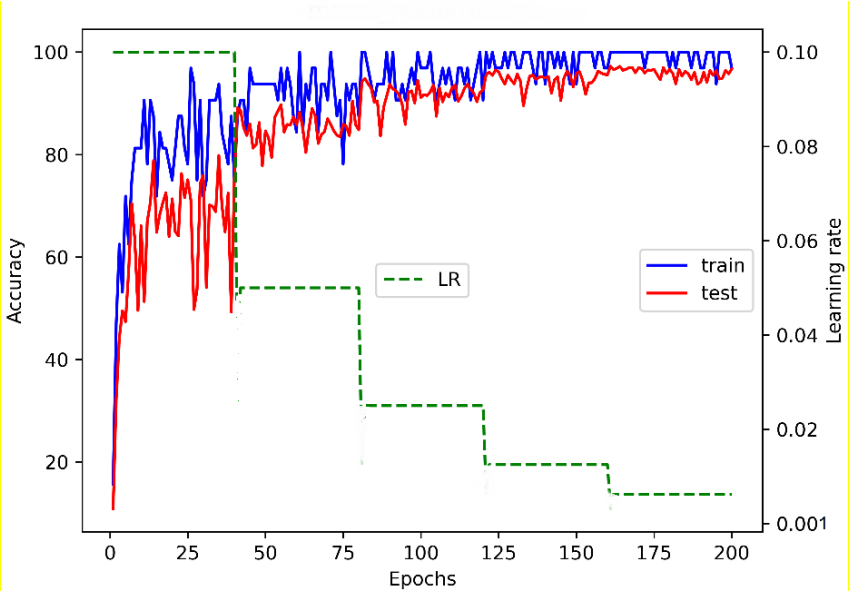


FIGURE 51, THE BEHAVIOR OF THE OPTIMIZED ARCHITECTURE ON THE KSU-ARSL DATASET.

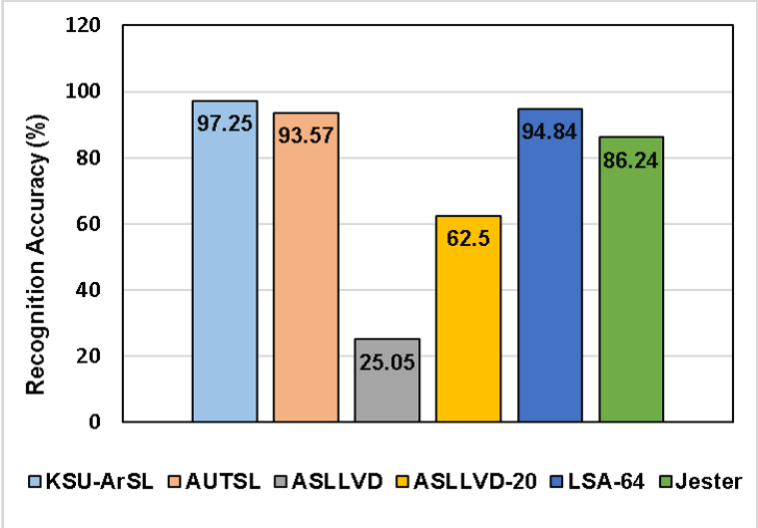


FIGURE 52, ENHANCED ARCHITECTURE ACCURACY ON DIFFERENT DATASETS.

Furthermore, it is noticed that the performance enhancement of the architecture on the LSA-64 is very slight, which might be attributed to the fact that LSA-64 was recorded under very restricted conditions regarding the background homogeneity, and there was light consistency in addition to using colored gloves. These restricted conditions enabled even conventional CNN models to

achieve high accuracies on this dataset, as the relevant patterns are easily distinguished from the non-relevant background. In such a situation, the contribution of the spatial attention layer is minimal. On the other hand, the performance of the enhanced architecture was worse on both versions of the ASLLVD dataset. This bad performance can be attributed to the size of the training data. As shown in

TABLE *11*, there are a limited number of samples for training relative to the number of classes in the dataset, which lead both the basic and enhanced architectures to overfitting. The situation became worse with the increased number of learnable parameters in the enhanced architecture, which makes the generalization more difficult.

The proposed architecture was compared with the state-of-the-art graph-based architecture on both AUTSL and ASLLVD datasets. In *TABLE 13*, the performance of the proposed architecture is compared with the reported results for different variants of the VTN architecture on the AUTSL dataset [113]. From *TABLE 13*, we can observe that the proposed architecture with spatial attention enhancement outperformed the best variant of VTN (VTN-PF) on both the validation and test datasets. Moreover, the number of trainable parameters in the proposed architecture is nearly one-hundredth the number of parameters of VTN architecture.

Similarly,

TABLE 14 shows that the proposed architecture outperformed the ST-GCN architecture [114], with large margins on both the entire ASLLVD dataset and the selected 20 classes dataset. Even though the performance of the architecture was degraded by adding the attention layer, it still outperformed the ST-GCN architecture on both datasets. The performance degradation after the addition of the attention layer is expected because of the increased effect of overfitting, since we nearly doubled the number of trainable parameters while keeping training on the same number of training samples.

TABLE 13, PERFORMANCE COMPARISON ON THE AUTSL DATASET.

Architecture	Validation Acc. (%)	Test Acc. (%)	Num of params ($\times \mathbf{10^6}$)
VTN	82.03	-	≈ 29
VTN-HC	90.13	-	≈ 51
VTN-PF	91.51	92.92	≈ 52
Basic 3DGCN (ours)	91.06	90.27	≈ 0.3
Enhanced 3DGCN (ours)	93.57	93.38	≈ 0.7

TABLE 14, PERFORMANCE COMPARISON ON THE ASLLVD DATASET.

Architecture	Dataset	Validation Acc. (%)
ST-GCN	ASLLVD-20	61.04
Basic 3DGCN (ours)		68.75
Enhanced 3DGCN (ours)		62.5
ST-GCN	ASLLVD	16.48
Basic 3DGCN (ours)		30.47
Enhanced 3DGCN (ours)		25.05

6.3. Space-Time Transformer

Vision Transformer (ViT) is a recent innovation in the field of computer vision, introduced in 2020 by Dosovitskiy et al. [115]. Unlike traditional convolutional neural networks, which have been the dominant approach in computer vision for several years, ViT is based on a completely different architecture, which employs self-attention mechanisms to process image data. ViT is based on the Transformer, a neural network architecture introduced by Vaswani et al. [116] in 2017 for natural language processing tasks. The Transformer is based on the idea of self-attention, which allows the network to attend to different parts of the input sequence during processing. In the case of ViT, the input sequence is an image, which is first divided into a grid of patches, each of which is treated as a separate input. The patches are then flattened and fed into a series of Transformer layers, which process them using self-attention mechanisms.

The key advantage of ViT is that it can process images at a much larger scale than traditional CNNs. CNNs typically require many layers to capture increasingly complex features in an image, which can lead to computational inefficiencies and difficulties in training. ViT, on the other hand, can process images at a much larger scale by breaking them down into smaller patches and processing them using self-attention mechanisms, which are more computationally efficient.

One of the major challenges in developing ViT was how to deal with the fact that the patches in an image are not inherently ordered. Unlike words in a sentence, which have a natural ordering, patches in an image can be arranged in any number of ways. To solve this problem, the researchers introduced an additional learnable positional embedding, which is added to the patch embeddings and allows the network to distinguish between different patches in the image.

Since its introduction, ViT has achieved state-of-the-art performance on several computer vision benchmarks, including the ImageNet dataset, which is widely used to evaluate the performance of computer vision models. ViT has also shown promise in other areas of computer vision, such as object detection and action recognition, where it has achieved competitive performance compared to traditional CNN-based models. While ViT is still a relatively new technology, its potential for achieving state-of-the-art performance on a wide range of computer vision tasks has already been demonstrated, and it is likely to play an increasingly important role in the field of computer vision in the years to come.

In this section, we propose a vision transformer architecture for Arabic sign language recognition. We adopt the vision transformer approach (TimeSformer) proposed in [117] as shown in Figure 53 and Figure 54. The architecture details are presented in the next section.

6.3.1. Methodology

Input clip. The TimeSformer takes as input a video $X \in \mathbb{R}^{H \times W \times 3 \times \mathrm{F}}$ consisting of F RGB frames of size $H \times W$.

Decomposition into patches. Following the research in [115], [117], each frame is split into N non-overlapping patches, each of size $P \times P$, such that the N patches span the entire frame, i.e., $\mathrm{N} = \mathrm{HW}/\mathrm{P}^2$. These patches are flattened into vectors $x_{(p,t)} \in \mathbb{R}^{3P^2}$ with $p = 1, \ldots, \mathrm{N}$ denoting spatial locations and $t = 1, \ldots, \mathrm{F}$ depicting an index over frames.

Linear embedding. Each patch $x_{(p,t)}$ is linearly mapped into an embedding vector $z_{(p,t)}^{(0)} \in \mathbb{R}^{D}$ by means of a learnable matrix $E \in \mathbb{R}^{D \times 3P^2}$:

$$z_{(p,t)}^{(0)} = E x_{(p,t)} + e_{(p,t)}^{pos}$$

Query-Key-Value computation. The Transformer consists of L encoding blocks. At each block l, a query/key/value vector is computed for each patch from the representation $z_{(p,t)}^{(l-1)}$ encoded by the preceding block:

$$q_{(p,t)}^{(l,a)} = W_Q^{(l,a)} LN\left(z_{(p,t)}^{(l-1)}\right) \in \mathbb{R}^{D_h}$$

$$k_{(p,t)}^{(l,a)} = W_K^{(l,a)} LN\left(z_{(p,t)}^{(l-1)}\right) \in \mathbb{R}^{D_h}$$

$$v_{(p,t)}^{(l,a)} = W_V^{(l,a)} LN\left(z_{(p,t)}^{(l-1)}\right) \in \mathbb{R}^{D_h}$$

Self-attention computation. Self-attention weights are computed via dot-product. The self-attention weights $a_{(p,t)}^{(l,a)} \in \mathbb{R}^{NF+1}$ for query patch (p,t) are given by:

$$a_{(p,t)}^{(l,a)} = SM\left(\frac{q_{(p,t)}^{(l,a)^T}}{\sqrt{D_h}} \cdot \left[k_{(0,0)}^{(l,a)} \left\{k_{(p',t')}^{(l,a)}\right\}_{\substack{p'=1,\ldots,N \\ t'=1,\ldots,F}}\right]\right)$$

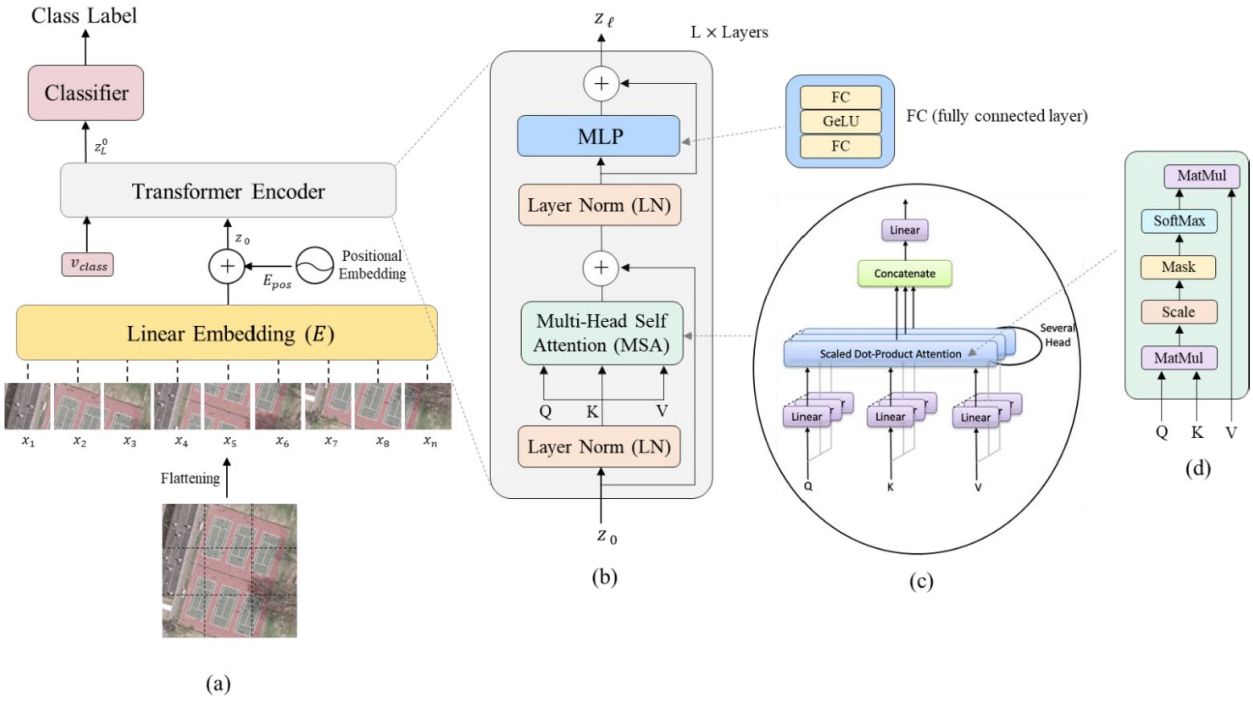


FIGURE 53, VISION TRANSFORMER (VIT) ARCHETECTURE

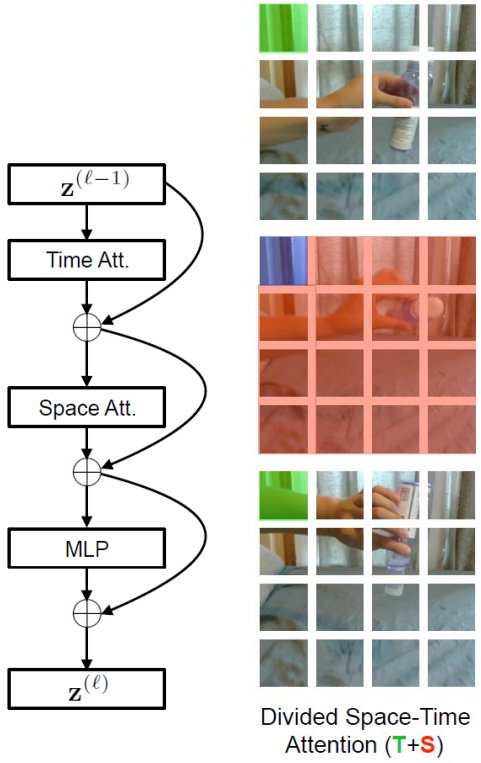


FIGURE 54, THE VIDEO SELF-ATTENTION BLOCK THAT WE INVESTIGATE IN THIS WORK.

6.3.2. Results

We used KSU-SSL in this investigation. We trained and evaluated the models on two GPUs, Nvidia 12GB 1080ti GTX and 24GB 3090 (CUDA 11.4), using Python 3.7 with the Pytorch framework. Anaconda 3 was used on Ubuntu 20.04.4 LTS. We used the following training settings across all experiments. The weights were initialized using the Glorot uniform initializer and trained using the Adam optimizer with a learning rate of 0.0009 and categorical cross-entropy loss. The models were trained using a batch size of 64 for 300 epochs with early stop patience of 50.

TABLE 15 presents the achieved accuracy of the TimeSformer model using KSU-SSL dataset. The results showed that the use of IR and RGB videos increased accuracy by 5.81%. The best accuracy was obtained by pertaining the proposed model on the Kinetics-600 dataset and then fine-tune the model using our KSU-SSL dataset.

TABLE 15, ARCHIVED PERFORMANCE USING THE SSL DATABASE

Method	Accuracy %
Pretrained on Kinetics-600 (RGB)	73.59
Pretrained on Kinetics-600 (IR & RGB)	79.4
Pretrained on Kinetics-600 (IR & RGB) with data augmentation	**95.6**

6.4. Conclusion

We presented three systems for dynamic hand gesture recognition via a combination of multiple deep learning techniques. The first system represents the hand gesture using the local hand shape features as well as the global body configuration features. This way of representation is very efficient for the complicatedly structured hand gestures of sign language. Openpose framework was utilized in this study for hand region detection and estimation. A robust face detection algorithm and the body parts ratios theory were utilized on the other hand for gesture space estimation and normalization. Two 3DCNN instances were used separately, to learn the fine-grained features of the hand shape and the coarse-grained features of the global body configuration. MLP and autoencoders were used to aggregate and globalize the extracted features and the SoftMax function for classification. Furthermore, to reduce the training cost of the 3DCNN module, we investigated domain adaptation and conducted extensive experiments to optimize the level of knowledge transfer.

The second system proposed a lightweight 3DGCN architecture for sign language recognition. The proposed architecture utilizes a few 3DGCN layers to avoid the common over-smoothing effect in deep GCN architectures, which results from the high repetitions of messages passing between the graph nodes. This shallow architecture is utilized to construct a graph representation from the most relevant MediaPipe landmarks of the signer body. This embedding step transfers the recognition problem to graph classification. Reducing the depth of the architecture might intuitively lead to less efficient representation, but we substitute that by enhancing the spatial representation with less computation cost. To achieve that, a spatial attention mechanism is added to the proposed architecture to enhance the modeling of the spatial patterns of gestures. The 3DGCN layers are decoupled into a spatial and temporal convolution, which are separated by a spatial multi-head self-attention layer. This added layer enhances the local representation in each frame rather than modeling the global dependence of the frames.

The third system used the vision transformer approach (TimeSformer) for Arabic sign language recognition. The proposed architectures were evaluated on the KSU-SSL dataset. Experimental results showed an encouraging recognition rate for the proposed systems compared to the state-of-the-art methods. For future work, the proposed architectures will also be prototyped and evaluated in a real-time sign language recognition scenario.

7. Design and development of speech recognition and speech synthesis modules

This component comprises a speech engine that recognizes the Arabic language and displays the recognized words on a screen. The recognition process allows the system to understand the spoken words, and. produce the corresponding sign using the Avatar.

7.1. Database Selection and labeling

Our intention at the time of proposing the project was to mainly use the KSU speech database [81] and label it, this is due to the richness of the KSU database and fact that it includes recording by Saudi nationals, male and female. Initial labeling of part of the KSU database was very costly and very time consuming, but we needed to keep most of the budget allocated to building the project databases to the video sign database, hence, we searched for available Arabic speech databases that are labeled. Through our research into the literature and state of the art Arabic datasets for an Arabic speech corpus, we selected the Arabic GALE corpus, which is very large and is the most suitable for our project because it was collected from broadcast news and conversations which is similar to the type of speech that our system will translate. Additionally, there are several existing researches in the literature using the GALE corpus**.** Gale has the following characteristics: (i) is publicly available, (ii) has a large collection of vocabulary and a very good amount of annotated data, transcribed, and annotated at the sentence level, and (iii) has published research using it. Gale has many phases, and each phase has conservation and news parts. In the first six months of the project, we used the parts of Gale initially available to us: GALE Phase 2 Arabic Broadcast Conversation Speech Part 1 and Part 2 and GALE Phase 2 Arabic Broadcast News Speech Part 1, as shown in Table 16.

TABLE 16, INITIAL GALE ARABIC TOTAL HOURS

Phase Name	Total hours
Gale Phase 2 Arabic broadcast conversation speech part 1	123
Gale Phase 2 Arabic broadcast conversation speech part 2	128
Gale Phase 2 Arabic broadcast news speech part 1	165
Total	**416**

After using the above parts of Gale in our initial investigation we were able to get all phases of Gale. TABLE 17 and TABLE 18 present the details of the different phases of Arabic Gale Dataset. The total number of hours of speech of all Gale phases is equal to 1227 hours.

TABLE 17, GALE ARABIC BROADCAST CONVERSATIONAL SPEECH

Phase Name	Total hours
Gale Phase 2 Arabic broadcast conversation speech part 1	123
Gale Phase 2 Arabic broadcast conversation speech part 2	128
Gale Phase 3 Arabic broadcast conversation speech part 1	123
Gale Phase 3 Arabic broadcast conversation speech part 2	129
Gale Phase 4 Arabic broadcast news speech part 1	75
Total Hours	**578**

TABLE 18, GALE ARABIC BROADCAST NEWS

Phase Name	Total hours
Gale Phase 1	17
Gale Phase 2 Arabic broadcast conversation speech part 1	165
Gale Phase 2 Arabic broadcast conversation speech part 2	170
Gale Phase 3 Arabic broadcast conversation speech part 1	132
Gale Phase 3 Arabic broadcast conversation speech part 2	128
Gale Phase 4 Arabic broadcast news speech part 1	37
Total	**649**

7.2. Overview of building an Arabic ASR system using KALDI

In this section, we will present the process of building an Arabic ASR system using the KALDI tool. We followed the chronological steps of KALDI to build the system, which covers the speech recognition task of the proposal.

In order to accomplish an Arabic Speech-To-Text (STT) system in the Kaldi platform, we need to train two models, a language model (from a dictionary) that defines the relation between the words and/or phonemes, and an acoustic model that contains the speech probability transitions.

In our work, we first trained the language model using the dictionary available from [118]. Next, we trained the acoustic model in two steps. The first step used the GMM/HMM model for the initial training of our models, as shown in Table 20. The alignment of the acoustic model and

language model utilizes the validation sets, where the diverse scores of the Word error rates show the correctness of the model. The speech model derived from this approach will be used in the second step. The second step involved the use of a Temporal Neural network, by using I-vector features and a Time Delay Neural network (TDNN).

7.2.1. Data preparation

The preparation of the data and the language model follows the commonly used steps of KALDI, where the speech and transcripts are split into train/test and dev, then the MFCC and CMVN features are generated and a language model is trained and a finite state graph is generated.

The GMM-HMM model speech features are MFCC, while the speech features for the DNN acoustic model are I-vectors. The MFCC features are extracted by the command (steps/make_mfcc.sh) for each set. Then, cepstral mean and variance statistics are computed per speaker using the script (steps/compute_cmvn_stats.sh).

For the deep neural network, I-vectors were used as a feature. The next step was to create high-resolution MFCC, then compute the diagonal UBM using 512 Gaussians, after that I-vectors were extracted. More information can be found in the KALDI script (local/nnet3/run_ivector_common.sh)

7.2.2. Training phase

We followed the “run.sh” program in gale recipe (s5b) in the Kaldi environment of this research [80]. The run program consists of 10 stages containing feature extraction, training, and decoding. In Table 19 we list all the training stages and the executed tasks at each stage.

TABLE 19: SUMMARY OF THE "RUN.SH" SCRIPT.

Stage	Function
0	Preparing data, lexicon, and language model
1	Generating MFCC features for train and test files
2	Training the monophone system (mono)

3	Align data using the mono system and training triphone system (tri1) using delta features
4	Building graph of the tri1 system and decoding the test files.
5	Align data using tri1 system, and training the triphone system using the LDA+MLLT features (tri2b)
6	Building graph of the tri2b system and decoding the test files.
7	Align data using tri2b system, and doing speaker adapting training (SAT) using the fMLLR-adapted features (tri3b)
8	Building graph of the tri3b system and decoding the test files.
9	Training and decoding the chain model using end-to-end alignments

In Table 20, we present the list of the acoustic models, generated at each of the stages of the training.

TABLE 20. GENERATED ACOUSTIC MODELS LIST

Acoustic model		Command used	Comments
GMM-HMM Stage 2 to 8	monophone	steps/train_mono.sh	
	triphone	steps/train_deltas.sh	Number of gauss = 30000
	tri2a	steps/train_deltas.sh	Using Deltas+(Delta-Deltas), #gauss = 40000
	tri2b	steps/train_lda_mllt.sh	Using LDA+MLLT, #gauss = 50000
	tri3b	steps/train_sat.sh	Using LDA + MLLT + SAT, #gauss = 100000
DNN Stage 10	TDNN	local/nnet3/run_tdnn.sh	Using chain lattice-free recipe #epoch = 3 , Activation function = relu ivector dim = 100, #hidden layers = 6

7.3. Initial investigation using 416 hours of GALE

As an initial investigation, we used Gale Arabic recipe (s5b) in Kaldi branch 5.0, which was developed by [80], with 416 hours of GALE as described in section 7.1. The run script consists of many steps that can be executed together. It starts by preparing the training data and converting the "flac" files to wave files. Then the data is prepared by splitting it into train, dev, and test sets for the news and conversation parts.

Table 21 presents the word error rate (WER) reported in the original Kaldi recipe and the WER for our trained system using different acoustic models. As expected TDNN gave the best results.

TABLE 21. WER OF AN ARABIC ASR USING GALE ARABIC SPEECH DATABASE.

Acoustic model	Type of speech	WER % (Our training)	WER % (amali recipe)[80]
Tri1	Report	31.33	26.38
Tri2a	Report	30.58	25.66
Tri2b	Report	27.83	23.32
Tri3b	Report	25.75	21.64
TDNN	Report	12.85	10.72
Tri1	Conversational	49.25	46.86
Tri2a	Conversational	47.94	45.92
Tri2b	Conversational	44.28	42.23
Tri3b	Conversational	41.52	39.26
TDNN	Conversational	27.48	24.77
Tri1	Conversational + Reports	43.54	40.35
Tri2a	Conversational + Reports	42.42	39.42
Tri2b	Conversational + Reports	39.05	36.17
Tri3b	Conversational + Reports	36.27	33.61
TDNN	Conversational + Reports	22.78	20.26

7.4. Building the Speech Recognition Module

Once we completed the initial investigation, we moved to use more data by including all the Gale data as per TABLE 17 and TABLE 18, resulting in a total of more than 1000 hours of speech. Since TDNN gave the best result in the initial investigation and when we used the whole GALE database, hence in Figure 55 we present the training and validation loss of the chain model (last stage) of the TDNN model. Figure 56 presents the training time of each iteration in the TDNN model.

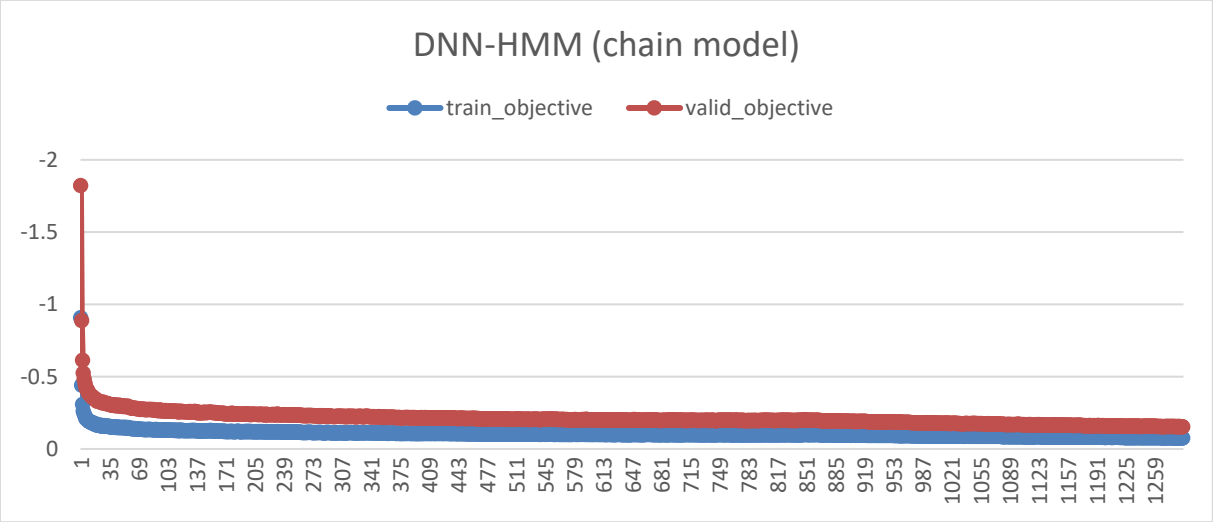


FIGURE 55: TRAINING AND VALIDATION LOSS OF CHAIN MODEL USING GALE CORPUS

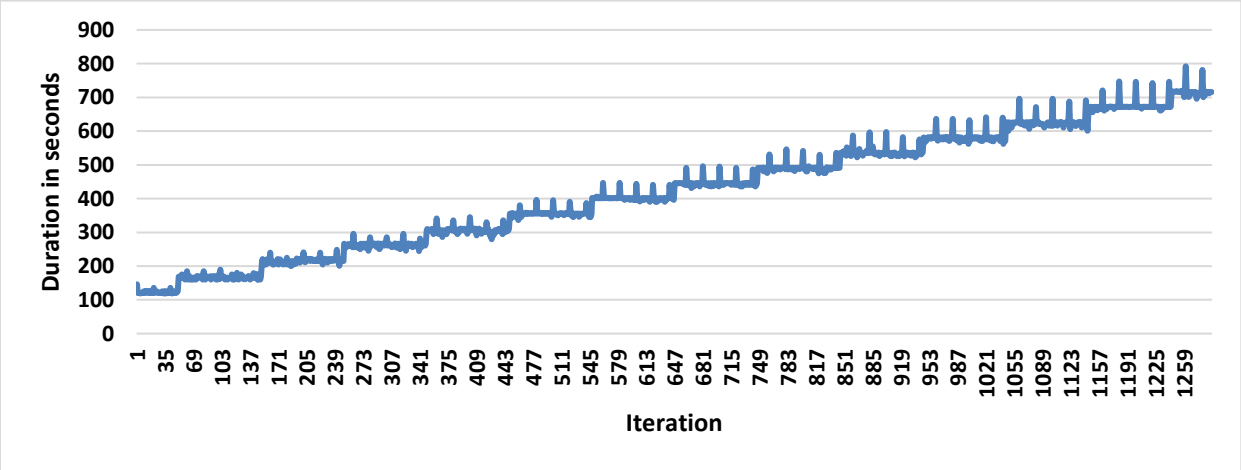


FIGURE 56: DURATION IN SECONDS FOR EACH OF TRAINING ITERATIONS OF THE CHAIN MODEL.

As shown in Figure 56, the total time increased after each iteration resulting in a very long training time, approximately 6.3 days for one model, and each hyper-parameter change required a similar simulation time.

Due to the huge size of the GALE corpus, the model took a lot of time to train, hence we used a powerful machine to train the model using this corpus. The machine that we used in the experiments has the following specifications: CPU: Intel Xeon E5-2660, GPU: TITAN RTX-24GB, and RAM: 220 GB.

To evaluate the trained model, we calculate the word error rate (WER) and character error rate (CER) for all testing files from the GALE database using each of the trained models as presented in Figure 57. We can see that the acoustic model based on TDNN (chain model) reached 14.65% WER, which is around 50% lower than the (Tri3b) model. Note that the WER presented in the original recipe [80] using the same model was 14.95%. As expected TDNN gave the best results.

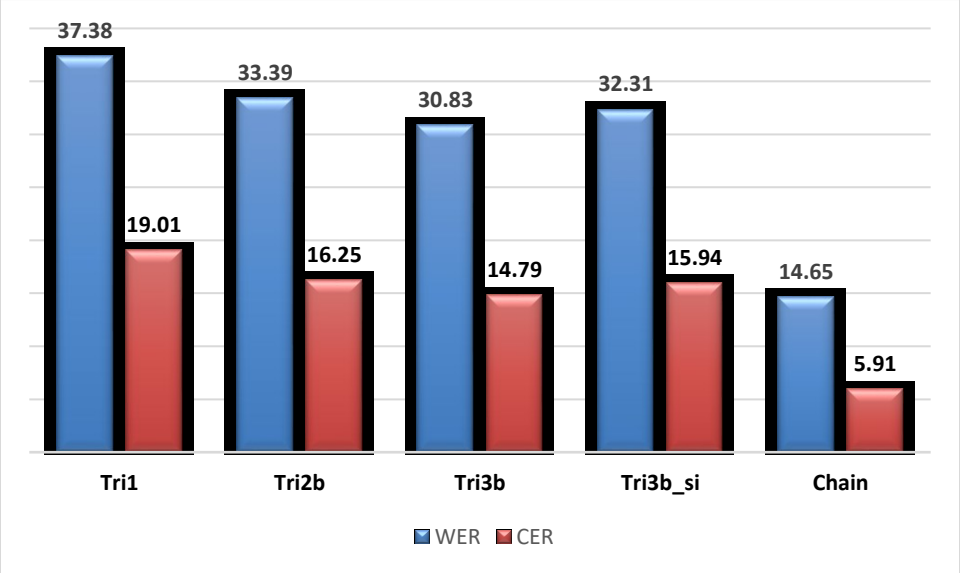


FIGURE 57: PERFORMANCE OF THE TRAINED MODEL USING THE TEST FILES OF THE GALE CORPUS.

7.5. Testing the AASR system on the speech corresponding to the KSU-SSL

As an initial test of our AASR we used the system to recognize the speech of two subjects pronouncing the signs of KSU-SSL, where one test was online and the other was offline. We used the best model which was the TDNN model (chain). The output of the model is the text of the pronounced speech in a Buckwalter format, which is converted back to an Arabic transcription. An initial interface of the recognition module is shown in Figure 58.

FIGURE *58*, INTERFACE OF THE RECOGNITION PROCESS WITH SOME RECOGNIZED SPEECH SAMPLES

In Table 22, we present the details of the performance achieved by the TDNN chain model for the speech of two different speakers (unknown to the model at training phase), who pronounced all the signs of KSU-SSL (293 words), in terms of the number of testing utterances, insertions errors, deletion errors, and substitution errors.

TABLE 22, DETAILS PERFORMANCE OF THE CHAIN MODEL

	# of Correct words	# of Insertions	# of Deletions	# of Substitutions	WER (%)
Speaker1	283	3	0	35	11.94
Speaker2	273	2	1	44	14.78

As we can notice from Table 22, most of the errors are substitution errors. After deep investigation, we found that the high WER can be attributed to the fact that our scoring considers a word as erroneous if only one phoneme is different. To display this remark, in Table 22, we present the output of the TDNN chain model for the speech of two speakers for some word examples, where we can see that mostly one letter was in error in the recognized words and the

erroneous letter was not in the root of the word. This can be corrected in post-processing which can be investigated in the future.

TABLE 23, WORD LEVEL SPEECH SAMPLE RECOGNITION

Sign words	Speaker 1	Type of error	Speaker 2	Type of error
هوائية	هوائية	Correct	هوائية	Correct
قطرة	قطرة	Correct	قطرة	Correct
قفص	قفصه	Substitution	قفصه	Substitution
صدري	الصدري	Substitution	الصدري	Substitution
قلب	قلب	Correct	قلب	Correct
كعب	كعبة	Substitution	وكعب	Substitution
لاصق	لاصق	Correct	لاصق	Correct
مخدرات	مخدرات	Correct	مخدرات	Correct

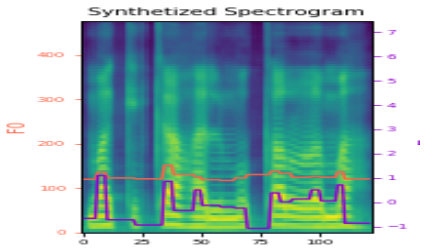
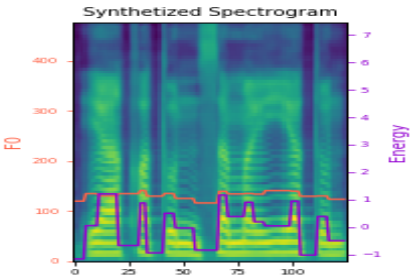
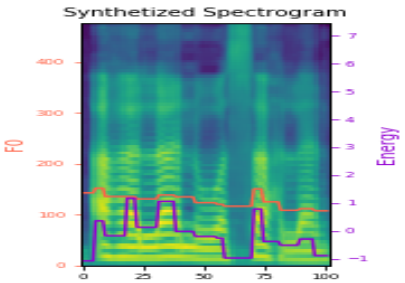
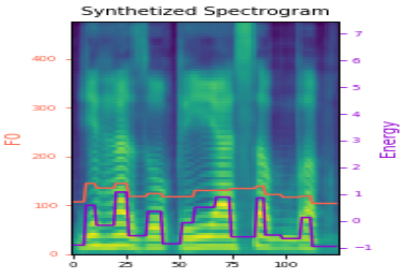
7.6. Building the speech synthesis module

In this module, we aim to design and build a text-to-speech module to generate high-quality speech from predefined text. We investigated state-of-the-art speech synthesis models to generate high-quality speech from the text, as stated in the proposal. As a result, we choose the state-of-the-art neural TTS model FastSpeech2 [97] based on a variety of experiments. We selected FastSpeech2 because of its fast properties and support for multi speakers. The main goal of this module is to take the text of the recognized signs from the sign language module and pronounce it to the non-deaf person.

To train the FastSpeech2 model for producing the speech for only the selected 293 signs we needed to record the speech of these signs many times, but this will limit the speech synthesis to only the 293 signs. Our future goal is to cover all the 3000 signs of the Saudi sign dictionary, hence instead we opted to use KSU speech database [81] to build a speech synthesis system that can vocalize any text. We trained FastSpeech2, with the speech of selected speakers from the KSU speech database who were also on the project team and agreed to use their speaking style in the system. The trained model produced good quality speech for any text including the words of the 293 signs.

In Table 24, the synthesized spectrograms for four signs from the project 293 signs are shown, where we can see that FastSpeech2 can generate high-quality spectrograms. After generating the spectrograms by FastSpeech2 we used the vocoder HiFi-GAN to generate the speech using the spectrograms.

TABLE 24: EXAMPLES OF THE GENERATED SPECTROGRAMS.

Word	Synthesized Spectrogram
مَكَّه المُكَرَّمَه	
إِعَاقَةٌ سَمَعِيَةٌ	
أَهَلاً وَسَهْلاً	
جِهَازُ قِيَاس الضَّغْط	

8. Design and building of the Avatar module

8.1. Design of the Avatar

To develop a Saudi sign language Avatar application, it is required to make or draw a personage with local clothes, from both genders, at a medium age, as shown in FIGURE 59. Other personages such as children might be developed in the near future.

The design of the sign language Avatar passed through many steps:

- Firstly, we took a men and women character from IClone character creator and changed their physiques to match the Saudi men and women. The physiques include facial characteristics, such as skin, nose, eyes colors, general face shape, body size…etc.

- Secondly, we exported only the bodies of those characters to Marvelous software [119], which is specialized in cloth design. In Marvelous, we designed and simulated the Arabic Thob for the man and the Hijab for the woman character.

- Thirdly, we exported only the clothes back to IClone character creator, and obtained the characters as shown in FIGURE 59.

FIGURE 59, LOCAL SIGN LANGUAGE CHARACTERS

8.2. Improvement of the Avatar

Improvements to the Avatar were done based on the feedback from our test team is repeatedly used to correct the actual movements. One main improvement was to keep the character movements smooth within the clothes, to solve this we exported the clothes and the bodies to Maya design software [120] where we fixed the mesh problems, especially when the character moves. Then we exported only the clothes back to IClone character creator and obtained the characters. Other improvements were also performed in the design step where building the Avatar took many cycles of improvements.

8.3. Design of the Avatar External Components

Some external components have been developed by our expert, where a classroom has been developed to virtualize the position of the character inside a school-like system, as shown in Figure 60.

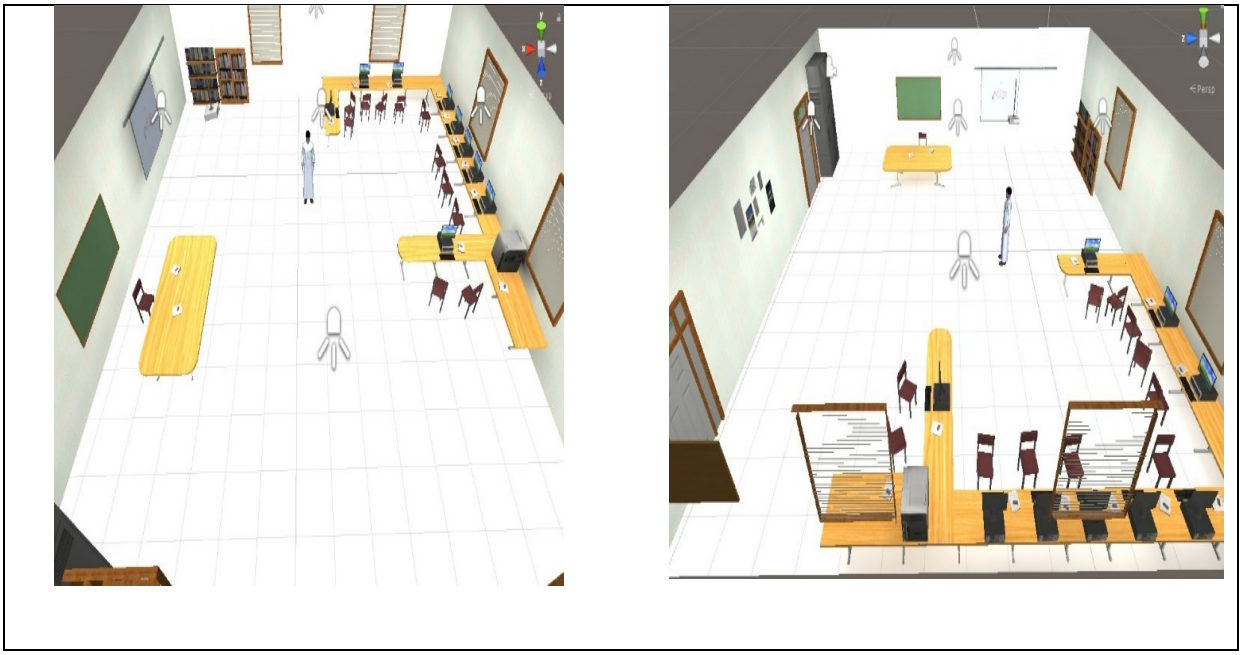


FIGURE 60, SAMPLE OF THE DEVELOPED EXTERNAL COMPONENTS – CLASSROOM

8.4. Recording the signs

The total number of signs within the Saudi dictionary is about 3000 signs, from which, we selected to record 293 signs to have a beneficial pilot sign translator. The selected categories and number of signs per category are shown in

TABLE 25.

TABLE 25, SELECTED RECORDED SIGNS PER DOMAIN

Section	Number of recorded signs
Alphabet	37
Common words	39
Days and Date	11
Family	8
Kings	9
Numbers	11
Pronouns and Adverbs	18
Verbs	20
Hospital and Medical	133
Regions	7
Total	293

As we mentioned in section 5.1, most of the signs are from the medical field, in the intent to make pilot tests, within a hospital or a medical clinic.

We are using the word “recording”, but it is not a recording as per the video part, it is a more complicated process that involves the mastering of diverse software for human gestures analysis and design. To record the motion of the sign, we used the IClone Pro [121], 3DXchange [122] and Unity [123].

First the sign designer uses IClone Pro to create the movements of a group of signs, then he transfers the recorded signs to 3DXchange to split each sign alone. Next, the designer exports the recorded signs as FBX file and use Unity to rename the file with Arabic names. We used Unity at this step because 3DXchange does not support Arabic letters. Naming the files in Arabic was to make the integration with the speech recognition module easier.

When the designer finishes a group of 20 to 30 signs he sends them to the technical team expert to check if the recording was done correctly. When the designer finishes 100 signs we send the recorded signs to an expert in sign language to verify them. The expert uses an application developed by our team to check if the Avatar correctly performed the signs and write notes about

the wrong signs so we can correct them. A sample view of the developed mobile app is shown in Figure 61. Some selected Avatar signs are presented in Figure 62.

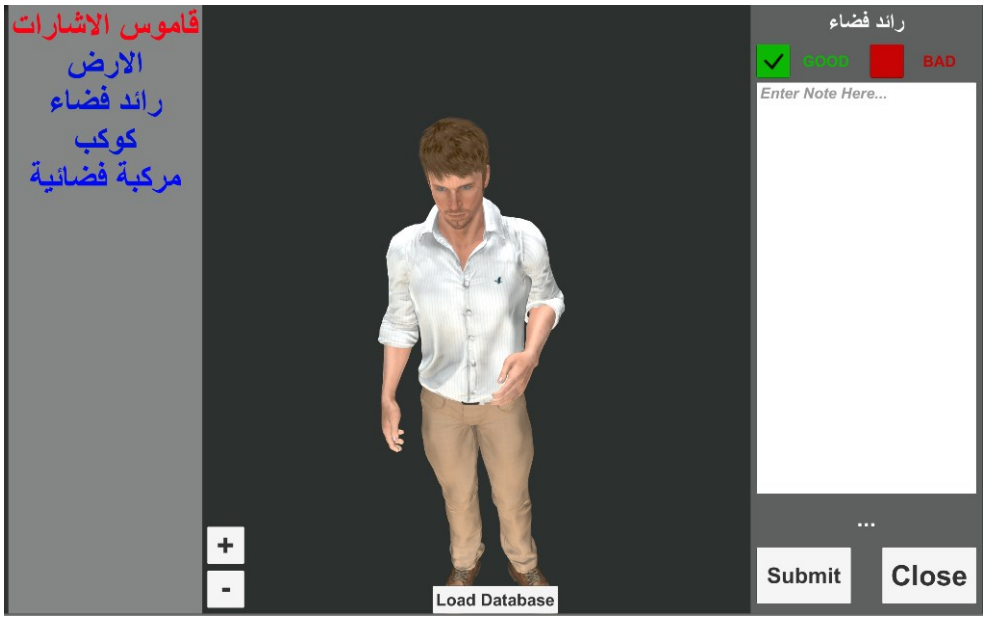


FIGURE 61, SIGN CHECKING IN THE NEWLY DEVELOPED MOBILE APPLICATION

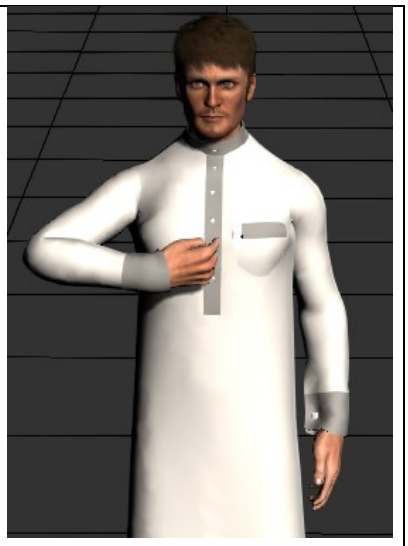
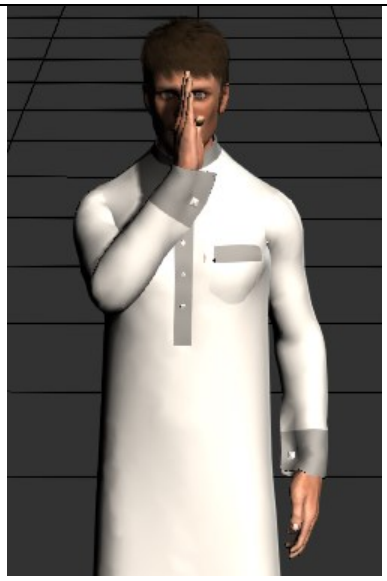
		
Father sign	Two sign	Me sign
		
Tuesday sign	Monday sign	Eight

FIGURE 62, AVATAR SAMPLE SIGNS

9. Integration between different modules

9.1. Integration between speech recognition module and avatar module

After the teams responsible of the speech recognition module and the avatar module finished building those two modules they started the integration between those two modules, and the difficulty was how to integrate two different modules developed with two or more than two different programing languages? The best solution was to use ROS Robot Operating System to send and receive data between the two modules.

ROS is an open-source, meta-operating system for robots. It provides the services expected from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes. A ROS system is comprised of a number of independent nodes, each of which communicates with the other nodes using a publish/subscribe messaging model as shown in example in Figure 63. These messages could be consumed by any number of other nodes. The nodes in ROS do not have to be on the same system (computers) or even of the same architecture. The developer could have an Arduino publishing messages, a laptop or an Android phone subscribing to them. This makes ROS really flexible and adaptable to the needs of the user, and this is why we choose ROS to integrate between the speech recognition module and avatar module.

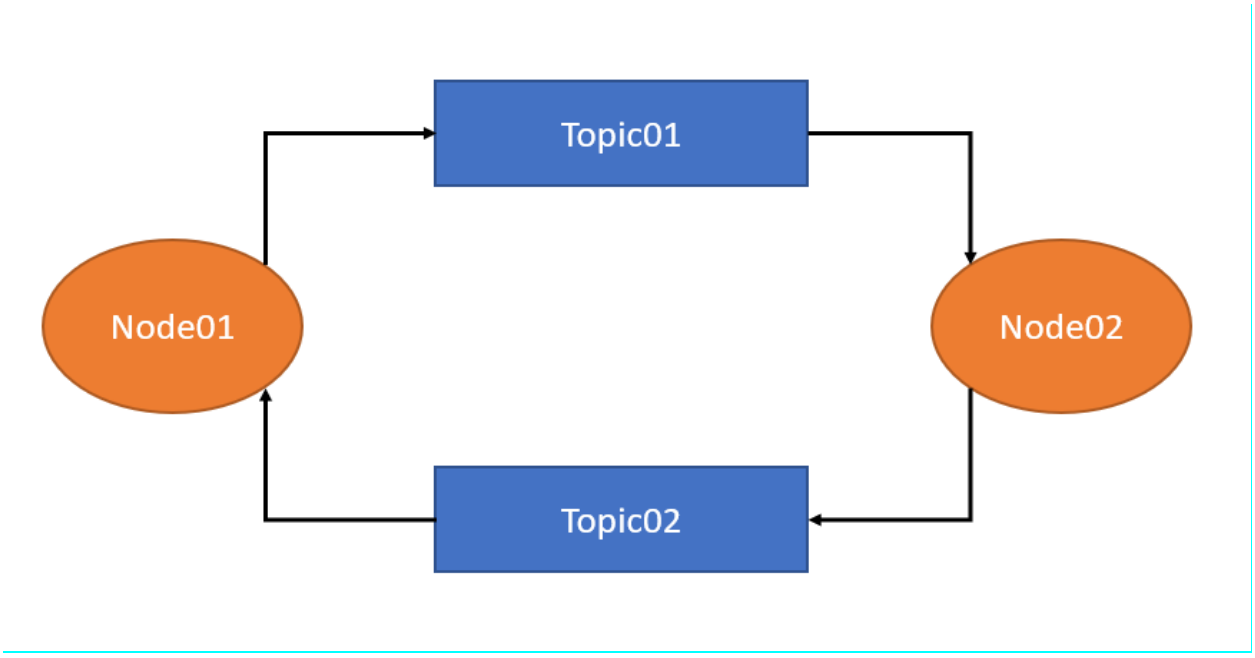


FIGURE 63 ROS STRUCTURE

For the speech recognition module, we need a trigger to start and stop the module when it is needed and also, we need to send the result of the speech recognition to the avatar module as a text, so the speech recognition module team created a speech recognition node that contained a subscriber to topic called “start_stop_REC” and the avatar module team also created an avatar node that has a publisher to the same topic. In this way the avatar module can send trigger signal to the speech recognition module to start or stop the recording of audio signal from the microphone. And in the other way the speech recognition node has a publisher to a topic called “text_from_speech” to publish the result of the speech recognition process at the end of the audio recording, in the other side the avatar module will subscribe to this topic to get the text result from the speech recognition module. Figure 64 shows the communication structure of the two modules.

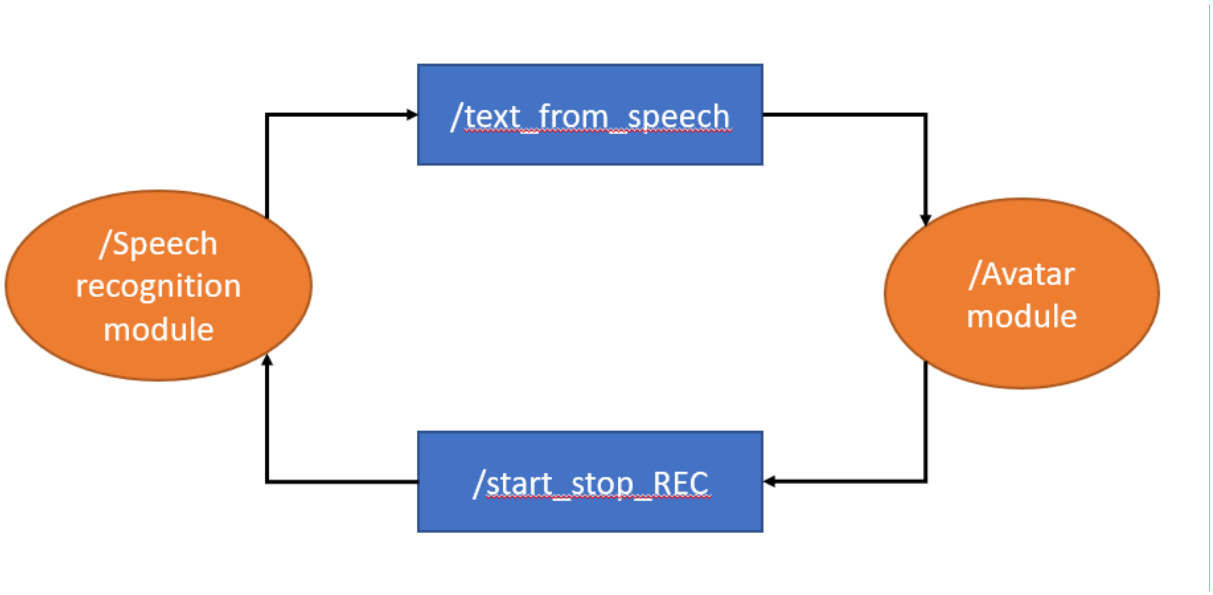


FIGURE 64 COMMUNICATION BETWEEN SPEECH RECOGNITION MODULE AND AVATAR MODULE

In the avatar module a graphical record button was created and linked to the “start_stop_REC” topic as shown in Figure 65, so when the user presses the button a “Start” message will be sent to this topic and when he releases it a “Stop” message will be sent to the same topic.

FIGURE 65 START SPEECH RECOGNITION BUTTON

In the other side, the speech recognition module has a callback function linked to the "start_stop_REC" topic, so when the "Start" message is received the speech recognition module will start recording audio from the microphone, then when a "Stop" message is received the speech recognition module stop the recording from the microphone and start the recognition process on the recorded audio file, and when it get the result of the speech recognition the result it will be sent to the "text_from_speech" topic. In the other side the avatar module has an avatar controller linked to this topic, so when the result from the speech recognition module is received on this topic the avatar controller split the text received into list of words, then start looking for most similar synonym in the avatar module database using DTW (Dynamic Time Warping) algorithm, if it find a similar synonym the controller will send the corresponding motion of each word to the avatar to perform it, else if it did not find a similar synonym the controller will split the word that was not found into letters and send the corresponding motion of each letter to the avatar so the word will be performed as sequence of letters.

If the text contains a number, the avatar controller will split the received number into digits using the result of division and the modulo. Then each digit will be performed according to its position in the number.

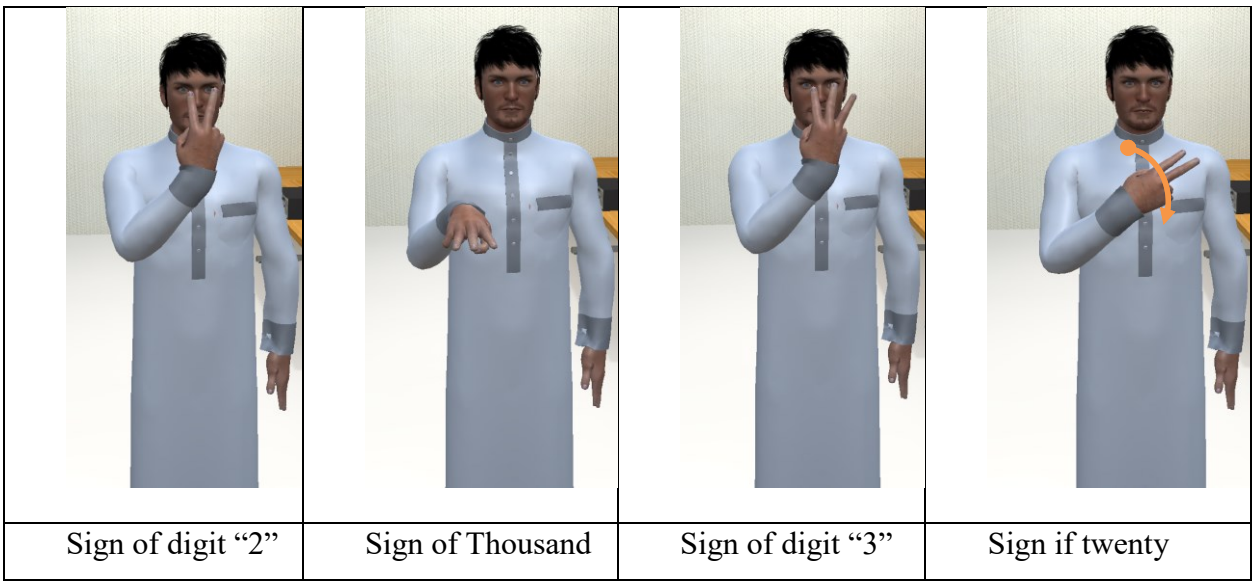
The avatar controller can handle numbers up to 12 digits **(123.456.789.123)**, and the process will be as follow:

First the avatar controller splits the number into groups of three digits and then insert the letters "T", "M" and "B" in the positions 3, 6, and 9 respectively if exist, **(123B456M789T123)** the letter "T" means thousand, the latter "M" means million and the latter "B" means billion.

Second the avatar controller takes the list of characters (the number after adding the letters to it) and switch between the first and second digit of each group of four characters

(132B465M798T132) because in pronouncing Arabic numbers the ones digit is pronounced before the tens digit.

Finally, the avatar controller split the list of characters now into groups of 4 characters **(132B 465M 798T 132)**, and start sending the corresponding motion of each character to the avatar to perform it, starting from left to right. If the digit is in forth position the avatar controller will add two zeros "00" to the digit so it will be performed in the form of hundreds, and if the digit is in the first position the avatar controller will add one zero "0" to it so it performs in the form of dozens. Example, the number 2023 will be transformed into 2T023 according to step one, then switching between first and second digit and switching between fifth and sixth according to step two 20T032. Finally, the avatar controller will send the motion of each digit as follow: motion of sign "2", zero will be ignored, then motion of sign "Thousand", also the second zero will be ignored, then motion of sign "3", then the digit "2" is in first position so the avatar controller will add one zero to it and the motion of sign "20" will send to the avatar, as shown in Figure 66.



Sign of digit "2"	Sign of Thousand	Sign of digit "3"	Sign if twenty

FIGURE 66 AVATAR PERFORMING THE NUMBER 2023 IN ARABIC SIGN LANGUAGE

We did the integration and is working and below are some examples shown in FIGURE 67 and FIGURE 68.

FIGURE 67 MOTION OF THE SIGN "SALAM ALAIKUM"

FIGURE 68 MOTION OF THE SIGN "KING SAUD UNIVERSITY"

9.2. Integration between sign recognition module and speech synthesis module

The main objective of this project is to offer two-way translation between the deaf and non-deaf. In this section, we present the integration between the sign recognition module and TTS module in order to allow the deaf to make the signs and the system will recognize the text and display it and generate the corresponding audio. Hence, a non-deaf person can read the text of signs or hear the audio of them.

Figure 69 shows the general diagram for the integration of TTS model and the sign language model. Due to the fact that the TTS and sign language modules are developed in the Python environment, the integration between the two modules is done in Python. We started by making the TTS module a function to call, then we put it with the sign recognition module in the same system. Once the sign recognition module recognizes the sign, the corresponding text will be fed

to the TTS module to generate the audio and run the generated audio. FastSpeech2 needs the Arabic text to have diacritics, hence we started feeding the recognized text to the Tashkeel model, as shown in Figure 69, to add the diacritics to the text and then sent it to the TTS.

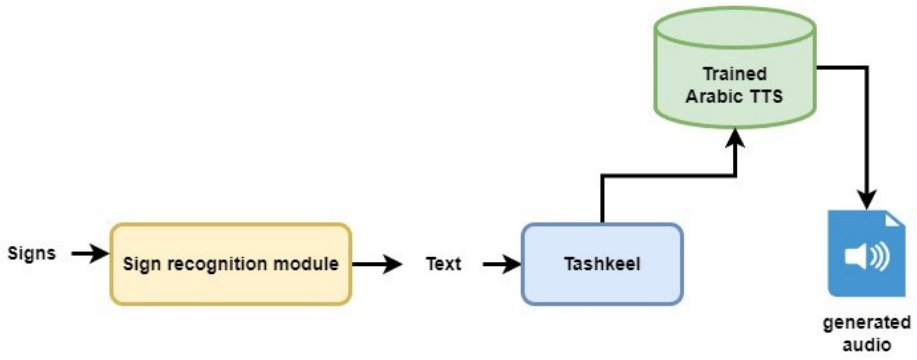


FIGURE 69: FLOW DIAGRAM OF INTEGRATION TTS WITH THE SIGN LANGUAGE MODEL.

The sign recognition system based on the proposed Transformer architecture was validated in real time using 20 randomly selected sings. The test was performed by three subjects, as shown in Table 26. The results showed that the system is able to recognize the singe in real time with high accuracy.

TABLE 26. THE RESULTS OF ONLINE REAL TIME VALIDATION OF THE SIGN RECOGNITION SYSTEM.

		Subject 1		Subject 2		Subject 3	
Class	**Sign**	Trial 1	Trial 2	Trial 1	Trial 2	Trial 1	Trial 2
Regions	Abha	√	√	√	√	√	√
Pronouns and adverbs	above	√	√	√	√	X (upon)	√
Numbers	Five	X (Ta)	X (Ra)	√	√	X (9)	X (6)
Common	Head of Department	√	√	√	√	√	√
	The light	X	X	X	X	X	X
	The good	√	√	√	√	√	√
Healthcare	heart beats	√	√	√	√	√	√
	abortion	√	X	√	√	√	√
	birth	√	√	√	√	√	√
	elevator	√	√	√	√	√	√

	massage	X	X	√	√	√	√
	spine	√	√	√	√	√	√
	trachea	X	X	√	√	√	√
Kings	King Khaled	√	√	√	√	√	√
Alphabets	alif mad	√	√	√	√	√	√
	lam	√	X	X	X	√	√
	yaa	√	√	√	√	√	X
Family	mother	X (Smoke)	X (Teeth)	√	X (Teeth)	X (Teeth)	X (Smoke)
Days	Saturday	√	√	√	√	√	√
Verbs	To enter	√	√	√	√	√	√
		75%	65%	90%	85%	80%	80%
		70%		**87.5%**		**80%**	

10. Future work

For future work, we will investigate three major challenges in sign language recognition. The first challenge is the real-time recognition of sign language, which includes online recognition with very low latency. Most of the studies focused on offline sign language recognition, as the dataset is collected and analyzed in offline mode. However, in real-world deployment, the sign language recognition systems need to deal with live streams of sign data and deliver real-time classification outcomes, which remains difficult. The second challenge is the real-environment employment of sign language recognition. A major issue with current sign language recognition systems is that most studies are performed in a controlled laboratory environment, regardless of the actual environment of the intended users. The third challenge is to detect and identify sign language from a continuous video stream.

For speech recognition and synthesis modules, during the writing of this report, Saudi Data and Artificial Intelligence Authority (SDAIA) released a huge Arabic speech database called SADA, which consists of more than 600 hours of recording. Most of the speech in SADA is in the Saudi dialect. Hence, we will investigate how to use it to enhance the STT and TTS modules so that it works well for the Saudi dialect.

For the Avatar, we will design more characters with different traditional clothing also create more background environments such as a hospital, and airport and add augmented reality to the Avatar interface where we can display the Avatar in any real place we want. In terms of the dataset, we will append more words to the dataset to cover other sections from the Saudi Sign Language Dictionary.

11. References

[1] “Disability Survey.” [Online]. Available: https://www.stats.gov.sa/sites/default/files/disability_survey_2017_ar.pdf.

[2] “Saudi Vision 2030.” [Online]. Available: https://www.vision2030.gov.sa/media/rc0b5oy1/saudi_vision203.pdf.

[3] A. Al-Nafjan, B. Al-Arifi, and A. Al-Wabil, “Design and Development of an Educational Arabic Sign Language Mobile Application: Collective Impact with Tawasol,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9176, pp. 319–326, 2015.

[4] R. G. Brill, B. MacNeil, and L. R. Newman, “Framework for appropriate programs for deaf children. Conference of educational administrators serving the deaf.,” Am. Ann. Deaf, vol. 131, no. 2, pp. 65–77, 1986.

[5] H. S. Al-Khalifa, “Introducing Arabic sign language for mobile phones,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6180 LNCS, no. PART 2, pp. 213–220, 2010.

[6] “Deafness and hearing loss.” https://www.who.int/en/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed Aug. 14, 2021).

[7] “Continued shortage of ASL interpreters causes strain as need increases | wzzm13.com.” https://www.wzzm13.com/article/news/asl-interpreters-needed-in-michigan/69-4fb88c9c-d114-47c0-8646-7b493c2f4d7f (accessed Aug. 28, 2021).

[8] “ASL Deafined | How to Learn ASL through video lessons online.” https://www.asldeafined.com/2019/04/shortage-of-asl-interpreters/ (accessed Aug. 28, 2021)..

[9] “Coronavirus: Lack of sign language interpreters leads to legal case against government - BBC News.” https://www.bbc.com/news/disability-52323854 (accessed Aug. 28, 2021)..

[10] M. Maschendorf Thomaz, V. M. Milbrath, R. I. B¬rtschi Gabatz, V. L. Freitag, and J. Cardoso Vaz, “Accessibility of adolescents with hearing impairment to health services,” Rev. Eletr. Enferm, vol. 21, pp. 55502–55503, 2019.

[11] J. Branson and D. Miller, “Chapter 5. Beyond ‘Language’: Linguistic Imperialism, Sign Languages and Linguistic Anthropology,” Disinventing Reconst. Lang., pp. 116–134, Nov. 2018.

[12] A. S. Al-Shamayleh, R. Ahmad, N. Jomhari, and M. A. M. Abushariah, “Automatic Arabic sign language recognition: A review, taxonomy, open challenges, research roadmap and future directions,” Malaysian J. Comput. Sci., vol. 33, no. 4, pp. 306–343, Oct. 2020.

[13] Alrayes Tareq, “Bilingual Students: Philosophy & Strategies. A theoretical research, submitted to the The 9th Arab Conference on Rehabilitation & Care of the Special Needs in Arab World, Present and Future.,” in The 9th Arab Conference on Rehabilitation & Care of the Special Needs in Arab World, Present and Future., 2006.

[14] M. Marschark, Educating deaf students : from research to practice. 2002.

[15] H. Knoors, G. Tang, and M. Marschark, “Bilingualism and Bilingual Deaf Education,” Biling. Biling. Deaf Educ., pp. 1–20, Aug. 2014.

[16] G. P. Berent and R. R. Kelly, “The efficacy of visual input enhancement in teaching deaf learners of L2 english,” Underst. Second Lang. Process, pp. 80–105, Jan. 2007.

[17] C. Shantie and R. J. Hoffmeister, “Why Schools for Deaf Children Should Hire Deaf Teachers: A Preschool Issue:,” https://doi.org/10.1177/002205740018200304, vol. 182, no. 3, pp. 42–53, Dec. 2017.

[18] R. Sutton-Spence and C. Ramsey, “What we should teach deaf children: Deaf teachers’ folk models in Britain, the USA and Mexico,” Deaf. Educ. Int., vol. 12, no. 3, pp. 149–176, Sep. 2010.

[19] S Foster et al., “Inclusive instruction and learning for deaf students in postsecondary education,” J. Deaf Stud. Deaf Educ., vol. 4, no. 3, pp. 225–235, Sep. 1999.

[20] A. K. Whyte and D. A. Guiffrida, “Counseling Deaf College Students: The Case of Shea,” J. Coll. Couns., vol. 11, no. 2, pp. 184–192, Sep. 2008.

[21] C. A. Bisol, C. B. Valentini, J. L. Simioni, and J. Zanchin, “Deaf students in higher education: reflections on inclusion,” Cad. Pesqui., vol. 40, no. 139, pp. 147–172, 2010.

[22] M. Hyde, R. Punch, D. Power, J. Hartley, J. Neale, and L. Brennan, “The experiences of deaf and hard of hearing students at a Queensland University: 1985–2005,” https://doi.org/10.1080/07294360802444388, vol. 28, no. 1, pp. 85–98, Feb. 2009.

[23] P. S. Kermit and S. Holiman, “Inclusion in Norwegian Higher Education: Deaf Students’ Experiences with Lecturers,” Soc. Incl., vol. 6, no. 4, pp. 158–167, Dec. 2018.

[24] TangAo, LuKe, WangYufei, HuangJie, and LiHouqiang, “A Real-Time Hand Posture Recognition System Using Deep Neural Networks,” ACM Trans. Intell. Syst. Technol., vol. 6, no. 2, Mar. 2015.

[25] J. Lukomski, “Deaf College Students’ Perceptions of Their Social-Emotional Adjustment,” J. Deaf Stud. Deaf Educ., vol. 12, no. 4, pp. 486–494, Oct. 2007.

[26] K. Nagle, L. A. Newman, D. M. Shaver, and M. M. Nagle, “COLLEGE AND CAREER READINESS: COURSE TAKING OF DEAF AND HARD OF HEARING SECONDARY SCHOOL STUDENTS,” vol. 160, no. 5, 2016.

[27] “College Students’ Sense of Belonging: A Key to Educational Success for.” https://www.routledge.com/College-Students-Sense-of-Belonging-A-Key-to-Educational-Success-for-All/Strayhorn/p/book/9781138238558 (accessed Sep. 12, 2021).

[28] P. A. Braswell-Burris, “Factors affecting the educational and personal success of deaf or hard of hearing individuals,” 2010, Accessed: Sep. 01, 2021. [Online]. Available: http://oatd.org/oatd/record?record=handle%5C%3A10211.10%5C%2F359.

[29] R. L. Ward, “The Experiences of Deaf College Graduates: Barriers and Supports to Earning a Post-secondary Degree.” 2015.

[30] D. Cokely, “The Effects of Lag Time on Interpreter Errors,” Sign Lang. Stud., vol. 53, no. 1, pp. 341–375, 1986.

[31] K. L. Sadler, “Accuracy of sign interpreting and real-time captioning of science videos for the delivery of instruction to deaf students,” ProQuest Diss. Theses, pp. 129-n/a, 2009.

[32] N. J, “University interpreting: linguistic issues for consideration,” J. Deaf Stud. Deaf Educ., vol. 7, no. 4, pp. 281–301, Oct. 2002.

[33] S. Hale, “Interpreting culture. Dealing with cross-cultural issues in court interpreting,” Perspectives (Montclair)., vol. 22, no. 3, pp. 321–331, 2014.

[34] B. Schick, K. Williams, and L. Bolster, “Skill levels of educational interpreters working in public schools.,” J. Deaf Stud. Deaf Educ., vol. 4, no. 2, pp. 144–155, Mar. 1999.

[35] M. Al-Ahmed, “Developing Accreditation practices for Sign Language Interpreters in Saudi Arabia: a proposed Conception. Unpublished PhD thesis,” College of Education, King Saud University, Riyadh.

[36] T. Al-Amri, “Verifying the performance Level of sing language interpreters in the Kingdom of Saudi Arabia, Unpublished PhD thesis,” College of Education, King Saud University, Riyadh, 2021.

[37] S. M. Halawani, D. Daman, S. Kari, and A. R. Ahmad, “An Avatar Based Translation System from Arabic Speech to Arabic Sign Language for Deaf People,” IJCSNS Int. J. Comput. Sci. Netw. Secur., 2013.

[38] G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo, and M. Khan, “An Automatic Arabic Sign Language Recognition System based on Deep CNN: An Assistive System for the Deaf and Hard of Hearing,” Int. J. Comput. Digit. Syst., vol. 9, no. 4, pp. 715–724, 2020.

[39] Y. Quan and P. Jinye, “Application of improved sign language recognition and synthesis technology in IB,” 2008 3rd IEEE Conf. Ind. Electron. Appl. ICIEA 2008, pp. 1629–1634, 2008.

[40] M. A. Alobaidy and S. K. Ebraheem, “Application for Iraqi sign language translation on Android system,” Int. J. Electr. Comput. Eng., vol. 10, no. 5, pp. 5227–5234, Oct. 2020.

[41] N. E. Abuzinadah, “An avatar-based system for Arabic sign language to enhance hard-of-hearing and deaf students’ performance in a fundamentals of computer programming course.,” Nov. 2020.

[42] T. Shanableh and K. Assaleh, “Telescopic Vector Composition and Polar Accumulated Motion Residuals for Feature Extraction in Arabic Sign Language Recognition,” EURASIP J. Image Video Process. 2007 20071, vol. 2007, no. 1, pp. 1–10, Oct. 2007.

[43] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, “Arasl: Arabic alphabets sign language dataset,” Data Br., vol. 23, p. 103777, 2019.

[44] S. M. Shohieb, H. K. Elminir, and A. M. Riad, “Signs World Atlas; a benchmark Arabic Sign Language database,” J. King Saud Univ. - Comput. Inf. Sci., 2015.

[45] M. Alfonse, A. Ali, A. S. Elons, N. L. Badr, and M. Aboul-Ela, “Arabic sign language benchmark database for different heterogeneous sensors,” in 2015 5th International Conference on Information and Communication Technology and Accessibility, ICTA 2015, 2016.

[46] M. Mohandes, S. I. Quadri, and M. Deriche, “Arabic sign language recognition an image - Based approach,” in Proceedings - 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW’07, 2007.

[47] M. ElBadawy, A. S. Elons, H. Sheded, and M. F. Tolba, “A proposed hybrid sensor architecture for arabic sign language recognition,” in Intelligent Systems’ 2014, Springer, 2015, pp. 721–730.

[48] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, and M. A. Mekhtiche, “Hand Gesture Recognition for Sign Language Using 3DCNN,” IEEE Access, vol. 8, pp. 79491–79509, 2020.

[49] M. A. Bencherif et al., “Arabic sign language recognition system using 2D hands and body skeleton data,” IEEE Access, vol. 9, pp. 59612–59627, 2021.

[50] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, “A framework for hand gesture recognition based on accelerometer and EMG sensors,” IEEE Trans. Syst. Man, Cybern. Part ASystems Humans, vol. 41, no. 6, pp. 1064–1076, Nov. 2011.

[51] V. E. Kosmidou and L. J. Hadjileontiadis, “Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data,” IEEE Trans. Biomed. Eng., vol. 56, no. 12, pp. 2879–2890, Dec. 2009.

[52] U. Côté-Allard et al., “Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 27, no. 4, pp. 760–771, Apr. 2019.

[53] C. Chansri and J. Srinonchat, “Hand Gesture Recognition for Thai Sign Language in Complex Background Using Fusion of Depth and Color Video,” Procedia Comput. Sci., vol. 86, pp. 257–260, Jan. 2016.

[54] D. Kelly, J. Mc Donald, and C. Markham, “Continuous recognition of motion based gestures in sign language,” 2009 IEEE 12th Int. Conf. Comput. Vis. Work. ICCV Work. 2009, pp. 1073–1080, 2009.

[55] D. Kelly, J. McDonald, and C. Markham, “A person independent system for recognition of hand postures used in sign language,” Pattern Recognit. Lett., vol. 31, no. 11, pp. 1359–1368, Aug. 2010.

[56] L. Ming, T. W.C., and T. Chiang, “A feature covariance matrix with serial particle filter for isolated sign language recognition,” Expert Syst. with Appl. An Int. J., vol. 54, pp. 208–218, Jul. 2016.

[57] H. Wang et al., “Sparse Observation (SO) Alignment for Sign Language Recognition,” Neurocomputing, vol. 175, no. PartA, pp. 674–685, Jan. 2016.

[58] K. Murakami and H. Taguchi, “Gesture recognition using recurrent neural networks,” Conf. Hum. Factors Comput. Syst. - Proc., pp. 237–242, 1991.

[59] J. Yang, J. Yuan, and Y. Li, “Parsing 3D motion trajectory for gesture recognition,” J. Vis. Commun. Image Represent., vol. 38, pp. 627–640, Jul. 2016.

[60] M. M. Zaki and S. I. Shaheen, “Sign language recognition using a combination of new vision based features,” Pattern Recognit. Lett., vol. 32, no. 4, pp. 572–577, Mar. 2011.

[61] J. Pu, W. Zhou, J. Zhang, and H. Li, “Sign language recognition based on trajectory modeling with HMMs,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9516, pp. 686–697, 2016.

[62] T. Li et al., “Recognition System for Home-Service-Related Sign Language Using Entropy-Based K-Means Algorithm and ABC-Based HMM,” IEEE Trans. Syst. Man, Cybern. Syst., vol. 46, no. 1, pp. 150–162, Jan. 2016.

[63] L. Quesada, G. López, and L. Guerrero, “Improving Deaf People Accessibility and Communication Through Automatic Sign Language Recognition Using Novel Technologies,” Adv. Intell. Syst. Comput., vol. 500, pp. 497–507, 2016.

[64] S. Aly and S. Mohammed, “Arabic Sign Language Recognition Using Spatio-Temporal Local Binary Patterns and Support Vector Machine,” Commun. Comput. Inf. Sci., vol. 488, pp. 36–45, Nov. 2014.

[65] M. R. Abid, E. M. Petriu, and E. Amjadian, “Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar,” IEEE Trans. Instrum. Meas., vol. 64, no. 3, pp. 596–605, Mar. 2015.

[66] A. Mohanty, S. S. Rambhatla, and R. R. Sahay, “Deep Gesture: Static Hand Gesture Recognition Using CNN,” Adv. Intell. Syst. Comput., vol. 460 AISC, pp. 449–461, 2017.

[67] B. Hu and J. Wang, “Deep learning based hand gesture recognition and UAV flight controls,” ICAC 2018 - 2018 24th IEEE Int. Conf. Autom. Comput. Improv. Product. through Autom. Comput., Sep. 2018.

[68] J. Huang, W. Zhou, H. Li, and W. Li, “Sign Language Recognition using 3D convolutional neural networks,” Proc. - IEEE Int. Conf. Multimed. Expo, vol. 2015-Augus, Aug. 2015.

[69] L. Pigou et al., “Beyond Temporal Pooling,” Int. J. Comput. Vis., vol. 126, no. 2–4, pp. 430–439, Apr. 2018.

[70] O. Köpüklü, N. Köse, and G. Rigoll, “Motion Fused Frames: Data Level Fusion Strategy for Hand Gesture Recognition,” IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2018-June, pp. 2184–2192, Apr. 2018.

[71] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, “Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition,” Pattern Recognit., vol. 76, pp. 80–94, Apr. 2018.

[72] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, “Hand gesture recognition with 3D convolutional neural networks,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2015.

[73] L. Zhang et al., “End-to-End Automatic Pronunciation Error Detection Based on Improved Hybrid CTC/Attention Architecture,” Sensors, vol. 20, no. 7, p. 1809, 2020.

[74] S. Escalera et al., “Multi-modal gesture recognition challenge 2013: Dataset and results,” in ICMI 2013 - Proceedings of the 2013 ACM International Conference on Multimodal Interaction, 2013.

[75] X. Chen and K. Gao, “DenseImage Network: Video Spatial-Temporal Evolution Encoding and Understanding,” May 2018.

[76] S. Aly and W. Aly, “DeepArSLR: A Novel Signer-Independent Deep Learning Framework for Isolated Arabic Sign Language Gestures Recognition,” IEEE Access, vol. 8, pp. 83199–83212, 2020.

[77] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, and M. S. Hossain, “Hand gesture recognition using 3D-CNN model,” IEEE Consum. Electron. Mag., vol. 9, no. 1, pp. 95–101, 2019.

[78] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” Dec. 2014.

[79] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, and M. A. Mekhtiche, “Hand Gesture Recognition for Sign Language Using 3DCNN,” IEEE Access, vol. 8, pp. 79491–79509, 2020.

[80] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, “A complete KALDI recipe for building Arabic speech recognition systems,” in 2014 IEEE spoken language technology workshop (SLT), 2014, pp. 525–529.

[81] M. Alsulaiman, G. Muhammad, M. A. Bencherif, A. Mahmood, and Z. Ali, “KSU rich Arabic speech database,” Inf., vol. 16, no. 6 B, pp. 4231–4253, 2013.

[82] S. Khurana and A. Ali, “QCRI advanced transcription system (QATS) for the Arabic multi-dialect broadcast media recognition: MGB-2 challenge,” in 2016 IEEE Spoken Language Technology Workshop (SLT), 2016, pp. 292–298.

[83] N. Zerari, S. Abdelhamid, H. Bouzgou, and C. Raymond, “Bidirectional deep architecture for Arabic speech recognition,” Open Comput. Sci., vol. 9, no. 1, pp. 92–102, 2019.

[84] B. Dendani, H. Bahi, and T. Sari, “Speech Enhancement Based on Deep AutoEncoder for Remote Arabic Speech Recognition,” in International Conference on Image and Signal Processing, 2020, pp. 221–229.

[85] A. Hannun et al., “Deep speech: Scaling up end-to-end speech recognition,” arXiv Prepr. arXiv1412.5567, 2014.

[86] V. Pratap et al., “wav2letter++: The Fastest Open-source Speech Recognition System,” CoRR, vol. abs/1812.0, 2018.

[87] M. Ravanelli, T. Parcollet, and Y. Bengio, “The PyTorch-Kaldi Speech Recognition Toolkit,” in In Proc. of ICASSP, 2019.

[88] O. Kuchaiev et al., “Mixed-Precision Training for NLP and Speech Recognition with OpenSeq2Seq.” 2018.

[89] H. Inaguma et al., “ESPnet-ST: All-in-One Speech Translation Toolkit,” arXiv Prepr. arXiv2004.10234, 2020.

[90] Aliwy, A. H., & Ahmed, A. A. (2021). Development of arabic sign language dictionary using 3D avatar technologies. Indonesian Journal of Electrical Engineering and Computer Science, 21(1), 609-616..

[91] Halawani, S. M., & Zaitun, A. B. (2012). An avatar based translation system from Arabic speech to Arabic sign language for deaf people. International Journal of Information Science and Education, 2(1), 13-20.

[92] M. M. El-Gayyar, A. S. Ibrahim, and M. E. Wahed, “Translation from Arabic speech to Arabic Sign Language based on cloud computing,” Egypt. Informatics J., vol. 17, no. 3, pp. 295–303, Nov. 2016.

[93] M. Brour and A. Benabbou, “ATLASLang MTS 1: Arabic Text Language into Arabic Sign Language Machine Translation System,” Procedia Comput. Sci., vol. 148, pp. 236–245, Jan. 2019.

[94] K. M. Bouzoubaa, Y. Souteh, and K. Bouzoubaa, “SAFAR platform and its morphological layer Interacting MAS View project LMF Lexicons View project SAFAR platform and its morphological layer,” Elev. Conf. Lang. Eng. ESOLEC, pp. 14–15, 2011.

[95] A. Boudlal et al., (2010). Alkhalil morpho sys1: A morphosyntactic analysis system for arabic texts. In International Arab conference on information technology (pp. 1-6). New York, NY: Elsevier Science Inc.

[96] Alobaidy, M. A., & Sundus, K. E. (2020). Application for Iraqi sign language translation on Android system. *International Journal of Electrical and Computer Engineering*, *10*(5), 5227.

[97] Y. Ren et al., “FastSpeech 2: Fast and High-Quality End-to-End Text to Speech,” in International Conference on Learning Representations, 2020.

[98] M. Alsulaiman, G. Muhammad, M. A. Bencherif, A. Mahmood, and Z. Ali, “King Saud University Arabic Speech Database,” Linguistic Data Consortium, 2014. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2014S02. [Accessed: 09-Jun-2022].

[99] T. A. Mesallam et al., “Development of the arabic voice pathology database and its evaluation by using speech features and machine learning algorithms,” J. Healthc. Eng., vol. 2017, 2017.

[100] H. Altaheri, M. Alsulaiman, G. Muhammad, S. U. Amin, M. Bencherif, and M. Mekhtiche, “Date fruit dataset for intelligent harvesting,” Data Br., vol. 26, p. 104514, Oct. 2019.

[101] H. Altaheri, M. Alsulaiman, M. Faisal, and G. Muhammed, “Date Fruit Dataset for Automated Harvesting and Visual Yield Estimation,” IEEE DataPort, 2019. [Online]. Available: http://dx.doi.org/10.21227/x46j-sk98.

[102] P. Dreuw, C. Neidle, V. Athitsos, S. Sclaroff, and H. Ney, “Benchmark databases for video-based automatic sign language recognition,” in Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08), 2008.

[103] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, “Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures,” IEEE Trans. Multimed., vol. 21, no. 1, pp. 234–245, 2018.

[104] “The United Arabic Sign Language Dictionary for the Deaf.” [Online]. Available: https://zho.gov.ae/en/UAESignLanguageDictionary/Pages/default.aspx. [Accessed: 09-Jun-2022].

[105] "الجمعية السعودية للاعاقة السمعية." https://shi.org.sa/ (accessed Aug. 28, 2021).

[106] M. Al-Hammadi et al., "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," IEEE Access, vol. 8, pp. 192527–192542, 2020.

[107] M. Al-Hammadi et al., "Spatial Attention-Based 3D Graph Convolutional Neural Network for Sign Language Recognition," Sensors, vol. 22, no. 12, p. 4558, 2022.

[108] E. P. Ijjina and K. M. Chalavadi, "Human action recognition using genetic algorithms and convolutional neural networks," 2016.

[109] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-scale video classification with convolutional neural networks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 1725–1732, Sep. 2014.

[110] R. Hou, C. Chen, and M. Shah, "An End-to-end 3D Convolutional Neural Network for Action Detection and Segmentation in Videos," Nov. 2017.

[111] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 1, 2001.

[112] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in Proceedings of the AAAI conference on artificial intelligence, 2018, vol. 32, no. 1.

[113] M. De Coster, M. Van Herreweghe, and J. Dambre, "Isolated sign recognition from rgb video using pose flow and self-attention," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3441–3450.

[114] C. C. de Amorim, D. Macêdo, and C. Zanchettin, "Spatial-temporal graph convolutional networks for sign language recognition," in Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings 28, 2019, pp. 646–657.

[115] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv Prepr. arXiv2010.11929, 2020.

[116] A. Vaswani et al., "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998–6008.

[117] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding,” arXiv Prepr. arXiv2102.05095, vol. 2, no. 3, p. 4, 2021.

[118] “Modern Standard Arabic Pronunciation Lexicon – ALT Website.” https://alt.qcri.org/resources/msa-dictionary/ (accessed Sep. 12, 2021).

[119] “Marvelous Designer.” https://www.marvelousdesigner.com/ (accessed Aug. 28, 2021).

[120] “Maya Software | Get Prices & Buy Official Maya 2022 | Autodesk.” https://www.autodesk.com/products/maya/overview (accessed Aug. 28, 2021).

[121] “Real-time Rokoko 3D Character Animation in Reallusion iClone & Cartoon Animator.” https://www.rokoko.com/integrations/3d-character-animation-in-iclone (accessed Aug. 28, 2021).

[122] “Animation Pipeline to 3D World | iClone and 3DXchange.” https://www.reallusion.com/iclone/pipeline.html (accessed Aug. 28, 2021).

[123] “Unity Real-Time Development Platform | 3D, 2D VR & AR Engine.” https://unity.com/ (accessed Aug. 28, 2021).

12. Publications/Presentations

This section of the report should include a listing of any scholarly works that resulted from the project activities. These items include, but are not limited to, journal articles, books or book chapters, conference proceedings, magazine articles, patent awards, theses or dissertations, major published reports, technical manuals, workshop or short course booklets, and presentations at professional meetings.

Within the scope of the project, we already published the following papers:

Al-Hammadi, Muneer, et al., "Deep Learning-Based Approach for sign language Gesture Recognition with Efficient Hand Gesture Representation". IEEE Access 2020. https://doi.org/10.1109/access.2020.3032140.

Al-Hammadi, Muneer, et al. "Spatial attention-based 3d graph convolutional neural network for sign language recognition." Sensors 22.12 (2022): 4558.

Mohamed Mekhtiche, et al. "Speech/Text to Avatar translator for Saudi Sign Language", 9th IEEE International Conference on Applied System Innovation 2023 (IEEE ICASI 2023), Japan, 21-25 April 2023. (Accepted)

Mansour Alsulaiman, et al. "Facilitating the Communication with Deaf People: Building a Largest Saudi Sign Language Dataset", Journal of King Saud University - Computer and Information Sciences. (Submitted).

Hamdi Altaheri, et al. "KSU-ArSL: Arabic sign language dataset and validation using the latest deep convolutional neural networks", Heliyon. (Submitted)

We also conducted a very successful online workshop where the project team presented their work to the research community worldwide. The workshop titled a Saudi sign language translation companion system was attended by 245 people.

This work was part of the Ph. D thesis of researcher Muneer Alhammadi who was supervised by CO-PI Prof. Ghulam Muhammad and Prof. Abdulwadood Abdulwaheed.

13. Appendices

APPENDIX A

Algorithm 1: Hand Region Estimation

Input: The elbow coordinates (x_e, y_e)
The wrist coordinates (x_w, y_w)
The square region length $(length)$

Output: The top left coordinate of the square region (x_B, y_B)
The bottom-right coordinate of the square region (x_E, y_E)

$Calculate\ the\ region\ length = \sqrt{(x_w - x_e)^2 + (y_w - y_e)^2}$

If $abs(x_w - x_e) < \alpha\ and\ abs(y_w - y_e) < \alpha$
$x_B = x_w - length/2$
$y_B = y_w - length/2$
$x_E = x_w + length/2$
$y_E = y_w + length/2$
Else If $abs(x_w - x_e) < \alpha\ and\ abs(y_w - y_e) > \alpha$
$x_B = x_w - length/2$
$x_E = x_w + length/2$
If $y_w < y_e$
$y_B = y_w - (length - \varepsilon)$
$y_E = y_w + \varepsilon$
Else
$y_B = y_w - \varepsilon$
$y_E = y_w + (length - \varepsilon)$
End If
Else If $abs(y_w - y_e) < \alpha\ and\ abs(x_w - x_e) > \alpha$
$y_B = y_w - length/2$
$y_E = y_w + length/2$
If $x_w > x_e$
$x_B = x_w - \varepsilon$
$x_E = x_w + (length - \varepsilon)$
Else
$x_B = x_w - (length - \varepsilon)$
$x_E = x_w + \varepsilon$
End If
Else If $(y_e - y_w) > \alpha\ and\ (x_w - x_e) > \alpha$
$y_{mid} = round(y_w + (y_e - y_w)/2)$
$x_{mid} = round(x_w - (x_w - x_e)/2)$
$y_B = y_{mid} - length$
$y_E = y_{mid}$
$x_B = x_{mid}$
$x_E = x_{mid} + length$
Else If $(y_w - y_e) > \alpha\ and\ (x_e - x_w) > \alpha$
$y_{mid} = round(y_w - (y_w - y_e)/2)$

$x_{mid} = round(x_w + (x_e - x_w)/2)$
$y_B = y_{mid}$
$y_E = y_{mid} + length$
$x_B = x_{mid} - length$
$x_E = x_{mid}$

Else If $(y_e - y_w) > \alpha \; and \; (x_e - x_w) > \alpha$

$y_{mid} = round(y_w + (y_e - y_w)/2)$
$x_{mid} = round(x_w + (x_e - x_w)/2)$
$y_B = y_{mid} - length$
$y_E = y_{mid}$
$x_B = x_{mid} - length$
$x_E = x_{mid}$

Else If $(y_w - y_e) > \alpha \; and \; (x_w - x_e) > \alpha$

$y_{mid} = round(y_w - (y_w - y_e)/2$
$x_{mid} = round(x_w - (x_w - x_e)/2)$
$y_B = y_{mid}$
$y_E = y_{mid} + length$
$x_B = x_{mid}$
$x_E = x_{mid} + length$

Else:

Wrong input values