POLITECNICO DI TORINO

Master's Degree course in Biomedical Engineering

Virtual Reality application for Italian and Spanish sign language recognition

Supervisor:

Prof. Luca Ulrich

Candidate:

Alessia BISIO

Co-supervisor:

Prof. Enrique Yeguas Bolívar

Academic year 2022-2023

Abstract

Sign language is a structured form of hand gestures involving visual motions and signs. For the deaf and speech impaired community sign language serves as useful tool for daily interaction, enabling them to participate fully in society. Sign language is not widely understood or used by hearing people, creating a significant communication barrier between the deaf community and the rest of society.

In the academic context, regarding deaf students, both parents and teachers play an important role by assisting them since childhood in learning sign language.

In recent years, the use of Virtual Reality (VR) as a teaching tool for sign language has been on the rise. This technology has been shown to offer various benefits, including enhanced empathy, improved communication skills, a better understanding of diversity, and the creation of inclusive learning environments. Additionally, VR has been proven to enhance retention, memory, and attention during the sign language learning process, making it a valuable tool for both deaf and hearing individuals seeking to bridge the communication gap.

The objective of this thesis is the development of an application for Spanish and Italian sign language recognition that exploits the VR environment to create a database of signs quickly and easily. Additionally, it aims to provide an educational tool for learning sign language and a skill assessment mechanism, all integrated with artificial intelligence-based software (AI) to classify and recognize signs.

This effort is an integral component of the ISENSE project, financed by the European Union, which aims to support students with deafness throughout their academic journey by introducing various technological tools for teaching sign language to the hearing community within academic settings. Notably, the application has been entirely constructed using Unity 3D and is deployed on Oculus Meta Quest 2.

Contents

List of Figures

List of Tables

1 Introduction

Sign language is a visual-based form of communication that consists of stable hand movements and postures that convey concepts [1]. Nowadays according to Ethnologue [2], there are 157 living sign languages and 24 million first language speakers in the world.

Sign language is essential for daily interaction within the deaf and speech-impaired community, because the structured signs help to reinforce communication and convey information in everyday conversations. However, sign language is rare among hearing people, and fewer are capable of understand and use it: this poses an important communication barrier between the deaf community and the rest of society [3]. Sign language has seen technological advancements in recent years, with Virtual Reality (VR) emerging as a powerful learning tool.

In this context has been demonstrated that VR has several benefits including enhanced empathy, improved communication skills, increased understanding of diversity, and the creation of inclusive learning environments. [4]. By providing immersive experiences, VR allows students to step into the perspectives of others, fostering empathy and promoting a sense of belonging. Additionally, technological applications leveraging AI and computer vision have been developed to facilitate sign language learning. These applications utilize computer vision algorithms to track and analyze hand and body movements, enabling real-time recognition and interpretation of sign language gestures. Learners can receive instant feedback, engage in interactive lessons with virtual avatars or characters, and track their progress over time. Through the integration of VR, AI, and computer vision, these technological applications aim to make sign language learning more accessible, engaging, and inclusive, empowering individuals to communicate and connect with the deaf community.

This thesis presents a standalone VR and AI-based recognition system for the Spanish and Italian sign language learning.

The system concerns the realization of specific and progressive training paths to learn the sign language and communication skills needed to interact with deaf students. In particular we created a tool for the hearing community (students, professors, technicians, relatives and friends of deaf students), that could facilitate and increase the learning of sign language and communication skills needed to interact with deaf students in the academic context. The advantages and benefits of VR applications were exploited, in particular the hand tracking done by the four cameras integrated into the Oculus Meta Quest 2 [15] to carry out an on-the-fly sign recognition. In the application, there are different VR scenarios by which users can add new signs to the database, learn new signs and test the acquired skills.

The software has been developed under the training program on national sign language for social inclusion through Virtual and Augmented Reality in ISENSE, an European project that aims at implementing supporting services to assist students with deafness during their academic life, with a particular focus on the enrolment phase, to increase the accessibility of deaf students in the academic context.

During the development of this thesis we wrote a paper for the IEEE MetroX-RAINE congress that will take place in Milan at the end of October 2023. The title of the article is "Training program on sign language: social inclusion through Virtual Reality in ISENSE project".

In the upcoming sections, we will delve into the core topics of this thesis: sign language, virtual reality, and artificial intelligence. we will begin by exploring the history and geographical context of sign language, providing insights into its development and regional variations. Subsequently, we will explain the virtual reality technology, examining its fundamental principles and applications. Moreover, we will analyze the state of the art about virtual reality for social inclusion purposes, sign language recognition systems and machine learning algorithms applied to this field. Moving forward, the framework and primary objective of this thesis will be presented, outlining the methodologies employed for movement tracking and implementing artificial intelligence algorithms for the recognition of signs. Furthermore, we will share the obtained results from the application's testing phase and offer conclusive insights, followed by a discussion on potential future research directions.

2 Sign Language and technologies

2.1 Sign language: historical and geografical background

Sign language encompasses a rich historical background that has had a profound impact on the education and communication of deaf individuals. Interestingly, there is evidence to suggest that visual language predates the development of auditory language in human evolution. This implies that the ability to communicate through visual gestures, expressions, and movements may have emerged before spoken language evolved. Indigenous sign languages and ancient cultures across different parts of the world recognized the vital role of visual communication and incorporated it into their societies [5].

During the 17th and 18th centuries, sign language gained recognition as a complete language with its own unique syntax; educators and scholars began to realize that sign language provided a natural and effective means of communication for deaf individuals, allowing them to express their thoughts, feelings, and ideas in a comprehensive and structured manner. This recognition marked a significant milestone in the history of sign language, as it solidified its position as a legitimate language deserving of academic study and inclusion in educational settings.

According to Ethnologue [2], there are 157 living sign languages with ISO-639-3 codes, representing approximately 24 million first language speakers. Sign language is independent of spoken language and it can develop independently in different deaf communities, even if differences in spoken languages can contribute to the variation between sign languages, in finger spelling, grammar, sintax and lexicon.

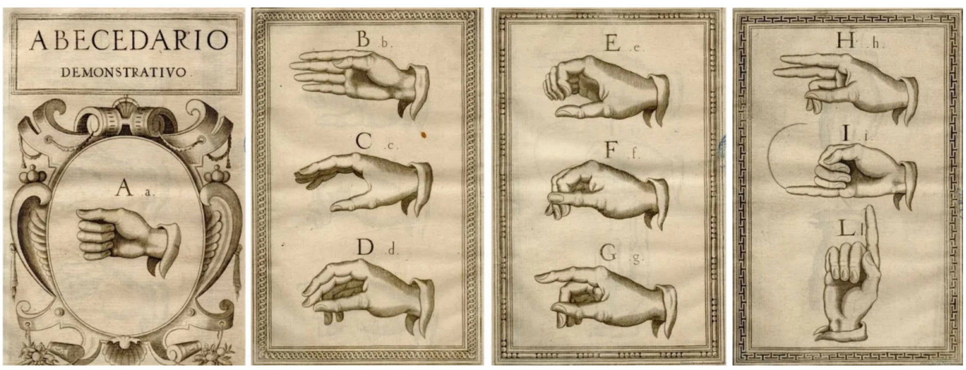

Figure 1: «Abecedario demonstrativo» de Reducción de las letras y arte para enseñar a hablar los mudos (1620), Spanish National Library, by Juan Pablo Bonet

This thesis project focuses specifically on Italian and Spanish sign languages: Italian sign language (Lingua dei Segni Italiana - LIS) was officially recognized as an official language in 2021; in Spain, each autonomous community has its own legislation pertaining to sign language, but it was recognized as an official language of the state (Lengua de Signos Espanola - LSE) in 2007 (Fig.1). These milestones underscore the growing recognition and significance of sign languages in embracing linguistic and cultural diversity.

Sign language, as said before, is essential for daily interaction within the deaf and speech-impaired community, because the structured signs help to reinforce communication and convey information in everyday conversations. Since few hearing people are able to understand and use sign language to communicate with the deaf, is very important to bridge this gap, facilitating smoother interactions, improving the inclusivity, indipendence and autonomy of deaf individuals. To do this, technologies can be a very useful tool, by providing real-time feedback, supporting interactive lessons, and enabling teachers and students to communicate effectively, enhancing the overall learning experience. They can be very useful for parents and teachers of deaf students too, because they play an important role by assisting them since childhood in learning sign language to communicate with others [6].

Some common examples of systems that can be used as a support tool for deaf people are online video platforms (such as YouTube or dedicated sign language learning websites), mobile applications, digital sign language dictionaries, gesture recognition systems, wearable devices, humanoid robots and Virtual and Augmented Reality.

2.2 Virtual Reality technology

Virtual reality (VR) refers to a computer-generated, interactive virtual environment that users can explore and interact with using specialized equipment such as head-mounted displays (HMDs). It can be categorized into non-immersive and immersive VR. Non-immersive VR uses screens surrounding the user to present virtual information, while immersive VR provides a fully immersive experience through wearable displays that track users' movements. Augmented reality (AR) overlays computer-generated imagery onto the real world, and mixed reality (XR) combines elements of both AR and VR.

The concept of VR originated in the 1960s with early prototypes, but it gained significant attention in the 2010s with the development of more accessible and consumer-friendly VR devices, such as Oculus. While initially popular in gaming, VR has expanded into various sectors, including education, training, simulations, and healthcare. However, challenges such as discomfort, cost, and technical limitations still exist and require further research and advancements to overcome [7]

The adoption of virtual and augmented reality (VAR) technology in higher education is still in its early stages, with STEM disciplines leading the way due to industry demands. VAR offers immersive learning experiences through augmented reality projections, virtual 3D realities, and interactive 360° videos. It enables students to explore physically restricted locations and interact with 3D models, fostering competencies such as spatial visualization, problem-solving, and critical thinking. However, successful adoption requires hands-on experience and the development of high-quality teaching content. Universities face challenges in investing in VAR infrastructure without a confirmed user base and ensuring the right technology selection and implementation.

Virtual reality (VR) technology has some limitations. Technological limitations include the lack of standardization, high hardware and software requirements, discomfort and potential health issues such as headaches and eye strain, as well as lag between user movements and visual display. Cybersickness, a form of motion sickness induced by VR, is another significant issue. It can occur due to the sensory conflict between visual perception and physical movement. Accessibility is also a concern, as the cost of VR headsets and VR-ready computers can be a barrier for many people. However, augmented reality (AR) and mixed reality (XR) applications on mobile devices offer a more accessible entry point to virtual experiences [8].

2.3 Virtual Reality for Social Inclusion

VR systems have emerged as promising tools for promoting social inclusion among various populations. With their immersive and interactive capabilities, VR applications offer unique opportunities to create inclusive environments and bridge social barriers. The VR for Good program [9] by Meta is an initiative aimed at utilizing virtual reality technology for positive social impact. The program offers resources, mentorship, and funding to participants, enabling them to leverage VR as a powerful storytelling medium. Through this initiative, VR experiences have been created to enhance education, support healthcare and mental well-being, raise awareness about important social topics, aid individuals with disabilities, preserve cultural heritage, and build virtual communities. In this way, users experience different perspectives and drive empathy.

In recent years, there has been a growing interest in leveraging VR technology to foster social inclusion, particularly among diverse groups such as neurodiverse individuals, marginalized communities and individuals with disabilities, that include hearing impaired people too. An interesting field of application of Virtual Reality is the treatment of Social Anxiety [10]. This study explores the effectiveness of Virtual Reality Exposure Therapy (VRET) as a therapeutic tool for Social Anxiety Disorder (SAD). Social anxiety is characterized by excessive fear of negative evaluation and rejection, and traditional therapies like cognitive-behavioral therapy (CBT) face challenges in replicating real social situations. VRET, by simulating these situations in virtual worlds, provides a unique opportunity for patients to confront and cope with their fears in a controlled environment. Virtual Reality Exposure Therapy (VRET) has shown promise in the treatment of Social Anxiety Disorder (SAD), while its effectiveness as a stand-alone treatment remains a subject of investigation. The study presented in [11] has as main purpose to create virtual reality (VR) games that cater to the diverse sensory needs and preferences of neurodiverse individuals, considering factors such as audio, visual, and haptic stimuli. The goal is to enhance the gaming experiences and inclusivity for neurodiverse children, allowing them to engage and interact with VR games in a way that accommodates their unique sensory profiles.

Another interesting application is the use of VR for speech-language pathology, focusing on two main types: non-immersive and immersive VR [12]. Non-immersive VR environments examples are Second Life®, were participants immersed themselves in virtual simulations of various scenarios, including social interactions, job interviews, conflict resolution, and making financial and social decisions, and EVA

Park, that provides a virtual world where individuals with aphasia and their therapists can interact, receive therapy, and enhance their communication skills through practice. On the other hand, immersive VR, such as CAVE system [13] presented in the cited review, has demonstrated greater engagement and potential for promoting communication skill development, particularly for individuals with autism. The goals of using VR in speech-language pathology are to enhance communication interventions, improve real-world interactions, and provide training opportunities for both individuals with communication disabilities and their communication partners.

In various social contexts (not only schools and universities, but also cinemas, theaters, museums, job places, public offices, supermarkets) individuals who are deaf or hard of hearing often face barriers that hinder their full participation and enjoyment. To enhance social inclusion of deaf people, many applications are developed for both young and adult targets.

An interesting example is the Toc-Tum mini-games [14], a novel medical expert system based on virtual reality, designed to make basic concepts of music accessible to the deaf culture. The main goals of the game are to provide an interactive and engaging platform for deaf children to learn about music and to validate the system's effectiveness through testing with the target audience and professionals in the field.

In [15] they discuss about the potential of VR as a tool to provide real-time captioning and sign language interpretation during theater shows, thereby improving the theater experience for the target audience.

Important aspects are also the ones that concern security of people with disabilities in general, and with hearing disability too. The work presented in [16] discusses the use of Immersive Virtual Reality and Serious Games (IVR SGs) as a training tool to enhance the preparedness of individuals, particularly deaf and hard-of-hearing (DHH) children that are not able to access audio information and warnings, for earthquake emergencies.

Then, many others works in this field focuses on application to teach sign language, the most useful and direct way to communicate with deaf people. For this reason in the next section I will present some related works that underline the importance of sign language detection and recognition exploiting different technologies.

2.4 Sign Language Detection

In the past years, most works related to sign language recognition did not exploit Virtual Reality (VR), but only simple acquisition systems and Artificial Intelligence (AI) based algorithms for sign classification.

2.4.1 Kinect and AI-based systems

An example of automatic sign language recognition is presented in [17]. With a dataset of 20 different Italian gestures captured using a Microsoft Kinect (Fig.2), the model comprises two Convolutional Neural Networks (CNNs) for hand and upper body feature extraction, followed by concatenation and classification using an Artificial Neural Network (ANN) with LCN and ReLU activation. In [18] the authors employed the Kinect sensor to capture depth data of 25 signs of German Sign Language, then they implemented a two-step classification approach: first, they used a depth-based feature extraction method to extract relevant features, then a k-nearest neighbors (k-NN) for gesture classification, evaluating the performance of their approach using cross-validation techniques.

(a) RGB (b) Depth map (c) User index (d) Skeleton

Figure 2: Example of kinect-based system: detection of human body using Microsoft Kinect and extraction of skeleton information.

2.4.2 VR-based systems

In the last years, instead, the use of VR for teaching purposes has increased, also for learning sign language. Recent studies confirmed that VR improves retention, memory, attention and motivation during learning, because the immersive nature of VR allows the interaction with the virtual world and the creation of a creative and engaging learning environment. Most of the works regarding sign language learning using VR exploit external acquisition systems to acquire videos of the signs and AI algorithms to perform the recognition in a VR environment. In [19] they combine the avatar's hand and body movements recorded using motion capture devices with a synchronized video of the face captured by a head-mounted camera. The avatar's movements are then reconstructed in the VR environment and used to teach sign language, but live sign recognition is not performed. A study on the recognition of American Sign Language (ASL) gestures in a VR environment using the Leap Motion controller with two IR cameras to track hand gestures in real-time is presented in [20]. A combination of Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) is used to classify signs. In this case, the different gestures correspond to the letters of the language.

Unlike the works presented so far, the VR-based recognition system object of this thesis is scalable in terms of the number of gestures required to define a specific sign, allowing for the recognition of both static signs, such as letters, and dynamic signs, such as words or phrases. The system achieves this scalability while utilizing only a single device, making it more accessible and cost-effective for users. Additionally, thanks to the integrated approach of the application, users have the capability to add new signs to the database of the system while "playing", it's all real-time.

2.5 Machine Learning algorithms for Sign Language recognition

There exist various Machine Learning (ML) algorithms suitable for hand gesture recognition in sign language, with several notable options for gesture classification including:

1. **Convolutional Neural Networks (CNN)**: well-suited for gesture recognition thanks to their hierarchical representation (from simple to more complex patterns in subsequent layers). This algorithm seek to optimize an objective function or loss function; by minimizing this function through iterative parameter updates, the CNN learns to make accurate predictions and extract relevant features from the input data [21].

2. **Support Vector Machine (SVM)**: a supervised learning algorithm that has good generalization capability (can handle complex decision boundaries) is effective in cases where the data is not linearly separable and is robust to noise and outliers [23, 25]. which main advantages for sign language recognition and classification are its good generalization capability (can handle complex decision boundaries), its effectiveness in high-dimensional spaces (gestures are often represented by a large number of features, such as joint angles or image descriptors: SVMs can effectively handle these high-dimensional feature spaces), robustness against overfitting, versatility in handling different data types, ability to work with limited training data, capability for non-linear classification using kernel functions, and interpretable results.

3. **k-Nearest Neighbour (kNN)**: it classifies objects based on their feature space and uses a supervised learning algorithm [23]. KNN is easy to understand and implement; it works well with small training sets and does not make strong assumptions about the underlying data distribution; if new sign gestures are added to the training ser, it doesn't require the retraining of the entire model, as it compares new instances with existing labeled data; kNN can handle noisy or imperfect data as it considers local patterns rather than global trends (outliers or mislabeled data points have less impact on the classification results).

4. **Decision Tree (DT)-based algorithms**: it can be a simple Decision Tree or a Random Forest, an ensemble learning method that combines multiple decision trees [23,24]. This model is particularly useful for this application because it offers an intuitive and easy-to-understand representation of the decision-making

process; it has fast training and prediction times compared to many other classification algorithms, thanks to the efficient splitting of feature values based on specific criteria; DTCs can also handle missing values, outliers and noisy data, making decisions based on majority voting; DTCs inherently perform feature selection by evaluating the importance of features in the classification process: important features are placed closer to the root of the tree, while less significant features are placed deeper in the tree. This feature selection capability helps in identifying relevant attributes for classification. Other advantages of a DT-based model are the ability to capture nonlinear relationships, versatility in handling different data types, and scalability to large datasets [26].

3 Materials and methods

3.1 Head-mounted device

The Oculus Quest 2 (Fig. 3), produced by Oculus Meta, is the chosen device for this sign language recognition application due to its advanced features and versatility. It's equipped with a four-camera internal tracking system strategically placed on the headset, allowing precise detection of hand and body movements.

Being a wireless and standalone device it does not require an external hardware (computer, cables or sensors), neither additional installation and setup. It's easy to use and allow higher mobility, enabling users to interact with the application naturally and without hindrance

Its high-resolution display and head movement tracking ensure clear and realistic representation of sign language gestures.

Figure 3: Oculus Meta Quest 2.

All information generated and saved within the application is securely stored in the internal storage of the Oculus headset. This data is readily accessible and can be directly retrieved from the internal storage whenever necessary, ensuring a seamless and efficient experience for users.

3.2 Conceptual implementation of the project: pseudocode

The methods employed in this study aim to accurately detect and interpret hand gestures in real-time. In this section, we will first present the pseudocode that summarizes all the steps that we followed for the conceptual implementation of the project.

The pseudocode presented in Algorithm 1 aims to perform real-time hand gesture recognition for sign language using multiple hand positions. The algorithm consists of different steps:

1. Obtain the number of poses involved in the sign gesture by capturing the sign pose for the specified type of gesture. Signs can be static, represented with a single pose, or dynamic, represented with the translation between two or more static poses.

2. Iterate over each pose in the poses list and collect the position data of finger bones for each hand.

3. Save the hand gestures captured into a defined structure.

4. Load the previously saved hand gestures from the database, considering the specific poses involved in the sign gesture.

Learn Mode:

5. Initialize the data structure to store the current hand positions recorded in real-time for each pose involved in that specific sign gesture.

6. Iterate over each pose in the poses list and collect the position data of finger bones for both hands and save the current hand position.

7. Calculate the hand features based on the collected hand positions for each pose of the current sign gesture reproduced.

8. If the current hand features match the saved hand features, validate the hand pose. Otherwise, consider it as not validated and give a feedback to the user.

Test Mode:

5. Calculate the hand features based on the collected hand positions for each pose involved in the sign gestures previously loaded.

6. Create a decision tree classifier using the loaded hand gestures as training data.

7. Initialize the data structure to store the current hand positions recorded in real-time for each pose involved in that specific sign gesture.

8. Iterate over each pose in the poses list and collect the position data of finger bones for both hands and save the current hand position.

9. Calculate the hand features based on the collected hand positions for each pose of the current sign gesture reproduced.

10. Initialize a dictionary to store the prediction results, associating each gesture with its confidence level.

11. Perform real-time classification for each saved gesture by utilizing the decision tree classifier and the calculated hand features.

12. Based on the predicted gesture, take appropriate actions or responses.

3.3 Overview of all methods employed

We will now provide a comprehensive overview of the methods used, including data collection, preprocessing, feature extraction, and the utilization of a decision tree algorithm for gesture classification. You can see a brief graphic resume in (Fig. 4).

Algorithm 1: Real-Time Hand Gesture Recognition for Sign Language (Multiple Positions)

```
posesList = CaptureSignPose(typeOfGesture)
foreach pose in posesList do
    foreach bone in fingerBonesList do
        handsDataToSave.Add(bone.position)
    end
end
SaveDataToJson(handsDataToSave, posesList, filePath)
listOfSavedSigns = LoadHandGestures(filePath, posesList)
savedHandsFeatures = CalculateFeatures(listOfSavedSigns, posesList)
if LearnMode then
    currentHandsData = new HandsData(posesList)
    foreach pose in posesList do
        foreach bone in fingerBonesList do
            currentHandsData.Add(bone.position)
        end
    end
    currentHandsFeatures = CalculateFeatures(currentHandsData, posesList)
    if currentHandsFeatures = savedHandsFeatures then
        ValidateHandPose;
    end
    else
        NotValidateHandPose;
    end
end
else if TestMode then
    decisionTree = CreateDecisionTree(savedHandsFeatures)
    currentHandsData = new HandsData(posesList)
    foreach pose in posesList do
        foreach bone in fingerBonesList do
            currentHandsData.Add(bone.position)
        end
    end
    currentHandsFeatures = CalculateFeatures(currentHandsData, posesList)
    prediction = new Dictionary<Gesture, Confidence>()
    foreach savedGesture in listOfSavedSigns do
        prediction[savedGesture] =
         decisionTree.Classify(currentHandsFeatures, savedGesture)
    end
    PerformAction(prediction)
end
```

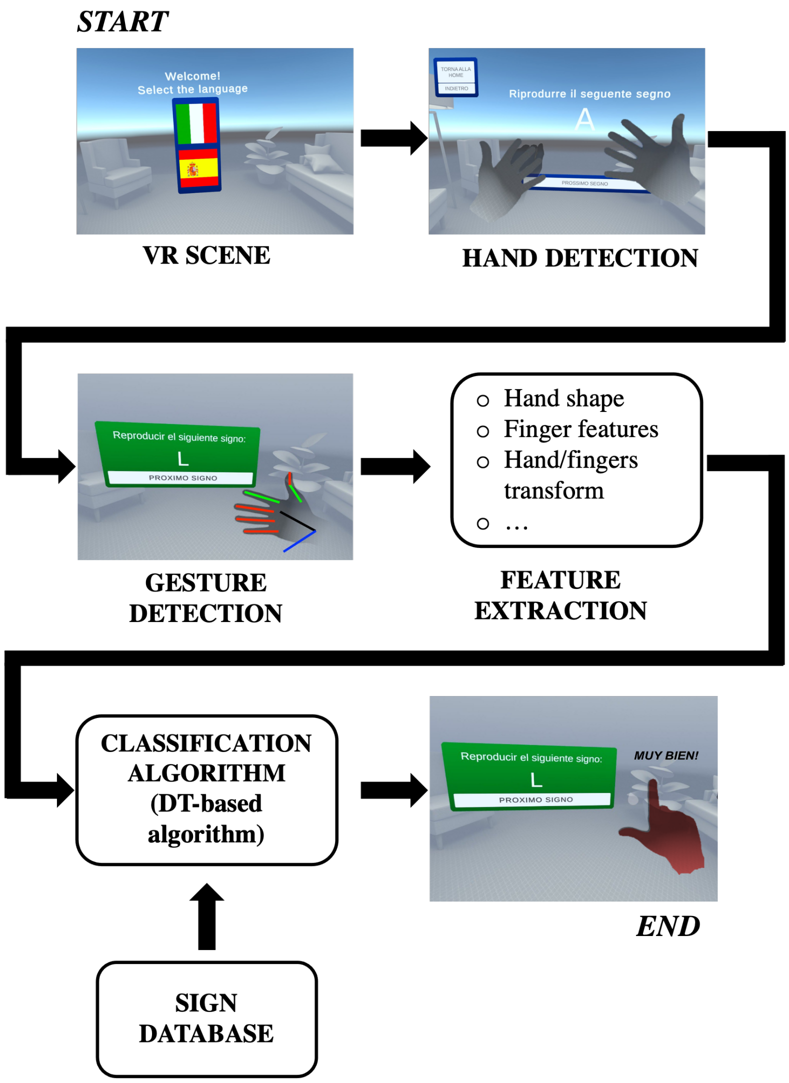

Figure 4: Overview of the sign detection system.

3.3.1 Hand Tracking: data collection

For the hand pose tracking, the Oculus device's hand tracking feature was leveraged using the Oculus Integration SDK package. Within this package, a 3D model of the human body, specifically the hand, is provided known as OVRSkeleton. This model encompasses all the various bones that constitute the hand (from the wrist to the fingers).

To generate and display the animated 3D model of hands, the OVR Mesh Renderer combines the data obtained from both OVR Skeleton and OVR Mesh. OVR Skeleton provides crucial information such as the bind pose, bone hierarchy, and capsule collider data. On the other hand, OVR Mesh loads a specified 3D asset from the Oculus runtime and presents it as a Unity Engine mesh.

By utilizing this integration, the system is able to accurately track and represent the hand poses, enabling users to interact with the virtual environment using their own hands instead of relying on Touch Pro or Touch controllers. This not only enhances the immersion and naturalness of the VR experience but also simplifies the user interface by eliminating the need for additional hardware.

To save hand pose data, a process known as serialization [27] has been implemented in C#. This means that the intricate details of hand movements and skeletal data are converted into a structured format that can be easily stored, transferred, or retrieved as needed. Serialization plays a pivotal role in ensuring that the dynamic and complex information related to hand poses can be efficiently preserved and utilized within virtual reality applications.

To save gestures, four nested structures are created. The innermost structure is used to store information about the ID, position, and rotation of each bone in every finger of the hand. The second structure is used to save the name of the gesture associated with that hand pose, the corresponding language, the hand's position and rotation relative to the base of the wrist, and the list of finger positions and rotations. The third structure is used to store the list of poses that define a single gesture, and the last structure is used to save all the gestures.

```
struct FingerBone
{
    public OVRSkeleton.BoneId BoneId;
    public Vector3 fingerBonePositions;
    public Quaternion fingerBoneRotations;
}

struct HandData
{
    public string Sign;
    public string Language;
    public OVRHand.Hand handedness;
    public Vector3 handPosition;
    public Quaternion handRotation;
    public List<FingerBone> fingerBone;
}
```

```
struct SerializableHandDataList
{
    public List<HandData> handDataList;
    public SerializableHandDataList(List<HandData> list)
    {
        handDataList = list;
    }
}

struct SerializableSignsDataList
{
    public Dictionary<string, List<HandData>> allSignsData;
    public SerializableSignsDataList(Dictionary<string,
        List<HandData>> data)
    {
        allSignsData = data;
    }
}
```

3.3.2 Dataset

To test the application, a custom dataset was created, trying to have a quite sign variation per language, with alphabet letters, words, and short sentences. The dataset specifically focuses on capturing static signs representing each letter of the alphabet, as well as two key pose signs for general concepts such as colors, measurements, emotions, characteristics, and more. Additionally, it incorporates three crucial pose signs for phrases, encompassing commonly used idiomatic expressions, greetings, questions, and popular phrases.

To ensure the dataset's accuracy and diversity, extensive efforts were made to compile a variety of sign examples. These examples were sourced from an array of videos, extensively referenced from the vast collection available at [28]. This methodology ensured that the dataset encompassed a wide range of sign variations, including variations in hand shape, movement and orientation.

To enrich the dataset comprehensively, the future plan includes recording each sign with contributions from multiple users, ideally aiming for a total of 10 instances for each sign. This approach acknowledges the significance of accommodating natural variations arising from diverse signing styles and hand sizes, ultimately resulting in a more inclusive representation of the sign language system.

However, it's important to note that, at present, due to time and resource constraints, the opportunity to collaborate with experts in sign language has not been realized. Nonetheless, this remains an important consideration for future dataset enhancements.

The following lines report an example of the data's structure saved.


```
{
  "Signs": [
    {
      "SignName": "Sign 1",
      "HandPoses": [
        {
          "Sign": "Sign 1",
          "Language": "italiano",
          "handedness": 0,
          "handPosition": {
            "x": 0.12247300148010254,
            "y": 1.054595947265625,
            "z": 0.38368916511535645
          },
          "handRotation": {
            "x": 0.5421946048736572,
            "y": -0.3408283591270447,
            "z": 0.565093994140625,
            "w": -0.5201248526573181
          },
          "fingerBone": [
            {
              "BoneId": 0,
              "fingerBonePositions": {
                "x": 0.12247300148010254,
                "y": 1.054595947265625,
                "z": 0.38368916511535645
```

```
  },
  "fingerBoneRotations": {
    "x": -0.565093994140625,
    "y": -0.5201248526573181,
    "z": 0.5421946048736572,
    "w": 0.3408283591270447
  }
},
{
  "BoneId": 1,
  "fingerBonePositions": {
    "x": 0.12247300148010254,
    "y": 1.054595947265625,
    "z": 0.38368916511535645
  },
  "fingerBoneRotations": {
    "x": -0.565093994140625,
    "y": -0.5201248526573181,
    "z": 0.5421946048736572,
    "w": 0.3408283591270447
  }
},
{
  "BoneId": 2,
  "fingerBonePositions": {
    "x": 0.10742449015378952,
    "y": 1.0742690563201904,
    "z": 0.38753214478492737
  },
  "fingerBoneRotations": {
    "x": -0.1114569902420044,
    "y": -0.36529383063316345,
    "z": 0.8845242261886597,
    "w": 0.26787084341049194
  }
},
{
```

```
            ...
          }
        ]
      }
    ]
  },
  {
    "SignName": "Sign 2",
    "HandPoses": [
      {
        "Sign": "Sign 2",
        "Language": "italiano",
        "handedness": 0,
        "handPosition": {
          "x": 0.10343188047409058,
          "y": 1.0496784448623657,
          "z": 0.3553816080093384
        },
        "handRotation": {
          "x": 0.02808574214577675,
          "y": -0.03404608741402626,
          "z": 0.7097265124320984,
          "w": -0.7030933499336243
        },
        "fingerBone": [
          {
            "BoneId": 0,
            "fingerBonePositions": {
              "x": 0.10343188047409058,
              "y": 1.0496784448623657,
              "z": 0.3553816080093384
            },
            "fingerBoneRotations": {
              "x": -0.7097265124320984,
              "y": -0.7030933499336243,
              "z": 0.02808574214577675,
              "w": 0.03404608741402626
```

```
              }
            },
            {
              ...
            }
          ]
        }
      ]
    }
  ]
}
```


3.3.3 Feature Extraction

For each pose a set of 22 features were calculated, including the euclidean distances among all the key joints of the hand skeleton's bones, represented in (Fig. 5): distance tip-base of a finger, distance between tips of different fingers and distance tip-intermediate or tip-proximal. The base point considered in this case is the wrist.

Figure 5: Key joints of the hand skeleton used to calculate distances.

The calculated features calculated are then saved in a .csv file, one for each sign. Distances are expressed in meters (see Table 1 and Fig. 6).

Sign Name	A	B	C
distThumb_Tip_Base	0.1060115	0.1064453	0.1075459
distIndex_Tip_Base	0.02585451	0.08317278	0.07978263
distMiddle_Tip_Base	0.03570991	0.09374061	0.09015281
distRing_Tip_Base	0.03219215	0.08861843	0.08596274
distPinky_Tip_Base	0.03196969	0.115004	0.1138574
distThumb_Tip_Meta	0.08596187	0.08625998	0.08877967
distThumb_Tip_Prox	0.05658751	0.05746192	0.05682051
distIndex_Tip_Inter	0.03793553	0.04585715	0.04589488
distMiddle_Tip_Inter	0.04385133	0.0515397	0.05177172
distRing_Tip_Inter	0.04191453	0.05018418	0.05010041
distPinky_Tip_Prox	0.02354414	0.07163912	0.07078069
distPinky_Tip_Inter	0.03466668	0.04160359	0.04154292
distThumb_Index_Tip	0.04551007	0.06303235	0.07990404
distThumb_Middle_Tip	0.05836258	0.07722645	0.09629998
distThumb_Ring_Tip	0.07062545	0.07742485	0.1049438
distThumb_Pinky_Tip	0.08108458	0.07757652	0.1082926
distIndex_Middle_Tip	0.0161744	0.01535455	0.02310537
distIndex_Ring_Tip	0.0312202	0.02573554	0.03999139
distIndex_Pinky_Tip	0.04501838	0.05138646	0.06483769
distMiddle_Ring_Tip	0.01529026	0.017779	0.01987761
distMiddle_Pinky_Tip	0.02946576	0.04895451	0.04996758
distRing_Pinky_Tip	0.01441556	0.03184233	0.0322117

Table 1: Example of numerical features calculated on poses of three signs.

Figure 6: Example of euclidean distances computed on three signs. "A" - distances between the tip of one finger and the other key points of the same finger; "B" - distances between the tip of each finger and the base of the wrist of the hand; "C" - distance between the tips of different fingers.

3.3.4 Real-time hand matching: learn mode

Our application employs a unique and engaging learning method. When users enter the learning mode, the application randomly selects a sign from our sign language dataset. It then presents the corresponding hand gesture, which users are tasked with replicating in real-time. To enable this process, we've implemented a crucial method called "displayHand." This method retrieves the sign's pose from the stored JSON data and seamlessly applies it to the game object equipped with the Hand Geometry component, essentially rendering the virtual hand within the immersive virtual reality environment.

The matching mechanism then compares the position and rotation of each finger of the user's real-time hand (represented in black in Fig. 7) with that of the saved reference hand (shown in green), allowing for a correct match if the discrepancies fall within a specified threshold. This approach ensures a dynamic and interactive sign language learning experience within the application and an immediate feedback.

Figure 7: Real time hand matching: current hand pose has to match the saved hand pose and the system validates the sign.

3.3.5 Machine Learning algorithms for Sign Language classification: test mode

Based on the literature review presented in the introduction chapter of this thesis, the algorithm implemented and used for testing the recognition system is a Decision Tree Classifier (DTC).

To do this, the dataset was firtsly divided into two sets: a training set and a test set with a 70%-30% distribution.

The ultimate goal of the experimentation is to achieve the optimal decision tree model and configuration for real-time prediction of a given sign (whether it is an alphabet letter, word, or sentence) in a virtual reality (VR) setting. Prior to the classification process, a careful selection of the best parameters has been conducted, including the criterion, maximum depth, minimum samples split, and minimum samples leaf.

To assess the performance of the final decision tree model, it will be compared across different outputs based on accuracy and F1-Score rate metrics. To ensure a fair comparison, the experimentation results for each output and configuration will be obtained using a 10-fold cross-validation approach. This technique involves dividing the dataset into ten subsets, training and evaluating the model ten times using different combinations of training and testing data, and then averaging the results to provide a comprehensive evaluation of the model's effectiveness.

In the construction of a DTC there are different factors and criteria taken into account. In our work, in particular, we set the stopping criteria to determine when to stop growing the tree and the splitting criterion for selecting the best attribute to split the data at each node. The algorithm uses information gain or entropy as the criterion for selecting the best attribute: the one with the lowest entropy and the highest information gain is selected as it provides the most significant reduction in entropy and maximizes the information gained about the class labels.

The algorithm was implemented and trained offline using Python library *scikit learn*; the resulting tree was then saved as a json model.

In a future upgrade of this project, the idea is to implement a k-Nearest Neighbour (kNN) algorithm to, in order to compare results and understand which method is more suitable for this kind of application.

3.4 Implementation of the project in Unity

To implement the application, the Unity 3D (2021.3.4f1) software was used. The steps taken for implementation are as follows:

1. The building settings were adjusted, particularly the platform and operating system to build for. Since the Oculus Quest 2 device was chosen, the OS used is Android. It's worth noting that with minor modifications, the project can be compiled for other devices as well (the project is scalable in this regard), such as Windows, Mac, Linux, WebGL, etc. Some of the parameters set include Texture Compression (ASCT) and Minimum API Level (Android 10.0 - API level 29).

2. The Oculus Integration package [reference to the download site] was imported, which contains key components for creating immersive VR experiences, including input controls, head and hand movement tracking, stereoscopic graphics, and other Oculus-specific features.

3. The scenes of the virtual environment were created. The application that I developed presents 3 different scenarios, through which the user can perform different actions. In (Fig. 8) you can see the workflow of the scenes.

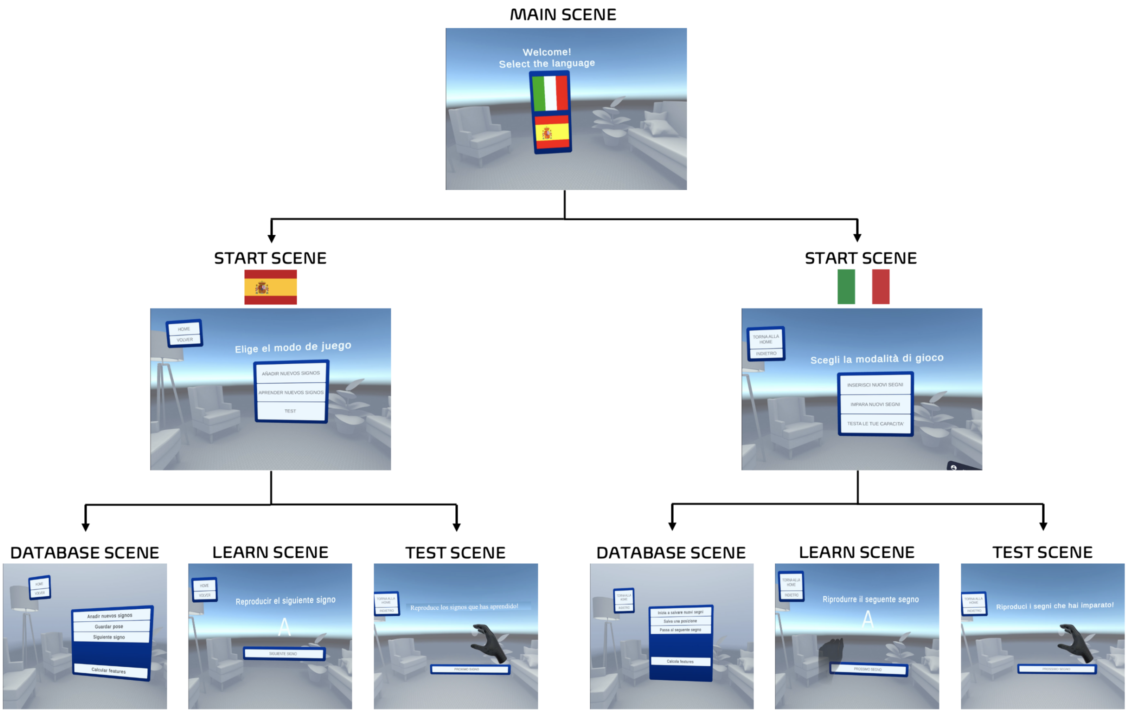

Figure 8: Workflow of the scenes into the application.

(a) **Main scene**. The initial scene where the user can choose the language (Italian or Spanish).

(b) **Start scene**. An introductory scene where the user can select one of the game modalities previously described. There is a spanish and an italian version of all the scenes from now on.

(c) **Database scene**. The user can add new signs to the database. This function is intended to be used by expert sign language speakers. Both static and dynamic signs are saved.

(d) **Learn scene**. Users can reproduce the sign and learn new signs by putting their hands to fit inside the virtual hands of the avatar rendered into the scene.

(e) **Test scene**. Finally, users can test the learned skills, by trying to sign a word from the dictionary without the help of the avatar. The application will give feedback according to the quality of the reproduced sign.

The main elements used to create these scenes were Game Objects and Canvas with Button UI to allow the user to call functions and trigger actions in the virtual environment, as well as simple TextMeshPRO screens for debugging the executed actions.

4. The C# codes were written to implement all the methods described before and then tested.

4 Results and Discussion

This chapter unveils the outcomes of our study, focusing on two primary aspects. Firstly, we present the results of the sign recognition tests, which scrutinize the accuracy and proficiency of our virtual reality application in recognizing sign language gestures. Subsequently, we delve into the qualitative assessments conducted to evaluate the usability of the virtual reality application designed for sign language learning by individuals. These evaluations encompassed inquiries aimed at understanding users' willingness to utilize the application, their interest in its features, and the underlying reasons driving their enthusiasm.

4.1 Signs database

The development of the virtual reality application for sign language recognition in Italian and Spanish has yielded promising results thus far.

Firstly, a comprehensive dataset of signs was captured directly using the four cameras of the VR device, specifically the Oculus Meta Quest 2, which was utilized for the development and testing of the project. This dataset serves as a valuable resource for sign language recognition research and further advancements in the field.

To create the dataset we focused initially on the letters of the alphabet. We have compiled a dataset of 20 signs, each with a single pose representing a specific letter. We obtained a json file with letters A, B, C, D, E, F, G, I, K, L, M, N, O, P, Q, T, U, V, W, Y.

Additionally, We have devised a method to accommodate signs with multiple poses, for example dynamic letters of the alphabet, single words or short sentences and expressions, although this feature has not been fully implemented at this stage. Instead, We plan to collaborate with deaf or sign language speakers in the future, particularly when distributing the application to universities, to ensure the accuracy and completeness of signs with two or more poses. These achievements mark a significant milestone in this project's development, laying the groundwork for more extensive sign language recognition capabilities in the future.

In this context, the creation of the sign language dataset represents a significant achievement, distinguishing itself by its direct implementation through virtual reality, setting it apart from the majority of existing literature, which predominantly relies on alternative devices such as Kinect.

4.2 Signs recognition: evaluation offline and developed methods

In the context of our application, another notable result we achieved is the recognition of signs in learning mode. The learning mode employs a real-time matching algorithm that performs a match between the virtual hand's current position and the saved pose, as elucidated in the preceding sections, conducted within the Oculus headset. This approach enables learners to receive immediate feedback on their sign language gestures, promoting an interactive and engaging learning experience. We conducted trials of our application in learn mode using simple signs, the same ones that we successfully incorporated into our dataset as previously mentioned. These trials yielded promising outcomes, demonstrating the effectiveness of our system in recognizing and matching these signs. However, it's worth noting that the matching process often required more time, particularly for signs in which certain fingers and key points of the hand were obscured or overlapped. We observed that the Oculus Quest 2's tracking system struggled to accurately associate these points, resulting in longer recognition times and distance calculation. Despite these challenges, our application exhibited robust performance.

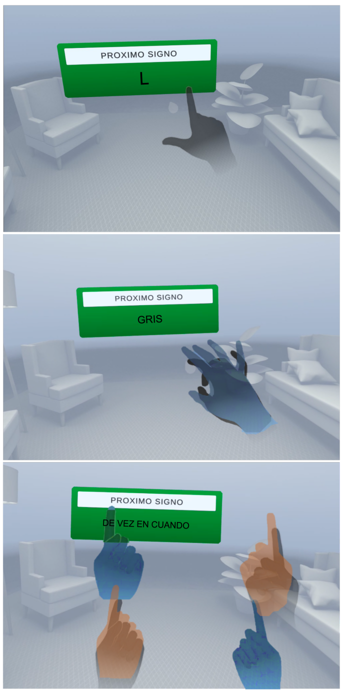

Figure 9: Sign language gestures categorized: alphabet signs, words and sentences.

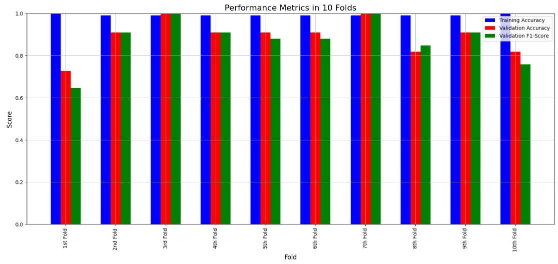

Figure 10: Performance metrics for alphabet signs.

On the other hand, in the test mode, we undertake the evaluation of sign recognition using a machine learning algorithm, specifically the Decision Tree Classifier. Due to challenges posed by outdated or non-functional libraries included inside Unity environment, we opted for an offline evaluation approach. To assess the performance of this recognition system, we employed RStudio, a powerful integrated development environment (IDE) for R, a programming language and environment used for statistical computing and data analysis. RStudio provides a user-friendly interface and tools for data visualization, making it an essential tool for evaluating the effectiveness of our sign language recognition algorithm.

To bring this functionality into the virtual reality environment, the people involved in the ISENSE European project have decided to collaborate with a specialized company with expertise in virtual reality development. This partnership will ensure a seamless integration of the recognition process within the VR environment: this part of the work will be done next year.

Figs. 10 show the classification results for the alphabet signs stored in the dataset. The performance metrics considered are Accuracy as a measure of the overall model precision in correctly classifying instances and F1-Score as a single measure of the model's performance by balancing both false positives and false negatives. Results are summarized in Table 2.

Dataset	**Metric**	**Mean Train**	**Mean Test**
Alphabet	Accuracy	0.991	0.890
	F1-Score	0.991	0.873

Table 2: Classification Results

The analysis of sign classifications highlights a notable distinction in performance when considering alphabet signs exclusively. In this context, the model demonstrates significantly higher Accuracy and F1-Score, underscoring its exceptional precision and overall effectiveness in classifying alphabet signs, due to their simplicity and static gesture.

Remarkably, both classifications, including the broader set of sign variations, consistently exhibit strong performance across the training and validation sets, surpassing the 85% threshold for both Accuracy and F1-Score for test set and 91% for train set. These findings underscore the model's ability to generalize effectively to unseen data and maintain consistent performance, showcasing its reliability across diverse sign variations.

4.3 Inclusive usage evaluation: results of the questionnaire

In the context of this thesis, during our pre-development assessment of the proposed application, we recognized the importance of obtaining insights from a diverse demographic, including individuals of varying ages, educational backgrounds, familiarity with sign language, and experience with virtual reality (VR). To gain a deeper understanding of opinions, expectations, doubts, and suggestions regarding the application we were developing, we designed a questionnaire that what submitted to a small selected population of people who belong to the deaf community, or have contacts with deaf people. The questionnaire was thoughtfully crafted by our project team, aiming to keep it simple with questions focused on key areas of interest and potential concerns related to the application's usability.

Participation in the questionnaire was entirely voluntary and anonymous, with respondents' input playing a crucial role in shaping the future development of the application to better cater to the needs and preferences of the target user community. In this section, we present the results of the questionnaire.

To provide context, we begin with a brief overview of the demographic information collected from the 31 people that filled out the questionnaire (see Fig.11).

<table>
<tr><td rowspan="2">Age</td><td>14-18</td><td>19-25</td><td>26-30</td><td>31-40</td><td>41+</td></tr>
<tr><td>0</td><td>21</td><td>3</td><td>3</td><td>4</td></tr>
<tr><td>Nationality</td><td colspan="5">Italian</td></tr>
<tr><td rowspan="2">Knowledge of Italian Sign Language (LIS)</td><td colspan="2">Yes</td><td colspan="3">No</td></tr>
<tr><td colspan="2">9</td><td colspan="3">21</td></tr>
<tr><td rowspan="2">Age at which they learned LIS</td><td>0-6</td><td colspan="2">7-12</td><td colspan="2">13+</td></tr>
<tr><td>3</td><td colspan="2">0</td><td colspan="2">6</td></tr>
<tr><td rowspan="2">Experience with VR</td><td>Yes, for fun with friends</td><td>Yes, for study and/or work purposes</td><td>Yes, to try inclusive applications</td><td colspan="2">No</td></tr>
<tr><td>7</td><td>2</td><td>3</td><td colspan="2">19</td></tr>
</table>

Figure 11: Demographic overview of the population involved in the study.

The questionnaire primarily consisted of two key inquiries: firstly, it sought to gauge respondents' interest in using the application and the reasons behind their choice, and secondly, it explored the potential impact of implementing such an application in secondary schools or universities. Specifically, we examined whether it could stimulate students to learn sign language and improve the social inclusion of deaf students in educational settings.

The response to the first question is summarized in Fig.12. Histogram in figure a) represents the response statistics to the question regarding what might capture the attention and drive the usage of a virtual reality application for learning sign language. Figure b) provides insights into the negative aspects, capturing the doubts, reservations, and concerns that individuals may harbor when considering the use of the application.

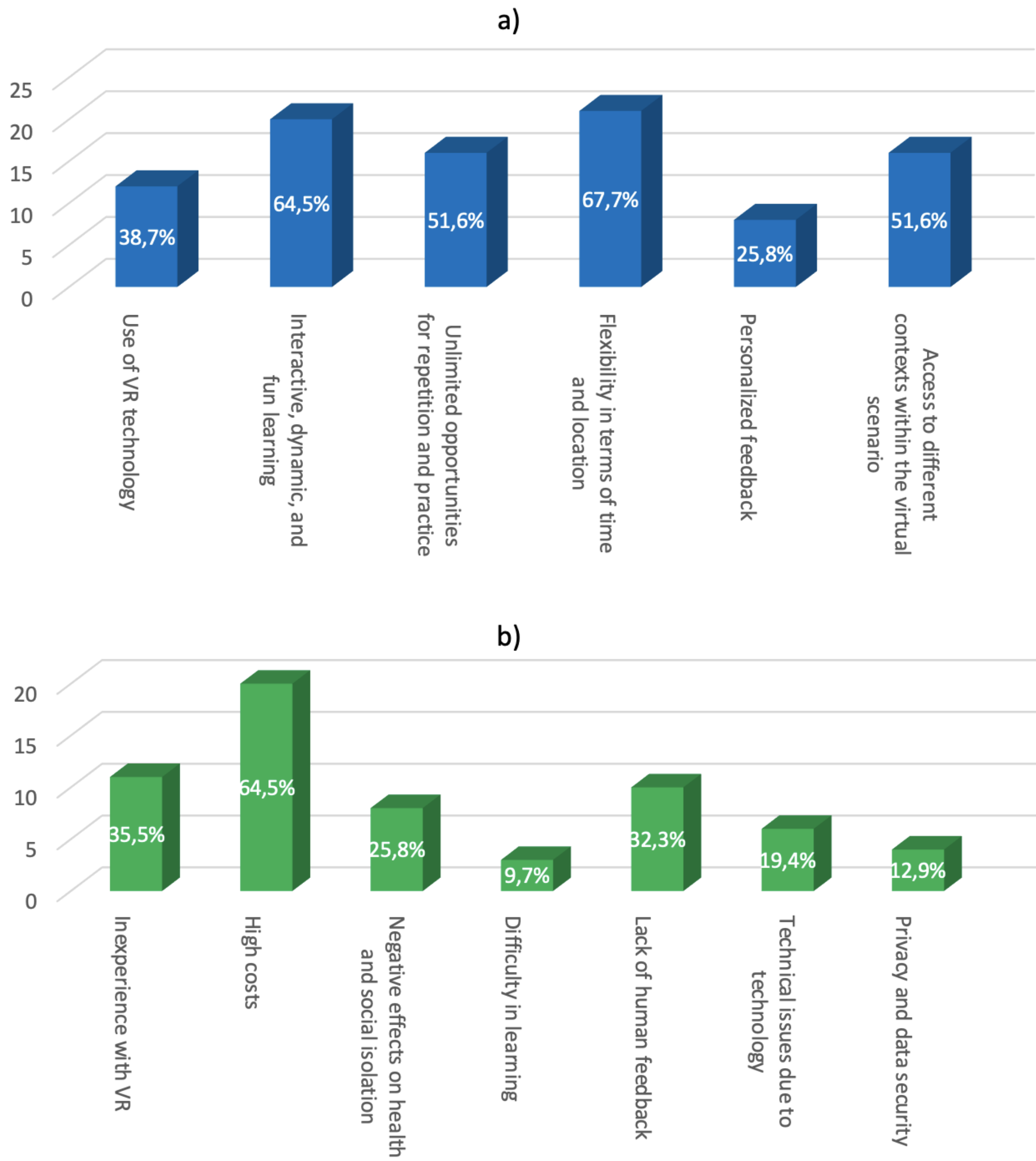

Figure 12: Results of questionnaire. a) Responses about aspects that might drive the usage of the VR application, b) Responses about doubts and negative aspects of the application.

The aspect that interests people the most in using this type of application is the flexibility in terms of time and space, precisely because a virtual reality headset can be taken anywhere, allowing for great flexibility in learning sign language without the need for a confined space. Furthermore, another response selected by more than 64% of respondents to the questionnaire is the interactivity and dynamism of the proposed application. This means that people appreciate the possibility of active engagement and the variety of experiences offered by the application, making the learning process more engaging and effective.

What is less convincing, however, are primarily the high costs associated with the technological equipment, the inexperience with VR headsets, and the lack of human feedback, which is an ethical topic that merits substantial discussion.

The objective of the ISENSE project is the social inclusion of deaf people into the academic environment, so we asked people what about the idea of insert this kind of application into that specific context. Regarding the distribution of the application in schools and universities, over 80% of respondents to the questionnaire have a highly positive view of using virtual reality for learning sign language. They see it as a valuable technology applied to this specific field, offering both practical utility and the potential to stimulate curiosity in comparison to traditional classroom instruction. The main reasons supporting these findings are depicted in the Fig.13 Undoubtedly, the use of VR headsets is the primary reason why people would recommend the use of this application for teaching sign language in educational settings.

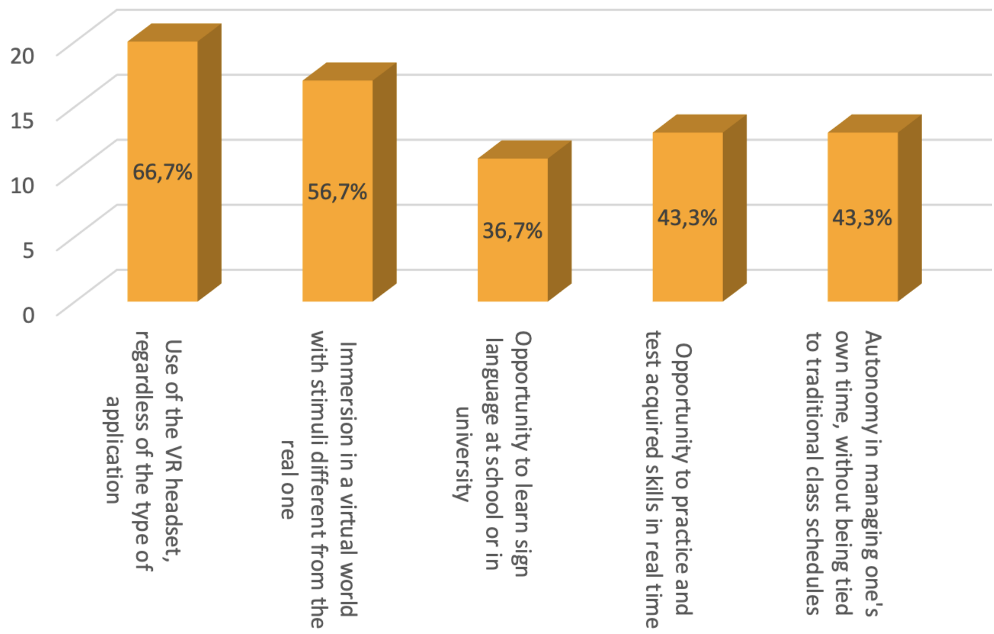

Figure 13: Results of questionnaire: opinion of respondents about distribution of the application into schools and universities.

In addition to exploring people opinion about distributing the application into schools and universities, respondents expressed diverse motivations for contributing to the creation of a sign language database in a VR application. Their primary reasons included a commitment to support the deaf community professionally, the importance of raising awareness and promoting inclusivity, a desire to bridge communication gaps, and a belief in the convenience and innovation of using technology to make sign language more accessible. Many also emphasized the civic duty and the stimulating aspect of building something useful for the community. In summary, these motivations collectively revolved around the idea of leveraging technology and shared knowledge to enrich society and facilitate the learning and understanding of sign language.

5 Conclusion and Future Works

In conclusion, the developed sign language recognition algorithm has shown promising performance in accurately recognizing simple signs, by utilizing the distances between joints of the skeleton's bones as features. This achievement highlights the potential of incorporating VR technology and AI in sign language education, providing an immersive and interactive learning experience.

However, it is important to acknowledge certain limitations and challenges encountered during the implementation of the algorithm. One such difficulty is the reliance on the Oculus device for accurate hand tracking. The algorithm requires a clear and unobstructed view of the user's hand, imposing restrictions on the testing environment (in terms of light and obstacles). Further optimizations and advancements are needed to ensure robust performance in various real-world scenarios.

Moving forward, several areas can be explored to enhance the functionality and applicability of the sign language recognition application. One potential avenue is the implementation of a mirror effect, allowing the application to recognize the hands of another person facing the user or the user's own hands while touching his/her face. This extension would facilitate interactive communication and learning between sign language users and non-users.

Another important aspect for future development is the continual expansion of the sign language database within the application.

As we look forward to the future development of this application, there are several key enhancements to be done. First and foremost, we're eager to integrate an in-app machine learning algorithm, a critical step in advancing the real-time sign language recognition capabilities of the application. It's important to note that the C# programming language, which we have used for the application, is not well-suited for this purpose due to the presence of outdated or non-functional libraries. Therefore, we need to find a more suitable programming language or framework to implement real-time machine learning within the application.

Simultaneously, we plan to elevate the immersive experience by modifying the default background in our virtual reality environment, providing users with a more personalized and engaging learning space. Until now we just used the background that came integrated in the Oculus Integration SDK package. Furthermore, we aim to empower users by allowing manual input of sign names through a virtual keyboard within the VR scene, enhancing their control and customization options. These future works signify our commitment to continuous improvement, ensuring that the application remains at the forefront of accessible and effective sign language learning tools.

6 Acknowledgements

These results are framed in ISENSE: Innovative Supporting sErvices for uNiversity Students with dEafness project funded by the European Union within the Erasmus+ programme (call 2022), Key Action 2: KA220 Cooperation Partnerships for higher education. AGREEMENT n. 2022-1-IT02-KA220-HED-000089554. This article has been funded with support from the European Commission.

References

[1] Schlesinger, I. M., & Namir, L. (2014). Sign Language of the Deaf: Psychological, Linguistic, and Sociological Perspectives. Academic Press.

[2] the world. (n.d.). Ethnologue (Free All). https://www.ethnologue.com/

[3] Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics, 10(1), 131–153. https://doi.org/10.1007/s13042-017-0705-5

[4] De Luca, V., Gatto, C., Liaci, S., Corchia, L., Chiarello, S., Faggiano, F., Sumerano, G., & De Paolis, L. T. (2023). Virtual Reality and Spatial Augmented Reality for Social Inclusion: The "Includiamoci" Project. Information, 14(1), 38. https://doi.org/10.3390/info14010038

[5] Ruben, R. J. (2005). Sign language: Its history and contribution to the understanding of the biological nature of language. Acta Oto-laryngologica, 125(5), 464–467. https://doi.org/10.1080/00016480510026287

[6] Naglot, D., & Kulkarni, M. (2016). Real time sign language recognition using the leap motion controller. https://doi.org/10.1109/inventive.2016.7830097

[7] Hamad, A., & Jia, B. (2022). How Virtual Reality Technology Has Changed Our Lives: An Overview of the Current and Potential Applications and Limitations. International Journal of Environmental Research and Public Health, 19(18), 11278. https://doi.org/10.3390/ijerph191811278

[8] Marks, B., & Thomas, J. (2021). Adoption of virtual reality technology in higher education: An evaluation of five teaching semesters in a purpose-designed laboratory. Education and Information Technologies, 27(1), 1287–1305. https://doi.org/10.1007/s10639-021-10653-6

[9] Meta, VR For Good, 2023, https://about.meta.com/community/vr-for-good/

[10] Emmelkamp, P. M. G., Meyerbröker, K., & Morina, N. (2020). Virtual reality therapy in social anxiety disorder. Current Psychiatry Reports, 22(7). https://doi.org/10.1007/s11920-020-01156-1

[11] Wasserman, B. A., Prate, D., Purnell, B., Muse, A., Abdo, K., Day, K., & Boyd, L. (2019). vrSensory: Designing Inclusive Virtual Games with Neurodiverse Children. https://doi.org/10.1145/3341215.3356277

[12] Bryant, L., Brunner, M., & Hemsley, B. (2019). A review of virtual reality technologies in the field of communication disability: implications for practice and research. Disability and Rehabilitation: Assistive Technology, 15(4), 365–372. https://doi.org/10.1080/17483107.2018.1549276

[13] Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V., & Hart, J. C. (1992). The CAVE: audio visual experience automatic virtual environment. Communications of the ACM, 35(6), 64–72. https://doi.org/10.1145/129888.129892

[14] Chaves, E. M., De Andrade Braga, P. B., Montenegro, Y. F. L., Rodrigues, V. E. S., Munguba, M. C., & Gupta, D. (2019). Toc-Tum mini-games: An educational game accessible for deaf culture based on virtual reality. Expert Systems, 38(1). https://doi.org/10.1111/exsy.12470

[15] Teófilo, M., Lourenço, A., Postal, J., & De Lucena, V. F. (2018). Exploring Virtual Reality to Enable Deaf or Hard of Hearing Accessibility in Live Theaters: A Case Study. In Lecture Notes in Computer Science (pp. 132–148). Springer Science+Business Media. https://doi.org/10.1007/978-3-319-92052-8-11

[16] Abdulhalim, I., Mutch, C., González, V. A., & Amor, R. (2021). Improving post-earthquake evacuation preparedness for deaf and hard of hearing children: A conceptual framework. International Journal of Disaster Risk Reduction, 62, 102360. https://doi.org/10.1016/j.ijdrr.2021.102360

[17] Pigou, L., Dieleman, S., Kindermans, P., & Schrauwen, B. (2014). Sign Language Recognition Using Convolutional Neural Networks. Springer EBooks, 572–578. https://doi.org/10.1007/978-3-319-16178-5-40

[18] Lang, S. C., Block, M., Rojas, R. (2012). Sign Language Recognition Using Kinect. In Lecture Notes in Computer Science (pp. 394–402). Springer Science+Business Media. https://doi.org/10.1007/978-3-642-29347-4-46

[19] Ghoul, O. E., & Othman, A. (2022). Virtual reality for educating Sign Language using signing avatar: The future of creative learning for deaf students. 2022 IEEE Global Engineering Education Conference (EDUCON). https://doi.org/10.1109/educon52537.2022.9766692

[20] Vaitkevičius, A., Taroza, M., Blažauskas, T., Damaševičius, R., Maskeliunas, R., & Woźniak, M. (2019). Recognition of American Sign Language Gestures in a Virtual Reality Using Leap Motion. Applied Sciences, 9(3), 445. https://doi.org/10.3390/app9030445

[21] Agha, R. a. A., Sefer, M. N., & Fattah, P. (2018). A comprehensive study on sign languages recognition systems using (SVM, KNN, CNN and ANN). https://doi.org/10.1145/3279996.3280024.

[22] Meta. (2023). Quest 2. Retrieved March 13, 2023, from https://www.meta.com/it/quest/products/quest-2/

[23] Bhushan, S., Alshehri, M., Keshta, I., Chakraverti, A. K., Rajpurohit, J., & Abugabah, A. (2022). An Experimental Analysis of Various Machine Learning Algorithms for Hand Gesture Recognition. Electronics, 11(6), 968. https://doi.org/10.3390/electronics11060968

[24] Devaraj, A., & Nair, A. K. (2020). Hand Gesture Signal Classification using Machine Learning. International Conference on Communication and Signal Processing. https://doi.org/10.1109/iccsp48568.2020.9182045

[25] Alnaim, N., Abbod, M. F., & Albar, A. (2019). Hand Gesture Recognition Using Convolutional Neural Network for People Who Have Experienced A Stroke. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). https://doi.org/10.1109/ismsit.2019.8932739

[26] Safavian, S., & Landgrebe, D. A. (1991). A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics, 21(3), 660–674. https://doi.org/10.1109/21.97458

[27] Gewarren. (2023). Serialization - .NET. Microsoft Learn. https://learn.microsoft.com/en-us/dotnet/standard/serialization/

[28] Spread the Sign. Thomas Lydell. European Sign Language Center. Retrieved June 2, 2023, from `https://www.spreadthesign.com/`

[29] https://www.researchgate.net/profile/Ashok-Sahoo-5/publication/262187093-Sign-language-recognition-State-of-the-art/links/02e7e53b246791ce2a000000/Sign-language-recognition-State-of-the-art.pdf