Cognitive Interaction with Robots

Final Project Report

Academic year: Fall 2022

Authors: Joaquim Marset, Ramon Mateo, Marta Gil

Contents

1 Introduction

The present report contains the final project of the subject *Cognitive Interaction with Robots*, which is about a desktop application to learn American Sign Language (ASL). The main point of this subject is human-robot interaction (HRI). Therefore, we had to do an AI-related project featuring an HRI, and we also had to evaluate that interaction.

We decided to develop an application to learn ASL as there are a lot of applications to learn languages like *Duolingo*, but there is almost nothing to learn sign language. And for those that might need to do it as fast as possible, it would be of great help.

The application applies the concept of gamification, a learning technique that transfers the mechanics of games to the educational-professional field to absorb better knowledge, improve some skills, or reward specific actions (which generates a positive reinforcement on the person), among many other objectives. This type of learning facilitates the internalization of knowledge in a more fun way, generating a positive experience for the user.

2 Target People

As we have said, there are not a lot of applications to learning ASL. We have *Ace ASL* [1], which presents a very similar concept to the one we wanted to develop. It is very powerful in terms of sign recognition, but the contents are a little limited. With this, we do not want to say that ours does it better, as we end up doing something that is not conceptually very difficult.

Sign language is a common tool for those learning it at early stages of their lives, especially when being kids. However, it is a complete paradigm change for the people whose main communication form is the spoken language. And it is a more stressful situation for those with some degenerative disease that causes the loss of hearing sense. Imagine having to deal with this situation of becoming deaf and not having learnt the most important tool to continue living as normally as possible. They have to deal with the fact that they will not be able to communicate as easily as talking is. Of course, they can still write things, but it is quite a hassle. Their closest people will probably need to learn it too.

Therefore, our goal is to create an application to facilitate this process of learning ASL for that group of people. And we will try to make it as funny and engaging as possible to maintain the user's motivation. However, we are not limited to that group. We are also considering those having some relative or friend who is deaf, and they want to communicate with them easily. And we even want to make it such that anyone that wants to learn ASL can.

We want to point out that we intend to focus more on young and adult people. We discarded kids because they do not suffer that much from this complete paradigm change. We also discarded the elderly, as they might have a hard time dealing with how the application works. Also, some might have a difficult time trying to learn new things.

3 American Sign Language

Each language and, sometimes, each country has its sign language, with different signs and grammar, which usually mirrors the grammar of the spoken language. It allows an easy adaption from spoken to sign language.

American Sign Language (ASL) is a complete, natural language with the same linguistic properties as spoken languages, with grammar that slightly differs from English. ASL is expressed by the movements of the hands, using either one or both.

ASL is not only used by deaf people and their environments, but others decide to learn it the same way they might want to learn any other language. Despite the wide use of ASL, there is no accurate count of their users. Reliable estimates for American ASL users range from 250,000 to 500,000 persons, including children of deaf adults and other hearing individuals.

[1] `https://www.signall.us/ace-asl`

Among all the possible signs we can find in ASL, we decided to focus on the following sets:

- Numbers between 0 and 9 (both included)
- All letters in the alphabet
- Common words like "book" or "chair"

We decided on these three sets and this particular order to teach them for various reasons. The first is because we start with static signs and then transition to dynamic ones. Static are those that only require a fixed position of the hand. Dynamic are those requiring some gesture. The second reason is that some signs require a single hand, and others two. The third reason is the complexity of the signs themselves.

The numbers between 0 and 9 are the only numbers having a static movement. Numbers greater or equal to 10 require some gesture with a single hand.

All the letters also require a single hand to perform the sign. However, among the 26 letters, "j" and "z" are dynamic and require movement. Also, we consider the letters more difficult, as multiple pairs can be confused, like "a" and "e". While we can use numbers to represent single-digit numbers, the letters are used as finger-spelling to spell out English words, especially proper names.

In Figure 1, we show how to perform both the numbers and the letters.

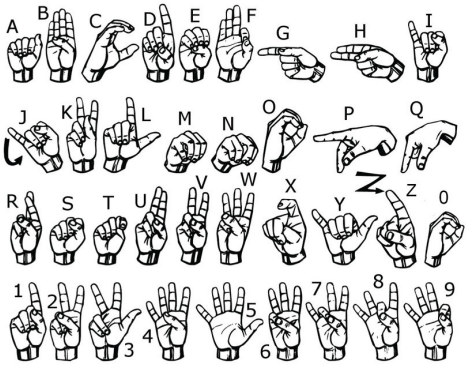


Figure 1: ASL letters and numbers between 0 and 9

Finally, we have a set of 100 common words, like "yes", "no", or "help". All the words require a dynamic gesture using both hands. For this third type, we do not present an example. We believe watching a video is better than looking at one image. Thus, we recommend going to *The Lifeprint Library* [2] and searching how to perform a particular sign. This web contains a database with all the signs in ASL, including the multiple versions each can have. Different versions of the same sign suppose different gestures to perform it.

4 Application

In this section, we will explain the different features of the application we developed. As mentioned, we developed a desktop application requiring only a computer and a webcam. Therefore, the interactions we consider are the user using the computer to navigate through the application and the webcam to record him performing signs. We developed it in *Kivy* [3], and in Appendix A we show multiple screenshots of the different menus and screens, and how the application looks.

We have been inspired by *Duolingo* and their learning system based on lessons. Each lesson features words that the user learns through a set of games. We have done the same, dividing the signs to learn into sets that we called lessons.

We created four lessons, each one featuring a different type of sign. In particular:

[2] `https://www.lifeprint.com/asl101/pages-layout/topics.htm`
[3] `https://kivy.org/doc/stable/gettingstarted/intro.html`

- Lesson 1: Numbers from 0 to 9 (both included)
- Lesson 2: Vowels
- Lesson 3: Consonants from "b" to "j" (both included)
- Lesson 4: Words "chair", "computer", "eat", "drink", "help", "book", "dog"

We created lessons consisting of a small number of signs to facilitate learning. For example, *Duolingo* considers a subset of four or five words. And we did something similar. The numbers are very easy to learn, so we put them together. However, the other lessons had at most seven signs. One could think seven is too much, but considering that most are static and unique (although confusing), we did not encounter any problems testing the application with real users.

Our lessons consist of four components. In particular:

- Theory: We present videos of how to perform each sign. In Figure 15, we show how this theory section looks.
- Games: We create a sequence of different games the user goes through to better learn the signs. We randomly generate the sequence, and each type of sign contains different games. The games we developed are as follow:
  - Memory: The user needs to match each sign with its corresponding video perofrming that sign.
  - Multi-choice question: The user needs to select the sign corresponding to the presented video performing that sign.
  - Inverted multi-choice question: The user needs to select the video corresponding to the presented sign.
  - Number gap-filling: The user selects the digits resulting from a simple multiplication.
  - Word gap-filling: The user needs to fill the gaps of a word with some masked letters. Of course, the masked letters are those featured in that lesson.

  In Figure 16, we show how all these games look.
- Practise: With the help of an AI-based sign detector, we allow the user to practise how to perform a sign. The user can select which to practise, and the application records the user doing it with the help of the webcam. The recorded video is sent to the corresponding detector to predict the sign. We will later explain, but we have one for each type of sign. In Figure 17, we show how the screen to practise signs looks.
- Exam: A 10-question exam merging games and detecting signs. We use the test to evaluate the learning performance in that lesson. We also use it to refute one of our null hypotheses we later explain. We featured both types of questions because the user needs to learn to perform the signs correctly.

5 Sign Detectors

Now, we will explain the sign detectors we used to detect each type of sign. We need to use different models as no model can predict all. It would be impossible. In particular, we implemented the one that detects numbers, and we searched on the Internet for the other two [6, 2].

5.1 Numbers Detector

We implemented this detector ourselves with the help of *Mediapipe Hands* [4]. The numbers we detect are static, so we only need to feed an image to the model to predict the number. Therefore, even though

[4] `https://google.github.io/mediapipe/solutions/hands.html`

we record a video, we pass each frame to the detector and generate a prediction. We compute the final prediction using a simple majority voting scheme.

From each frame (i.e. an RGB image), we use *Mediapipe Hands* to extract the 21 landmarks from each hand together with the angles between each pair of fingers. Each number in ASL only needs one hand, but we do not know which hand the user uses, so we compute it for both.

The model is a fully-connected network with two dense layers and the softmax layer. To train the model, we created a dataset by merging images from [7] and [5]. As you can see, the training dataset only contains one hand per image, but we can have two. For this reason, when training the model, our batch dimension represents different images. However, when testing our model with a frame from a video, we consider each hand an element in a batch, so our batch size can be at most 2. To work with this setting, we need to feed the frames separately to the model. However, given that we record a video for a very short time, we do not have that many frames, and the inference of a whole video is fast.

5.2 Letters Detector

MiCT-RANet [6] is the detector to recognize letters. It was the state of the art for real-time recognition of ASL finger-spelt video sequences. As we can see in their GitHub, they can detect all the letters in the ASL alphabet, including the dynamic "j" and "z". We decided to use this model because we could not find any other model capable of detecting the dynamic signs. We searched a lot, and in all cases, the models either discarded these two letters or detected them as static signs. In the latter, they simply trained the model with one frame of the dynamic movement.

This model was trained on the ChicagoFSWild dataset [9], which is the first collection of ASL finger-spelling naturally occurring in online videos. The data consists of short clips of finger-spelling sequences.

The model architecture combines optical flow [4], an attention mechanism [1], an LSTM cell, and a MiCT-ResNet. Optical flow determines the movement between adjacent frames. The attention mechanism weighs elements of a sequence according to its importance for the task.

MiCT-ResNet is the implementation the author of the sign detector developed when reproducing the work of [11]. The latter developed a hybrid model combining a 2D CNN backbone and 3D convolutions to perform human action recognition in videos, naming it Mixed Convolutional Tube Network, or MiCT-Net. Combining both types, they can avoid the training problems of 3D convolutions but still extract both spatial and temporal features to separate similar classes in action recognition.

With all these components, they created MiCT-RANet. With MiCT-ResNet, they generate feature maps from a set of frames from the input video sequence. Then, the attention mechanism reflects the importance of features at the different spatial locations to recognize finger-spellt sequences. The resulting attention weights are weighted using optical flow to indicate the regions in motion within the image. Finally, the LSTM cell generates a new representation by taking into account the information from the past, and a classification layer produces the output letter. In Figure 2, we present the architecture of MiCT-RANet.

[5] `https://www.kaggle.com/datasets/lexset/synthetic-asl-numbers`

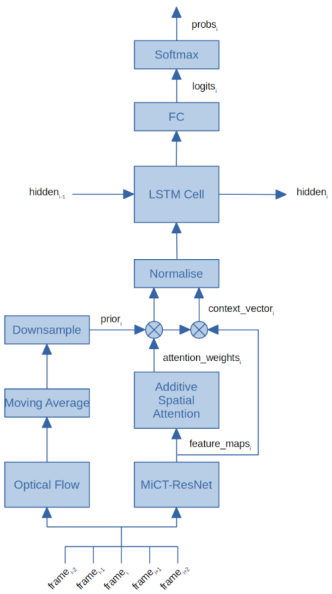


Figure 2: MiCT-RANet architecture (taken from [6])

5.3 Words Detector

Sign Pose-based Transformer (SPOTER) [2] is the sign detector to recognize words. In particular, we used the version trained with the WLSL dataset [5], featuring 100 common words like "eat", "computer", or "dog".

The model extracts 54 pose landmarks (i.e. head, body, and hands) from each video frame using *MediaPipe*. To improve generalization, they perform spatial augmentation like rotations or squeezing to the extracted landmarks' coordinates. Then, they normalize the coordinates to account for differences between videos, like the distance from the camera.

The architecture is a slightly modified transformer [10]. The input to the transformer encoder is a sequence of normalized body pose landmarks from different frames. The encoder works like the original transformer, adding positional encoding and going through a set of encoder layers consisting of multi-head self-attention and fully-connected sub-layers.

The decoder is the one presenting differences. The input is a query later decoded as the class. First, it goes through a multi-head projection module. It is a multi-head self-attention that only receives one element in the sequence. Therefore, the softmax is useless, and the result is a simple projection into the space of the value vectors of the self-attention module. Thus, the query and key vectors are not learned as there is no sequence. Like the usual multi-head self-attention, it uses multiple heads, so we have multiple parallel projections later concatenated. After this first multi-head projection, it follows the usual transformer decoder. Again, we have multiple decoder layers with the same structure.

We use the output of the decoder to generate the sign prediction. In Figure 3, we present the architecture of SPOTER, and we can see how it is almost the same as the original transformer.

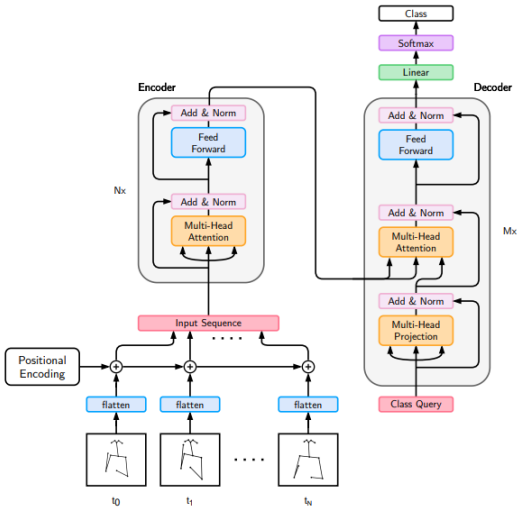


Figure 3: SPOTER architecture (taken from [2])

6 Hypotheses

We wanted to implement an application to facilitate the autonomous learning of ASL. For this reason, we wanted to demonstrate that learning through our application brings some benefits over other methods, like simply watching videos. In particular, we wanted to answer the following research questions:

- Does the learning progress depend on the learning methodology used?
- Does the learning easiness depend on the learning methodology used?

We understand progress as how well the user learns the different signs and ease of learning as how easy the learning becomes.

For each research question, we created a null and an alternative hypothesis. For the first question:

- Null hypothesis: Learning progress is independent of the learning methodology
- Alternative hypothesis: Learning progress is dependent on the learning methodology

For the second research question:

- Null hypothesis: Ease of learning is independent of the learning methodology
- Alternative hypothesis: Ease of learning is dependent on the learning methodology

To evaluate the human-robot interaction, we tried to refute the null hypothesis for each research question by using statistical tests.

7 Variables

After the research questions and the hypotheses, we defined the variables of our study. We have independent variables, the ones conditioning the study and the ones we can manipulate. As their name implies, they are not affected by other variables. Then we have the dependent variables, the study outcomes and the ones we measure. They are influenced by the independent variables. Considering our research questions, we defined the following set of variables:

- Independent:

  - Learning methodology: Using the lessons in our application or simply watching videos plus doing tests.

- Dependent
  - Learning progress: Quantitatively measured using different metrics stored in the application.
  - Learning easiness: Qualitatively measured using a questionnaire.

We compared how learning ASL differs between using our application or watching videos plus doing tests. We need the second methodology to perform tests because we extract some metrics from each exam to measure the learning progress. And we need those exams to be the same for both groups. We measured the second dependent variable using a questionnaire we later present.

Considering this independent variable, we divided our users into two groups, each using one of the two learning methodologies. And to make the comparisons even, we made them learn the same signs.

8 Experiment Procedure

As we have explained, we wanted to compare our application with learning ASL by watching videos. For this reason, considering our independent variable, we selected two groups of people to perform the experiments, each learning using each methodology.

Given that we wanted to facilitate later the process of hypothesis testing, we integrated everything in our application but blocked some parts for those learning videos. If we remember, a lesson has a theory section consisting of videos of performing the signs. And we imagined learning through videos as watching how to perform them. Also, we needed all the groups to perform the tests. For this reason, we handled the same application for both groups but disabling the features of games and practice in a lesson.

Once the users had the application, they learned in order the four lessons. That is, starting from lesson 1, lesson 2, and so on. Each was a little more difficult than the previous one. For example, all the numbers are static signs and very simple to remember because most of them are performed like anyone would think when they are told to make a number with their hand. This is why we placed them at the beginning. Second, we have vowels that are also static, but their signs are not as intuitive as the numbers. Third, we have consonants, introducing the first dynamic sign, "j". Finally, we have the words, all dynamic and involving both hands.

We also visualized each lesson to be done sequentially. That is, starting with the theory section, playing the games and practising signs, and finally, doing the exam. Each lesson is completed when the exam is finished. The exam is only done once, given that it is always the same and not randomized as the games. However, the games and the practice parts can be used anytime to better learn the signs. As a reminder, these two extra components are the differentiating point in our application to facilitate learning ASL.

Once the user completed a lesson, it moved to the next one until doing all four. Then, we considered the interaction finished, and they had to fill a satisfaction questionnaire rating their experience. Finally, they had to return us a file containing different metrics rating their performance. In the next section, we will explain the questionnaire and these metrics.

When handling the application to the users, we also attached an instructions manual explaining what they had to do, the different features the application had, and what they should expect. For example, for the group of people learning through videos, we explained all the games and how the sign detection system worked, given that the exam used both.

9 Evaluation Measurements

In this section, we will explain the different measurements we created to test the hypotheses using the results from the experimentation on the users. In particular, we decided to use quantitative data to test the first hypothesis and some qualitative data to test the second one.

9.1 Quantitative Measurements

For each lesson, the application saves multiple measurements from the different components. The relevant ones we used to test our hypothesis are the following:

- Time to complete a lesson
- Time to complete the exam
- Score of the exam

The exam score is between 0 and 10, and we computed it by penalizing each mistake. The sign detectors are imperfect, so we decided to have at most one question of this type and deduct less from each mistake. In particular, each error in a game-related question penalizes 0.25 and 0.15 in the sign detector-related ones.

As we mentioned, the user completes a lesson when he finishes the exam. Therefore, we counted the time spent on the different lesson components until completing the exam.

We used these three metrics to demonstrate how one can learn the signs better using our application than learning through videos. This is why we expected the first ones to achieve better results in the exams and maybe need less time to complete a lesson, as the games and the sign practice should allow for learning the signs better. However, those using videos do not have many features to try, so it depends on the time they consider spending watching videos. In any case, we also expected the second group to spend more time watching videos, as remembering the signs from only videos should be more difficult.

With these, we would end up with three metrics per lesson. However, we wanted to demonstrate the overall learning progress, not per lesson. For this reason, we computed a weighted average for each metric using the data from the four lessons. We have explained how we considered each lesson slightly more difficult than the previous one. Thus, the weights increase as we move to a later lesson. In particular, we set 0.20 for the first one, 0.25 for the second and third, and 0.3 for the fourth one.

We tested our hypothesis using the three metrics separately. We could have computed some composite metric like another average. However, the values each metric could take are different, needing to scale them. Also, we were not sure if all the metrics would be really useful, so we preferred to maintain them separately.

9.2 Qualitative Measurements

We evaluated the ease of learning using a satisfaction questionnaire we created. It contains three types of questions. First, demographic questions to know the user's profile. Second, questions to evaluate the ease of learning. Third, an open-ended question to receive feedback from the users.

The second group of questions is based on the NASA Task Load Index [6]. As the name implies, it is a questionnaire to assess the mental workload to perform a task. It features different questions, each related to a particular dimension, but all having the same type of answer. That is, all the questions are rate-scale questions. In particular:

- Mental demand: How mentally demanding was performing the task?
- Physical demand: How physically demanding was performing the task?
- Temporal demand: How hurried or rushed was the pace of the task?
- Performance: How successful were you in performing the task?
- Effort: How hard did you have to work to accomplish that level of performance?
- Frustration level: How insecure, discouraged, irritated, stressed, and annoyed were you when performing the task?

[6] `https://en.wikipedia.org/wiki/NASA-TLX`

We picked those dimensions that we thought could apply to our application. In particular, we discarded the physical one because the task did not have any physical-related activity. We also removed the temporal dimension because the users had all the time they needed to perform the task. And we were not interested in asking about the time they spent, as we were already recording it with the quantitative measurements.

Another change we made was trying to specify a little more the questions. In some cases, we thought the questions were quite broad, and we wanted to point them out in some particular direction. Also, this would avoid any confusion among the users trying to understand what we are trying to ask.

A third change we made was adding an extra question asking about the user's motivation to continue learning. We expected those learning using the whole application would be more motivated as they have more features to use and should learn easier than the other group.

Finally, we also changed the rating scale, as the original questions had a scale of 100, with steps of 5. Therefore, we changed to a typical rate scale of 1 to 5.

Below we present the complete questionnaire we handled to the users, specifying the type of answer each question has. We first have the demographic questions, followed by the questions to assess the ease of learning. Finally, we have an open-ended question:

- Age (Number)
- Gender (Set of options)
- Education level (Set of options)
- Are you familiar with some deaf people? (Yes/No)
- Did you already know some Sign Language? (Yes/No)
- How mentally demanding was the completion of the different lessons? (1 (Very Low) - 5 (Very high))
- How successful were you in learning the signs from the lessons? (1 (Very Unsuccessful) - 5 (Very Successful))
- How hard did you have to work (reviewing and/or practising signs) to reach that level of learning? (1 (Very Easy) - 5 (Very Hard))
- How insecure did you feel before performing the tests? (1 (Very Confidently) - 5 (Very Insecure))
- Do you feel motivated to continue learning as you did? (1 (No Motivation) - 5 (Very Motivated))
- What did you like, did not like, or missed in the application? (Open-ended)

With the second group, we can evaluate the ease of learning. However, we wanted a single metric to test our hypothesis, given that the five questions should be correlated and refer to the same concept. For this reason, we decided to use a mean as a simple composite score to generate a single metric.

We assumed the questions were correlated and should be related to what we wanted to evaluate. We cannot validate the latter, but at least we can try to find the correlated subset. For this reason, we decided to use Cronbach's Alpha [3], to get that subset. This metric takes a value between 0 and 1. Zero means no correlation between the items, and one perfect correlation. Usually, surpassing a threshold of 0.7 is considered acceptable. Thus, we searched for the subset giving the highest value surpassing that threshold.

We achieved these two conditions with the first and third questions of the second group. That is the question belonging to the mental demand and effort dimension. Of course, we still handled the complete questionnaire to the users. But when testing our hypothesis, we used those two.

10 Experiment Results

In this section, we explain the results of testing our application with different users, together with trying to reject the null hypotheses.

We handled the application to 14 people, 7 using the whole application and 7 learning through videos.

10.1 Data Distribution

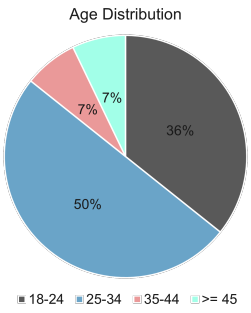
Let's start by looking at the distribution of the profiles of the different users who tested the application. In Figure 4, we show the distribution of the answers to each demographic question. In the question about the education level, we only show the options for which we have results, as we offered a few more options.

We can see how we mainly have people between 18 and 34 years old. The youngest was 23 years old, and the oldest was 31. Then we have one person belonging to each of the other two groups. The one in the last group was under 60. We have this age distribution because we mainly asked people we know, and we did want to focus on young and adult people, as we explained in section 2.

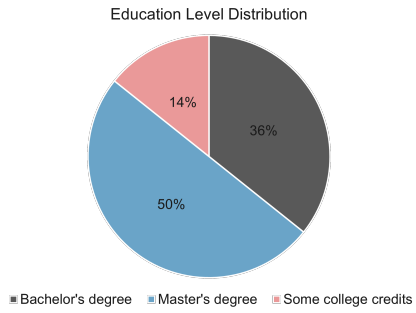
Related to the education level, we can see how we mainly have people with a bachelor's degree or a master's degree. We were not concerned about the education level, given that the signs we made them learn were not difficult. However, we also expected to find these results considering our target ages.

Third, we have the question related to knowing some deaf person. Only one participant answered "Yes". Therefore, most candidates simply wanted to learn ASL but not having any need to do it. This is our fault, as we mainly asked people we know, but we ended up with very short time to properly conduct the experimentation and search for people belonging to the other target groups we mentioned. In future work, we would like to find people belonging to those groups.

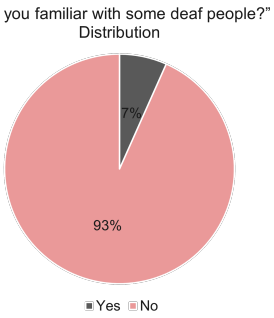
Finally, we have the question asking about already knowing some sign language. We asked this question because we mainly wanted people not to know any. If not, we believed they could have some advantage. Either because some signs might overlap or because they would be more used to sign language itself. As we can see, no participant already knew some sign language.



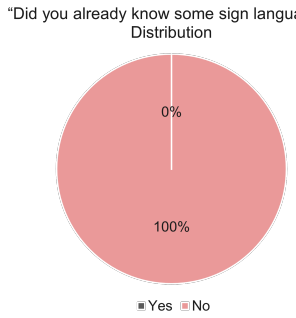
(a) Age



(b) Education Level



(c) "Are you familiar with some deaf people?"

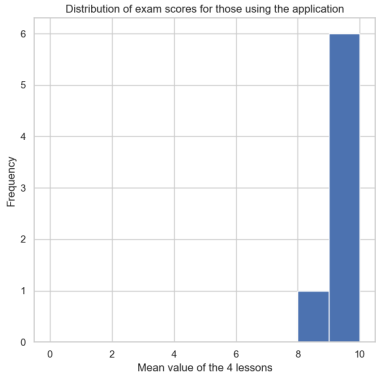


(d) "Did you already know some sign language?"

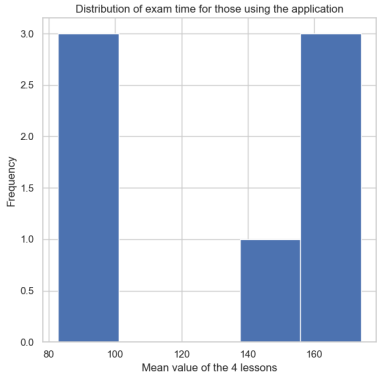
Figure 4: Distribution of the different questions asking for the user's profile

Let's now look at the distribution of the weighted averages we computed using the metrics the application saved for the different lessons. Figure 5 shows the distribution of the three explained metrics for those using the full application. Figure 6 shows the same for those learning through videos.

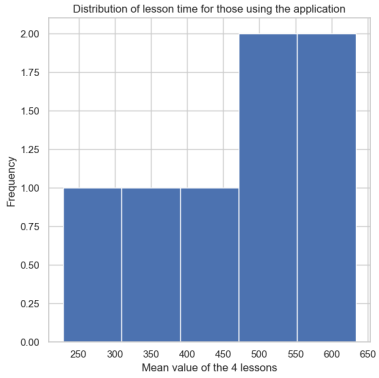
We want to mention that we used 11 bins to generate the distributions of the exam scores to account for all possible scores an exam could have but only used 5 for the time distributions to avoid one-element bins. In any case, the distribution remained almost the same.



(a) Mean Exam Score



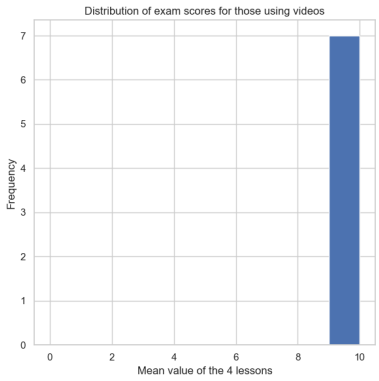
(b) Mean Exam Time



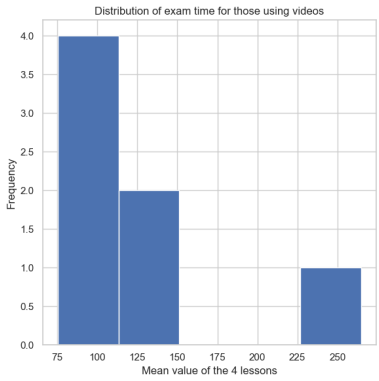
(c) Mean Lesson Time

Figure 5: Distribution of the three quantitative metrics for those using the full application

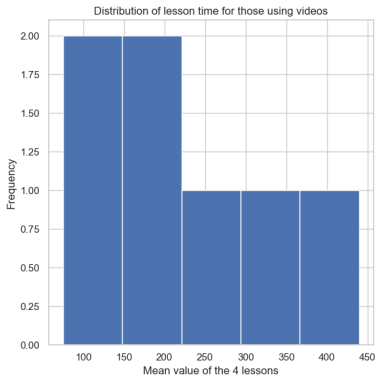
Looking at the results of those using the whole application, we can see how almost all except one achieved a mean exam score of at least nine. If we look at the average exam time, the users completed them in less than 3 minutes, meaning that maybe the exam was too easy for this group. Finally, if we look at the time to complete the lesson, we see more variability, but more users spend more time. However, we do not know if they spent it to learn the signs better or for simple curiosity to try the different features in depth. We cannot tell using these metrics.



(a) Mean Exam Score



(b) Mean Exam Time



(c) Mean Lesson Time

Figure 6: Distribution of the three quantitative metrics for those using videos

If we look at the results for those using videos, we can see how all the users got an average of 9 in the different exams, which seems better than the other group. However, considering we only have seven people per group, the difference is very small. Looking at the average time to complete the exams, we have similar times. We have one that spent quite a lot of time in the different exams compared to the others. But in general, we see that the users of this group spend less time in the exams than the other group. Finally, if we look at the average time to complete the lessons, we can see how the users of this group spent less time. They did not have that many features to try. However, this also means they did not need to spend more time watching the videos in the theory section. Therefore, maybe the signs they learned were a little simple.

Finally, let's look at the distribution of the mean score of the best subset of answers of the questionnaire. If we remember, we kept only two questions out of the initial five evaluating the ease of learning. In Figure 7, we show the distribution of the mean score of the two questions. In Figure 8, we show the same for the group learning through videos.

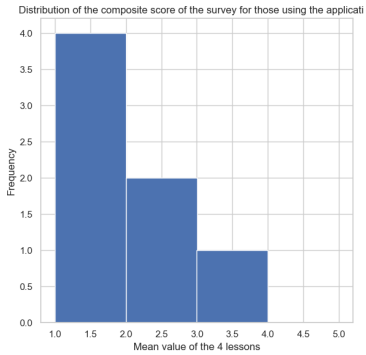


Figure 7: Distribution of the mean score of the subset of answers for those using the whole application

If we remember, the two questions had a rating scale from one to five. In both cases, one means a low value (i.e. not mentally demanding and not supposing effort), and five, a high one (i.e. very mentally demanding and supposing a lot of effort).

We can see how most users found the task of learning the signs in the different lessons not mentally demanding or supposing a significant effort. Only one user had a mean score of 3, meaning that he found it more or less mentally demanding and supposing some effort. These results are the ones we expected, given that all the features in the application should facilitate learning, which is what we wanted to evaluate with the questionnaire.

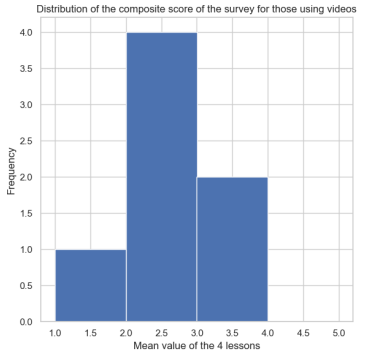


Figure 8: Distribution of the mean score of the subset of answers for those learning through videos

In the group of videos, we can see the results are a little higher, meaning the users found it a little more mentally demanding and needed more effort. Now, the most frequent value is two instead of one, and more users gave it a three in both questions. Again, we expected these results, as those learning through videos should have a hard time, as they do not have the other features to facilitate learning. Maybe, they found the exam difficult, as they could not play with the games and the sign detector. We explained the type of questions they should expect in the exam. They did not complain about it in the open-ended question, so we assume it was not the case.

10.2 Testing the first null hypothesis

Now that we have shown the distributions, we can explain how we tested our hypotheses. We mentioned that we wanted to test the first null hypothesis about the learning progress using the three metrics from the application statistics. We can consider the exam scores as ordinal data and the times as interval data. And as we have seen, the data in all cases does not follow a normal distribution (maybe in one case, but we did not consider it normal to make the tests). Together with the fact that we do not have a lot of data,

we decided to perform the Mann-Whitney U test [8].

The Mann-Whitney U test is a non-parametric test that tries to reject the null hypothesis of two samples coming from the same population (i.e. having the same distribution). Thanks to being non-parametric, we do not need any assumption on the samples' distribution because the null hypothesis does not involve any statistic to compare like, for example, the mean. This test compares each member of one group with all the members of the other. If the two samples come from the same population, each member of the first group will have the same chance of being larger or smaller than each sample of the second group.

The null hypothesis of this test is rejected if one group is significantly larger than the other without specifying the direction of that difference. In that case, we are performing a two-sided test because we do not care about that direction.

If we care about it, we need to modify the alternative hypothesis. Instead of specifying that the two samples are different, we specify the particular direction we are interested in (i.e. specify which distribution we want to be larger than the other). In this second case, we say that we are performing a one-sided test.

We started testing the null hypothesis for the first research question by performing a two-sided test, checking if there was some difference between the two samples. We used *SpaCY*, given that it already implements the test, and we only need to call a function to get the p-value. As we saw in class, we considered a significance level of 5%, meaning that we can reject the null hypothesis if the p-value is lower or equal to 0.05. In particular:

- Using the mean exam score, we got a p-value of 0.04, so we can reject the null hypothesis.
- Using the mean time to complete the exam, we got a p-value of 0.80, so we cannot reject the null hypothesis.
- Using the mean time spent in a lesson, we got a p-value of 0.01, so we can reject the null hypothesis.

With the mean exam scores and the mean time to complete the lessons we could reject the null hypothesis of the learning progress being independent of the learning methodology used. However, as we performed a two-sided test, we did not know exactly the direction of the difference between the two groups.

For this reason, we repeated the test with those two metrics but now doing a one-sided test. We also generated some box plots to understand the difference. In Figure 9, we show the box plots we computed using the exam scores of all four lessons for both groups. We preferred not to use the weighted average, as it would be easier to see it that way. We repeated the same for the time to complete the lessons, and in Figure 10, we show the generated box plots for this other metric.

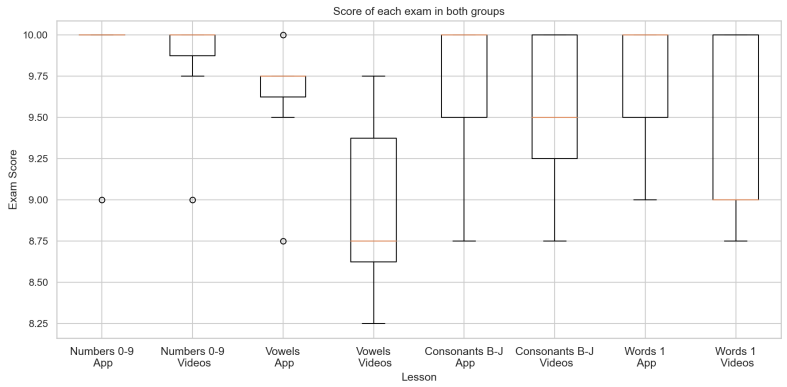


Figure 9: Box plots of the exam scores for both groups

Looking at the exam scores, we can see how the scores from the ones using the whole application are higher than those using videos in all the lessons. The box plot median in all cases is equal to or higher than the one from the same lesson. Also, we can see how the interquartile ranges are smaller in the full application-related results. We have outliers or whiskers that reach similar values to the same box

plot from the videos. In any case, we can say that the results are better overall when using the whole application. And we can statistically demonstrate it by doing the mentioned one-sided tests. Looking for the group using the whole application to be significantly larger than the other, we obtain a p-value of 0.02, rejecting the null hypothesis of being the same population.

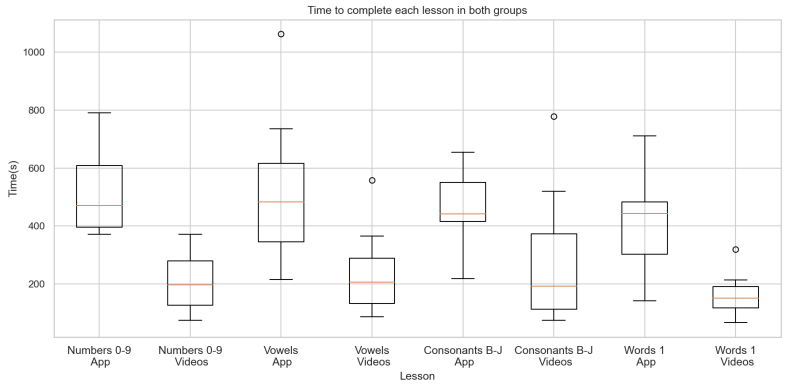


Figure 10: Box plots of the times to complete each lesson for both groups

Looking at the time to complete a lesson, we can see how the times for those using videos is lower. It is easier to see in this case, as the box plots are more separated. We also performed the one-sided test, looking again for the group using the whole application to have a distribution significantly larger than the other. We got a p-value of 0.01, meaning that the times from those watching videos are significantly shorter. We have already mentioned it, but we can expect these results considering that those learning through videos cannot play games or practise signs. However, we expected them to need more time than the other group as they did not have the features to facilitate the learning. We believe the signs we made them learn were not that difficult, and most users could do it by looking at the videos very few times. Also, this confirms that most users of the other group, played the games and practised the signs to see how it worked.

Before concluding this demonstration, we also present the results we got for the time to complete the exams in both groups. In Figure 11, we show the same box plots for this other metric. There is no significant difference, explaining why we got a very big p-value.

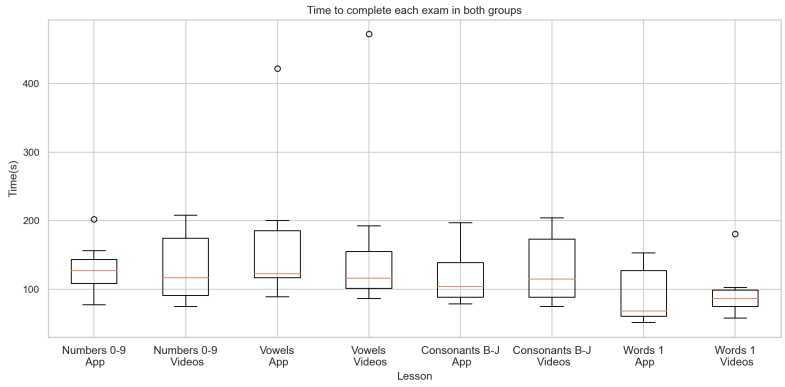


Figure 11: Box plots of the times to complete each exam for both groups

After all these tests, we can say that we could reject the first null hypothesis thanks to the exam scores. We could do it with the time to complete the lessons, but we do not feel confident making that assumption because we were more interested in a particular direction rather than simply looking for a significant difference in the distribution.

10.3 Testing the second null hypothesis

To test the second null hypothesis, we followed the same procedure, performing a two-sided test to check if there was some difference between the two distributions. In this case, the distributions were those from the mean scores we computed from the best subset of questions. We obtained a p-value of 0.07, meaning we cannot reject the hypothesis with a confidence level of 5%. However, we were very close, and with more people, we could reject it.

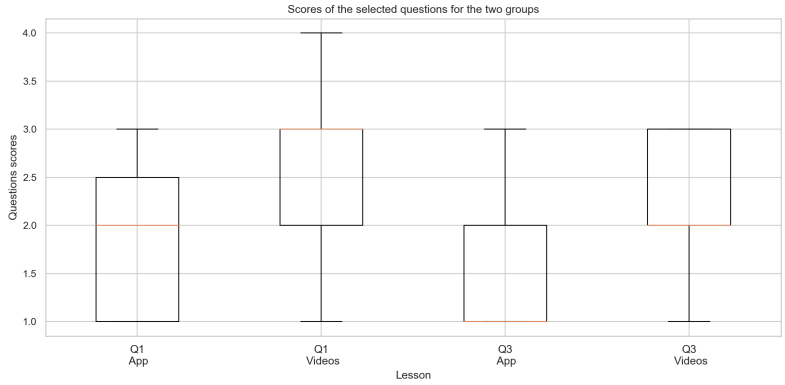


Figure 12: Box plots of the answers to the best subset of questions for both groups

We also plotted the same box plots but now consider the answers to the best subset of questions. In Figure 12, we show the results for both groups. If we remember, lower values in both questions are better. We can see some differences, in particular, the results we were expecting. Again, the median of the box plots is lower for the ones using the full application, as well as the interquartile range starting and ending below the same question for the group using videos. These results confirmed our beliefs about rejecting the null hypothesis if we had more data.

11 Limitations

After testing our application with real users and receiving feedback, together with the results of our statistical tests, we can point out some aspects in which we should improve.

First, our main problem was that we did not find enough people to try our application. We did not have enough time to perform the experimentation, and we could not search for people other than those close to us. This is why we could not test it with some of the potential groups we wanted to focus our application on.

Another problem is with the signs we wanted the users to learn. The complexity level was probably too simple. We indeed observed some significant differences between the two learning methodologies, but maybe, with more people, we could lose this difference. However, we were very limited by the sign detectors, as we wanted to offer the possibility to practise signs.

A third problem that comes from the feedback from the user is the sign detectors and the games. Most of them pointed out the sign detectors failed to predict the signs, especially with the letters. The problem with the letters and the words was the signs were dynamic and required a more complex model to predict them. And considering the time we had to do everything, creating one from scratch was out of the question. And some of them also pointed out that the games were quite repetitive.

Fourth, the application taught ASL, but the people testing it were from Spain. Therefore, they probably did not find it particularly useful. We could not find enough research on this language (i.e. models and datasets), so our only option was to create everything from scratch, which was out of the question.

Finally, we want to mention some problems with the experimentation, which is evaluating the users only once. We made them perform the exam after learning the signs, which facilitated getting better results in the exam. It would be better to repeat this same evaluation more than once to assess better if

the users really learned the signs in each lesson. And another problem was that after learning one lesson, they moved to the next one and never returned. We could have made extra tests merging all the signs.

12 Conclusions

In this project, we have implemented a desktop application to learn ASL autonomously, with the help of *Kivy*, *Pytorch*, and *Mediapipe*. We have created a system based on lessons similar to *Duolingo*, offering the possibility to watch videos of how to perform a sign, play games to get used to them, and even practise how to perform them yourself. To run our application, we only need a computer, a webcam, and the motivation to learn ASL.

We have tested our application with real users, focusing on the aspects of human-robot interaction, even partially demonstrating that our application brings benefits when learning ASL compared to other methodologies like simply watching videos. However, we have seen that our application is far from becoming something that could bring a real benefit, especially for those that need to learn it as fast as possible due to some degenerative disease that causes permanent hearing loss. At least, it currently serves to introduce people to learning this language.

In future work, we would like to improve our application by adding more signs with more complexity, adding more variety of games to make the learning funnier, and improving our sign detectors which are far from optimal. We would also like to be able to detect sentences, allowing the users to learn how to communicate using sign language.

We also want to carry out a more in-depth study with a more significant population, where they have to take the exam several times, leaving enough time between each try to be able to evaluate whether there has been proper learning.

Finally, our application only runs on a PC. It would be interesting to transfer it to mobile devices so people can use it everywhere without needing a computer. And we could also add new features like daily reminders, a reward system to generate positive reinforcement for learning, etc.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL].

[2] Matyáš Boháček and Marek Hrúz. "Sign Pose-Based Transformer for Word-Level Sign Language Recognition". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Jan. 2022, pp. 182–191.

[3] Lee Joseph Cronbach. "Coefficient alpha and the internal structure of tests". In: *Psychometrika* 16 (1951), pp. 297–334.

[4] Berthold K.P. Horn and Brian G. Schunck. "Determining optical flow". In: *Artificial Intelligence* 17.1 (1981), pp. 185–203. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(81)90024-2. URL: https://www.sciencedirect.com/science/article/pii/0004370281900242.

[5] Dongxu Li et al. "Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison". In: *CoRR* abs/1910.11006 (2019). arXiv: 1910.11006. URL: http://arxiv.org/abs/1910.11006.

[6] Florent Mahoudeau. *MiCT-RANet for real-time ASL fingerspelling video recognition*. https://github.com/fmahoudeau/MiCT-RANet-ASL-FingerSpelling. 2020.

[7] Arda Mavi and Zeynep Dikle. *A New 27 Class Sign Language Dataset Collected from 173 Individuals*. 2022. arXiv: 2203.03859 [cs.CV].

[8] Nadim Nachar. "The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution". In: 2008.

[9] Bowen Shi et al. "Fingerspelling recognition in the wild with iterative visual attention". In: *CoRR* abs/1908.10546 (2019). arXiv: 1908.10546. URL: http://arxiv.org/abs/1908.10546.

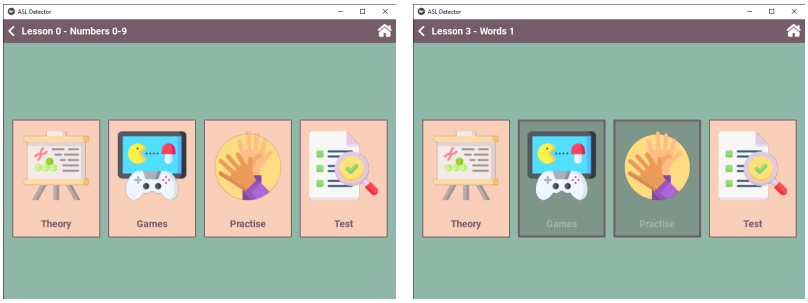
[10] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[11] Yizhou Zhou et al. “MiCT: Mixed 3D/2D Convolutional Tube for Human Action Recognition”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 449–458. DOI: 10.1109/CVPR.2018.00054.

A Screenshots of the application

(a) Main Screen (b) Lesson Selector Screen (c) Score Selector Screen

Figure 13: Main Screen, Lesson Selector Screen, and Score Selector Screen

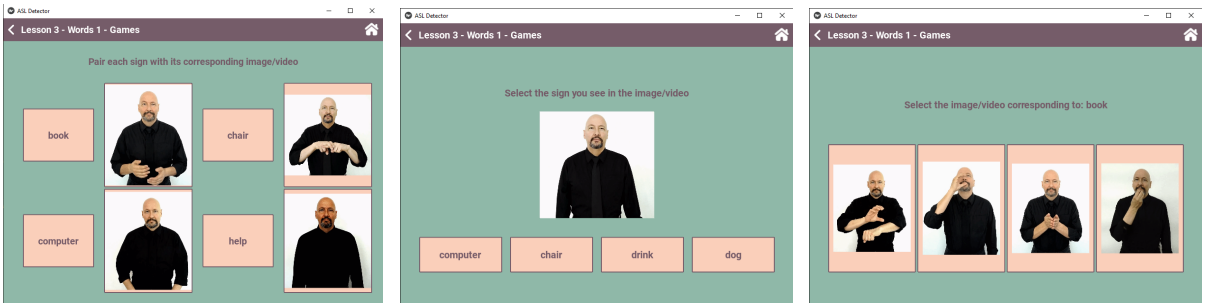


(a) Lesson Screen (b) Lesson Screen for those learning through videos

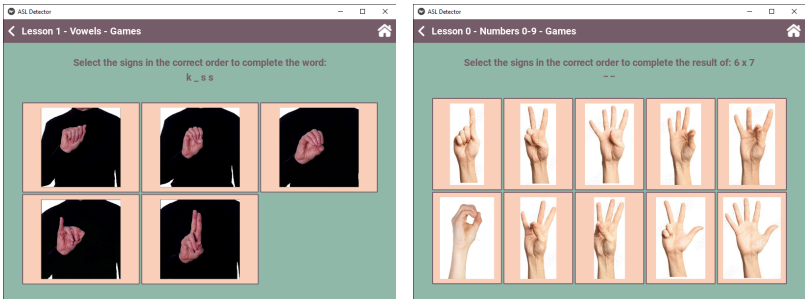
Figure 14: Lesson Screen for those using the full application and those learning through videos

(a) Theory Screen (b) Video Player Popup

Figure 15: Theory-related screens

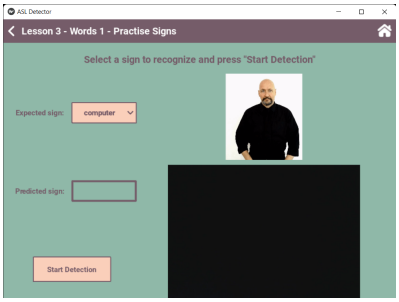


(a) Memory Game (b) Multi-choice Game (c) Inverse Multi-choice Game

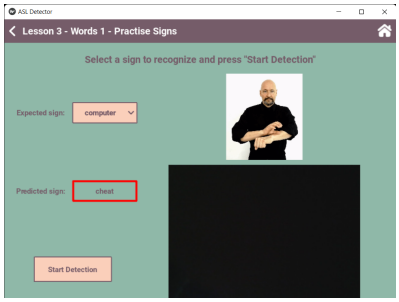


(d) Word Gap-filling (e) Number Gap-filling

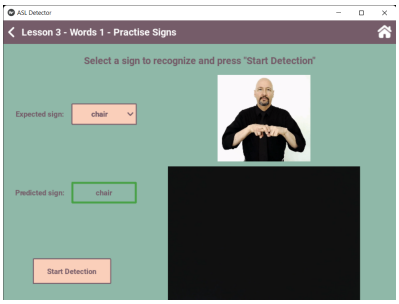
Figure 16: Games-related screens



(a) Sign Detector Screen

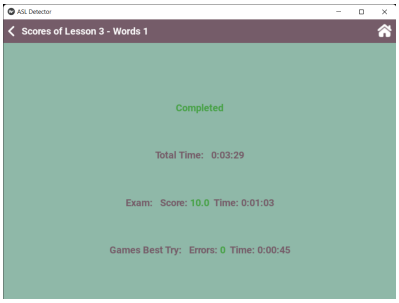


(b) Sign Detector Screen (wrong prediction)



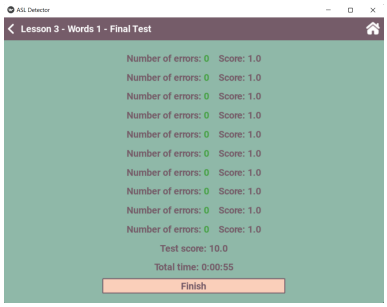
(c) Sign Detector Screen (correct prediction)

Figure 17: Sign Detector Screen



(a) Lesson Score Screen

(b) Games Score Screen



(c) Exam Score Screen

Figure 18: Score-related screens