Prediction of Choice Using Eye Tracking and VR

Master Thesis

Carlos Gomez Cubero

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Prediction of choice using eye tracking and VR

Theme:
Human-Robot Interaction

Project Period:
Spring Semester 2020

Project Group:
VGIS 1046

Participant(s):
Carlos Gomez Cubero

Supervisor(s):
Matthias Rehm

Copies: 1

Page Numbers: 35

Date of Completion:
June 4, 2020

Abstract:

In this thesis, its used Virtual Reality (VR) and eye tracking together to test if it is possible to obtain an intention recognition from the human gaze, that could be used in the future for human-robot interaction with collaborative robots. To do this, first is studied the related work for eye tracking and gaze detection, then the machine learning algorithms for time series analysis. Later, a data set is gathered using a VR game where the participant have to choose between different objects while the gaze is being logged. This data set is eventually put to the test with the cascade effect hypothesis and training a Long Short-Term Memory (LSTM) model, showing interesting results, with an accuracy significantly better than the random guess for a few seconds before the participant has chosen an object in the game.

Contents

Preface

Aalborg University, June 4, 2020

Carlos Gomez Cubero
<cgomez18@student.aau.dk>

Chapter 1

Introduction

This chapter contains the information needed for the reader to get into the topic. It starts with the context, that slightly describe the problematic and make sense with follow section, the motivation and goals.

1.1 Context

Animals are incredible machines of self-adaptation and compensation to overcome situations,it might be for instance, an hazard that can compromise the integrity, or playing a specific role in a social situation where remain part of a pack determine survival. The adaptation or compensation happens thanks to two main elements, sensors, that give information of the environment and actors involved in the situation, and a previous knowledge, that measure the situation and adjust the behaviour to overcome it. Examples can be: flying or fighting response in a live-threatening situation, sharing goods with a member of the group in need, or leaving ice cream for your little sister so your parents don't scold you.

Collaborative robots, also known as cobots, are a branch of robotic systems that are meant to be placed in a shared environment with humans, without the need of a cage to protect the operators. Most common sensors in cobots are the force sensor in the joints, that allow the robot to detect anomalies when it moves due to an unexpected force in any of the joints, this is the example of the robot UR3 and UR5 from Universal Robots. When it comes to adaptation, as the previous paragraph, the scenario is equivalent. The robot must have sensors that measure the environment and the actors, and a previous knowledge to react to the situations. Current collaborative robotic systems rely in the force sensors to stop or react when it hit an unexpected object, but the industry is pointing beyond, hoping to adjust the behaviour of the cobots to become more human-friendly, even anticipating human actions.

Human gaze gives a lot of information in a relation human-human, and there-

fore is a big indicator to detect or anticipate human reactions, from how someone look to the supermarket shelf when shopping to understand if someone feels confident in a social situation. Measuring the gaze path and find patterns in certain situations can help to anticipate human reaction in an interaction with cobots, this task can be assisted by machine learning models that perform exceptionally good when dealing with time series data.

1.2 Motivation and Goal

This thesis has been made in the HRI-Lab (Human-Robot Interaction Laboratory) of the AAU where the interaction between robots and human is studied. This laboratory is run and used by a multidisciplinary team composed of professors from Robotics to Mediology or Psychology. The lab counts with some cobots and a VR environment as well as a VR set with eye tracking equipment among others, the perfect environment to bring all this components together. Before the thesis the writer has spent dozens of hours in this lab using some of this equipment in different projects and developing a tool for the department to bring one of the existing robots, Sawyer robot, to a VR scene so it is possible to experiment with a virtual avatar of the robot in VR. For this previous experience the writer is familiar with cobots and VR.

All information that may be relevant for a robot to understand the environment or actors in an interaction are precious. Having all this equipment available the goal of this thesis is to study the intention recognition of humans using the gaze, which can be captured with the eye tracking equipment, and analyze if is possible to get insights from the gaze and what degree of accuracy can be achieved, so these findings can be used in future projects with cobots.

Chapter 2

Gaze in intention recognition

This chapter contains an analysis of the state of the art (SotA) of gaze detection and intention recognition. Starts by covering what is gaze detection, continue with how the eye tracker work, the approaches to tackle the gaze analysis, the cascade effect hypothesis, the stages in decision making, and ends with a practical example of differences between results in the lab and results in the field.

2.1 Gaze Detection

Imagine controlling your computer just with the movement of your eyes, this might sounds like something from science fiction movies but the truth is that gaze detection is a reality. Already in 1989 Thomas E. Hutchinson et al.[1] presented Erica, a computer work space that could be controlled with the eye movement, intended to be used by users with physical disabilities. The system consisted in an infrared (IR) camera, that discriminate the ambient light, positioned together with the computer's screen. An IR torch illuminate the face of the user that stare to the screen and an algorithm with traditional computer vision detect the position of the pupils. With this information they could determine the region of the screen where the user is looking at with a precision of a 3x3 matrix over the screen. This approach for gaze detection and controlling a computer has been improved since that in multitude of publications, offering better calibration and pupil detection techniques as well as using different camera sets[2, 3, 4]. Nowadays exist multitude of low cost and "do it yourself" solutions available to replicate this projects using for instance the Kinect or a webcam[5, 6], and its possible to find open source project done by the community.

A second approach to the eye tracking is in a form of a wearable. The camera is attached to the user, normally using a helmet or different type of glasses. This open the variety of possibilities to research with gaze tracking in other fields apart from using a computer. Y. Wang et al. [7] propose a method for a low-cost

head-mounted eye tracking system to control the computer, where it combines the position of the pupil using a camera attached to a diadem and complemented with a gyroscope and accelerometer module to compensate the movement of the head. Andreas Bulling and Hans Gellersen covered in 2010 this topic in their article "Toward Mobile Eye-Based Human-Computer Interaction" for the IEEE journal [8] showing some of the prototypes in the market at that time. Now ten years later it is possible to find commercial equipment as the one provided by the company Pupils Lab, that can map the gaze in a video.This product consist in the eye tracking cameras pointing to the eyes and a camera recording the view of the user, after the calibration its possible to map the gaze on the users view. This philosophy is the one used for this project, with the particularity that the test runs in a Virtual Reality (VR) environment with a solution from Pupils Lab optimized for VR headsets.

2.2 Common Components in the Eye Tracking Solutions

After analyzing different methods and approaches for eye tracking there are common points that all of them share. Stopping on them may help to understand better how this systems work and clarify possible misunderstandings.

2.2.1 Hardware

All eye tracking consist in one or multiple cameras, in the modern systems is usual 2 cameras placed right in front of the eye, what gives a better resolution of the Region Of Interest (ROI), together with an IR light torch composed of one or more single points of light. The cameras count with and IR filter what help to get the lightning conditions out of the equation.

2.2.2 Software

After the camera feed there is a software that process the image, this can be done by traditional computer vision or SotA machine learning models trained to detect images. The video feed is cropped to the ROI what is the area of the eye including the eyelids, and here is where the tracking take place. The goal is to estimate the Point-Of-Gaze (POG), E. D. Guestrin et al [9] presented a theory for remote estimation of the POG by reconstructing the visual axis, for this is used the optic axis of the eye, that is the line connecting the center of the pupil to the center of the eye, and the reflections of the IR light (glints) in the cornea. Contrary of what it might seem, the visual axis deviate from the optical axis and is the line connecting the fovea and the center of the cornea. For each eye the visual axis is projected into the screen of destination and the projection is interpolates to acquiring an estimation of the POG.

2.2.3 Calibration and Verification

The calibration has not been mention yet, but is a crucial part in the use of these systems, due to its nature, the process of estimating the POG has an immense variability, and therefore very sensible to any change, taking in account the placement of the hardware and the physical characteristics of the user, for this reason the system must be calibrated at the beginning of each use and re-calibrated if any component is misplaced while it is used. The calibration in most cases consists in staring to different points across the field of view and adjust the POG algorithm parameters to maximize the accuracy. With the system calibrated is important to run a control test to verified the accuracy of it, so the data gathered can be reliable.

2.3 Eye Fixation and Saccades

To analyze the gaze first is needed to understand how is the human gaze behaviour. Exist 5 different types of eye movements: saccades, smooth pursuit, vergence, vestibulo-ocular movements and Optokinetic response movements [10]. Saccades, figure 2.1,is the most popular eye movement studied in research, it consist in a rapid movement of the eyes to change the point of focus of the gaze, also known as eye fixation, this eye movement is the one you reader are doing right now jumping from one word to the next. It can be done at your own will but most important is that it is done mostly unconsciously. To study this eye movement are commonly used two approaches, eye fixation analysis and saccades analysis. In 2000 D. Salvucci et al. [11] compared different methods to identify what is an eye fixation and what saccades. Eye fixation analysis consist in record the points where the gaze stops and focus the attention, here the image projected in the eye and processed by the brain is clear in the region around the POG while during a saccade the image is blurry. L. Cooke [12] used this approach studying the fixation duration and fixation frequency for usability tests. Saccades analysis consist in tracing the path of the gaze. This approach involves more time resolution since the saccadic movements are extremely fast, is uncertain the exact sample rate needed to ensure a perfect record. In 2008 R.Wierts et al. [13] proposed a method and postulate that 50Hz, what is consider low frequency, should be enough for clinic tests.

2.4 Gaze Cascade Effect Hypothesis and intention recognition

In 2003 Shimojo et al. [14] introduced the term "Cascade Effect", in their experiments they found first that in a situation where a participant hast to choose between different options, the attention is first random between the options and as

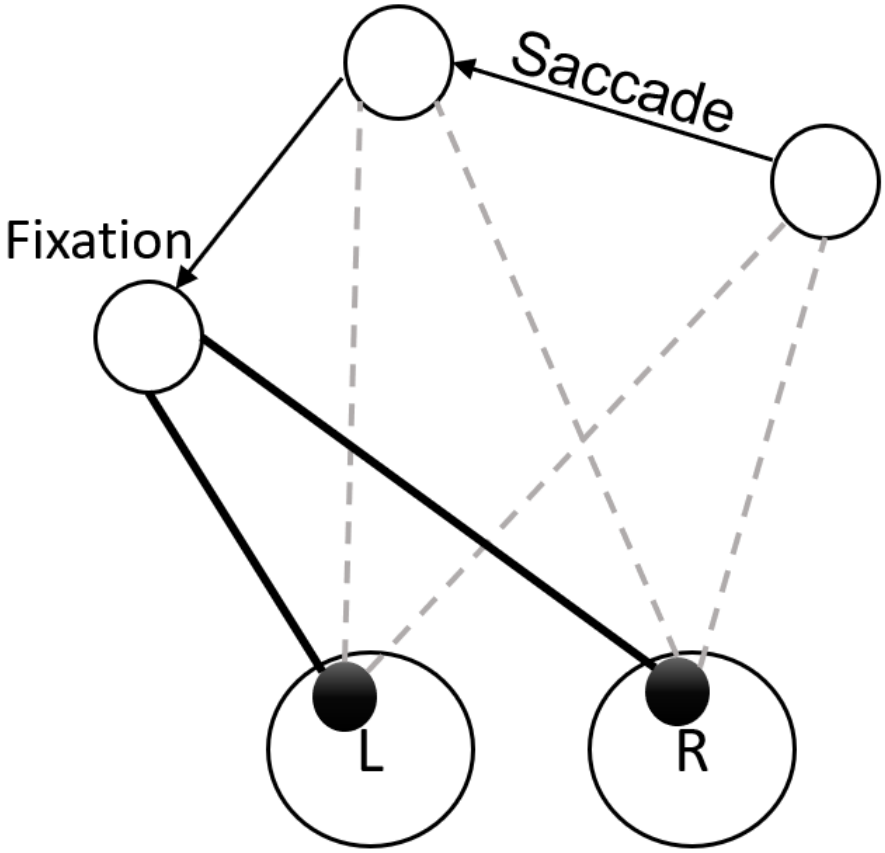

Figure 2.1: "Representation of saccadic movement. Circles in the top are the eye fixation points and the arrows the saccades."

the time runs, gradually, the attention is shifted towards the more liked option. Also they carried out other experiments where they tried to bias the participant by showing and hiding 2 images varying the exposure time, the results turned out that the image that was exposed for a longer time was more likely to be chosen at the end of the experiment. D.Bird et al. [15] studied this phenomenon and its relation with the eye movement, they carries out two experiments, in both they hide and show intermittently 2 pictures varying the duration of exposure to again bias the participants. In one experiment they show each picture in one side, forcing the participant to move the eyes, and in the other experiment they show the images in the same place. The results from their experiments conclude with a similar outcome for both experiments, and followed the cascade effect hypothesis with a 54% of probability of choosing the image with longer exposure.

2.5 Stages in Decision Making Relying on Gaze

Russo et al. [16] in their study of the gaze path analysis state that the process of choosing between objects using the gaze has 3 stages, figure 2.2.

- Orientation

  Is when the participant in a seeking and choosing activity has to direct the attention to the place where the available options are. Its characterized by having the POG out of the scope of where the choosing action will take place.

- Evaluation

  Is when the participant is focused on the task, looking at the objects moving the POG between one and other.

- Verification

  It occurs instants before the object is picked with the hand or is speak aloud, the participant already knows what is the object chosen and is staring at him.

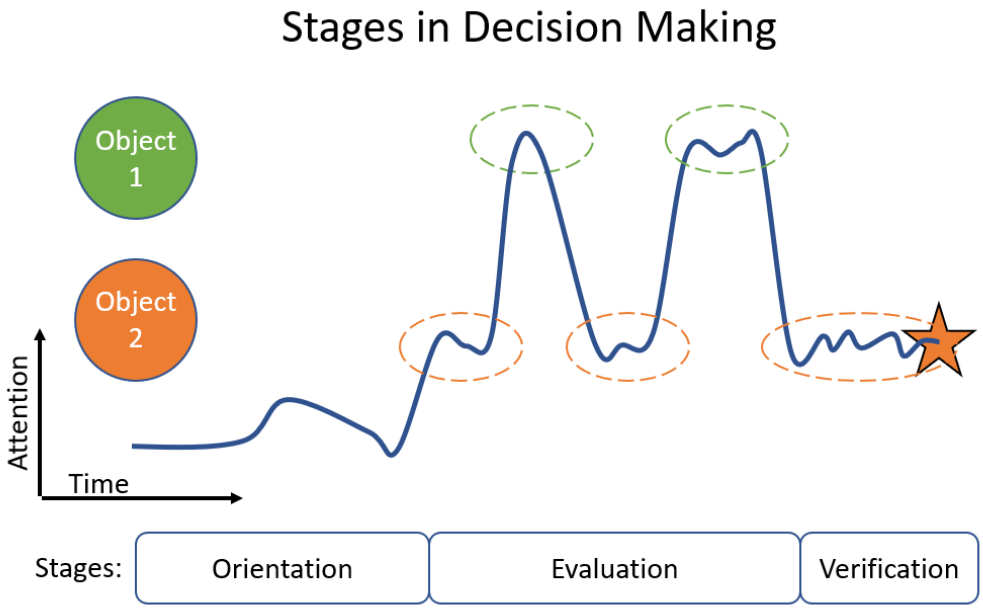

Figure 2.2: "Stages in decision making when gaze is involved, relating the attention with the time. The colored circles represent the position of the objects, the blue line the attention. The dash-lined ellipses mark down the fixations on an object and the star is where the action of picking the object occurs."

2.6 Moving from the laboratory to the real world

For The experiments carried out in this project due to its characteristics of using a VR environment with a fixed virtual set-up and the way it have been conducted, the results may differ from a future real world application of it. Russo and Leclerc [16] experimented with the visual behaviour of consumers buying products in a supermarket shelving in a laboratory simulation, testing two different tasks, searching for a product and deciding a product to buy. To analyse the gaze behaviour they recorded a sequence of the eye-fixation, this eye-fixations where coded by location and not for duration, Kersiting et al. [17] call this eye-fixation dwells, and repeated the same experiment in a real supermarket, calling this the "Natural Environment", as opposed to the "Laboratory Environment". From Kersting et al. experiment it appears that the results in the different tasks differ in the two experiments. While in Russo et al. experiment there were no differences found between the two tasks, in Kersting et al. there were significant differences in the number of dwells between tasks. This leads to think that as part of the cognitive process, the results can differ from a natural environment to a laboratory environment, and this has to be taken in consideration and be cautious when drawing conclusions from the laboratory results.

Chapter 3

Time series analysis and machine learning

This chapter has a brief introduction to the digitization problematic and time series, and gives a small introduction to machine learning, what should be helpful to follow why LSTM was chosen for the experiments.

3.1 From analog to digital

The world we live in is analog, and when it comes to measure it what we got are continuous values, this means that we can fraction the time in infinitesimals portions and the values can be in the whole domain of the real numbers. To bring a signal from the real world to a computer they have to go through a process of analog to digital conversion(ADC). In this process the converter has to digitalize the signal to be understandable for a computer, two main features characterize the converter. The sample rate, which is how often a sample is recorded in the system and depends on how fast is the signal, in order to not lose information it has to be sampled at least 2 times the fastest frequency of interest, theorem of Nyquist, and the bit depth which is how much precision can be achieved between a range of values, and depends on the error that is acceptable, this error or noise is measured as the uncertainty of the measure and its maximum value is half of the read step, in the literature the bit depth can be also found as the dynamic range. This two features determine the goodness of a discrete signal to reassemble its analog one and they are compromised by power of process, memory capacity and utility. An example is the standard Compact Disc Digital Audio (CDDA) also know as Audio CD, that set the parameters of digitization to ensure the a level of quality in digital audio. The main points are a sample rate of 44.1kHz, minding that human audition range up to 20kHz, and a bit depth of 16 bits, noticing that this gives a dynamic range enough for a exceptional signal-noise ratio. All this combined and after

adding metadata and redundancy ensure that 80 min of music (what is expected for an album) can be stored in a physical compact disc.

3.2 Time series

A time series is a discrete sequence of data that is distributed over the time and indexed with a time stamp. The series can be distributed evenly spaced in time, with an equal time slot between samples, for instance the music, or unevenly spaced in time without a constant time slot, for instance a record of clinical trials. They are widely used for logging data and its worth is in most cases not the individual logs but the trends.

3.3 Machine learning

Machine learning (ML) is a branch of the Artificial Intelligence (AI). The ML algorithms has the ability to learn from a given data set and perform different tasks that otherwise could be too complicated with traditional coding in terms of human workload or nearly impossible to design and code. Different ML algorithms are designed for different functionalities as regression, classification or clustering among others, in [18] there is a brief introduction to the different ML methods. Exists different types of algorithms, the most popular are included in the Supervised Learning (SL) and Unsupervised Learning (UL), while in SL the data set for training is labeled in UL the data set for training is not labeled.

Machine learning algorithms are nothing new, the concepts behind their functions where studied in the previous century, but is now thanks to the computational power, fast memory available and the introduction of GPUs that this algorithms can be carried out[19].

3.3.1 Supervised Learning

Supervised learning algorithms are focused on finding an existing relation between a certain input variables and certain output variables, the input is normally name as "data" and the output as "label". After showing to the algorithm a big amount of examples of this data and labels, so called training, if certain conditions exist that relate the data and their label the algorithm will be capable to generate a label for a future data that it did not train with, this is mostly used for classification and regression. Examples of problems solved with this kind of algorithms could be object recognition, spam e-mail classification or stock market prediction[20, 21].

3.4 NN - Neural Networks

Neural Networks (NN) are one of these SL algorithms which has become the most popular in the present. A NN has 3 different type of layers, input layer, output layer and hidden layer. These hidden layers are formed at the same time for perceptrons also called neurons, a perceptron contains weights and a bias, a transfer function and an activation function, figure 3.1. The parameters of the perceptrons in the hidden layers are learned during the training process by using optimizer algorithms, the hidden layers can be stacked one after another, what gives the model a bigger capability of abstraction.

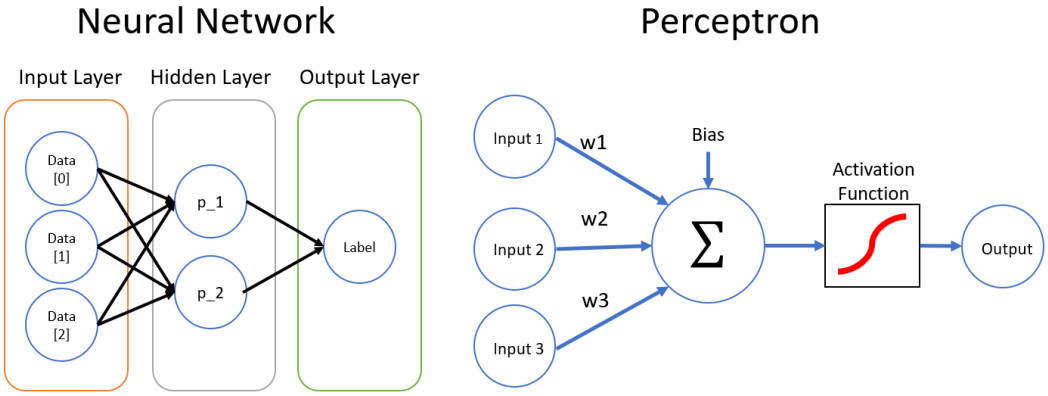

Figure 3.1: "At the left a schema of a simple Neural Network with 3 inputs and 1 output, and one hidden layer of dimension two. At the right a schema of a Perceptron"

3.4.1 RNN - Recurrent Neural Networks

A Recurrent Neural Networks (RNN) is a class of NN that loops some of their perceptrons using a delay, this peculiar architecture gives the model recurrence and the possibility to analyze sequences in the time [22]. While basic NN generate an output based only in the present input, RNN can use also a function of the previous inputs in a variable called hidden state. This architectures are useful for different cases where the context of the data is more valuable than the data itself, for example text analysis or voice recognition. In the figure 3.2is shown an abbreviated schema of a vanilla RNN node.

3.4.2 LSTM - Long Short-Term Memory

Long short-Term Memory (LSTM) models are the most popular RNN nowadays, a vanilla RNN, as stated before, loops the nodes of the NN to save previous states, as seen in figure 3.3, the hidden stated propagates from the first input to the last,

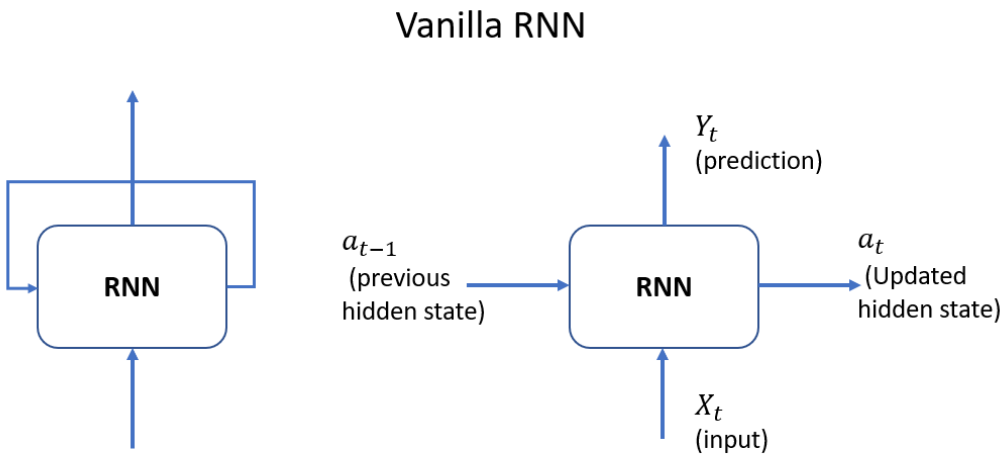

Figure 3.2: "Schema of a vanilla RNN, at the left the sybol, at the right the detail of inputs and outputs"

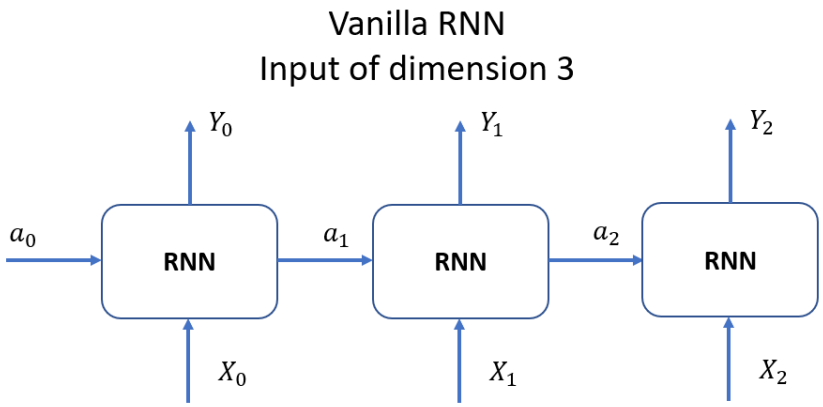

Figure 3.3: "Schema of a RNN layer with an input of size 3"

this causes that this values loose effect as they travel thru the loops,therefore close hidden states in a node have more effect than the far ones, this is called short-term memory [23]. The difference in a LSTM is that is designed in a fashion that store the most valuables hidden states in a new output called cell state, this cell state is loaded and cleared by the LSTM nodes and thanks to that states from far nodes can remain intact thru the entire layer, this gives the name to the model long short-term memory. In the figure 3.4 is shown the function that runs inside of a LSTM. It consists in 3 gates composed by a NN, a sigmoid function, and a multiplier. The parameters of these gates are learn during training and act like valves that let the

information in the cell state be updated or erased. LSTM are used in a wide variety of fields where time series are involve, from detecting cyber-security attacks [24] to stock fluctuation forecast [25]. As other NN architectures, the number of layers in a model increase their power of abstraction, this means extracting complex features, depending on the data distribution and how the trend behave the number of layers may variate to obtain an optimal solution.

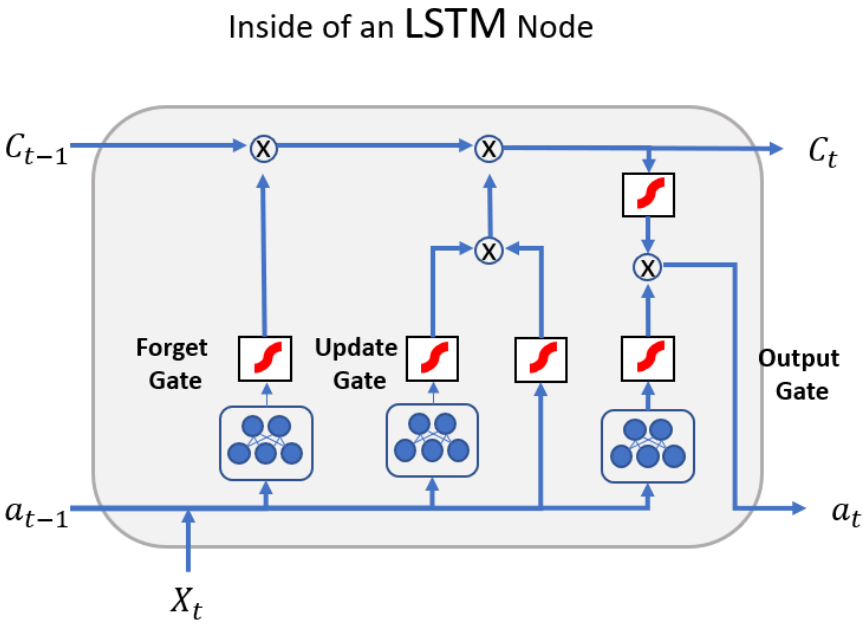

Figure 3.4: "Schema of an LSTM node. C is the cell state, α is the hidden state, X is the input, all dependant of t what is the position in the time series"

Chapter 4

Tools

In this chapter is explained briefly the tools used for this project.

4.1 Unity

Unity is a game engine created by the company Unity Technologies. It allows to create high end games for free to indie artist and researchers, and come with a variety of tools to cover most of the computer graphics projects, including VR and augmented reality in cross platforms. Games in unity are coded in C#, and the assets as 3D objects, images or sprites must be created in an external software and then imported.

4.2 The Workstation

The workstation used to run the experiments is a laptop from the brand MSI. It comes with a CPU intel CORE i7 7th generation, 16 Gb of RAM, and a GPU GTX 1060 GEFORCE from the brand Nvidia. This is a VR ready laptop since a consumer laptop would not be able to run VR applications. The performance of this laptop while running the experiment was around 40-50 frames per second (FPS), this value is not bad but FPS on the order of 100 would be desirable, this can be achieves by using VR ready desktop computers.

4.3 VR and Eye Tracking

The VR system used for the experiment was the HTC Vive. It consist in a wired headset, two controllers and 2 light houses. The HTC Vive is a reliable VR system which provides state of the art features and excellent performance and it is fully compatible with unity using the library Steam VR. In the top of that it is mounted a Pupils Lab solution for eye tracking in VR headsets. It consist in two cameras

placed beneath the headset's screen, facing the eyes and which are connected to the PC using the USB port used by the headset, figure 4.1.

Figure 4.1: "HTC Vive headset used for the experiment. In the top, view of the head set and the controllers, in the center, view of the lenses, in the bottom, close up view of one lens where it can be appreciated the camera for eye tracking"

Chapter 5

Experiments

As mentioned in the goal of the project, we aim to study if is possible to predict the intention of a human by its gaze. To gather the data for the intention recognition it was designed a small game using a virtual reality environment and it is described in this chapter.

5.1 Considerations

It is important to mention that this experiment was carried out between April and May of 2020, coinciding with the outbreak of the COVID-19 and its corresponding lock-down. This means maintaining social distancing and extreme the hygienic measures in order to ensure the safety, remember that the VR headset is something that you put on your face. During this time finding participants became a hard task and at the same time the university laboratories were closed. All these is translated in a lack of participants and powerful equipment, but thankfully the experiment could be done according to what is available in chapter 4. For these reasons caution should be taken when interpreting the results.

5.2 Description of the Virtual Environment and the Game Mechanics

The virtual environment for this experiment is run in Unity and it simulates a small closed room, the participant is sitting in a chair and has a table in the front, this resemble the collaborative robot assembly task where a participant is sitting in a table and have to do a task together with a robot, but in this case without robot,in the wall in front of him will appear a number of objects with the same shape and dimensions but with different color, in figure 5.1 is shown the virtual scene for one of the experiments. The participant then can look at the objects, select the one

they like the most and bring it to the table, in the table they can place it at their own will and then repeat the process, the selection, also called as picking action in this report, is done using the VR controller. During the entire process the gaze is tracked and logged for the following study.

The mechanics of the game are very simple, it has 2 stages that are repeated circularly: choosing between the objects and bringing the object to the table. The participant only have a controller to interact with the game and only one button, the trigger. During the first stage the controller cast a visible ray which function is to aim for one of the objects, when the participant pulls the trigger and if the controllers ray is touching an object then the first stage ends. Then the object selected come to the controller as if it was grabbed with the hand, and the participant can move it around and eventually pull the trigger again to drop it in the table finishing the second stage. When the first stage start again the objects in the wall are replaced for a new ones with new colors.

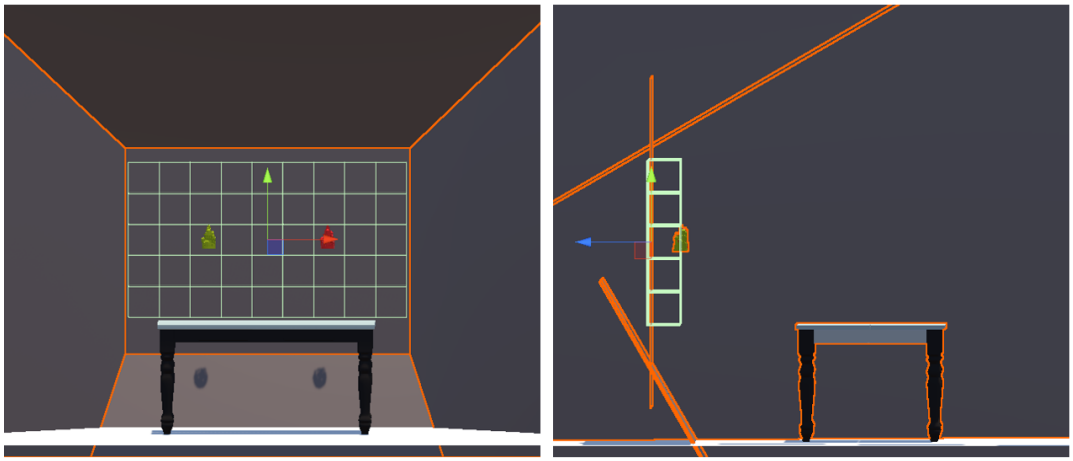

Figure 5.1: "Views of the unity scene of the experiment with 2 objects. In the left the front view, in the right the lateral view."

5.3 Data Acquisition

For the data acquisition is used the saccades analysis mentioned in section 2.3 as the time for every iteration is very short and the number of choices is small. From the eye tracking system we get the position of the gaze (POG) in the participant's screen in a form of pixels position. With this position is casting a ray that log the name of the object hit by it, this way its possible to know where is the participant's attention. In the front wall is placed a set of cubes in a grid distribution of shape 5x9, figure 5.2, each cube with a name related to its 2D position in the grid. The table and the rest of the walls has a key name as well that can be post-processed

after but does not count with any grid distribution of cubes or such as its not consider relevant.

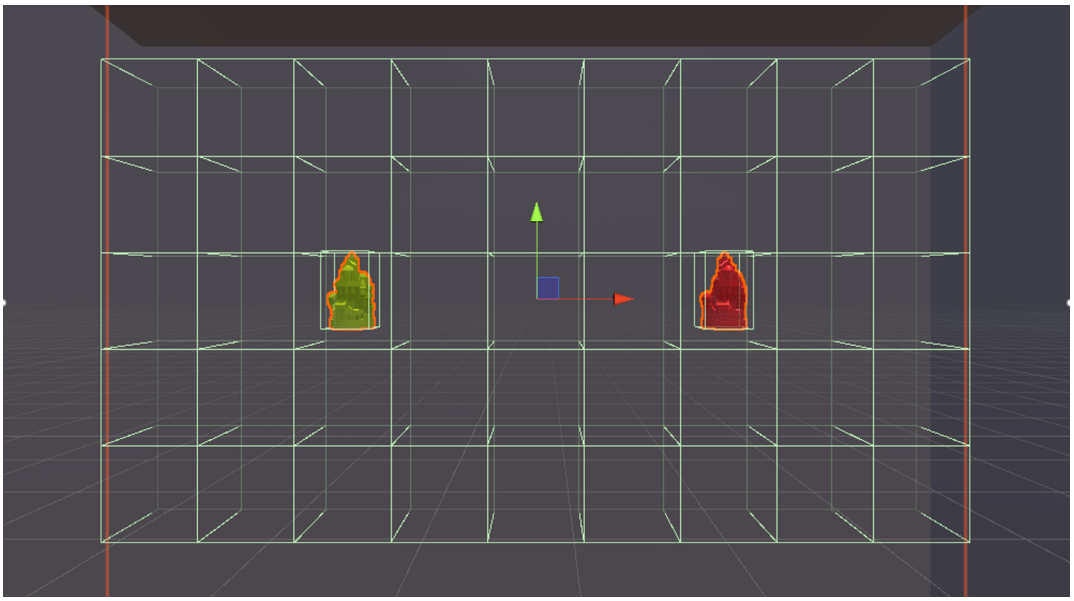

Figure 5.2: "Distribution of the grid of cubes, used to record the POG of the participant when performing the test."

Now that we know what we it is being logged, its time to talk about how fast. The sample rate has been set to a conservative 10 Hz, this value is very small and a desirable value as mentioned must be in the order of 50Hz or above, but due to the workstation computational power constrain that let to run the experiment at a maximum of 45 fps this value feels save to get a stable reading. regarding the sample rate to the number of cubes in the grid and its size, the resultant reading is a continuous discrete path that jumps from one cube to the following one without skipping cubes when the movement of the eyes is normal In case of a higher sample rate a grid with more resolution would be desirable.

Together with the gaze position is also logged auxiliary data:

- End-Start.

  That is triggered at the beginning of the experiment and every time an object is placed in the table.

- Pick.

  That is triggered when the participant select one of the objects from the wall.

- Selected object.

  That record the position of the object picked by the participant.

In the post-processing all this recorded data is first split generating a set of series with only the action of picking the objects and then arranged to be input in the machine learning algorithms, in a set of series form of a 3D array,2 dimensions correspond to the X and Y of the position in the grid and the 3rd dimension is the time. The values move from [0,0] for the top-left corner to [8,4] for the bottom-right corner. For the surrounding walls is given the values [-1,2], [9,2], [4,-1] and [4,5] for the walls left, right, top and bottom respectively. At the same time another array is created with the position of the object picked for each series.

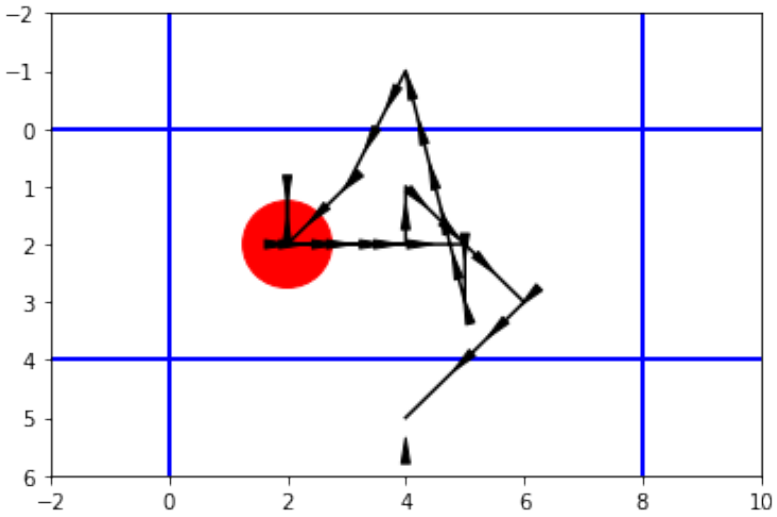

Figure 5.3: "2D representation of the gaze path for one series. In black the line that trace the eyes over the grid with arrows pointing the direction, in red the position of the object picked"

5.4 How the Test was Conducted

The test was divided in two experiments, in the first experiment the participants had to choose between two object and in the second experiment between three. Every experiment had a duration of 10 to 15 minutes per participant. The indications before starting the experiment were simple, after a calibration they must play the game, where they have to pick the object more appealing for them and place it in the table, without time constrains. During the course of the experiment they are asked about their attention to the game, and the experiment stops when the participant feel tired. The two experiments are carried out for each participant one after another with 5 minutes of rest and re-calibrating the system for the second experiment.

5.5 Observations in the Raw Data

After performing the test were gathered 425 series for the 2 objects experiment and 302 series for the 3 objects experiment, from the analysis there is some relevant information that can be taken.

5.5.1 Duration of Each Action

The length of each series is different but with the median of the set its possible to get an idea of what could be the normal behaviour, in the figure 5.4 is shown the histogram of the series length for each experiment.

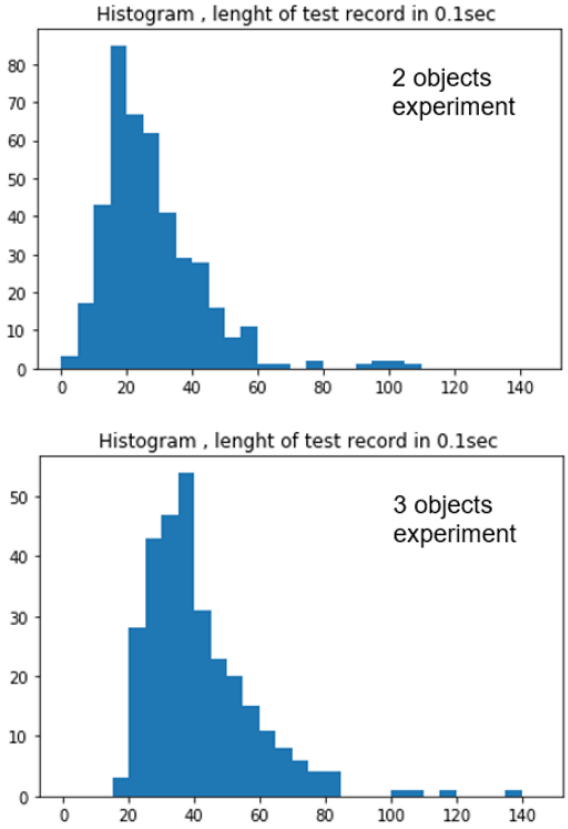

Figure 5.4: "Histogram of length of the series for the different experiments"

5.5.2 Attention and Dedication Throughout the Experiment

From the observation of the recorded actions it was noticed a decrease in its length as the experiment continues over time. This can lead to a 2 possible scenarios, the

participant is getting more confident with the game and therefore it can do the task faster, or the participant is getting bored and is loosing the concentration. This is not tacking into account but with a bigger data set it could be possible to rule out some data or shorter the duration of the experiments.

5.6 Algorithms to Test

- Cascade Effect

  As stated in section 2.4, according to the cascade effect hypothesis the object most likely to be picked should be the one that got more attention throughout the process of choosing the object. To test this the algorithm consist in just comparing the attention to each object and the object picked.

- LSTM

  LSTM is nowadays the most popular model for classification and regresion of time series. The algorithm tested consist in a vanilla LSMT model of 2 layer, with an input size according to the type of experiment.

Chapter 6

Results

This chapter describes the results obtained using the data extracted from the experiments into the different algorithms.

6.1 Stages in Decision Making

As it was stated in the section 2.5, it was expected to found 3 stages in the series: orientation, evaluation and verification. This was confirmed by analyzing the beginning and the end of the series available. The orientation is the stage where the subjects rise their head from the table and redirect their attention to the front wall, in the figure 6.1 it is shown the histogram with the times for orientation, extracting that in 65% of the cases the orientation took 0.6 seconds or less. Verification is the stage where the subjects stare to the object they like and pick it, for an instant they have to aim the controller to the object. This was studied using an average of the attention of the object picked respect to the time in the end of the series. From figure 6.2 it seems that in the last 0.5 seconds the attention was focused in the object picked in the 70-85% of the series what leads to think that is in this 0.5 seconds where the controller is aimed.

These results add valuable information for the training process, acknowledging that for the majority of cases the first 0.5 seconds is happening the orientation and the data is not useful, at the same time for how it is logged, as a constant, we will see in next section that this is not much important since the data is clipped or padded depending on their length, but more important is that 0.5 seconds before the end of the series the subject has already decided what object to pick and its only aiming the controller to pick it.

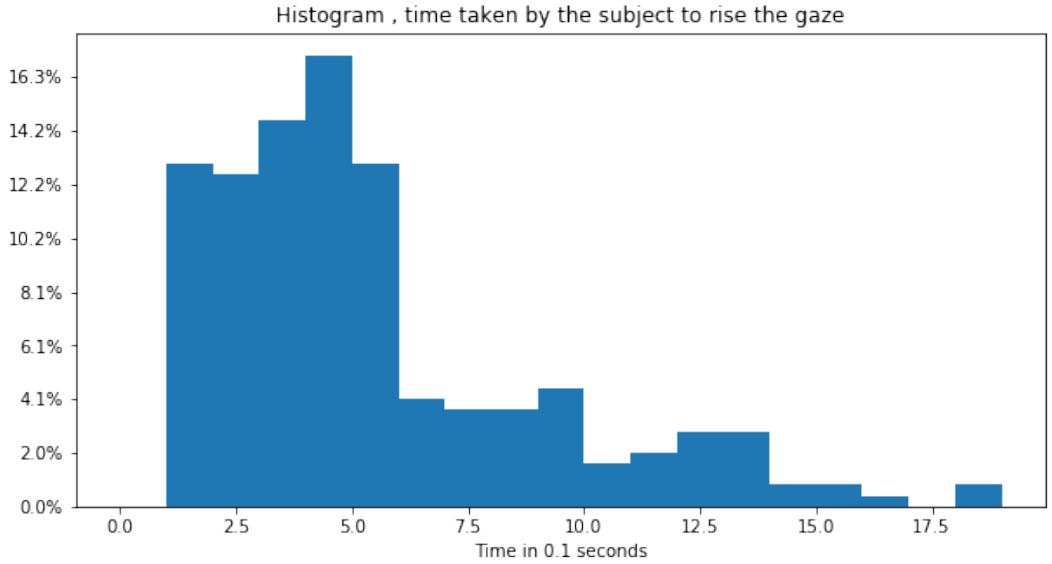

Figure 6.1: "Histogram of orientation time for the data set of the experiment."

6.2 Experiment With 2 Objects

For the experiment with two objects 425 series were gathered, as shown in the figure 5.4 the distribution of the length is concentrated between 2,5 and 6,5 seconds, for this reason the data set was decimated to the series in this range.The data post-processed was then cut to 40 samples padding at the beginning the series with less than 40 samples and clipped at the beginning the series with more than that. In total it was used a data set of 200 series of 40 samples each.

6.2.1 Gaze Cascade Effect Check

Applying the gaze cascade effect hypothesis mentioned in section 2.4, it should turn out that the object that got more attention is the one picked. It was studied the accumulation of attention of the object picked, as shown in the figure 6.3. This measure is the number of seconds of difference between the attention to the picked object minus the other object. The graph shows the average in the entire data set and it is evident how the attention on the chosen object increases as the action progresses. As well as is shown in figure 6.2 in average the cascade effect hypothesis is verified as valid to predict the object to be picked with a conservative 55-60% of success.

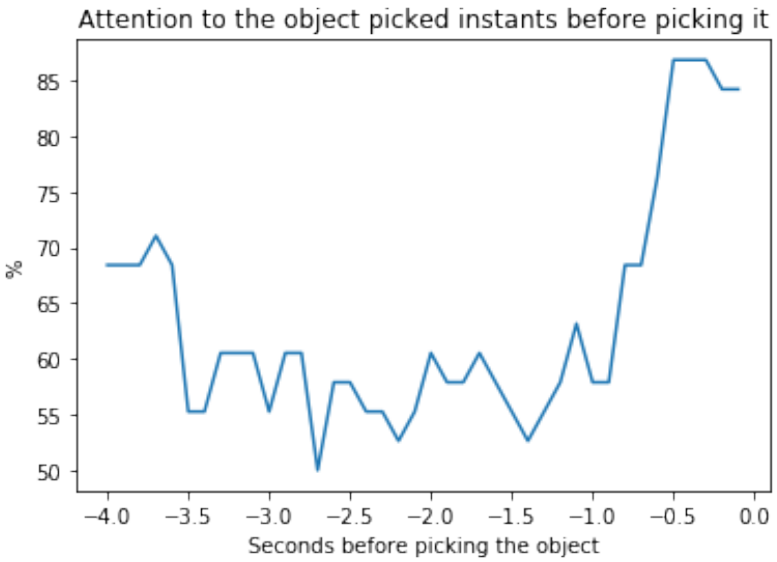

Figure 6.2: "Attention to the object picked seconds before picking it in the experiment with 2 objects. The X axis represents the time in the action remaining for the object to be picked, the Y axis represents the percentage of series in which the attention is on the object picked."

6.3 LSTM Check

To test the LSTM the data set has been divided in 60% for training, 20% for validation, and 20% for testing. The training data has been augmented by flipping the axis of the records (see figure 5.3) to have more data to work with, increasing it in a x4 ratio. The model is a vanilla LSTM with 2 layers, as input receive an array of 40 values and outputs a single value in the range of 0 to 1, this value is used for classification. The algorithm did not perform very well using this training data, showing a spike of accuracy at the end of the series while performing poor in the middle of it. It was put to the test training the model with a renewed data set which doesn't contain the last 0.5 seconds of the series, as previously mentioned, this is the verification stage where the attention is in the picked object and therefore the relation between gaze and object is obvious, what could make the training process very easy and biased. Pulling out this 0.5 seconds the accuracy successfully increased. In the figure 6.4 the accuracy of the different training can be compared, with a red line marking the 0.5 what tell us the accuracy of a random guess and in green the start of the verification stage. In blue is the model trained with the base data set, it can be seen how the algorithm perform worse than a random choice and increases as the attention to the object get maximized. In the other hand in orange is the model trained with the clipped data set, its performance is way better and it maintains an accuracy above 0.5 in the entire time line, reaching the 0.75 of accuracy around 2 seconds before the action ends, over-matching the cascade effect

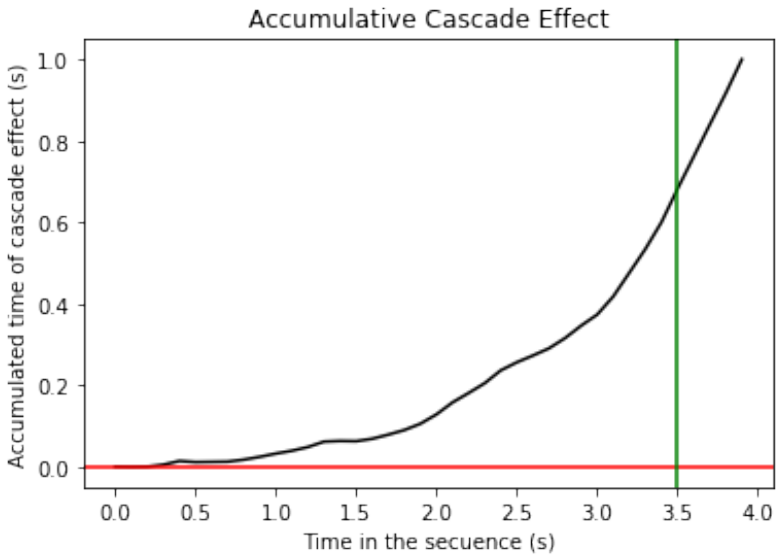

Figure 6.3: "Accumulated attention to the object picked in the experiment with 2 objects. The X axis represents the time in the action, the Y axis the seconds of attention ahead for the object picked. In red is the accumulation 0 that means equal attention, in green the second 3.5 as stated in section 6.1 the start of verification stage."

accuracy.

For classification is needed a boundary of decision at the end of the last LSTM layer, this is typically implemented with a fully connected neural network. For this test where there are only 2 classes, and then one boundary of decision, it has been used a simple binary discriminator with a threshold of 0.5, training with a fully connected neural network at the end did not increase the accuracy of the model.

6.4 Experiment with 3 Objects

For the experiment with 3 objects 302 series were gathered. As shown in figure 5.4 the distribution of the length of action is between 3 and 8 seconds. with most of them around 4 seconds. For this reason the series shorter than 3 seconds and longer than 8 seconds were ruled out, at the same time it was decided to use a length of 60 samples , remember sample rate of 10Hz, as in the previous experiment clipping the series longer than 60 samples and padding the beginning for the series shorter than that. In total it was used 219 series of 60 samples each.

6.4.1 Gaze Cascade Effect Check

As in the previous experiment, it was studied the cascade effect for the 3 objects case. In figure 6.5 is shown the percentage of series where the attention is on the

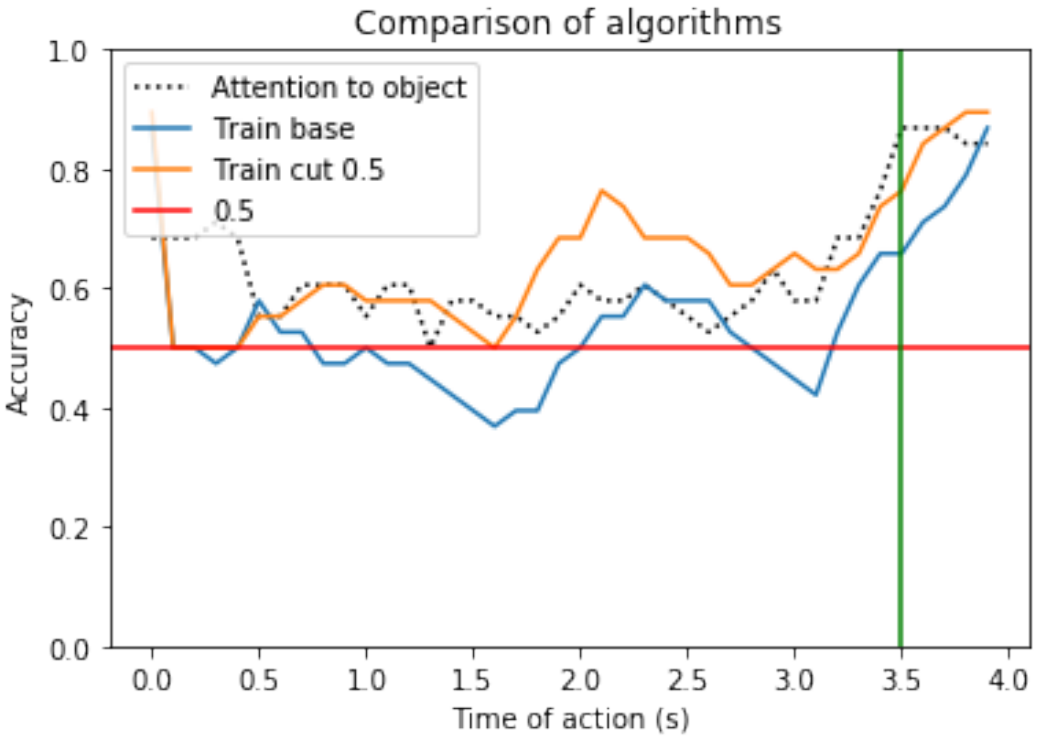

Figure 6.4: "Comparison of 2 training methods for the experiment with 2 objects. In blue training with the normal data, in orange training without the last 0.5 seconds. In black dash-line the attention to the object picked, in red the 0.5 value and in green the start of the verification stage."

object picked in the time of the action, minding that there are 3 objects a value of 33% means a random attention. For the first 3 seconds the attention fluctuates between 25-45% when the second 3 seconds the attention moves between 40-60%. The insight is that in the first half the participant is exploring the different options and in the second half is taking the decision. In figure 6.6 is shown the accumulated cascade effect that plots the seconds of attention that the object picked is ahead compared to the rest of the objects, this value is weighted for 3 objects. This come to reaffirm what is stated in the previous lines, that is in the second half of the series where the fixation on the object to pick is higher.

6.4.2 LSTM Check

As in the previous experiment, the data has been divided in 60% for training, 20% for validation and 20% for testing. The training data has been augmented in the same fashion. The model is again a vanilla LSTM with 2 layers receiving as input an array of 60 samples and outputs a single value in the range of 0 to 1 that is used for classification. It was trained with the base data set and with the data

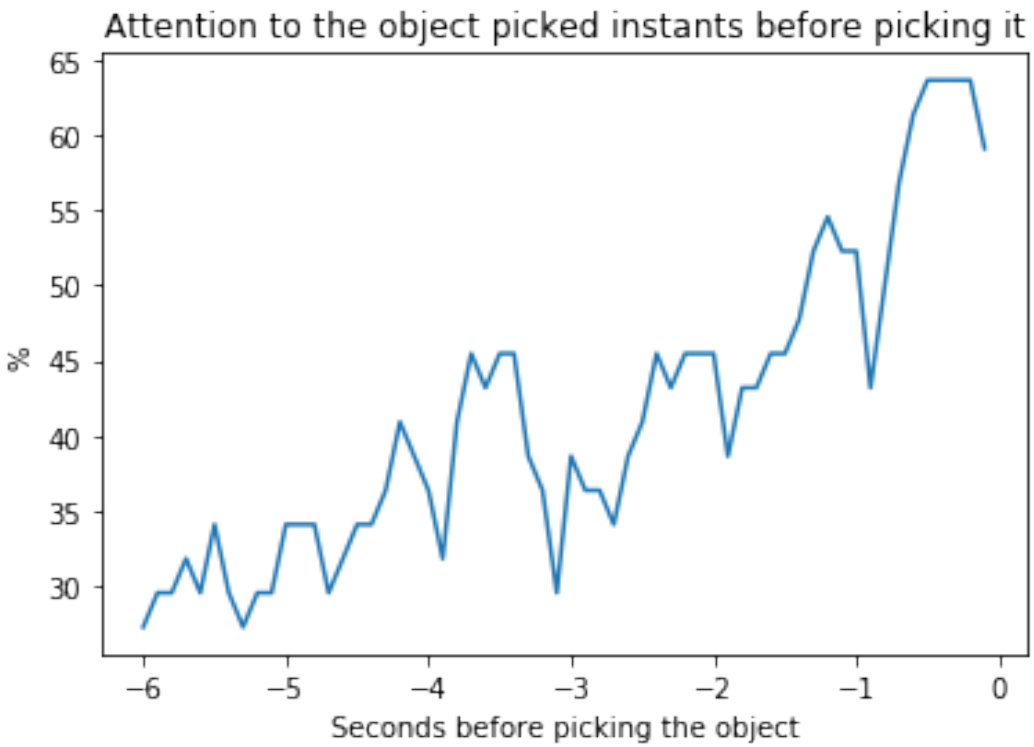

Figure 6.5: "Attention to the object picked seconds before picking it in an experiment with 3 objects. The X axis represents the time in the action remaining for the object to be picked, the Y axis represents the percentage of series in which the attention is on the object picked."

set pulling out the last 0.5 seconds. In the figure 6.7 its shown the results for the two training methods, it is presented in relation to the time in the action, how the models predict at this point. In blue is the model trained with the base data set and in orange the one trained with the data set without the last 0.5 seconds, the black dash-line shows the average attention on the object that is picked and in red the line at 0.33 which is the value for a random guess when there are 3 possibilities. The graph show that the results for both training are tight and perhaps the one without the last 0.5 seconds perform slightly better. The best perform of the models occur after the second 3 as mentioned in the previous point this could lead to think that in the first 3 seconds in the series the POG is somehow random and there isn't a pattern while the first scan of the objects. At the same time the result of the attention in the picked object is in the same order as the LSTM models results. The final outcome is that using any of these methods we can achieve around 0.5 of accuracy 2.5 seconds before the object is picked, what is an improvement from the 0.33 of accuracy in a random guess.

As in the previous experiment, for classification it has been used fixed decision

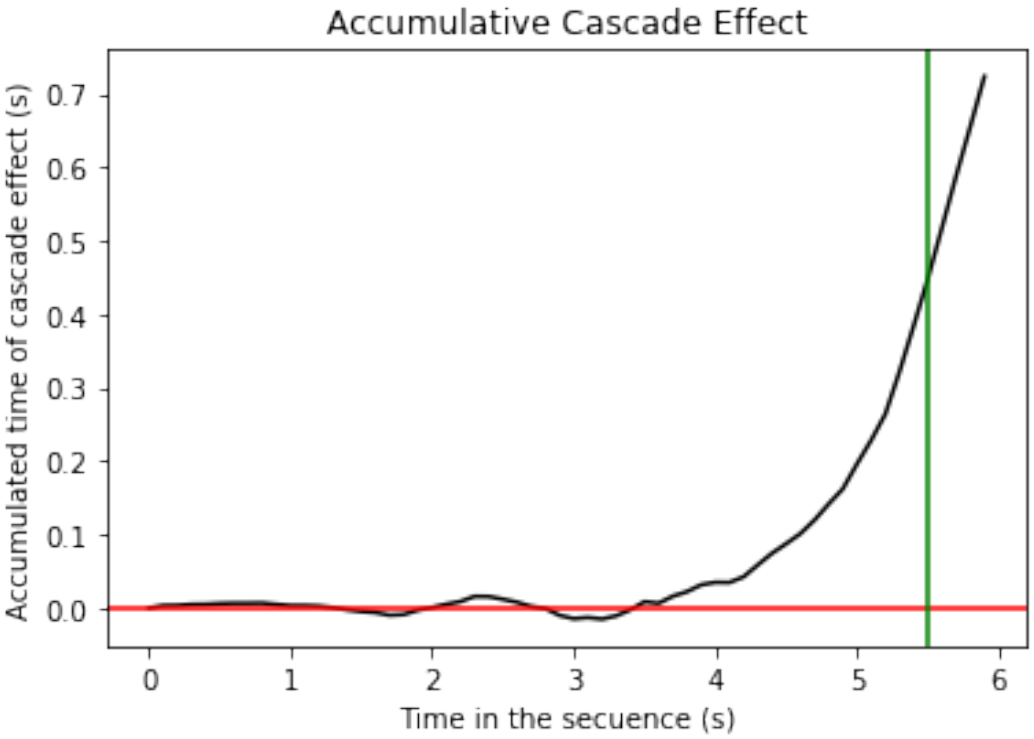

Figure 6.6: ""Accumulated attention to the object picked in the experiment with 3 objects. The X axis represents the time in the action, the Y axis the seconds of attention ahead for the object picked. In red is the accumulation 0 that means equal attention for the 3 objects, in green the second 5.5 as stated in section 6.1 the start of verification stage."

boundaries in this case 0.33 and 0.66. An optimization of this boundaries using the training data didn't improve the outcome of the final models. At the same time using a fully connected neural network at the end of the last LSTM layer didn't improve the accuracy either.

6.5 Other Machine Learning Configurations Tested

Apart for the model tested in both experiments, as mentioned, a vanilla LSTM with 2 layers, it was tested other distributions with one or more than 2 layers. The results for using a single layer are very poor, giving an accuracy around the random guess. The models with more than 2 layers performs with same accuracy as the model with only 2 layers. This may be expected as the data is not too complex in terms of dynamic range and also in time. This might variate with a more detailed data set. Lastly it was also tested using different configurations of fully connected neural networks at the end of the model to improve the classification, this didn't bring better accuracy than using naive boundaries of decision, and it is understandable

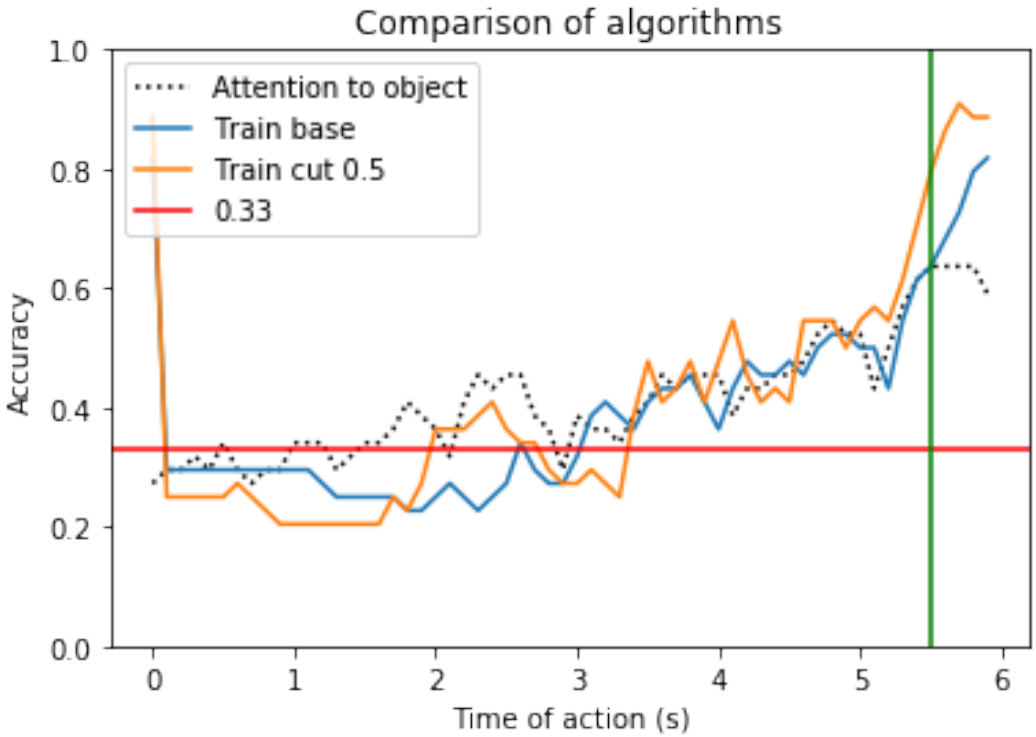

Figure 6.7: "Comparison of 2 training methods for the experiment with 3 objects. In blue training with the normal data, in orange training without the last 0.5 seconds. In black dash-line the attention to the object picked, in red the 0.5 value and in green the start of the verification stage."

since the amount of classes is very low (2 or 3).

Chapter 7

Conclusion

After all the exposed in this project and with the results from the experiments it is difficult to give an easy conclusion. In one hand it is fair to admit that the data set is short in quantity and it could be more detailed in time and dynamic range. However, in the other hand the results with the methods exposed give a shine of hope for future investigation in this matter, if is used more powerful tools. With a 0.75 of accuracy 2 seconds before the object is picked for the 2 objects experiment, when a random guess accuracy is 0.5, and 0.5 accuracy 2.5 seconds before the object is picked for the 3 object experiment, when a random guess is 0.33, we can conclude that with the data available it is a big success. Also minding that the algorithms used are very simple and not high computational demanding, therefore they can run in real time on a conventional CPU. But in the overall more experiments have to be done to find an accuracy more generalized and see how far is it possible to get in the matter.

Without a doubt this open a door for the intention recognition in the field of collaborative robots, where humans and robots share the work space and collaborate in repetitive tasks. Having real time information of the POG in a controlled environment, and an algorithm that give insights of a possible incoming human reaction, it could be possible for a robot to anticipate with certain accuracy what the human is going to do next: picking an object, leaving the table, grabbing a tool, even feelings like fear to be hit by the robot. The use of this inputs to adjust the robot behaviour, or in an ideal case, been able to interact with a human just by his intention recognition, are features that can improve the relation human-robot. While we are waiting for the technology to bring eye tracking to a work bench for professional use, it turns out possible to study and analyse possible scenarios using VR, it is matter of gathering enough data, as much detailed as possible, and put to the test these or other algorithms, taking advantage of the opportunity that virtual reality simulation offers us.

Carlos Gomez Cubero
cgomez18@student.aau.dk
Frederikstorv 1, 3Th
9000 Aalborg

Bibliography

[1] T. E. Hutchinson et al. "Human-Computer Interaction Using Eye-Gaze Input"". In: (1989). URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=44068.

[2] Y. Ebisawa et al. "Non-invasive eye-gaze position detecting method used on man/machine interface for the disabled". In: *[1991] Computer-Based Medical Systems@m_Proceedings of the Fourth Annual IEEE Symposium*. 1991, pp. 374–380.

[3] C. A. Hennessey* and P. D. Lawrence. "Improving the Accuracy and Reliability of Remote System-Calibration-Free Eye-Gaze Tracking". In: *IEEE Transactions on Biomedical Engineering* 56.7 (2009), pp. 1891–1900.

[4] Y. Ebisawa and K. Fukumoto. "Head-Free, Remote Eye-Gaze Detection System Based on Pupil-Corneal Reflection Method With Easy Calibration Using Two Stereo-Calibrated Video Cameras". In: *IEEE Transactions on Biomedical Engineering* 60.10 (2013), pp. 2952–2960.

[5] M. Z. C. Azemin, M. I. M. Tamrin, and A. A. Arshad. "Validation of low-cost eye tracking setup for smooth pursuit application". In: *2014 IEEE Conference on Open Systems (ICOS)*. 2014, pp. 123–127.

[6] H. Ho. "Low cost and better accuracy eye tracker". In: *2014 International Symposium on Next-Generation Electronics (ISNE)*. 2014, pp. 1–2.

[7] Y. Wang, H. Zeng, and J. Liu. "Low-cost eye-tracking glasses with real-time head rotation compensation". In: *2016 10th International Conference on Sensing Technology (ICST)*. 2016, pp. 1–5.

[8] A. Bulling and H. Gellersen. "Toward Mobile Eye-Based Human-Computer Interaction". In: *IEEE Pervasive Computing* 9.4 (2010), pp. 8–12.

[9] E. D. Guestrin and M. Eizenman. "General theory of remote gaze estimation using the pupil center and corneal reflections". In: *IEEE Transactions on Biomedical Engineering* 53.6 (2006), pp. 1124–1133.

[10] B. Farnsworth. "Types of Eye Movements[Saccades and Beyond". In: (2019). URL: https://imotions.com/blog/types-of-eye-movements.

[11] Dario Salvucci and Joseph Goldberg. "Identifying fixations and saccades in eye-tracking protocols". In: Jan. 2000, pp. 71–78. DOI: 10.1145/355017.355028.

[12] L. Cooke. "Is Eye Tracking the Next Step in Usability Testing?" In: *2006 IEEE International Professional Communication Conference*. 2006, pp. 236–242.

[13] R. Wierts, M. J. A. Janssen, and H. Kingma. "Measuring Saccade Peak Velocity Using a Low-Frequency Sampling Rate of 50 Hz". In: *IEEE Transactions on Biomedical Engineering* 55.12 (2008), pp. 2840–2842.

[14] Shimojo et al. "Gaze bias both reflects and influences preference". In: (2003). URL: https://doi.org/10.1038/nn1150.

[15] D.Bird et al. "The role of eye movements in decision making and the prospect of exposure effects". In: (2012). URL: https://doi.org/10.1016/j.visres.2012.02.014.

[16] J. E. Russo and F. Leclerc. "An Eye-Fixation Analysis of Choice Processes for Consumer Nondurables". In: (1994). URL: https://www.researchgate.net/publication/24098870_An_Eye-Fixation_Analysis_of_Choice_Processes_for_Consumer_Nondurables#fullTextFileContent.

[17] Kerstin Gidlöf et al. "Using Eye Tracking to Trace a Cognitive Process: Gaze Behaviour During Decision Making in a Natural Environment". In: (2013). URL: https://bop.unibe.ch/JEMR/article/view/2351/3547.

[18] Jorge Castañon. "10 Machine Learning Methods that Every Data Scientist Should Know". In: (2019). URL: https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9.

[19] Tim Hwang. "Computational Power and the Social Impact of Artificial Intelligence". In: *CoRR* abs/1803.08971 (2018). arXiv: 1803.08971. URL: http://arxiv.org/abs/1803.08971.

[20] Sidath Asiri. "Machine Learning Classifiers". In: (2018). URL: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623.

[21] Apoorva Dave. "Regresion in Machine Learning". In: (2018). URL: https://medium.com/datadriveninvestor/regression-in-machine-learning-296caae933ec.

[22] Tony Yiu. "Understanding RNNs (Recurrent Neural Networks)". In: (2019). URL: https://towardsdatascience.com/understanding-rnns-recurrent-neural-networks-479cd0da9760.

[23] Michael Phi. "Illustrated Guide to LSTM's and GRU's: A step by step explanation". In: (2018). URL: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

[24] Y. Li and Y. Lu. "LSTM-BA: DDoS Detection Approach Combining LSTM and Bayes". In: *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*. 2019, pp. 180–185.

[25] Y. Zeng and X. Liu. "A-Stock Price Fluctuation Forecast Model Based on LSTM". In: *2018 14th International Conference on Semantics, Knowledge and Grids (SKG)*. 2018, pp. 261–264.