THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Gaze Based Human Intention Analysis

Supported by Virtual Reality and AI

Julius Pettersson

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2023

Gaze Based Human Intention Analysis
Supported by Virtual Reality and AI

Julius Pettersson
ISBN 978-91-7905-860-9

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5326
ISSN 0346-718X

This thesis has been prepared using LaTeX.

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Digitaltryck
Gothenburg, Sweden, May 2023

To family and friends, eye thank you.

Abstract

The ability to determine an upcoming action or what decision a human is about to take, can be useful in multiple areas, for example during human-robot collaboration in manufacturing, where knowing the intent of the operator could provide the robot with important information to help it navigate more safely. Another field that could benefit from a system that provides information regarding human intentions is the field of psychological testing where such a system could be used as a platform for new research or be one way to provide information in the diagnostic process. The work presented in this thesis investigates the potential use of virtual reality as a safe, measurable, and customizable environment to collect gaze and movement data, eye tracking as the non-invasive system input that gives insight into the human mind, and deep machine learning as one tool to analyze the data. The thesis defines an experimental procedure that can be used to construct a virtual reality based testing system that gathers gaze and movement data, carry out a test study to gather data from human participants, and implement artificial neural networks in order to analyze human behaviour. This is followed by two studies that gives evidence to the decisions that were made in the experimental procedure and shows the potential uses of such a system.

Keywords: Virtual reality (VR), time series analysis, human intention prediction, eye tracking, deep machine learning, uncertainty estimation, collaborative robots, psychological testing.

List of Publications

This thesis is based on the following publications:

[A] **Julius Pettersson** and Petter Falkman, "Human Movement Direction Classification using Virtual Reality and Eye Tracking". *Published in Procedia Manufacturing, Volume 51*, (pp. 95-102), 2020.

[B] **Julius Pettersson** and Petter Falkman, "Intended Human Arm Movement Direction Prediction using Eye Tracking". *Re-submitted to:* IJCIM International Journal of Computer Integrated Manufacturing, 2022.

[C] **Julius Pettersson** and Petter Falkman, "Comparison of LSTM, Transformers, and MLP-Mixer Neural Networks for Gaze Based Human Intention Prediction". *Accepted in:* Frontiers in Neurorobotics, 2023.

[D] **Julius Pettersson**, Anton Albo, Johan Eriksson, Patrik Larsson, Kerstin W. Falkman, and Petter Falkman, "Cognitive Ability Evaluation using Virtual Reality and Eye Tracking". *In 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, (pp. 1-6), 2018.

[E] **Julius Pettersson**, Kerstin W. Falkman, and Petter Falkman, "Exploring the usability of Virtual Reality and Eye Tracking for Psychological Testing using Raven's Progressive Matrices". *Submitted to:* Frontiers in Psychology, 2023.

Other publications by the author, not included in this thesis, are:

[F] Dahl, M., Albo, A., Eriksson, J., **Pettersson, J.**, and Falkman P., "Virtual Reality Commissioning in Production Systems Preparation". *In 2017 22nd IEEE International Conference on Emerging Technologies and Automation (ETFA)*, (pp. 1-7), 2017.

[G] **Pettersson, J.**, and Falkman P., "Human Movement Direction Prediction using Virtual Reality and Eye Tracking". *In 2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, virtual, 2021.

Acknowledgments

I would like to extend my deepest gratitude to Petter Falkman, my main supervisor, who played a decisive role in both the start, continuation, and finish of my PhD. Your advice, encouragements, and our sincere conversations have been crucial to my work and my personal growth. I would also like to thank my co-supervisor Martin Fabian and the senior members of our research group.

Writing papers can be difficult. Writing them with a target audience outside of one's own area of expertise is harder. I could not have ventured into the field of psychology without the guidance of Kerstin Falkman, it has been a privilege to collaborate with you.

I am also grateful to Ludvig, Rikard, and Alvin for our long discussions (and rants), which have been an important part of my progress and daily work. Likewise, the coffee breaks with the rest of my colleagues, Anton, Endre, and Sabino to mention a few. I would also like to thank Tobias for his contribution to our work, which later led me here.

The journey of a PhD is all about curiosity, overcoming obstacles, and learning new things. During my early years in school there are three persons that I feel especially grateful for: Bodil, the librarian, who always made sure to keep a fresh stack of books ready for me to read, and my two teachers, Margareta and Annika, who invested their time and energy into my development, and kept my young mind challenged.

Thank you to my dear friends: Emil, Johan E, Johan Ö, Pontus, Oscar, and Fredrik, for making sure that these years were filled with a healthy mix of adventures, laughs, and a wide selection of beer. Having you by my side kept me sane (sort of) when things got tough.

I could, of course, not have done this without my group of loyal fans, my family: my mom Helén, my dad Dick, my dear sister Ottilia, Tessie, my grandmother Anette, and the family at no. 120, Anders, Maria, Beatrice, and Alicia. Thank you for cheering me on and for providing a nurturing space for my holiday breaks! I will also take this opportunity to pay tribute to those who are no longer with us: Berne, Birgit, Arne, Inger, and Elov. I know you always believed in me.

Finally, thank you to my beloved Julia for your patience, kind words, and endless support. I could not have finished this work without you!

Acronyms

AI:	Artifical Intelligence
ANN:	Artifical Neural Networks
AOI:	Area Of Interest
CNN:	Convolutional Neural Networks
DML:	Deep Machine Learning
ET:	Eye Tracking
FNN:	Feedforward Neural Networks
HMD:	Head Mounted Display
I-VT:	Velocity-Threshold Identification
LSTM:	Long-Short Term Memory
ML:	Machine Learning
MLP:	Mult-Layer Perceptron
RNN:	Recurrent Neural Networks
RPM:	Raven’s Progressive Matrices
SML:	Supervised Machine Learning
SPM:	Raven’s Standard Progressive Matrices
UE:	Uncertainty Estimation
VR:	Virtual Reality
VRE:	Virtual Reality Environment

Contents

Part I

Overview

CHAPTER 1

Introduction

The ability to determine what actions or decisions a human is about to make can be useful in multiple areas, for example, in manufacturing where humans working with collaborative robots is becoming increasingly more popular [1]. The advantages of having humans and robots in the same workspace interacting with each other are many, such as; increased flexibility [2] and increased productivity for complex tasks [2]. However, the robots are still not that interactive since they cannot yet interpret humans and adapt to their swift changes in behaviour in a way that another human would do. The main reason is that the collaborative robots today are limited in their sensory input, which makes it the responsibility of the human to stay out of the way.

Another field that could benefit from a system that provides information regarding human intentions is the field of psychological testing. Testing of mental capacity has been around since the early 1900s and has been greatly extended since then [3]. These tests can be used, for example, to evaluate special abilities, intelligence, and social attributes as described in [3]. There is, however, potential to improve these methods even further using the technology that is at hand today. It is often, during certain tests where the participant is asked to complete a specific task, as important to observe the person's behavior during the experiment as to obtain the actual test results [4]. The authors of [4] further describe that the results will be affected if the person taking the test is anxious, showing signs of speech or language difficulties, or has difficulties concentrating.

Other fields that have been rapidly expanding and that may be used to provide an understanding of human behaviours and intentions are; virtual reality (VR), eye tracking (ET), gathering and management of large datasets, and artificial intelligence (AI).

Eye-tracking (ET) is an objective, painless, and noninvasive [5] way to gather more insight into how a person is reasoning from measurements and analysis of where the person is directing their gaze [6]. It is possible to gain insight into the alternatives a person is considering or what strategy is used while performing a task. ET has, for example, been used in industrial contexts with gaze as a machine control input [7], to evaluate new ways to facilitate human–robot communication [8], to analyze the navigational intent in humans and how they interact with autonomous forklifts [9], and to investigate pedestrians' understanding of an autonomous vehicle's intention to stop at a simulated road crossing [10]. It has also been used as a tool to diagnose autism [11], where children participated in different games and social activities on a tablet while their gaze was observed.

VR can be described as a technology through which visual, audible, and haptic stimuli is able to give the user a real world experience in a virtual environment [12]. Benefits such as being able to provide more relevant content and present it in a suitable context are reasons [13] uses to promote the use of VR in neuropsychological testing. The authors highlight the possibilities and benefits of measuring data using VR, such as accuracy, timing, and consistency to enhance the analysis. VR has proved a useful tool, for example when observing the level of distraction in children with ADHD [14], [15]. It can also be used in an industrial context; when making prototypes [16], to train operators in assembly [17], and improve remote maintenance [18].

The use of modern technologies such as ET and VR makes it possible to collect larger amounts of data, with higher accuracy, and at a higher pace than before [19]. These large volumes of data, created at high speed, and with great variety [20] is referred to as Big Data. One area of AI that can be used to process these huge datasets is called deep machine learning [21]. Big data and AI has been shown to be important tools for the future to improve industrial manufacturing [22]–[24], as well as providing benefits in the field of psychology, for example, when analyzing how students perform on cognitive diagnostic assessments [25] and to determine if a person has ADHD [26].

Human intention prediction can be achieved using camera images and probabilistic state machines [27] with the goal of determining between explicit and implicit intent. It can also be achieved using 3D-vision, speech recognition, and wearable sensors with the objective of predicting intention in hand-over tasks [28]. It was proposed by [29] to use a Gaussian Mixture Model and data from a Kinect camera to predict human motion, reporting about 80% classification accuracy, on 8 movement classes, after 60% of the trajectory has been observed. Other ways are to monitor eye gaze to predict an upcoming decision [30] for robot control or analyze bioelectric signals, such as electromyography, to predict human motion [31]. In the paper by [32] it is shown that monitoring human eye gaze can be used to recognize actions related to pouring and mixing a powder based drink. [33] presents a way of using Earth Mover's Distance to calculate the similarity score between the hypothetical gazes at objects and the actual gazes to determine if the human visual intention is on the object or not. It was shown by [34] that it is possible to use a Kinect camera to capture eye gaze and arm movements, and use that information to predict the goal location of a reaching motion, reporting a success rate of above 80% after 40% of the trajectory has been observed. The work by [35] shows that it is possible to use an artifical neural network, together with wearable ET, to predict intention regarding which object is about to be picked out of three objects in a VRE. They achieve an accuracy between 70-80% for test sequences that are 3-14 s long, using the gaze projected on the surface where the objects are placed.

1.1 Research Questions

The ability to determine an upcoming action or what decision a human is about to take, can be useful in multiple areas, for example, in manufacturing where humans are working with collaborative robots and in psychological testing where such a system could be used as a platform for new research or be one way to provide information in the diagnostic process. A way to gather more insight into how a person is reasoning is to measure and analyze where the person is looking [6] using ET. There are multiple ways of tracking gaze and one of them is through VR. The data that is collected needs to be analyzed and one area of AI that can be used to process these datasets is called deep machine learning [21] (DML). This is the basis for the following research questions:

RQ1: *Is it possible to analyze and predict human intention through the study of eye gaze?*

Understanding human intention is becoming increasingly important, as described earlier. There are several ways of achieving this, for example, using camera images, electromyography, or a combination of eye gaze and movement tracking. The eye gaze can reveal what alternatives a person is considering or what search strategy is used while performing a task. The goal of RQ1 is to investigate if it is possible to analyze and predict human intention through the study of eye gaze.

RQ2: *Is DML a suitable tool to analyze the connection between eye gaze and intention in humans?*

DML has shown to be a powerful tool to analyze large amount of complex data in multiple research fields, including industrial applications and psychological research. RQ2 aims at exploring if the combination of eye tracking and DML could be used as a flexible tool to analyze the connection between eye gaze and human intention as different tasks are being carried out.

RQ3: *How can a VRE-test be designed to gather the necessary eye gaze and movement data to be used for human intention analysis?*

VR has successfully been used in both industry and psychological research. The benefits of using a VRE includes, for example, being able to continuously gather data from both the user and the environment, and it gives the developer of the test full control over all events in the VRE while providing the user with an experience that is similar to a real world application. RQ3, therefore, aims at determining a procedure for how a VRE can be designed to gather eye gaze and movement data, from human participants, that can be analyzed to improve the understanding of human intention.

1.2 Thesis Outline

The thesis consists of two parts, Part I features an overview of the research and Part II contains the publications that constitute the basis of the first part. Part I starts off in Chapter 1 with an introduction to the field that has been researched, followed by theoretical introductions to the areas of psychological testing, virtual reality (VR), eye tracking (ET), and supervised machine learning (ML) in Chapter 2, Chapter 3, Chapter 4, and Chapter 5 respectively. The experimental procedure of using VR, ET, and ML for a data-driven problem is introduced in Chapter 6, followed by Chapter 7 that covers the presentation of two studies of human intention. The summary of the included papers is given in Chapter 8 and the thesis ends with concluding remarks and suggestions for future work in Chapter 9.

CHAPTER 2

Psychological Testing

In [36] the aim of a psychological test is described as a method to measure different abilities and conditions that cannot be directly observed, such as intelligence, psychopathology or neuropsychological disorders. Psychological tests are often standardized to ensure validity and reliability.

A psychological test is usually designed with a particular population in mind. An individual's result on the test is always presented in relation to that population, on an appropriate scale, for example IQ in cases where intelligence is measured. In a process called standardization, the test is used with a representative sample of the population [4]. From this group's mean values and variance, you then generate a function from raw points to the desired scale.

The reliability and validity of the test, i.e. if the same results are achieved as the measurements are performed multiple times and how well it measures what it intends to measure [4], also has to be calculated. One way to ensure reliability is to standardize the test procedure, for example making sure that the instructions given to the test person are always the same and that the environment in which the test is performed is the same [4], i.e. there is no external interference.

Another key element is to inform the participants about the premise of the testing and what their information will be used for to make them feel comfortable before giving their consent to participate [4]. There are additional factors, described by [4], that might affect the test results and/or the behaviour of the individual being tested such as anxiety, difficulties to concentrate or to communicate.

When collecting data for psychological research through the use of psychological testing this is mostly done manually. This means that researchers are often limited in the amount and types of data that can be collected. Observations of behaviour are, for example, made in real-time or through watching video recordings [37] of the test participant.

2.1 Digitalization in the Field of Psychology

Virtual Reality (VR) is a technology that has proved useful within psychology, for example as a tool to observe the level of distraction amongst children with ADHD [14], [15]. The authors of [13] highlights the possibilities and benefits of measuring data using VR, such as accuracy, timing, and consistency, to enhance the analysis. The research in [38] shows that VR can be used to interact with children through facial emotions and expressions. It can also be of great use in the process of treating and rehabilitating arachnophobia [39].

Another field of technology, that is already part of psychological research today, is the study of eye gaze movement. The eyes contain multiple levels of information, for the sender as well as the receiver, about the environment, emotional states, and mental states [40]. Assessing eye movement through ET is already widely used today. It is, for example, used for research purposes, in areas such as theory of mind [41] (the ability to imagine other peoples feelings and perspective), diagnosing autism [11], as an assistive tool for people with mobility difficulties [42], evaluating responsiveness to joint attention in infants, as well as in diagnosing Williams syndrome, ADHD, and reading disabilities [43]–[45].

Previous research has also shown that ML has potential within psychology to predict and increase our understanding of behaviour [46]. Furthermore, a study has shown that ML is efficient in facial recognition to determine facial expressions [47]. Consequently this could provide another parameter towards the purposes of analyzing an individual's behaviour since facial expressions are closely tied to emotion [48]. Another study by [25] shows that both supervised and unsupervised artificial neural networks (ANNs) can be used to analyze how students perform on cognitive diagnostic assessments. It has also been shown in [26] that ANNs can be used to determine if a person has attention deficit hyperactivity disorder (ADHD) and results by [49] indicate that ML can be used for automated test scoring of a novel story recall task.

2.2 Raven's Progressive Matrices

The concept of general cognitive ability, the *g* factor, was first introduced by the English psychologist Charles Spearman in 1904 [50]. To distinguish the differences between general intelligence and specific abilities while performing different tasks, [50] also states a second factor named *s*. The *g* factor has two main components; the capacity to think clearly and make sense of complex data, called educative ability, as well as the capacity to store and reproduce information, called reproductive ability [51]. The *s* factor is often represented by a circle with four elements; spatial, logical, mechanical and arithmetical abilities [52].

Raven's Progressive Matrices (RPM) are a set of tests designed to measure abstract reasoning and *g* factor [53]. They are well known and widely used since they are easy to administer and to interpret in a clear way [54]. The RPM are graphically easy to implement in a virtual environment, and are thus well suited to implement in VR. These tests are available in three different forms; Standard Progressive Matrices (SPM), Colored Progressive

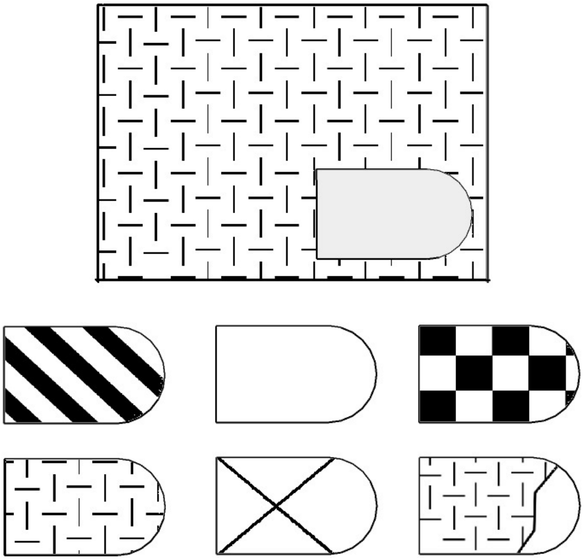


Figure 2.1: A figure that shows an item from Raven's Standard Progressive Matrices (SPM).

Matrices (CPM) and Advanced Progressive Matrices (APM). These different versions are intended to be used for testing people with varying cognitive and physical abilities where SPM is the most widely used and was intended to be used once the intellectual capacity to reason has developed, age 8 and above. The CPM, on the other hand, was designed to be used before this ability has developed [55], age 5-11, and the APM was developed to be used on adults and adolescents with over-average intelligence.

The SPM test consists of 60 items divided into 5 sets (A-E) of increasing difficulty and was first published in 1938 [51]. Each set follows a different logic that progressively increases in difficulty [56] with each set becoming more difficult than the previous. Each item has a logical pattern where one piece is missing and the task is to select the correct alternative amongst a given set of alternatives, which varies from six to eight depending on the item and level of difficulty. An example from SPM of how these items may look can be seen in Figure 2.1.

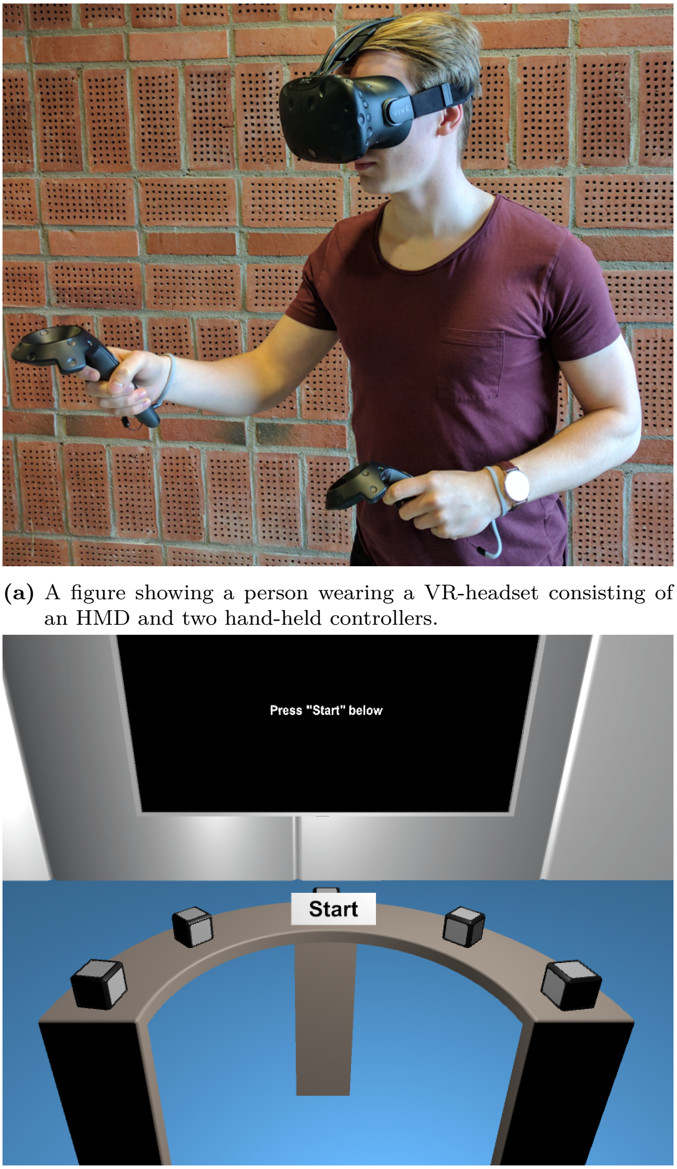
In the summer of 2022, Raven 2 was launched, which is a revised version [57]. A unified version of the test that replaces all previous versions - Standard, Coloured and Advanced, and both classic, parallel, and plus versions. All tasks in Raven's 2 are newly designed. The test can be administered as a traditional pen-and-paper test but is also available in a digital, i.e computerized version. As the revised version of RPM was not yet available at time of data collection the previous version was used in the thesis.

CHAPTER 3

Virtual Reality

Virtual Reality (VR) is the technique of 3D immersion in a computer created environment. A device that can be used to visualize the VR environment (VRE) to the user is a head mounted display (HMD) [12]. The HMD is equipped with sensors that measure the user's head motions and a display that is responsible for providing the user with the visual content. The system also provides the user with audible and haptic stimuli to immerse the user in a real world experience [12] of the VRE. An example of a person wearing an HMD can be seen in Figure 3.1a and the user's view of the VRE from **Paper A** is shown in 3.1a. There are other ways that can be used to visualize a VRE, e.g. CAVE [58] that projects images on the walls of a physical room.

VR technology is spreading to new areas with a steady increase in overall usage [59], for example, in the field of psychological testing [60]. It has been used to measure the distraction level of children with attention deficit hyperactivity disorder (ADHD) [14], [15], in a virtual classroom. The research by [38] shows that VR can be used to interact with children through facial emotions and expressions and it can also be used in the treatment of phobias, e.g., arachnophobia [39]. Benefits such as being able to provide more relevant content and present it in a suitable context [13] are other reasons to promote the use of VR. It can also be used in other areas, for example; when making prototypes [16], to train operators in assembly [17], and to improve remote maintenance [18].



(a) A figure showing a person wearing a VR-headset consisting of an HMD and two hand-held controllers.

(b) The user's view of the VRE from **Paper A**.

Figure 3.1: An example of a VR-headset and the view from inside the HMD.

CHAPTER 4

Eye Tracking

Eye tracking (ET) is defined by [61] as the technique of measuring what a person is looking at, in what order the objects are gazed upon, and for how long the eye gaze stays fixed on that object. The eye gaze is an interesting biological marker because it is possible to analyze underlying neurophysiology based on the movement of the eyes [5]. Tracking gaze is therefore an appealing test method and also because it is objective, painless, and noninvasive [5]. ET can give an insight into the individual's problem solving, reasoning, and search strategies [61]. However, ET is only capable of tracking visible movements of the eye and not the hidden mental processes of visual attention [62]. This makes for the simplified assumption, when using ET for attention analysis, that attention is associated with gaze direction [62] even though that is not always the case.

One way of tracking the eyes, as described in [61], is achieved by illuminating them with infrared light, which is used to prevent the user from being dazzled, to get a clear reflection that is captured using a camera. The reflections are then used to calculate a vector of the relationship between the cornea and the pupil [61], which in turn is used to calculate the gaze direction.

ET has, for example, been used to analyze the navigational intent in humans and how they interact with autonomous forklifts [9], to analyze the prospective memory for delayed intentions in children [63], to investigate pedestrians' understanding of an autonomous vehicle's intention to stop at a simulated road crossing [10], to allow people with severe speech and motor impairments to move a robotic arm [64], and to predict which one out of four tasks, where the participants aligned two cubes in various ways in a VRE, that was carried out [65].

4.1 Eye Movements

The way humans react to a visual stimuli is dependent on many factors [66], e.g. for a simple task we may be interested in determining if something is present or what it is, called detection and identification respectively, and for a more complex situation the goal might be to detect a target in a larger visual field of many targets.

In order for us humans to observe an object in the real world, we have to fixate our gaze at it for long enough time so that the brain's visual system is able to perceive it [67]. We are only able to see a very narrow visual scene with high acuity at any point in time [67] and to observe a larger area with acuity we need to continuously scan it with small rapid movements so called saccades. The fovea is a small area on the retina that is responsible for providing this high-acuity vision [67] using the lens that focuses the light coming from the pupil on this area that is densely populated with a type of photoreceptive cells, called cones, that are sensitive to small objects, color, and contrast [62]. However, the density of these cells decreases rapidly in the periphery, reducing acuity. The periphery on the other hand mostly contains another type of cells called rods, these are sensitive to light, shade, and motion [62], [67]. The peripheral vision is, instead of providing high-acuity, giving us information [67] about where to look next and what changes or movements that occur in the visual field.

There are three types of positional eye movements that are of interest when observing the visual attention [62]: fixation, saccades, and smooth pursuits. These movements are defined as follows:

- **Fixations** are tiny movements resembling random noise no larger than 5° visual angle that are stabilizing [62] over a specific area of interest and are said to correspond to one's desire to maintain the gaze on a specific object. These movements range between 150–600ms in duration and about 90% of the viewing time is spent on them [62].
- **Saccades** are [62] rapid eye movements, ranging between 10-100ms in duration, that are used to reposition the fovea to a new location such that a new area of the environment can be visualized. These movements occur as both corrective adjustments of the eye as well as voluntarily controlled eye movements [62] that are used to change the focus of attention.
- **Smooth pursuits** are movements that are used to visually track a moving target [62] and refers to the fact that the eyes, depending on the target movement range, are able to keep up with the velocity of the target.

Other, nonpositional, eye movements are adaptation and accommodation [62] (i.e., pupil dilation, lens focusing).

Detection of Fixations and Saccades

The main goal of ET is, according to [62], to distinguish between the three positional movement types mentioned above. This is done through the localization of regions where the ET signal switches between two stationary values, i.e. fixations, where the sharp edges of the changes are the saccades. There are several metrics that can be used to extract further information from the fixations and saccades, e.g. [62] fixation duration, fixation count, saccade amplitude, and saccade count.

There are mainly two automatic ways [62] to perform this analysis, the first one being averaged summations and the second one is through differentiation. The first one, also referred to as the "dwell-time" method, averages the ET signal over time and if it remains within what can be seen as low variance for longer duration than a specific threshold it is classified as a fixation [62]. The second method, on the other hand, subtracts consecutive data points to estimate the velocity of the eye movements [62], which requires that the ET is performed using a fixed sampling rate. Fixations are extracted from these velocities either as the segments that occur between saccades, or as the segments where the velocity falls below a predefined threshold [62]. There are indications that the second method is better for real-time detection of saccades [62] due to faster calculations. The thresholds, for both methods, are often determined through empirical studies [62].

One of the main issues of ET analysis it that the recorded signal is inherently noisy [62] due to the eye's constant movements and also as a result of eye blinks. Filtering the data before it is used is therefore of importance. Eye blinks should, according to [62], generally be easy to distinguish since they create a large disturbance in most eye trackers.

I-VT filter

The velocity-threshold identification (I-VT) filter is a spatial (velocity-based) algorithm [68] that is used to distinguish fixations from saccades in eye gaze data. The intuition behind the algorithm is that fixations have low velocities ($< 100°$/sec) while saccades have high velocities ($> 300°$/sec) [68]. The I-VT algorithm works as follows [68];

1. Calculate point-to-point velocities.
2. Classify each point as either a fixation, if its below a specified threshold, or as a saccade if its above it.
3. Group consecutive fixations together and calculate the center point of each group based on the center of mass.
4. Set the start time for the fixation as the time of the first point in the group and the duration of the fixation as the time between the first and last point in the group.

The velocity threshold that is used is the only parameter that needs to be specified [68] and it can be set to what is considered a reasonable angular velocity based on computation of angular velocities (requires the distance from eye to visual stimuli to be known) or simply using the sampling frequency in conjunction with empirical data. A study by [69] shows that a threshold between $20 - 40°$/s are suitable values to try for specific eye trackers whereas a threshold of 30°/s may be a suitable trade-off when working with a multitude of eye trackers.

Other things to consider for the use of the I-VT filter, apart from the selection of the threshold, are [70]:

- **Lack of smooth pursuit detection** - There is no distinction between fixations, or saccades, and smooth pursuits [70] and the latter will therefore always be classified as either a saccade or a fixation, depending on the velocity threshold.
- **Noisy data requires filtering** - All systems that are designed to perform measurements are generally noisy to some extent [70], these disturbances may come from the

equipment as well as from the environment. The way the eye movement velocities are calculated in the I-VT filter, as the fraction between the difference in angular position and the sampling frequency [70], means that if the eye tracker makes even the smallest miscalculations this will introduce significant noise in the velocities calculated from data collected at a high frequency. On the other hand, with eye trackers sampling at lower frequency, the noise introduced by measurement issues will typically still have the same amplitude as for higher frequencies, but as the time between each sample is greater, applying a filter introduces the risk of distorting [70] the original gaze data. Noise generally appears as random spikes in the data [70] and since it has a higher frequency than the signal the I-VT filter aims to detect, it is possible to reduce the noise using a low pass filter [70] that smooths the data by removing signals of high frequency. An alternative to the low pass filter is to calculate the average eye movement velocity over several samples, which is less sensitive to noise than using just two measurements [70].

- **Gap fill-in** - Another issue is that some loss of data is almost always present in digital measurement systems [70] occurring when a sample cannot be collected as the measurement is performed. When it comes to ET in a worn eye tracker this is mostly caused by the participant blinking, resulting in gaps of a few hundred milliseconds. Other reasons that gaps appear, for shorter durations, include delays in data transfers, temporary reflections caused by prescription glasses [70], etc. This could potentially split a fixation in two [70] if the data is not replaced by valid information and one does, therefore, need an algorithm that fills in the gaps.
- **Eye selection** - The eyes are often behaving slightly different when it comes to the start and end time of fixations and eye blinks [70]. This may lead to gaps in the data from one of the eyes and this requires a decision to be made regarding how the data from both eyes should be merged into a single data set for the I-VT filter [70]. Two examples are; averaging between eyes or using only the left or the right eye as the base for calculating fixations.
- **Close fixations** - Imperfections, such as short gaps or noise, results in data points being misclassified [70], which in most cases means that long fixation gets separated into two shorter ones with a saccade in between them that is short in both travelled distance and duration. This can be countered thorough a post-processing procedure that merges fixations [70] that are close in time and space.
- **Short fixations** - The basic I-VT filter does not limit how short a fixation can be [70], but due to the cognitive processes that processes the visual information, that occurs during fixations, there is a limit to how short these can be. This requires the implementation of a filter that removes data points [70], labeled as fixations, that last too short time.

CHAPTER 5

Supervised Machine Learning

The use of modern technology, such as sensors and computer programs, makes it possible to collect more data at a higher accuracy and a higher pace than previously. It is, however, difficult to analyze these large datasets, sometimes referred to as Big Data [20], using traditional methods.

Machine learning (ML) is a tool that can be used to process these huge datasets and solve practical problems using statistics and probability theory [71]. Supervised ML and unsupervised ML are the two most common types of algorithms [72]. The former means that the algorithm learns from examples of the output that is expected from a given input, i.e. it is given labels or targets for each input [72]. The latter type lacks this information, for example in the task of clustering, where the goal is to retrieve information on underlying patterns or to group data into categories [73].

An ML algorithm generally consists of the following components; a model, a cost function, and an optimization algorithm [72]. These are then coupled with a dataset to solve a specific problem. Different types of learning tasks are, for example, classification, regression, machine translation, anomaly detection, and denoising.

The main challenge in the field of ML, according to [72], is to train a model that performs well on previously unseen inputs, which is called generalization, and not just on the samples that were used during training, called training dataset. This dataset is used to determine the model's performance, called training error, and the parameters of the model are then altered in order to reduce the error [72]. This can be seen as an optimization problem, and what separates ML from optimization is that the goal is to also obtain a small generalization error [72], i.e. the estimated performance on the test set, a dataset collected separately from the data used in training. The performance of an ML model is, therefore, dependent on the model providing a small training error while it at the same time keeps a small difference between training and test error. The two factors corresponds to two important challenges in ML [72]: the first one, underfitting, occurs when the model is not able to achieve a low enough training error and the second one, overfitting, occurs when the discrepancy between the training and test error is too large. The likelihood of a model to under- or overfit can be managed through the alteration of its capacity [72], which can be seen as its ability to fit a large variety of functions, for example by adjusting the width and/or depth of an artificial neural network (ANN). Models with a low capacity may experience difficulties learning the training set whereas the ones with high capacity memorizes properties of the training set that are not transferable to the test set.

The following sections in this chapter will cover some ML approaches in the realm of ANNs and some of their areas of application.

5.1 Feedforward Neural Networks

Feedforward neural networks (FNNs) are the basic building blocks for deep learning models [72]. The goal of an FNN, as described in [72], is to approximate some function f^*, for example, $y = f^*(x)$ that maps an input x to a category y. An FNN defines a mapping $y = f(x; \theta)$ [72] and learns the value of the parameters θ that gives the best approximation of f^*. Feedforward comes from the fact that the information in these models only flow in one direction [72], from the input x, through the intermediary computations that define f, and finally to the output y. No information from the outputs [72] is fed back into the model again, when FNNs are extended to include feedback connections, they are called recurrent neural networks (RNNs), further described in Section 5.3.

The “network” component in FNN comes from the models typically being composed of many different functions. One example, given by [72], is the chain of the three functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$, that forms $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$. This is a common structure in ANNs [72] where $f^{(1)}$, in this case, is called the first layer, $f^{(2)}$ is called the second layer, and so on. The total length of the chain is what determines the depth of the model and this is what inspired the “deep” part of deep learning [72].

During the training of ANNs, the goal is to make $f(x)$ match $f^*(x)$ as closely as possible [72] using the training data that provides noisy, approximate examples of $f^*(x)$ evaluated at different training points. Each data point, x, has a corresponding label $y \approx f^*(x)$ [72] and the model shall, for each x, produce a value from the output layer that is close to y. The algorithm must learn to decide how to combine the intermediary layers, and the output layer, to approximate $f^*(x)$ as accurately as possible [72]. However, the training data does not contain any information regarding the desired output from the intermediary layer, the reason for why they are called hidden layers [72], and the dimensions of these hidden layers determines the width of the model. Each layer is composed of several units, acting in parallel, each representing a function that transforms a vector to a scalar [72]. The units are similar to neurons in the way that they take inputs from multiple other units and uses that to compute their own activation value [72]. This, and the fact that the networks contain features loosely inspired by neuroscience, is why they are called neural networks. However, FNNs should be seen as ways to approximate functions rather than as models of how the human brain operates [72].

5.2 Convolutional Neural Networks

CNN:s are a type of feedforward neural networks that are more robust to shift, scale, and distortion invariance [74] than fully connected neural networks, and therefore better at detecting spatial and temporal features. This is achieved by convolving or sub-sampling the input to the layer with local receptive fields [74] (filters) of a given size [n x m]. Each filter has $n \cdot m$ number of trainable weights + a trainable bias and these are shared [74] for all filter outputs.

5.3 Recurrent Neural Networks

Recurrent neural networks (RNNs) are a subgroup of ANNs that are used to process sequences of data [72]. An RNN shares its weights across several timesteps [72], whereas a fully connected neural network would have separate weights for each part of a sequence. In an RNN, the current timestep is not only computed as a function of its input, which is the case for regular feedforward neural networks, but also previously output states [72]. This gives the network access to historical data and how it changes over time. RNNs generally also allow for processing of sequences of variable length.

Long-Short Term Memory

Traditional RNN:s tend to suffer from problems with exploding or vanishing error gradients [72], [75], which prohibits proper learning over longer time instances. Long Short-Term Memory (LSTM) cells [75] are designed to provide a solution to this problem using a constant error flow [75] through the network, together with three gates that open and close in order to access the error flow [75]. The input gate decides when the internal state of the LSTM cell should be affected by the input to the cell, the forget gate determines when the cell's internal memory should be reset, and the output gate controls whether the current state of the cell should influence the error flow or not [75]. An LSTM network may contain multiple cells and the network learns to control each individual gate [75] in each cell. The GRU-unit [76] is another type of gated cell that is similar to the LSTM, however, it uses only two gates, a reset gate that determines when to ignore the previous state and an update gate that decides if the state shall be updated or not.

5.4 Dropout

Dropout is a deep machine learning method that is used to reduce overfitting [77]. This is done by randomly ignoring, with probability p, each neuron in a network every time a training case is presented to the network. The goal of randomly excluding some neurons for every training case is to make sure that the network learns generalized features instead of a co-adaptation between neurons [77]. The probability to be used for fully connected layers, suggested by [77], is $p = 0.5$. However, there exist other types of dropout [78] and suitable values for p varies with both the dropout type and the architecture.

Dropout as a Bayesian Approximation

The dropout method described above can, according to [79], be used to approximate Bayesian inference. This is done by enabling dropout at all times, not only during the training of the network, which means that the network will randomly omit some neurons also when making predictions causing variation. The mean prediction as well as the model uncertainty can be obtained by making N number of predictions [79] on the same data and collect the results. [79] claims that $N \in [10, 1000]$ should give reasonable results. Using this approach is useful since it provides a way to reason about model uncertainty that is easy to implement and less computationally expensive [79] than alternative methods. [79] suggests that the probability p for dropping a neuron should be in the range of $p \in [0.1, 0.5]$.

5.5 Transformers - Encoder

The original Transformer by [80] is an attention-based neural network architecture with an encoder-decoder structure, mapping one set to another, to solve natural language processing tasks. Since then, the Transformer has been adjusted in order to perform image classification with the Vision Transformer (ViT) [81], which only uses the encoder part. The first part of the ViT splits the image into a sequence of non-overlapping patches [81] and each patch is projected to a hidden dimension C that acts as the linear trainable embedding. A learnable positional encoding is then added to the embedding [81], in order to learn the ordering of patches since self-attention inherently lacks this capability. A class token that is used to obtain a classification that does not favor any one of the particular inputs is also added. This is then fed into the first encoder, the ViT is made up of N_x number of encoder blocks that are identical in size, that consists of a multi-head attention that performs self-attention in H parallel tracks followed by two position-wise feed forward layers separated by a non-linear activation. Self-attention is, according to [80], a function that maps a query and a set of key-value pairs to an output, computed as a weighted sum of the values. The particular attention function used in the Transformers encoder is called Scaled Dot-Product Attention and it computes the dot products of a set of queries Q with all keys K, divide this by the square root of the dimension of the queries and keys, $\sqrt{d_k}$, and then apply a `softmax` function in order to get the weight for the values V, which can be summarized as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.1}$$

The network ends with a hidden fully connected layer and a linear classifier [81]. The ViT also utilizes skip-connections [82] and layer normalization [83].

5.6 "Multi-Layer Perceptron"-Mixer

The "Multi-Layer Perceptron"-Mixer (MLP-Mixer) by [84] was proposed as an alternative to using CNN:s or Transformers-based architectures for image classification. Two selling points are that the Mixer network is able to achieve mostly comparable prediction results while using less memory and having less computational complexity. This gives a faster training procedure and a higher throughput (number of predictions per second) at inference. The main idea of the MLP-Mixer is to provide a simple architecture that performs two operations, mixing of features at a given spatial location and mixing between different spatial locations, in a separated way [84]. These two types of mixing are present in both CNN:s and attention-based networks but in a way that is less distinct. The input to the Mixer is a sequence of non-overlapping patches that represents one image and each patch is projected to a hidden dimension C using the same projection matrix. The Mixer is made up of N_x number of Mixer-blocks that are identical in size, where each block consists of two MLP-blocks [84]. The first one performs mixing between different spatial locations on the rows of the transposed input X and the second one mixes features at row of the input X. The weights of each MLP are shared for all rows and the MLPs consist of two fully-connected layers with a non-linear activation in between [84]. The parameters D_S and D_C are the hidden sizes for the two MLPs respectively. The network ends with global average pooling and a linear classifier, a common way of performing classification [84]. The Mixer network also utilizes skip-connections [82] and layer normalization [83].

5.7 Alternative Neural Network Architectures

There exists a wide range of other network architectures that can be used to analyze sequences of data. A few examples are; auto-regressive neural networks such as PixelCNN [85] and Wavenet [86] that have been used to generate images pixel-by-pixel or raw audio respectively and graph CNNs, e.g. STGCN [87], that has been used in human action recognition tasks that model the human body using skeleton key positions.

CHAPTER 6

Experimental Procedure

This chapter defines an experimental procedure that can be used to analyze human behaviour using VR, ET, and ML. It starts off with the formulation of the objective, what data is needed, and how the performance of the ML solution will be evaluated. This is then followed by three steps that describes how VR with ET can be used to collect human movement data, namely the experimental setup that covers the hardware, the test development that describe how a VRE can be designed in order to collect gaze and movement data, and then the test study and the selection of participants will be covered. The test related sections are followed by suggestions for ways to preprocess the data before use with ANNs, the design of an ANN architecture that may solve the objective, and what to consider when the results are obtained. Each section will also provide a brief description of what has been used in **Paper A-E**.

6.1 Objective - What is of interest and why?

The first thing to consider is what problem(s) is(are) of interest, how these may be solved through an ML approach, and what the end goal is or what the final product will look like. Once the objective is clearly defined it might be useful to identify possible subgoals that can be used to explore the overarching objective through an iterative process that may provide partial solutions. Both the main objective and its subgoals need to be measurable in such a way that they can be evaluated in a meaningful way. Evaluation will be further explored in Section 6.3.

The objective in **Paper A** was to investigate whether human eye gaze data can be used to classify which object out of 5 boxes that was selected after the test procedure was completed. The objective from **Paper A** provides the foundation for **Paper B**, which had the objective to continuously classify, ahead of time, which out of 18 possible boxes, that the test participant is about to reach for, utilizing the ANN's notion of uncertainty. **Paper C** had the same objective since the goal was to compare the results against **Paper B**.

The objective for **Paper D** was to explore the possibilities to further standardize cognitive testing through the use of a VRE with ET to increase the amount of information that is available after a test and gather that information in a automatically generated report. Lastly, the goal of **Paper E** was, in a similar way to **Paper A**, to classify what alternative a person selected on an item based on where the participant had been looking.

6.2 Data - What data is needed?

Once the objective is clearly specified it is time to figure out what data is needed to build the ML model in order to provide a solution to the problem. This involves the following steps:

- If the data is not already available, it has to be gathered somehow and this is both a time consuming and possibly costly procedure if it involves, for example, human participants.
- The amount of data required to develop a working ML solution varies and one must determine the minimum number of data points that gives a working solution. However, the quality of the data is also of importance and this requires good equipment, a suitable test design that accurately represents the objective that is to be solved, and that the data points with the most valuable information are collected. In order to make the most of the data during the training procedure one might also want to employ some type of data augmentation, further described in Section 6.7.
- The next step is to figure out what measurements are of interest, how and to what degree these capture the different aspects of the problem, and how these may be used to solve the objective.
- The available architectures and the possible problem formulations are also affected by the type of data that is collected and how it is arranged, e.g. sequences of numerical data, matrices, images, etc. The formatting of data will be covered in Section 6.7.
- The problem formulation will also affect the required data, for example a many-to-many sequence problem might require as many labels as input data, whereas a many-to-one classification problem might be solved with a single label for each set of data points.

The data that is used in the two studies in Chapter 7 has been collected from volunteer test participants and in most cases with the author of the thesis as the test leader. The dataset in **Paper A** features 720 data points collected from 24 participants and the dataset used in **Paper B-C** contains 3192 data points from 21 participants. **Paper D** did not involve any data collection beyond continuous testing throughout the development process. Finally, **Paper E** contains data from 27 participants that each have answered the 60 items in SPM. These datasets can be seen as quite small in ML context, however, the results shows that small datasets might be sufficient to develop proof of concept, especially for tasks of lower complexity.

The type of data is almost the same for all studies. It involves eye gaze, HMD movements, controller movements, and test specific data, such as selected boxes or alternatives, which is were the differences come into play. **Paper A** and **Paper E** employ a many-to-one problem formulation whereas the networks in **Paper B-C** were trained as a many-to-many problem used in a many-to-one context.

6.3 Evaluation - How will the network results be evaluated?

In order to accurately determine whether the ML solution successfully solved the objective or not it is important to know beforehand how to evaluate the network's performance. Depending on the problem to be solved there may be different ways of doing the analysis, for example if it is a common or previously tried experiment one should first consider to use the same metrics or benchmarks in order to make it possible to quickly do comparisons. In other applications the use of common metrics such as mean squared error or classification accuracy etc., may falsely evaluate the network due to a system dynamic that is not clearly captured in the metric. An example of this could be to incorporate slack in the error measurement if the system itself does not require pinpoint precision.

Paper A used a traditional classification accuracy score in conjunction with prediction filtering based on the standard deviations of the predictions that allows the system to be evaluated according to its certainty as well as performance. **Paper B-C** was evaluated slightly different in order to simulate the performance on a continuous stream of data. **Paper D** contains no ML and thus lack evaluation metrics. **Paper E** used a traditional classification accuracy score using the mean values of the UE.

6.4 Experimental Setup - What hardware can be used to collect the data?

There are several ways of measuring eye gaze, e.g. camera-based ET methods [88] and wearable eye-tracking glasses [89], as well as human movements, e.g. camera-based methods [90] and wearable inertia based methods [91]. One way to merge gaze and movement tracking into one system is through the use of a VR-headset with hand controllers and built-in ET. It is possible to design a fully controllable VR environment (VRE) that gives access to information about where the user has been looking, moving their head and hands, while simultaneously limiting visual distraction through the immersion that the headset gives. Performing various experiments in VR is also much safer than in the real world since there is no risk that the operator is hurt nor that the equipment, inside the VRE, is damaged. There is also an endless supply of material since new parts can be generated from a piece of code.

Paper A-E all use the same setup, namely a consumer grade VR-headset, "*Tobii Eye Tracking VR Devkit*" [92], that has built-in ET and utilizes two handheld controllers to navigate the VRE. The system is capable of tracking the position and orientation of the HMD and the hand held controllers. The eye gaze is tracked with *Binocular dark pupil tracking* at a frequency of 120 Hz. This type of eye-tracking is achieved by illuminating the eyes, off-axis compared to the cameras that are used to capture images of the reflected light as it bounces off the retina and exits the eye, causing the pupil to appear darker than the rest of the eye. The images are used to calculate a gaze direction vector based on the positional relationship between the cornea and the pupil. The ET can be performed in the entire 110° field of view of the HTC-Vive HMD [92], with an accuracy of ∼0.5° and a delay of ∼10 ms from the illumination of the eye until the data is available in the SDK. The eye tracker is individually calibrated to each test participant using a 5-point calibration strategy available in the SDK. The calibration is based on the user being instructed, visually and audibly, to focus her/his gaze on 5 pre-defined points in the VRE and that gaze data is used in the SDK to calculate a 3D-model of the eye.

6.5 Test Development - How can the test be designed with regards to the data objective and available hardware?

Developing a VRE test can be broken down into the following steps:

- Language - Choose one or multiple language options that can be used to present written instructions during the test procedure and make sure that the test leader is able to deliver the spoken instructions in the chosen languages. The instructions should preferably be customized to fit the target group that is going to perform the test.

- Design the test in a way that makes it as clear as possible to follow the different steps of the tests including for example; language selection, calibration of the equipment, input of extra information such as age, gender, etc. (to use for basic demographics), and the start of the test itself. It is also convenient if the test is easy to restart if something goes wrong, the participant has additional questions during the test, or simply to make it easy to move on to the next participant.

- Consider using anonymous participant IDs in order to store the collected data in a way that preserves the privacy of the participants.

- Limit distraction - Limiting or controlling distractions from the test itself is good way to reduce or introduce noise in the data depending on what is desired. Visual stimuli is easily controlled in a VRE and this is an important strength that should be utilized to make the test as standardized as possible. Adding audible and haptic stimuli is also possible.

- Warm-up - If the test procedure is unfamiliar to the participant then it might be useful, in order to reduce the bias from inexperience with for example the equipment, to have a warm-up segment that ensures that the participant gets some experience regarding what is to be done.

- Finally, consider developing the VRE in a modular fashion such that modifications can be made if necessary or in order to be able to reuse the parts of the environment in a different context.

The main VRE structure for **Paper A-E** uses Swedish and English as the two instructional languages, it incorporates anonymous participant ID:s that are randomly generated as the test is launched, the ET calibration steps and the gathering of general information follows the same procedure, and its ET, movement tracking, and visual distraction limiting features are the same. This makes for a modular design that is slightly modified as the objective changes through the papers.

The test stage in **Paper A** features a table in the form of half a circle where cubes will appear at random in 5 different zones with 45° spacing. The test stage has been designed in a way that is meant to force the test participant to look in the direction of the cube, make a movement towards the cube, and acknowledge that movement by touching the cube. The zone that gets a cube is randomly selected every time a new cube is to be created and the positioning of the cube is also randomized, in the interval $x \in [-x_s, x_s], y \in [-y_s, y_s]$, where x_s, y_s are the maximum deviations allowed around the center of each zone. The test has a 1s time delay between each cube appearing that helps to slow down the pace of the test execution.

Paper B-C use the same test environment featuring two even distributions of 9 cubes, each at two different heights and radii. The cubes appear at two different radii, based on the participant's arm length, and requires the test person to touch it while simultaneously pressing a button on the controller to make the cube disappear. After a cube has disappeared, and a delay of 0.2s, the next cube in the randomized sequence is lit. The delay is used as a way to force a slower pace throughout the test and data is collected during this time.

The VRE in **Paper D** is designed to model a subset of 10 different items from SPM and takes place in a sparsely furnished, square space with calm colors to prevent the user from being distracted. The user can move freely in the room throughout the test, both in the real world and in the virtual, but it is recommended to remain seated/standing still.

The test stage in **Paper E** starts with three simple training items that are used to make the participant more accustomed to the VR-part of the test. The training items are inspired by the first difficulty level (A) of SPM. The decision to add training items was based on the fact that new participant during the development of the VRE spent an unreasonable amount of time on the first item despite its simplicity. Note that the data from the training items is not included in the final dataset. The actual test contains the 5 levels of progressively higher difficulty with 12 items each from the SPM, 60 items in total. The test takes place in a VRE that consists of a room-like space with a gray and blue color scheme without any extra cosmetics apart from a window and a door. The limited aesthetics were chosen to prevent the user from being distracted by the VRE. The main feature of the room is a black "screen" that presents all the information. The dark colors of the room and the screen were chosen to reduce eye strain from the bright displays of the HMD and to provide contrast to the Raven items that are white with black features. The user can move freely in the room throughout the test, both in the real world and in the virtual, however, it is recommended to remain seated. The items were created as two different 3D-models in Blender, with 6 or 8 alternatives respectively. The paper-based SPM was scanned using a printer and the images were cut into pieces corresponding to the item and all the different alternatives generated automatically using a Python-script. The images were transformed into textures suitable for 3D-models in Unity and are continuously coupled with the correct model during the test. The 3D-models are light grey instead of white in order to further reduce eye strain caused by an otherwise too bright environment. There is also a separation segment between each test item achieved using a +-sign that appear in the middle of the screen, as the previous item is removed. The +-sign decreases in size for the duration of 1.5 seconds until it disappears and the next item is displayed. This serves the purpose of resetting the user's attention to the middle of the screen.

6.6 Test Study - How will the data collection take place and who will participate?

- Limit external distractions - External distractions or disturbances may have a negative impact on the quality of the data since that could give some of the participants unfair disadvantages. It is, therefore, crucial to limit these (the ones that have not already been taken care of during the test development) as much as possible, for example by using a room with low noise levels or making sure that the participants does not feel stressed about the upcoming task. The latter may be reduced by clearly explaining the goal of the test, going through what is expected of them, and answer any questions.

- Standardized instructions - Make sure to use the same instructions for every participant in order to reduce bias from the instructional phase, however, keep in mind that some people may need some additional help in order to be able to carry out the test as intended.

- Feedback - Ask the participants if they are willing to give some feedback that can be used to improve the test procedure and/or the VRE.

- Selection of participants - In order to create a robust system that works in various conditions and for as many users as possible one needs to consider the test group diversity. It could be possible to design a system that successfully learns how a few participants behave, that is not transferable to others. A larger test group mitigates this as well as other biases towards, for example, age or gender. The importance of this may, however, vary depending on whether the goal is to create a system as a proof of concept or if it is supposed to be production ready.

The data in **Paper A-E** has been gathered at mostly quite places, however, there have been occasions where there have been other people present. The instructions, for each test study, have been given in the same way to every participant. The test procedures and the VREs have successfully been improved through feedback, from early participants, before the actual test studies were carried out and between the studies. **Paper A-E** are all simplified test studies that are designed as a proof of concept. The selection of participants is, therefore, limited to volunteers and mainly people who work or study at Chalmers. All participants volunteered and gave their consent orally.

6.7 Preprocessing - How will the data be processed to fit the ML solution?

In order to train the ANN with the most useful information possible it is important to preprocess the data such that it is presented in its most usable format. This includes selecting the appropriate labels to be used for both training and evaluation to determine the success of the network. The following steps should be considered:

- Filtering of outliers - If the data consists of unwanted outliers these should be removed before training in order to reduce the likelihood that these guide the training of the network in the wrong direction.
- Selection of features - The features are the input to the ANN and selecting these will greatly influence the success of the training of the network. Features that are too similar may, for example, drive the training of the network into a local optimum due to an over-representation of redundant information. Analyzing the correlation between different features or utilizing domain knowledge are two possible ways of determining which ones that provide valuable information.
- Data augmentation - Training ANNs generally requires a lot of data and if it is difficult and/or expensive/time consuming to gather more data, for example when dealing with human test participants, it may be possible to augment the training data in order to achieve a better performance. Ways of augmentation could be to add duplicates of the data, with or without noise, shuffling of the data, and other transformations that slightly perturbs the data such that it aids the networks generalization capabilities.
- Network specific preprocessing - Depending on the task and the network architecture the data may need to be formatted in a specific way in order to solve the objective. If sequences require equal length one may consider for example zero-padding the data or applying some kind of up- or downsampling to give them equal length. Other contexts could require that the data is parsed using a sliding window.
- Normalization/standardization - The network may over emphasize the importance of some features over others if they are of different magnitude. This can be countered through feature wise normalization or standardization of the data. The former referring to re-scaling the data, for example between its minimum and maximum value, and the latter to re-scaling the data to have zero mean and unit variance.
- Simplifying the problem - Before it is time to select the labels for the supervised learning problem one may consider reformulating the problem to reduce the number of different variations that the network has to learn. One example of this in a classification problem could be if there are 2 classes that contain 10 similar subclasses. This could either be formulated as a multi-class problem with 2*10=20 classes or as a binary classification problem coupled with a multi-class problem with 10 classes.
- Selecting labels - Supervised ML requires suitable labels to learn how to solve the objective. Selecting the appropriate labels is the difference between a successful and a failed project. A clearly defined objective and evaluation procedure should, therefore, be the starting point for the selection of labels along with a careful analysis of the available data.

- Class re-balancing - The data that is collected may sometimes be unbalanced, i.e. there are one, or a few, classes that constitute the majority of the data. This may cause the network to become biased towards guessing these classes and in worst case rendering the network useless since it learns to always guess the majority class, as the cost of a wrong guess is small compared to all the right answers. Class unbalances can be mitigated through re-weighting the loss of being wrong during training, such that the loss of guessing wrong is higher for rare classes. It could also be possible to collect more data that is targeted towards the minority classes or, if it is possible, redesign the test procedure to ensure balanced data.
- Train/validation/test split - The last step of the data preprocessing is to split the data between training and test data. The size of these may vary but the important thing is that the test set is large enough to determine that the network generalizes well, while on the other hand a larger training set usually improves the training procedure and reduces the risk of overfitting, thereby providing a better generalization. A common strategy when experimenting with network parameters is to also split the data a third time into a validation set. This is useful both during the training phase to monitor the loss on unseen data, but also to mitigate the risk that repeated experiments make the solution tailor-made towards the test set, which should only be used for the final evaluation.

Paper A-C use the same way of discovering and removing outliers. The datasets have been approximated using Beta-distributions and then a maximum threshold, maximum duration (samples) of a test segment, has been set according to the mean plus three standard deviations of this distribution. All data points that contained more samples than the threshold were discarded. **Paper E** filtered the data based on a threshold determined from observations.

The next step filters out each sample, within each data point, that contained NaN values. These data points were discarded in **Paper A** whereas they were replaced with the previous valid sample for **Paper B, C & E**. NaN values occur when the ET fails to read the eye properly, most commonly as a result of the participant blinking.

In **Paper A**, one of the goals was to investigate the neural network's ability to handle raw gaze data and the features that were used are, therefore, left and right gaze direction vector, left and right pupil diameter, and a variable that contains the duration of the test. The features that were used in **Paper B** are the combined eye gaze direction vector (x, y, z), obtained as an average of the separate gaze vectors from each eye, the y- and z-coordinates of the HeadPosition, and the pupil diameter, averaged between left and right eye. The x-coordinate of the HeadPosition was removed as it corresponds to the participant's height, which is constant during the entire duration of the test due to the fact that they remain standing and does, therefore, not provide much information to the network since the boxes are individually calibrated to the participant's height and reach. The HeadRotations were removed since the focus point of the gaze is more interesting and because of the fact that the head is often rotated in conjunction with the eyes, therefore, providing limited information to the network. The reason that information such as EyeHitPoint and EyeHitObject are not used is because they require specific knowledge of all objects in the environment, something that is possible to know in a VRE but would limit the possibility to implement the system in a real-world scenario. The same features were used for the comparison in **Paper C**. The features for **Paper E** that were selected are the two coordinates in the plane of the item, calculated as a projection on to the plane using the average gaze direction vector and the HMD-position vector together with the distance from origo to the plane. One additional feature is a boolean value that is true for difficulties A & B (that always have 6 answers) and false for C-E that have 8. This supports the network's ability to distinguish between the amount of possible answers.

Paper B-C are the only studies where data augmentation was added and it is achieved by stacking copies of movements after each other in the creation of the training dataset.

The data in **Paper A** was featurewise normalized in the range of [0, 1] and the data points that were of shorter length than the decided threshold were padded with zeros (ZP) at the end to guarantee the structure of the data point that is fed to the network. **Paper A** also evaluates using linear upsampling (US) of the data points to achieve the desired length. US is, however, not an alternative for a continuous data and since ZP performed better, it was chosen as the preferred method. The data in **Paper B-C** was featurewise normalized in the range of [-1, 1]. The data in **Paper E** was featurewise normalized in the range of [0, 1] and the sequences that were shorter than the maximum length were padded with values of -1 such that all sequences have the same length. The value of -1 was chosen arbitrarily outside the range of the normalized data and it can, therefore, be masked (ignored) in the ANN.

The problem formulation for **Paper A** is a multi-class classification problem that uses the id of the boxes as labels. **Paper B** was rewritten from a 19 class classification problem to a 10 class problem, with the ids of the boxes at the lower level as the labels. Nine of these classes, the box in the centre is excluded, are also binary classified as either 0 or 1, corresponding to the lower or the upper level of boxes. The same problem formulation was used in **Paper C**. **Paper E** is a multi-class classification problem that uses the different alternatives to the test as labels.

Neither of the papers use any class re-balancing. **Paper A-C** used a data split of roughly 45%/5%/50% (train/validation/test). **Paper E** used k-Fold cross-validation (kFCV), $k = 10$, in order to adjust for the fact that the dataset is small. kFCV is an iterative way of training and evaluating the network on all parts of the data to reduce bias from a lucky initial selection of data for evaluation. $k = 10$ means that the data was split into 10 non-overlapping subsets and for each iteration one subset is selected for testing and the other 9 are used for training. The process is then repeated, re-training the network for each iteration, until all subsets have been used for testing exactly once. Finally, an average accuracy score from the k iterations is obtained.

6.8 Neural Network Design - What network architecture(s) can be used to solve the task?

Once the data is formatted in a proper way it is time to create the ANN that will perform the analysis of the data. A first step is to consider if there are any specific ANN properties that are suitable for the specific task, such as CNNs for images or fixed time series, or RNNs for more complex time series problems. Start off with a simple network and add more complex structures later, this makes the network easier to analyze if it is not working. It trains faster, and lowers the computational cost. This is also when one may consider whether there are any hardware limitations that come into play when using the trained network in its live environment. Fast online predictions are easier to achieve using a smaller network since it requires less computational resources. The choice of intermediary activation functions may affect the performance of the network [72] and a few common alternatives are; `ReLU` [93], `GELU` [94], and `tanh`.

When the network architecture is in place one needs to apply an output activation that transforms the network output to its desired format [72] and then couple it with the appropriate loss function that controls the learning process of the network. A `softmax` output is commonly used for multi-class classification and is often paired with a `categorical crossentropy` loss whereas a `sigmoid` activation is paired with `binary crossentropy` loss for binary classification, etc.

The training of the network also requires an optimizer that is used to find the appropriate error gradients to learn from [72] and some optimizers may work better for a specific architecture than others. It is also important to determine when the network should be considered fully trained. One approach is to monitor the networks performance on the validation data and terminate the training once the validation loss stops decreasing, sometimes referred to as early stopping. The reason that the performance on the validation data is considered is because the objective is to train a network that works well with unseen data and not only optimize towards known data.

Paper A uses a CNN approach, inspired by the inception modules from `Inception-v3` [95], adapted to 1D time-series data as the basis for the classification coupled with the uncertainty estimation described by [79]. The architecture, Figure 6.1, is utilizing `ReLU` activation functions, a `softmax` output activation, and is trained with `categorical crossentropy` loss. This structure is then reused for **Paper B** in a time distributed way with the addition of an LSTM-layer, Figure 6.2, and the intermediary activations have been swapped from `ReLU` to `tanh`. The network has two outputs, the first one uses a softmax activation together with a `categorical crossentropy` loss whereas the second one uses a `sigmoid` activation paired with a `binary crossentropy` loss.

There are two networks in **Paper C**, one that is based on the encoder part of the Transformer architecture and one based on the MLP-Mixer. The attention encoder, based on the ViT [81], can be seen in Figure 6.3 and works as follows: it starts with a Conv1D-layer that is responsible for formatting the input data into subwindows. It simplifies the preprocessing, by moving the subwindowing of the data into the network, and enables the network to learn from this stage, compared to **Paper B**, where the subwindows were formatted during the preprocessing. The positional encoding is a layer of trainable parameters that are responsible for learning the order of the data since that information is otherwise lost in the following attention layers. The first encoder block starts with a multi-head attention layer (MHA) that performs self-attention in parallel heads/tracks followed by two Conv1D layers, which processes the output from the attention calculations. Both of these Conv1D-layers apply a Gaussian Error Linear Unit (GELU) [94] activation. The encoder also contains two skip connections as seen in Figure 6.3. A skip connection is a summation of the output from a layer and the output from a previous layer. The encoder layer is repeated N_x number of times (including the first block) before the network ends with the TimeDistributed UE and the two outputs, the first one uses a softmax activation together with a `categorical crossentropy` loss whereas the second one uses a `sigmoid` activation paired with a `binary crossentropy` loss. The Mixer network, based on the MLP-Mixer by [84], can be seen in

Figure 6.4 and works as follows: it starts with a Conv1D-layer that has the same function as the one in the encoder network above. The output from this layer is fed to the first Mixer-block that consists of two MLP-blocks and two **T**-blocks that transposes their respective inputs. The first MLP-block performs mixing between different spatial locations on the rows of the transposed input X and the second one mixes features at row of the input X. The MLPs consists of two fully-connected layers, with dropout, and a non-linear activation, `tanh`, in between. Each mixer block also features two skip connections followed by a layer normalization [83] - not illustrated in the figure. The mixer block is repeated N_x number of times (including the first block) before the network ends with the TimeDistributed UE and the two outputs, the first one uses a softmax activation together with a `categorical crossentropy` loss whereas the second one uses a `sigmoid` activation paired with a `binary crossentropy` loss.

The ANN from **Paper E**, Figure 6.5, uses masking such that all padding values are ignored. The masked input is then fed to a convolutional (Conv1D) layer that transforms the input into subwindows of non-overlapping patches of size 10. These patches are the input to the LSTM layer that tries to find patterns regarding the time aspect of the data. This is followed by the implementation of the uncertainty estimation by [79]. The architecture, Figure 6.5, is utilizing `GELU` activation functions, a `softmax` output activation, and is trained with `categorical crossentropy` loss.

Paper A-C & E were all trained with the `Adam` optimizer [96] until the validation loss stopped decreasing.

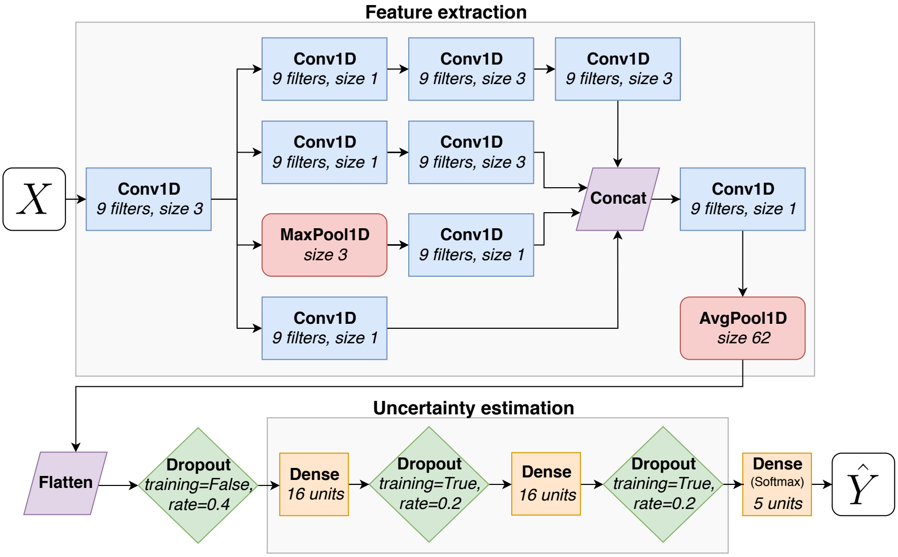


Figure 6.1: A flowchart that describes the network architecture used in **Paper A**.

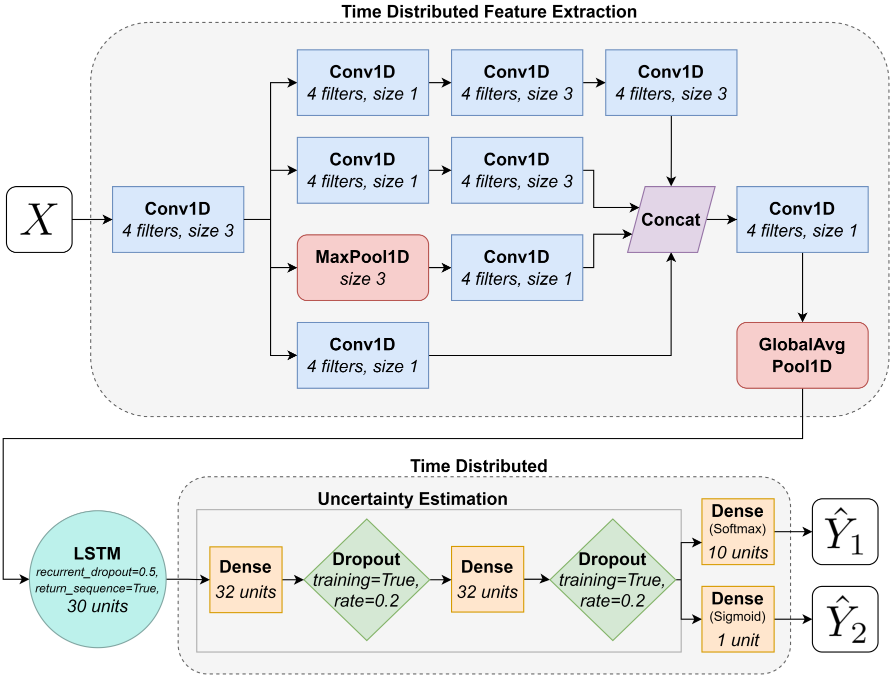


Figure 6.2: A flowchart that describes the network architecture used in **Paper B**.

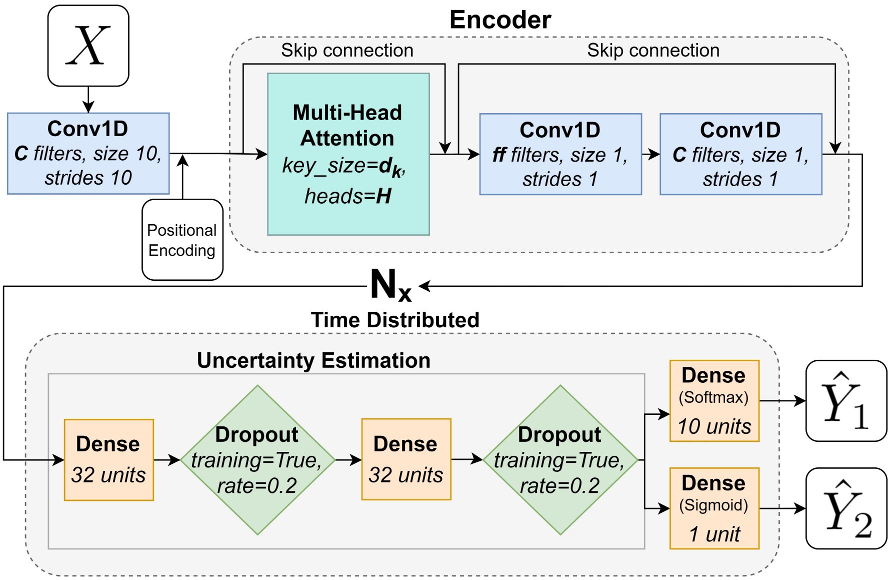


Figure 6.3: A flowchart that describes the Transformers encoder architecture used in **Paper C**.

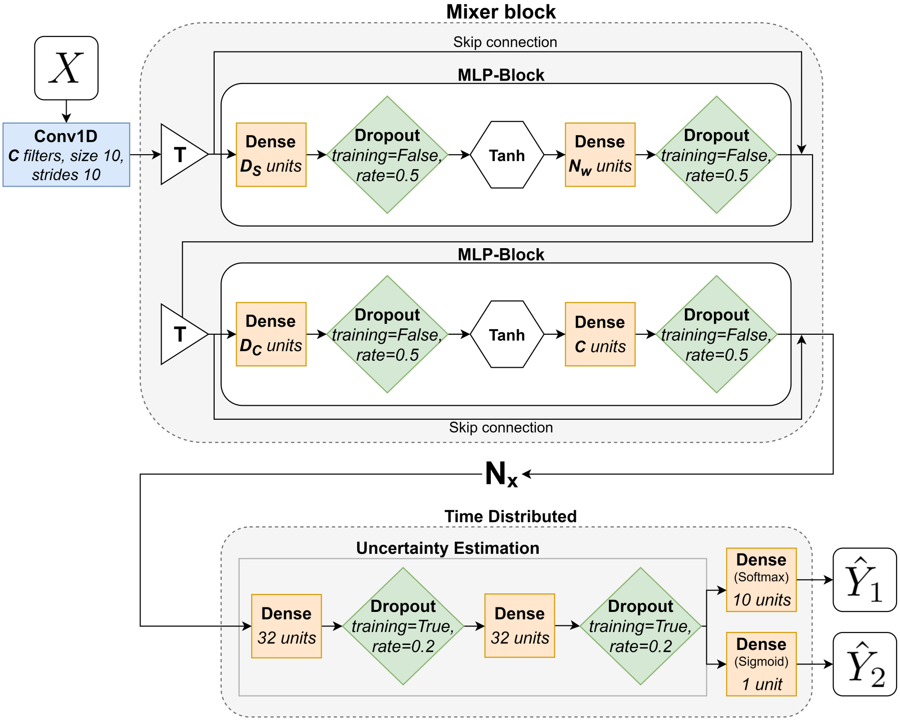


Figure 6.4: A flowchart that describes the MLP-Mixer architecture used in **Paper C**.

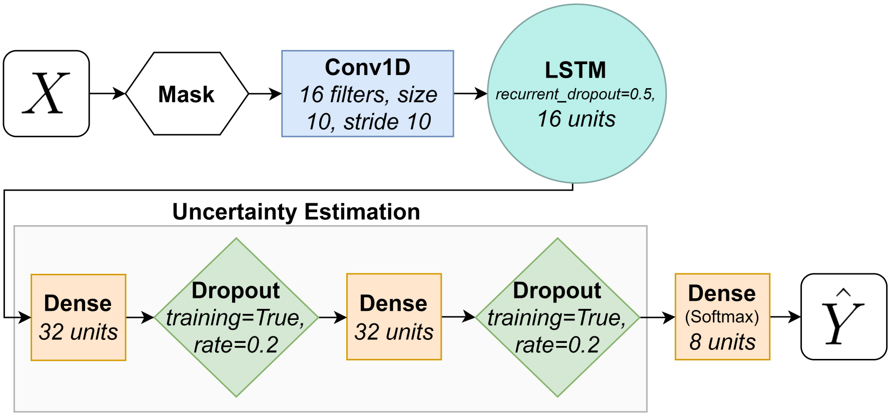


Figure 6.5: A flowchart that describes the LSTM architecture used in **Paper E**.

CHAPTER 7

Human Intention Analysis

This chapter will present the results that were obtained from the two test scenarios explained in **Paper A-B** that investigates the ability to determine human movement intention, in two steps of complexity, based on eye gaze. These results are accompanied by a comparison between three different ANN architectures **Paper C** using the same data as in **Paper B**. The movement prediction is followed by two other test scenarios, detailed in **Paper D-E** that utilizes the same tools to perform intention analysis in a different context, namely in the realm of psychological testing where the task, for the participant, is to provide an answer to a logical pattern. The main takeaways include that it is possible to analyze human intent in similar ways regardless of application as long as the eye gaze is used as the input data. It is also clearly visible that modelling the uncertainty of the ANN is greatly improving the analysis and discussion of the networks performance, both from safety and usability perspectives.

7.1 Study 1 - Intended Human Arm Movement Direction Analysis

This section will provide the combined results from **Paper A-C** and show the progression from a simple classification task, through the addition of uncertainty estimation (UE), to prediction of human arm movement direction.

The classification results from **Paper A** without UE can be seen in Figure 7.1. The graph shows the largest contributor from the softmax output for each sample that was classified. The samples are sorted in increasing order, left to right, based on this value. A green bar represents a correctly classified sample whereas a red bar indicates that the sample was incorrectly classified.

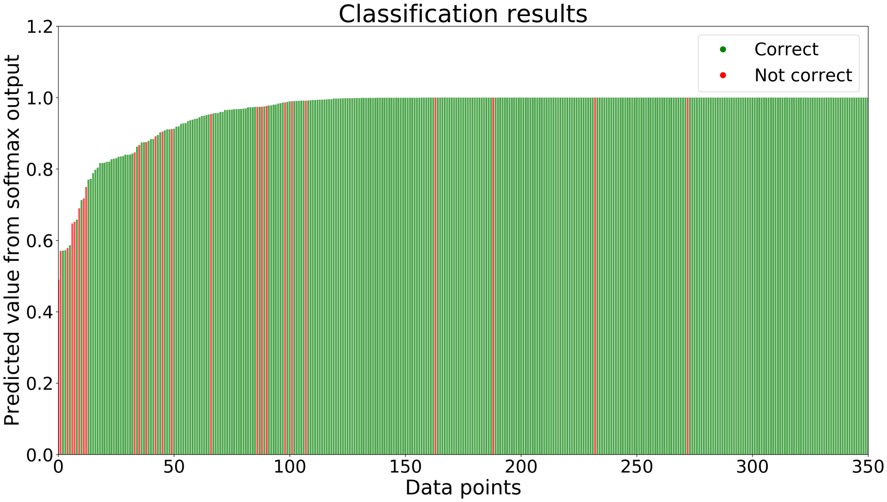


Figure 7.1: A graph of the classification results without UE.

These results can be improved through the addition of UE as explained in **Paper A**. The difference when making predictions for UE is that many predictions are done on the same data such that it is possible to obtain a mean value and a standard deviation of the prediction. The pseudo code for this is shown in Algorithm 1.

The results obtained from the network using UE, with $nrOfPredictions = 1000$ as suggested by [79] to ensure good plots, can be seen in Figure 7.2. The graph shows the largest contributor from the softmax output for each sample that was classified. The samples are sorted in increasing order, left to right, based on this value. The black interval displays two standard deviations of the prediction around its mean value. A green bar represents a correctly classified sample whereas a red bar indicates that the sample was incorrectly classified.

Algorithm 1 Pseudo code for predicting with UE.

```
Input: X, nrOfPredictions
Output: $\hat{Y}$, $\hat{Y}_{STD}$
predictions = []
for $i = 0$ to nrOfPredictions do
    predictions[i] = model.predict(X)
end for
$\hat{Y}$, $\hat{Y}_{STD}$ = mean(predictions), std(predictions)
return $\hat{Y}$, $\hat{Y}_{STD}$
```

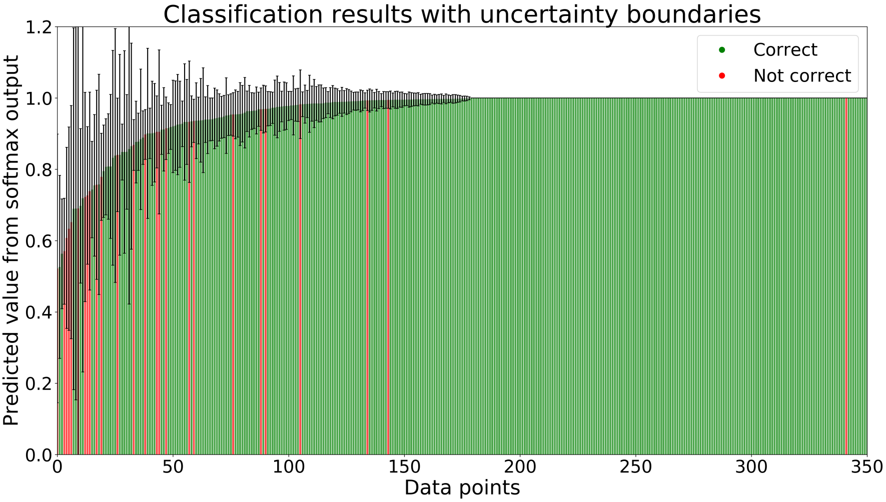


Figure 7.2: A graph of the classification results with UE.

Once the mean and standard deviation has been obtained from the network these can be used to determine if the network is confident enough, high mean and low standard deviation, to make an accurate prediction. This was implemented as shown in Algorithm 2 where a prediction is accepted if the mean minus two standard deviations is larger than a chosen lower limit, later referred to as Th_L.

The classification results for different lower limits can be seen in Table 7.1. It is clear that the classification accuracy can be increased with this approach, however, at the cost of the network not being able to classify all samples.

Algorithm 2 Pseudo code that accepts or discards a prediction.

```
Input: $\hat{Y}$, $\hat{Y}_{STD}$, lowerLimit
Output: $\hat{Y}$
if $\hat{Y} - 2 * \hat{Y}_{STD} >$ lowerLimit then
    Accept $\hat{Y}$ as the prediction for this sample.
else
    Discard $\hat{Y}$, the network is not confident enough.
end if
```

Table 7.1: Comparison of classification results for different levels of filtering using UE and zero padding (ZP).

Th_L	Accuracy	% samples classified
0	93.28%	100.00%
0.10	93.52%	99.74%
0.20	93.73%	98.97%
0.30	93.70%	98.45%
0.40	94.44%	97.67%
0.50	95.39%	95.35%
0.60	95.59%	93.80%
0.70	96.31%	90.96%
0.80	96.76%	87.60%
0.90	98.33%	77.26%

These classification results were the foundation for the work in **Paper B-C** where the objective shifted from classification of a direction after a movement completed, to continuous prediction of movement direction without the notion of start or finish. The movement prediction was achieved using three network architectures suitable for time-series namely LSTM, Transformers encoder, and MLP-Mixer. All of the networks also utilized the concept of UE.

The performance of the networks has been evaluated using the following custom metrics:

- A_P = Accuracy of predictions that are above UE threshold,
- A_M = Accuracy of how many movements are correctly classified at least once,
- A_{VP} = Vertical accuracy, evaluated whenever there is a box prediction.

These are more suitable to use to evaluate the network on how well it is able to utilize its notion of UE in order to predict the intended movement direction, compared to a standard accuracy metric that does not capture the aspect of UE. The reason to consider these metrics can be described as follows: A_P is the metric that keeps track of the accuracy of predictions that are being made, however, it is possible to achieve an accuracy of $A_P = 100\%$ for a very high threshold with just a single correct prediction. A result of this kind is not considered valuable since such a network would not sufficiently solve the primary objective. A_M on the other hand keeps track of how many movements that were correctly classified at least once. However, one way to achieve $A_M = 100\%$ is through a network that makes predictions all the time, without regard for the accuracy of each prediction, eventually one will through randomness be correct. This type of result is, on its own, not useful either for the same reason. Through the combination of the two metrics, A_P and A_M, it is possible to evaluate how well the network is able to handle this contradictory task of being both fast to predict and correct in its prediction. A_{VP} is the accuracy score for the secondary classification objective. One way to select the threshold, Th_L (called lowerLimit in Algorithm 2), where a network gives the best compromise between a high accuracy and covering all movements (high A_P and high A_M) is to calculate the intersection of these, illustrated in Figure 7.3, on the validation set using thresholds varied between $[0, 1]$ with a step size of 0.01. Due to the fact that predictions are filtered out based on an increasing threshold, once it reaches above the highest certainty of the network, all predictions will be filtered out i.e. no predictions are made, and the accuracy therefore goes to zero, as seen in Figure 7.3.

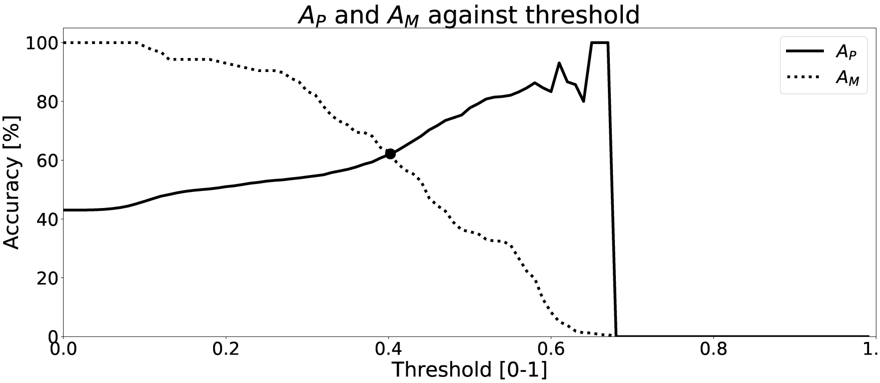


Figure 7.3: A figure showing the selection of the optimal threshold for a network.

In addition to the metrics described above, the comparison also includes the number of parameters, P, that make up the networks and the execution time (T) of each network, defined as the time in milliseconds that it takes to perform a single prediction. All networks were trained and evaluated on the same hardware in order to ensure that the execution times are comparable. The experiments were performed on a laptop with an Intel(R) Core(TM) i7-8650U CPU and 16GB of RAM.

The best performing networks from **Paper B-C**, *LSTM*, *Enc1*, and *Mix3*, were evaluated on the test set and the results are presented in Table 7.2. It is shown that the *Enc1* network is the best performing one overall with a prediction accuracy $A_P = 82.74\%$, movement accuracy $A_M = 80.06\%$, and vertical accuracy $A_V = 89.10\%$. A good alternative to *Enc1* is *Mix3* with balanced and slightly lower accuracy scores, both of them outperformed *LSTM* in terms of accuracy. The execution time for a single prediction was measured using the python function `time.perf_counter()` and the measurement was repeated $n = 10^5$ times to obtain a fair estimate. The results are presented as a mean and a standard deviation in Table 7.2 for each network. It is clear that the difference is, in this case, negligible since the variance is larger than the difference between the fastest and the slowest network. However, larger networks with more trainable parameters might show more clear differences.

Table 7.2: Table showing a performance comparison between the top performing network from each network type, evaluated on the test set.

Best network from each architecture - Test set						
Network	Th_L	A_P	A_M	A_{VP}	P	$T[ms]$
Enc1	0.53	**82.74%**	**80.06%**	**89.10%**	7.02k	31.43 ± 4.05
Mix3	0.25	76.97%	77.86%	87.61%	4.99k	29.42 ± 3.88
LSTM	0.38	70.70%	67.89%	81.29%	6.99k	30.39 ± 4.10

A segment of predictions on the test set for *LSTM*, *Enc1*, and *Mix3* are shown in Figure 7.4-7.6 respectively. The black lines with squares correspond to the true label for an entire movement, the blue dots are the unfiltered predicted labels at each timestep, the green X's are the predicted labels when the certainty is above Th_L, and finally the black line with the dotted black lines in the bottom graph corresponds to the mean softmax output plus/minus two standard deviations. It can be seen that all of the networks, after filtering on certainty, makes few mistakes and manages to correctly classify most of the movements for this segment. The bottom part of the figure displays the certainty fluctuating over time and it shows that *Enc1* often rapidly rises and falls in certainty for each movement, which indicates that the network is swift to update its certainty once it receives a new data sample. The certainty of *Mix3* fluctuates more aggressively than the other two networks and there is no clear pattern in the unfiltered predictions, however, once filtered it still predicts most movements correctly. The *LSTM* has the smoothest certainty plot but larger confidence bounds than the *Enc1*. The comparison of this segment indicates that the behaviour of the certainty is not that important as long as the predictions are filtered. The fact that both the *Mix3* and the *LSTM* have larger confidence bounds is likely the reason that they have lower thresholds that gives the highest A_I.

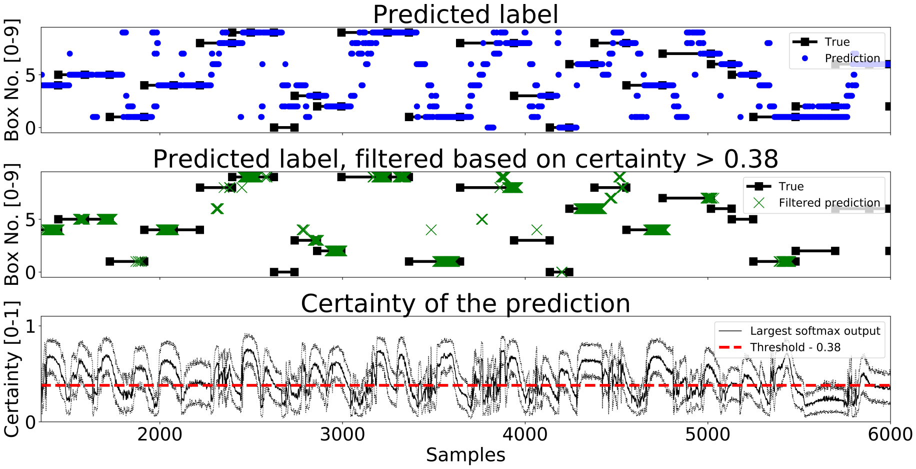


Figure 7.4: A figure that shows a prediction segment from *LSTM* obtained on the test set.

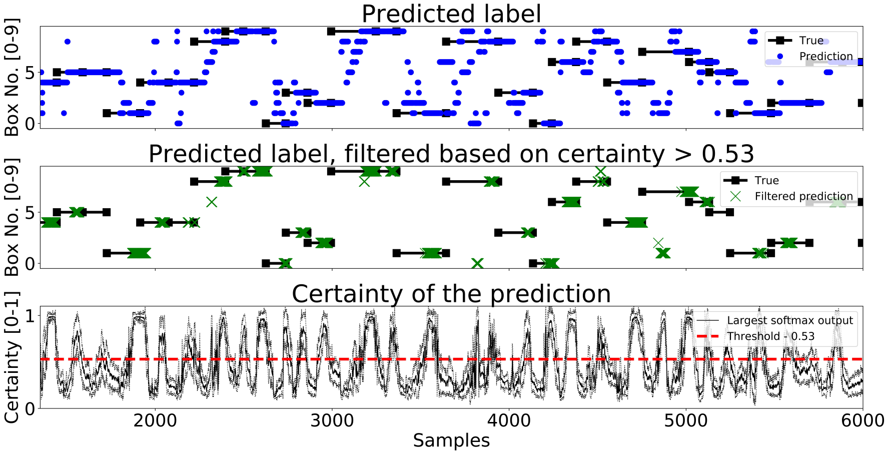


Figure 7.5: A figure that shows a prediction segment from *Enc1* obtained on the test set.

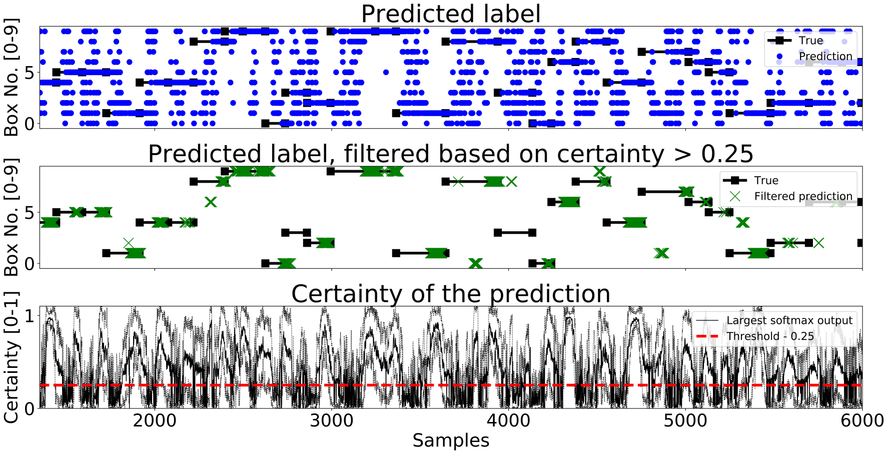


Figure 7.6: A figure that shows a prediction segment from *Mix3* obtained on the test set.

The accuracy of the predictions are not all that matter, how early a movement can be predicted is also of high importance and to evaluate that, it is essential to know what the hand movements look like. Figure 7.7 shows an aggregation of all hand movements from the test set (**Paper B**) created in order to further evaluate the performance of the networks with regards to time. The upper graph shows the normalized distance, d_i, that the controller travelled from the moment the previous box was clicked until the next one, calculated at each sample i for each movement as:

$$d_i = 1 - \frac{|\mathbf{p}_{end} - \mathbf{p}_i|}{|\mathbf{p}_{end} - \mathbf{p}_{start}|} \tag{7.1}$$

where $\mathbf{p}_{start}$ is the coordinate $(x, y, z)^T$ of the controller for the first sample of the movement and $\mathbf{p}_{end}$ is the last one. The normalized distance was then plotted with an alpha of $0.03 \in [0, 1]$ and normalized time in order to show the characteristics of all movements on the same scale. The lower graph shows the velocity, v_i, towards the target, $\mathbf{p}_{end}$, at each sample i for each movement, calculated as:

$$v_i = f_s \cdot (\mathbf{p}_i - \mathbf{p}_{i-1})^T \cdot \frac{\mathbf{p}_{end} - \mathbf{p}_i}{|\mathbf{p}_{end} - \mathbf{p}_i|} \tag{7.2}$$

where f_s is the sample frequency of the eye tracker. The velocity towards the target was then plotted with the same alpha and the same normalized time as described above. The results from the velocity calculations sometimes, due to positional tracking errors, result in unreasonable values. The velocity v_i was therefore removed if it exceeded 2.5 m/s.

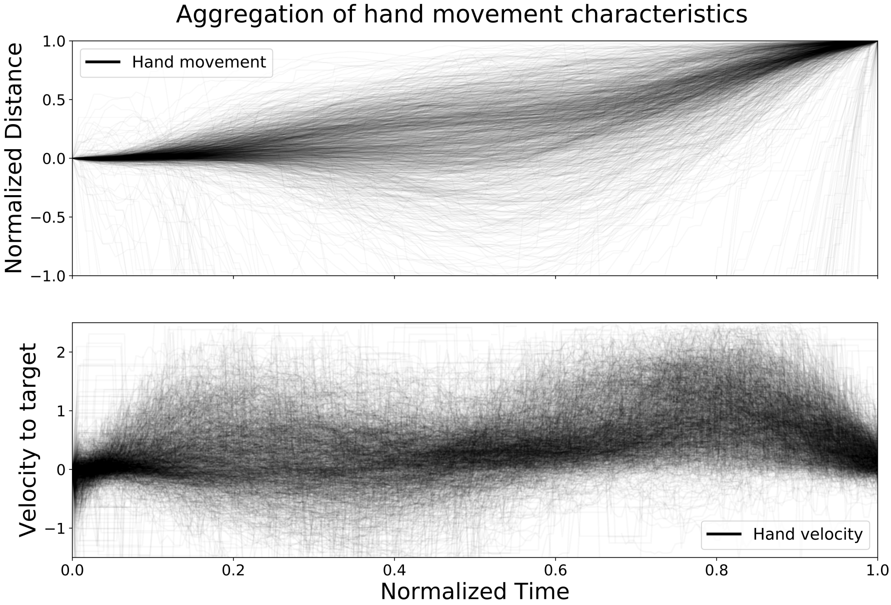


Figure 7.7: The figure shows an aggregation of all hand movements from the test set, with respect to distance left to target and velocity towards target, plotted with normalized time.

From the figure it is clear that the data is noisy and with some variation, however, a few trends are clearly emerging as well. The figure shows that there is little to no movement in the beginning of each time series followed by a segment with varying amount of movement, both towards and away from the target, up until about the halfway point. Around the halfway point the combination of the distance and the velocity graph shows a stationary segment followed by a new segment of movement that slows down towards the end. However, during the second movement segment almost all movements have positive velocity towards the target the entire time.

The normalized distance data from Figure 7.7 was used in Figure 7.8-7.10 to further investigate how early the movements were first correctly predicted, referred to as time ahead of movement completion (TAMC). The figures shows the first correct prediction from each of the movements that were correctly classified at least once. The histograms, top and right, shows the distributions of correct predictions with regards to the normalized time and normalized distance respectively. The 5th, 25th, 50th, 75th, 95th, and 99th percentiles were added in order to give a more nuanced description of the TAMC and the normalized time values for these are summarized in Table 7.3. The distributions of *LSTM* are more spread out, for both time and distance, compared to the other two networks and it is slightly faster than the other two since the concentration of points are shifted lower to the left. *Enc1* has the most compact distribution of points, concentrated to the upper right. This means that it is slightly slower at detecting movements. *Mix3* looks like a combination of the other two networks since its time distribution is similar to *LSTM* and the distance distribution is more concentrated to the upper half as in *Enc1*. The results in Table 7.3 shows that *LSTM* is the fastest at detecting movements for the lower percentiles, followed by *Mix3*, and then *Enc1*. However, the differences between the three networks decrease towards the end of the movement durations.

The first 5% of the correct predictions of the *LSTM* are most likely “lucky-shots”. These network predictions occur as the network sticks to the same prediction for the next movement, which due to the random box sequence sometimes was the same target twice in a row. They are called lucky since the test person, and therefore the network, can not know the next target for the first 0.2 s, i.e. first 5.4% to 27.0% of the movements for the max/min duration, due to the delay that was inserted between each task in the VRE. The “lucky-shots” are less prominent in both *Enc1* and *Mix3*.

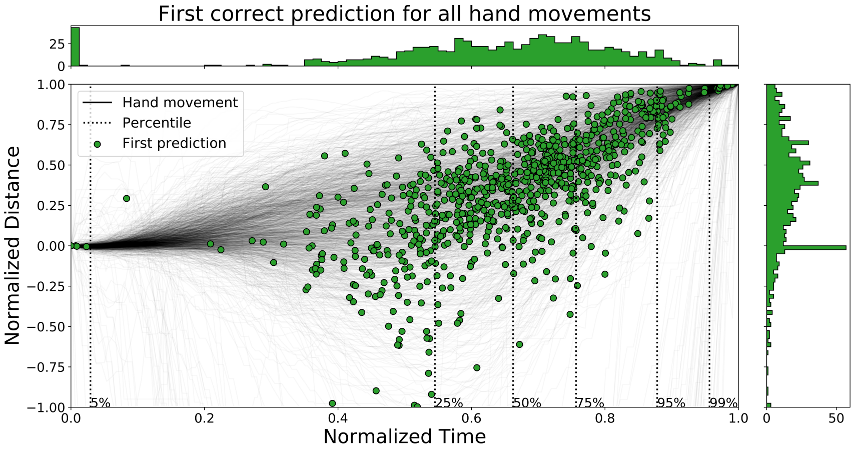


Figure 7.8: A figure that shows the first correct prediction for all hand movements from *LSTM* on the test set, plotted with normalized time and distance to target.

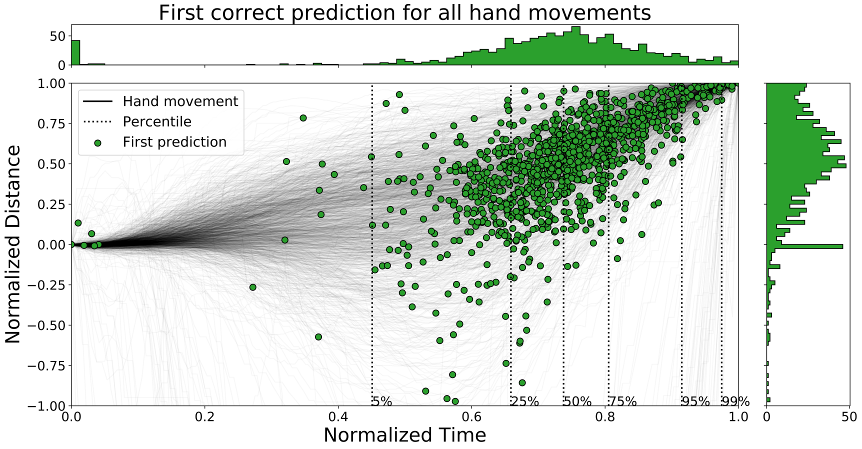


Figure 7.9: A figure that shows the first correct prediction for all hand movements from *Enc1* on the test set, plotted with normalized time and distance to target.

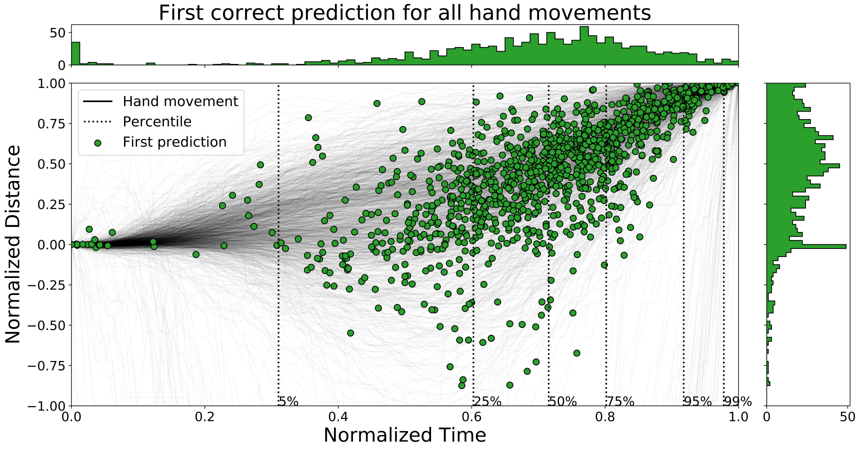


Figure 7.10: A figure that shows the first correct prediction for all hand movements from *Mix3* on the test set, plotted with normalized time and distance to target.

Table 7.3: A table that summarizes the normalized time values for the percentiles of all the evaluated networks.

Percentile	5	25	50	75	95	99
Enc1 - Normalized time $[0, 1]$	0.45	0.66	0.74	0.81	0.91	0.97
Mix3 - Normalized time $[0, 1]$	0.31	0.6	0.72	0.8	0.92	0.98
LSTM - Normalized time $[0, 1]$	0.03	0.55	0.66	0.76	0.88	0.96

7.2 Study 2 - Digitization of Raven's Progressive Matrices for Gaze Based Analysis using Virtual Reality and Eye Tracking

This section will provide the combined results from **Paper D-E** and show how gaze data from a psychological test can be visualized and analyzed. This is followed by an attempt at classification with the goal of investigating whether it is possible to determine what alternative a person selected based on where he/she was looking.

Paper D presented a way of gathering data in VR from a subset of Raven's Standard Progressive Matrices (SPM). The data was automatically processed and used to generate a report that displays the test results with detailed information about the performance on each item after the completion of the test. Table 7.4 shows an example of a test result summary.

Table 7.4: A summary of test results for a participant.

	Answer	Correct Answer	Is Correct	Time [s]
Item 1	4	4	True	9.53
Item 2	5	5	True	5.15
Item 3	1	1	True	4.17
Item 4	2	2	True	2.22
Item 5	2	6	False	4.01
Item 6	5	1	False	2.85
Item 7	8	8	True	1.69
Item 8	2	3	False	1.61
Item 9	6	3	False	2.68
Item 10	2	4	False	1.98

The report also contains information about the eye gaze movements for each item, presented as a trajectory that shows the path that the gaze travelled and a heatmap that shows the aggregated intensity of the gaze. An example of a trajectory can be seen in Figure 7.11 and the corresponding heatmap is presented in Figure 7.12.

The VR implementation was then extended in **Paper E** to include all 60 items in SPM and additional tools for visualization and analysis were added.

Each item can be separated into two parts, the board where the pattern is presented and the alternatives to chose from. This has been used to define areas of interest (AOIs), i.e. areas that contain information valuable to the test participant. The board, with the pattern to be completed, was divided into 9 zones in a 3x3 grid, Figure 7.13, due to the fact that the later difficulties of SPM are structured this way. These 9 zones, labeled Z1-Z9, together with the alternatives to choose from, T1-T6/8 (depending on difficulty level), where deemed the AOIs and all other areas (non-AOIs) were labeled as Background. The fixation points

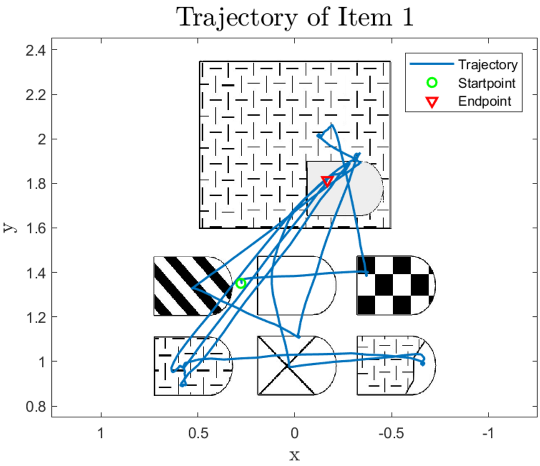


Figure 7.11: A trajectory that shows the path where the user's eye gaze was directed during the test. The x-axis is flipped because of the room placement in the VRE.

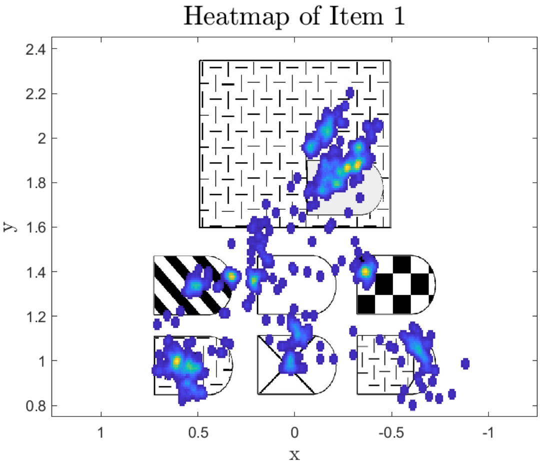


Figure 7.12: A heatmap that shows the intensity of the user's eye gaze during the test. The x-axis is flipped because of the room placement in the VRE.

were projected onto the plane of the item in the VRE and were labeled according to within which AOI-borders they lie.

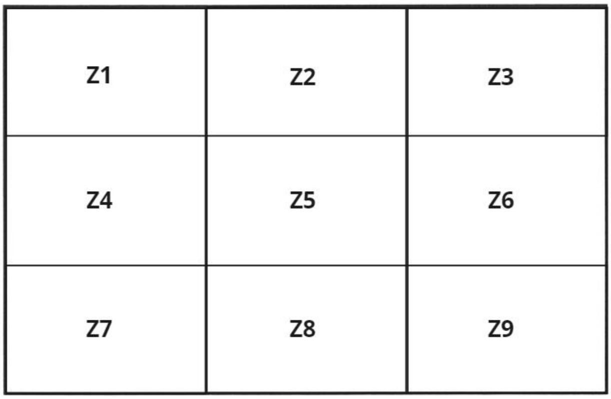


Figure 7.13: A graphical illustration of the 9 zone allocations used together with the item alternatives to define AOIs.

An analysis toolbox, summarized in Table 7.5, with a graphical interface was developed for analysis and visualization of data. The toolbox consists of two different views; an item view where a participant can be analyzed in detail on one specific item at a time and a participant view where it is possible to view a participant's test metrics for the entire test. The item view consists of the following analysis tools: the trajectory that shows the path that the gaze travelled and the heatmap that shows the aggregated intensity of the gaze from **Paper D**, the fixation points plotted as an overlay to the item, the ability to change how much of the trajectory that is shown at a time, a directed graph obtained using AOIs (Figure 7.14), and a Gantt chart that shows the AOIs as time allocations across the span of an item (Figure 7.15). The second view is the participant view where it is possible to view a participant's test metrics for the entire test, such as score per item, time spent per item, and amount of fixations on the board vs the alternatives for each item (Figure 7.16). Each of these metrics also provides a comparison to the population mean of the data set.

Table 7.5: A summary of the analysis toolbox with explanations of the different tools.

Tool	Explanation
Trajectory	An overlay to the item image that shows the path that the gaze travelled during an item, obtained using raw gaze data.
Heatmap	An overlay to the item image that shows the aggregated intensity of the gaze during an item, obtained using raw gaze data.
Fixation overlay	An overlay to the item image that shows all the fixation points.
Fixation graph	A directed graph that shows the cumulative time spent fixating on AOIs and how the gaze was moved between these.
Fixation time allocation	A Gantt chart that shows how much time was spent fixating on AOIs, at what time, and in what order these were gazed upon.
Score per item	A line chart that shows the score on each item, including the maximum obtainable score, for the participant and the population mean.
Time per item	A line chart that shows the time spent on each item for the participant and the population mean.
Fraction of fixation time on the board	A line chart that shows how large percentage of the total fixation time that was spent fixating on the board.

The two tools, fixation graph and fixation time allocation, are new ways of visualizing fixations that are particularly interesting because they are less visually complex than, e.g. scanpaths, while at the same time providing a lot of the same information. The key difference is that they both use predefined AOIs to aggregate the fixations, thereby limiting the number of nodes in the graph and rows in the Gantt chart to a fixed amount. This sacrifices some level of detail, since it does not distinguish between different fixation locations on the same AOI. However, this simplification makes it possible to more easily observe the spatial and temporal order of the fixations for longer sequences of data. The directed graph in Figure 7.14 presents an overview of the gaze fixations of a participant in the process of deciding which alternative to choose when completing the pattern on the board. The graph was calculated using each switch in AOI during an item. Note that the points labeled as

Background were excluded from the graph to improve readability since they mostly occur when the participants switch between AOIs as seen later in Figure 7.15. A larger node means that it was visited more and the percentage within the node shows how large a fraction of time was spent here. A small node without arrows means that the node was not visited. The circles correspond to the zones of the board and the rectangles are the alternatives. A transition between two nodes occurred in the direction of the arrow (edge) and the thickness of the arrow shows how many times that edge was travelled. The green node is the first visited node of the graph and the blue is the last one. Using Figure 7.14 as an example we can see that this particular individual on this specific item started out by looking at alternative 5, then moved to Z8, followed by Z5, and so on, spending most of his/her time on alternative 6. This way of visualizing data thus gives us a very clear and concise overview of the result, however, the exact ordering of the events is sometimes lost for graphs with more transitions.

The Gantt chart in Figure 7.15 shows the AOIs as time allocations across the time span of an item, measured in number of ET samples. The duration of an allocation is determined by the fixation duration for that particular fixation point. The ones that are just a sample or two wide are more likely transitions between AOIs than actual fixations. Note that the chart only displays the AOIs that were fixated upon at least once. This chart also has the added benefit of being able to display the precise amount of time spent fixating on each alternative, for each fixation, and the correct order of the fixations. Using Figure 7.15, the same observations as in Figure 7.14 can be made, however, this time it becomes clear that the time spent on alternative 6 is actually separated into three different segments.

These ways of visualizing a participant's eye gaze makes it possible to analyze the behaviour during the entire duration of an item from the participant's perspective, compared to using a traditional pen-and-paper or a digitized version without ET that relies on external observation and the final score. Monitoring eye gaze makes it possible to, for example, investigate whether the correct alternative, to a wrongly answered item, was considered at all, if it was discarded early, or if it was observed a lot even though it did not end up being the final answer. This introduces more nuances to the analysis in an objective way since it does not disturb the participant nor require any additional questions to be answered.

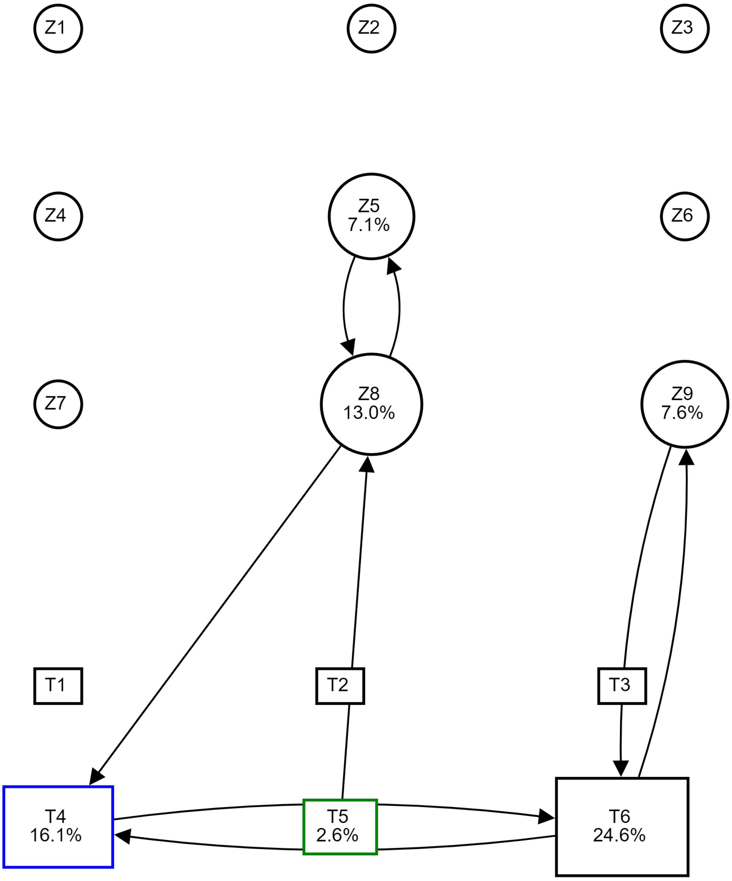


Figure 7.14: A figure that shows a directed graph corresponding to the path of AOIs travelled by the gaze during the test.

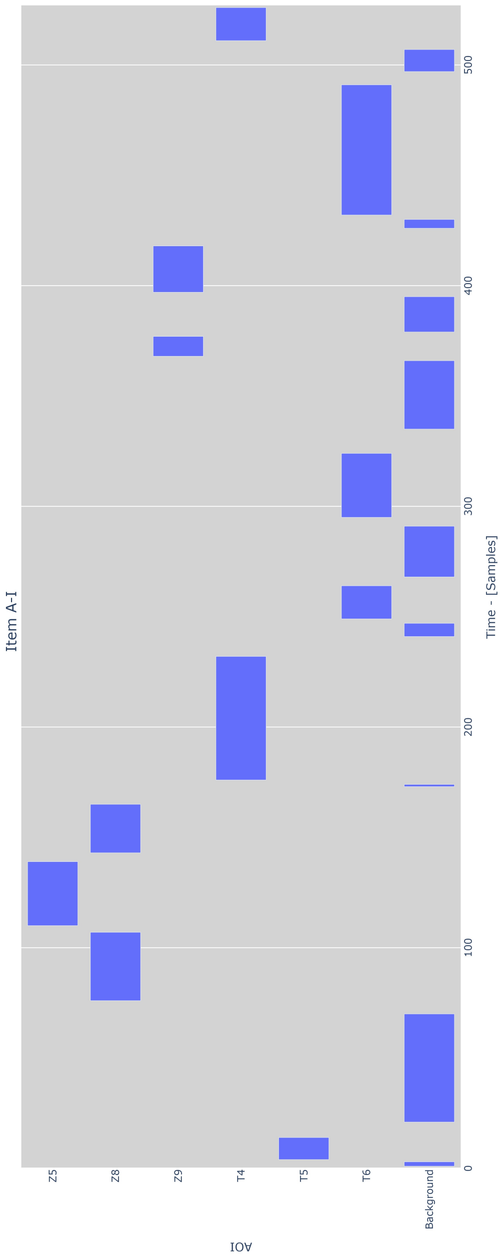


Figure 7.15: A figure that shows the participant's fixation time allocation on an item with the number of ET samples as the x-axis.

In the toolbox there are also graphs for visualizing data on group level. The fraction of fixation time spent on the board versus the alternatives on each item for the population mean is shown in Figure 7.16. The solid line is the average time values of the participant and the dotted line is composed of two averages one from 1-19 and the other one from 20-60. The plot shows that there is a lot of variation between items, however, it is clear that the average participant suddenly spend a larger portion of their fixation time on the board from item 20 and onward.

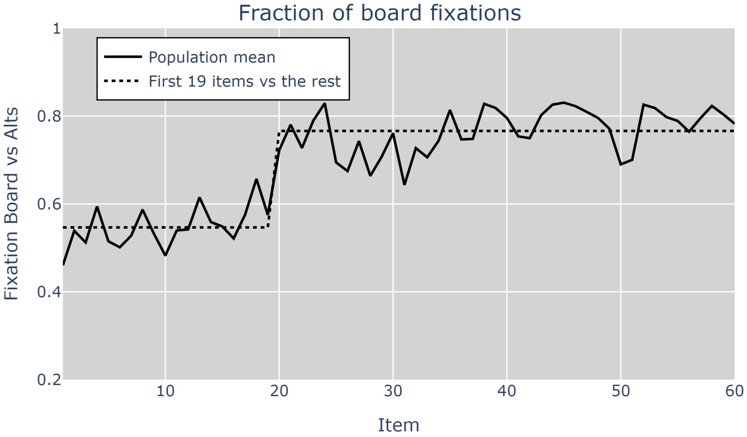


Figure 7.16: A figure that highlights the mean behaviour for the fraction of the total fixation time that occurred on the board for each item. It also shows the possible strategy shift that happens in B-VII.

Two interesting observations were made once the population mean (of the quite small and homogeneous dataset) was plotted. First, there is a clear jump after item 19 indicating that there is some change in how the average participant solves the items from this point onwards. Secondly, the average time spent on each item can be seen in Figure 7.17. The solid line shows the average time values plotted with a logarithmic y-axis. This highlights three things: the first item takes an unexpectedly long time to solve, even though the participants had time to “warm-up” on a few similar items, the trend of a steadily increased time per item is even more pronounced, and finally, it is clear that the time drops significantly at the start of every difficulty level (item 13, 25, 37, and 49) except from the first one. Observing the results on a group level thus also shows interesting characteristics of the test itself.

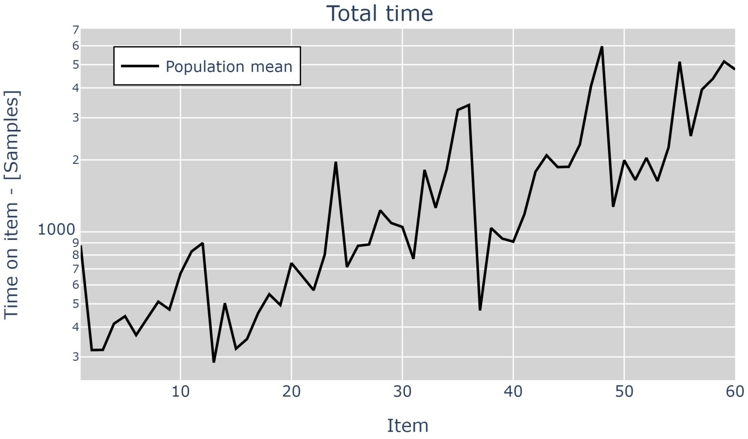


Figure 7.17: A figure that shows a saw-like pattern of the mean total time spent on each item with a logarithmic y-axis. It highlights the varying and increasing difficulty of the test.

An interesting question to investigate is whether artificial neural networks (ANNs) could aid in providing even more insights into the decision-making processes and strategies that precede the choices during a problem-solving task. A starting point could be to use ANNs to be able to classify what alternative a participant selected as an answer to an item from the SPM, based only on where he/she had been looking. A successful classification would prove that the eye gaze contains the information that leads to a decision to select a particular answer. This is the first step into investigating at what point in time a person selected an answer and possibly what other alternatives that were considered.

The results from the kFCV show that the network is able to use eye gaze data to classify what alternative a participant chose as an answer for an item with an accuracy of 47.81% ± 8.39. The different levels of difficulty for SPM (A-E) had slightly varying levels of accuracy: A = 45.99%, B = 54.32%, C = 45.37%, D = 50.63%, and E = 42.63%. This shows that level B was slightly easier to classify than the others and E slightly more difficult.

CHAPTER 8

Summary of included papers

This chapter provides a summary of the included papers.

8.1 Paper A

Julius Pettersson and Petter Falkman
Human Movement Direction Classification using Virtual Reality and Eye Tracking
Published in Procedia Manufacturing, Volume 51, (pp. 95-102), 2020.

Combining the areas of virtual reality, eye-tracking and machine learning can be one way to increase the intelligence of collaborative robots. This could be broken down into three stages, **Stage One:** *Movement Direction Classification*, **Stage Two:** *Movement Phase Classification*, and **Stage Three:** *Movement Intention Prediction*, described in the introduction. This paper gives a solution to the first stage and shows that it is possible to collect eye gaze data and use that to classify a person's movement direction. The results clearly shows that it is possible to combine virtual reality and eye tracking into a platform for testing and analysis of human behaviour, which can be beneficial in multiple areas of research. It is also shown that the implementation of uncertainty estimation improves the network and provides a way to improve the classification accuracy, at the cost of the percentage of samples classified, to obtain a more confident network.

8.2 Paper B

Julius Pettersson and Petter Falkman
Intended Human Arm Movement Direction Prediction using Eye Tracking
Re-submitted to: IJCIM International Journal of Computer Integrated Manufacturing, 2022.

The goal of this paper was to provide a system for intended human arm movement prediction and the two classification objectives, **Primary** - *determine the discrete horizontal direction corresponding to the box that was clicked* and **Secondary** - *distinguish between whether the movement occurred on the upper or lower level of boxes.* The best network reached an accuracy of 70.70% for the primary objective, correctly classifies 67.89% of the movements at least once, and an accuracy of 81.29% for the secondary objective. These results might seem far from 100%, however, it is important to remember that human behaviour is complex and difficult to capture. It is, therefore, perhaps impossible to reach 100% and maybe not a requirement for intention prediction to provide value. Considering that the system predicts upcoming movement directions, before the completion of these events, solely based on eye gaze and without knowing the directions specifically only the number of directions (10) it becomes easier to see the benefits of the system. A robot could receive warnings regarding in which direction an operator is likely to move and adjust its behaviour accordingly.

8.3 Paper C

Julius Pettersson and Petter Falkman
Comparison of LSTM, Transformers, and MLP-Mixer Neural Networks for Gaze Based Human Intention Prediction
Accepted in: Frontiers in Neurorobotics, 2023.

This paper builds upon the work presented in **Paper B** where eye gaze and movement data was gathered and used to train an LSTM network to perform gaze based arm movement prediction. Using the same data, this paper has provided two additional solutions to the classification objectives: **Primary** - *determine the discrete horizontal direction corresponding to the box that was clicked* and **Secondary** - *distinguish between whether the movement occurred on the upper or lower level of boxes.* A comparison with respect to accuracy for a given uncertainty threshold, time ahead of movement completion, and the execution time of a single prediction using the three methods is also presented.

8.4 Paper D

Julius Pettersson, Anton Albo, Johan Eriksson, Patrik Larsson, Kerstin W. Falkman, and Petter Falkman
Cognitive Ability Evaluation using Virtual Reality and Eye Tracking
In 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), (pp. 1-6), 2018.

The aim of this paper was to implement a simplified version of Raven's Progressive Matrices in a virtual reality environment where the eye gaze data is saved and compiled to a document that can be used by psychologists during a diagnostic process or in research. The data has potential to indicate how the test persons reason while solving the different problems and could be implemented as an extension of the psychologists current toolbox. Furthermore, it has been found that test participants are less distracted by external disturbances due to the virtual environment implementation. The virtual environment could also be extended, as part of future work, to include external disturbances that could be manipulated by the test conductor in order to investigate how the different test participants respond to different types of disturbances.

8.5 Paper E

Julius Pettersson, Kerstin W. Falkman, and Petter Falkman
Exploring the usability of Virtual Reality and Eye Tracking for Psychological Testing using Raven's Progressive Matrices
Submitted to: Frontiers in Psychology, 2023.

The paper provides a full implementation of Raven's Standard Progressive Matrices in VR with ET. It presents the execution of a pilot study and the data that was obtained. Finally, the data was used to investigate what information can be extracted from monitoring the gaze during the test and presents a toolbox with examples of possible ways to analyze the data. The different tools incorporated in the toolbox show that it is possible to use VR and ET to visualize data in a way that gives an immediate, clear, and concise overview of the results on several levels, both for each individual participant and on a group level, as well as for each item and for the test as a whole. It is also possible to compare these individual results with the group mean. Visualizing a participant's eye gaze using these tools makes it possible to analyze the behaviour of a participant, from her/his perspective, during the entire duration of an item, compared to using a traditional pen-and-paper or a digitized version without ET that relies on external observation and the final score.

CHAPTER 9

Concluding Remarks and Future Work

The ability to determine an upcoming action or what decision a human is about to take, can be useful in multiple areas, for example during human-robot collaboration in manufacturing, where knowing the intent of the operator could provide the robot with important information to help it navigate more safely. Another field that could benefit from a system that provides information regarding human intentions is the field of psychological testing where such a system could be used as a platform for new research or be one way to provide information in the diagnostic process. The work that was presented in this thesis investigated the potential use of virtual reality as a safe, measurable environment suitable to gather gaze and movement data, eye tracking as the non-invasive system input that gives insight into the human mind, and deep machine learning as one tool to analyze the data. The thesis defined an experimental procedure that can be used to construct a virtual reality based testing system that gathers gaze and movement data, carry out a test study with human participants, and implementation of artificial neural networks in order to analyze human behaviour. This was followed by two studies that gave evidence to the decisions that were made in the experimental procedure and demonstrated the potential use cases of such a system.

It is possible that the VREs that are used in Study 1, Chapter 7, contain biases. For example, the nature of the tasks that have been implemented guarantees that the participant has to direct the gaze towards interesting areas to complete most of the movements. The randomness that is present in **Paper A-C** is also inhibiting a learning process that could potentially move participants from using the foveal vision to utilizing more of the periphery as, for example, the stationary placement of parts in a picking station is learnable. The goal with the simplicity of the tasks was, however, to enable a discussion regarding the performance of the different systems and the behaviour of the participants.

In Chapter 4, the concept of fixations and how these can be calculated was explained. From experimental results these showed little to no performance gain in Study 1. However, this should be investigated further since the lack of importance might either be due to that the network learns to calculate these on its own or that the tasks that was used are not complex enough to make the fixations provide any additional information to solve the objective. Finding, for example, a box that has been lit may mostly rely on peripheral vision, which excels at detecting changes in brightness whereas a more complex search task would probably rely more on using foveal vision to distinguish between objects or patterns. One other important aspect to consider is that a system based on eye gaze will always have limited abilities to analyze decisions made from peripheral vision since the ET hardware is only capable of measuring the direction of the foveal vision.

The eye tracker used in the presented studies collects data at 120 Hz, which means that it should capture most eye movements, including the faster saccades that typically range from 10-100 ms in duration. A faster tracker, up to maybe 200 Hz, could give some additional information regarding the fastest eye movements. However, it is not certain that it will improve the results of the intention prediction as it will also give more data samples that are similar to each other for the slower movements, i.e. there will be a trade-off between new information and overflow of data, and this problem is likely becoming more prominent for even faster tracking systems. A slower tracker would deal with the issue of too much data, however, due to the nature of the eye and the rapid movements, saccades, the lower limit to capture the majority of eye movements is probably around or slightly above $\frac{1}{10\text{ ms}}$, i.e. about 120 Hz.

RQ1: *Is it possible to analyze and predict human intention through the study of eye gaze?*

It was shown in **Paper B** that it is possible to predict the intended arm movement direction the reaches an accuracy of 70.7%, for predictions with high certainty, on a continuous stream of eye gaze data. These results were further improved in **Paper C** where the best ANN network achieved an accuracy of 82.74%, for predictions with high certainty, on continuously incoming data and correctly classifies 80.06% of the movements at least once. The movements are, in 99% of the cases, correctly predicted the first time, before the hand reaches the target and more than 19% ahead of movement completion in 75% of the cases, which corresponds to about 239 ms for the median movement duration of the task.

RQ2: *Is DML a suitable tool to analyze the connection between eye gaze and intention in humans?*

In **Paper A** and **Paper E**, it is shown that DML can be used to identify behavioural patterns in eye gaze data for classification of movements and answers to psychological tests respectively. However, the complexity of the objective in **Paper B-C**, continuously predict the discrete horizontal direction that a human is about to/is moving, is what really shows the capabilities of DML. **Paper B-C** also demonstrates that there are multiple ways of reaching the same goal using different network architectures.

RQ3: *How can a VRE-test be designed to gather the necessary eye gaze and movement data to be used for human intention analysis?*

Paper D showed that it is possible to use VR to gather eye gaze and movement data from humans performing the task of solving a logical pattern. This formed the basis for **Paper A-C**, where this concept is transferred from logical patterns to tasks involving reaching for specific objects in the VRE and the data is successfully used to first classify and then predict human intention. Finally, in **Paper E** it is shown that gaze data from a psychological test, obtained using VR, can be used to classify what answer a participant selected.

9.1 Future Work

The test studies described in Chapter 7 are somewhat limited in the way the data has been collected. In order to fully ensure that the methodology works on a more generalized scale one should redo these tests, or similar tests, with groups that are larger and where the participants have been selected by people that are experts in creating diversified test groups.

The VREs used to achieve the results in **Paper A-C** contains simplified tasks that are similar to the ones present in, for example, a pick-and-place station in a manufacturing environment. In order to further evaluate the described procedure and the results from **Paper B-C**, the implementation of a VRE with more complex tasks would be of interest. This could include an assembly station were the operator collaborates with a virtual robot with an external control system that receives the predicted intentions and makes the robot adapt accordingly.

A natural extension to the suggestion above, if the results are determined successful, would be to implement the system in a real world application, preferably similar to the industrial application that is used for the step above. This would include using safety glasses with built-in ET instead of a VR-headset and the evaluation of the validity of using such a system in a real-time environment.

Another topic to explore would be to use the full version of RPM from **Paper E** to gather data in a study formed by researchers in the field of psychology, and together with them determine a few interesting objectives that may be solved using the procedure described in this thesis, for example to implement the functionality from **Paper B** to perform analysis on continuous data and use that to determine when the participant decided on what to answer.

References

[1] I. El Makrini, K. Merckaert, D. Lefeber, and B. Vanderborght, "Design of a collaborative architecture for human-robot assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 1624–1629.

[2] J. Krüger, T. K. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP annals*, vol. 58, no. 2, pp. 628–646, 2009.

[3] R. M. Bakwin, A. Weider, and H. Bakwin, "Mental testing in children," *The Journal of pediatrics*, vol. 33, no. 3, pp. 384–394, 1948.

[4] A.-C. Smedler and E. Tideman, *Att testa barn och ungdomar : om testmetoder i psykologiska utredningar*, 1. utg. Stockholm: Natur & kultur, 2009, ISBN: 978-91-27-11692-4 (inb.)

[5] T. D. Gould, T. M. Bastain, M. E. Israel, D. W. Hommer, and F. X. Castellanos, "Altered performance on an ocular fixation task in attention-deficit/hyperactivity disorder," *Biological psychiatry*, vol. 50, no. 8, pp. 633–635, 2001.

[6] C. Karatekin, "Eye tracking studies of normative and atypical development," *Developmental review*, vol. 27, no. 3, pp. 283–348, 2007.

[7] F. Jungwirth, M. Murauer, M. Haslgrübler, and A. Ferscha, "Eyes are different than hands: An analysis of gaze as input modality for industrial man-machine interactions," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, ACM, 2018, pp. 303–310.

[8] G. Tang, P. Webb, and J. Thrower, "The development and evaluation of robot light skin: A novel robot signalling system to improve communication in industrial human–robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 56, pp. 85–94, 2019.

[9] R. T. Chadalavada, H. Andreasson, M. Schindler, R. Palm, and A. J. Lilienthal, "Bi-directional navigation intent communication using spatial augmented reality and eye-tracking glasses for improved safety in human–robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101 830, 2020.

[10] M. Hochman, Y. Parmet, and T. Oron-Gilad, "Pedestrians' understanding of a fully autonomous vehicle's intent to stop: A learning effect over time," *Frontiers in psychology*, vol. 11, 2020.

[11] N. I. Vargas-Cuentas, D. Hidalgo, A. Roman-Gonzalez, M. Power, R. H. Gilman, and M. Zimic, "Diagnosis of autism using an eye tracking system," in *Global Humanitarian Technology Conference (GHTC), 2016*, IEEE, 2016, pp. 624–627.

[12] M. Dahl, A. Albo, J. Eriksson, J. Pettersson, and P. Falkman, "Virtual reality commissioning in production systems preparation," in *22nd IEEE International Conference on Emerging Technologies And Factory Automation, September 12-15, 2017, Limassol, Cyprus*, IEEE, 2017, pp. 1–7.

[13] A. A. Rizzo, M. Schultheis, K. A. Kerns, and C. Mateer, "Analysis of assets for virtual reality applications in neuropsychology," *Neuropsychological Rehabilitation*, vol. 14, no. 1-2, pp. 207–239, 2004.

[14] R. Adams, P. Finn, E. Moes, K. Flannery, and A. Rizzo, "Distractibility in attention/deficit/hyperactivity disorder (adhd): The virtual reality classroom," *Child Neuropsychology*, vol. 15, no. 2, pp. 120–135, 2009.

[15] Y. Pollak, P. L. Weiss, A. A. Rizzo, *et al.*, "The utility of a continuous performance test embedded in virtual reality in measuring adhd-related deficits," *Journal of Developmental & Behavioral Pediatrics*, vol. 30, no. 1, pp. 2–6, 2009.

[16] M. Abidi, A. Al-Ahmari, A. El-Tamimi, S. Darwish, and A. Ahmad, "Development and evaluation of the virtual prototype of the first saudi arabian-designed car," *Computers*, vol. 5, no. 4, p. 26, 2016.

[17] A. M. Al-Ahmari, M. H. Abidi, A. Ahmad, and S. Darmoul, "Development of a virtual manufacturing assembly simulation system," *Advances in Mechanical Engineering*, vol. 8, no. 3, p. 1 687 814 016 639 824, 2016.

[18] D. Aschenbrenner, N. Maltry, J. Kimmel, M. Albert, J. Scharnagl, and K. Schilling, "Artab-using virtual and augmented reality methods for an improved situation awareness for telemaintenance," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 204–209, 2016.

[19] J. Pettersson, A. Albo, J. Eriksson, P. Larsson, K. Falkman, and P. Falkman, "Cognitive ability evaluation using virtual reality and eye tracking," in *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, IEEE, 2018, pp. 1–6.

[20] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: The management revolution," *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.

[21] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[22] K. Nagorny, P. Lima-Monteiro, J. Barata, and A. W. Colombo, "Big data analysis in smart manufacturing: A review," *International Journal of Communications, Network and System Sciences*, vol. 10, no. 3, pp. 31–58, 2017.

[23] O. Morariu, C. Morariu, T. Borangiu, and S. Răileanu, "Manufacturing systems at scale with big data streaming and online machine learning," in *Service Orientation in Holonic and Multi-Agent Manufacturing*, Springer, 2018, pp. 253–264.

[24] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018.

[25] Y. Cui, M. Gierl, and Q. Guo, "Statistical classification for cognitive diagnostic assessment: An artificial neural network approach," *Educational Psychology*, vol. 36, no. 6, pp. 1065–1082, 2016.

[26] G. Deshpande, P. Wang, D. Rangaprakash, and B. Wilamowski, “Fully connected cascade artificial neural network architecture for attention deficit hyperactivity disorder classification from functional magnetic resonance imaging data,” *IEEE transactions on cybernetics*, vol. 45, no. 12, pp. 2668–2679, 2015.

[27] M. Awais and D. Henrich, “Human-robot collaboration by intention recognition using probabilistic state machines,” in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, IEEE, 2010, pp. 75–80.

[28] W. Wang, R. Li, Y. Chen, Y. Sun, and Y. Jia, “Predicting human intentions in human-robot hand-over tasks through multimodal learning,” *IEEE Transactions on Automation Science and Engineering*, 2021.

[29] J. Mainprice and D. Berenson, “Human-robot collaborative manipulation planning using early prediction of human motion,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 299–306.

[30] C.-M. Huang and B. Mutlu, “Anticipatory robot control for efficient human-robot collaboration,” in *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*, IEEE, 2016, pp. 83–90.

[31] L. Bi, C. Guan, *et al.*, “A review on emg-based motor intention prediction of continuous human upper limb motion for human-robot collaboration,” *Biomedical Signal Processing and Control*, vol. 51, pp. 113–127, 2019.

[32] A. Haji Fathaliyan, X. Wang, and V. J. Santos, “Exploiting three-dimensional gaze tracking for action recognition during bimanual manipulation to enhance human–robot collaboration,” *Frontiers in Robotics and AI*, vol. 5, p. 25, 2018.

[33] L. Shi, C. Copot, and S. Vanlanduit, “Gazeemd: Detecting visual intention in gaze-based human-robot interaction,” *Robotics*, vol. 10, no. 2, p. 68, 2021.

[34] H. chaandar Ravichandar, A. Kumar, and A. Dani, “Bayesian human intention inference through multiple model filtering with gaze-based priors,” in *2016 19th International Conference on Information Fusion (FUSION)*, IEEE, 2016, pp. 2296–2302.

[35] C. Gomez Cubero and M. Rehm, "Intention recognition in human robot interaction based on eye tracking," in *IFIP Conference on Human-Computer Interaction*, Springer, 2021, pp. 428–437.

[36] Psykologförbundet, *Hantering och förvaring av psykologiska test inom hälso- och sjukvården*, [Online], `https://www.psykologforbundet.se/globalassets/omforbundet/hantering-och-forvaring-av-psykologiska-test.pdf`, 2013.

[37] J. H. Elder, "Videotaped behavioral observations: Enhancing validity and reliability," *Applied Nursing Research*, vol. 12, no. 4, pp. 206–209, 1999.

[38] M. Dyck, M. Winbeck, S. Leiberg, Y. Chen, R. C. Gur, and K. Mathiak, "Recognition profile of emotions in natural and virtual faces," *PLoS One*, vol. 3, no. 11, e3628, 2008.

[39] P. Lindner, A. Miloff, W. Hamilton, *et al.*, "Creating state of the art, next-generation virtual reality exposure therapies for anxiety disorders using consumer hardware platforms: Design considerations and future directions," *Cognitive Behaviour Therapy*, pp. 1–17, 2017.

[40] N. J. Emery, "The eyes have it: The neuroethology, function and evolution of social gaze," *Neuroscience & Biobehavioral Reviews*, vol. 24, no. 6, pp. 581–604, 2000.

[41] A. I. Goldman *et al.*, *Theory of mind*, 2012.

[42] A. Armanini and N. Conci, "Eye tracking as an accessible assistive tool," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, IEEE, 2010, pp. 1–4.

[43] A. Navab, K. Gillespie-Lynch, S. P. Johnson, M. Sigman, and T. Hutman, "Eye-tracking as a measure of responsiveness to joint attention in infants at risk for autism," *Infancy*, vol. 17, no. 4, pp. 416–431, 2012.

[44] D. Riby and P. J. Hancock, "Looking at movies and cartoons: Eye-tracking evidence from williams syndrome and autism," *Journal of Intellectual Disability Research*, vol. 53, no. 2, pp. 169–181, 2009.

[45] P. Deans, L. O'Laughlin, B. Brubaker, N. Gay, and D. Krug, "Use of eye movement tracking in the differential diagnosis of attention deficit hyperactivity disorder (adhd) and reading disability," *Psychology*, vol. 1, no. 04, p. 238, 2010.

[46] T. Yarkoni and J. Westfall, "Choosing prediction over explanation in psychology: Lessons from machine learning," *Perspectives on Psychological Science*, vol. 12, no. 6, pp. 1100–1122, 2017.

[47] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: Machine learning and application to spontaneous behavior," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, vol. 2, 2005, pp. 568–573.

[48] J. A. Russell and J. M. Fernández-Dols, *The psychology of facial expression.* Cambridge university press, 1997.

[49] C. Chandler, T. B. Holmlund, P. W. Foltz, A. S. Cohen, and B. Elvevåg, "Extending the usefulness of the verbal memory test: The promise of machine learning," *Psychiatry Research*, vol. 297, p. 113 743, 2021.

[50] C. Spearman, ""general intelligence," objectively determined and measured," *The American Journal of Psychology*, vol. 15, no. 2, pp. 201–292, 1904, ISSN: 00029556.

[51] J. C. Raven *et al.*, *Raven's progressive matrices.* Oxford Psychologists Press Oxford, 1998.

[52] J. W. Kalat, *Introduction to psychology. pacific grove, ca: Brooks*, 1996.

[53] J. Raven, "The raven's progressive matrices: Change and stability over culture and time," *Cognitive Psychology*, vol. 41, no. 1, pp. 1–48, 2000, ISSN: 0010-0285.

[54] J. Raven, "The raven's progressive matrices: Change and stability over culture and time," *Cognitive psychology*, vol. 41, no. 1, pp. 1–48, 2000.

[55] J. C. Raven, J. Court, and J. Raven, *Manual for Raven's Progressive Matrices and Vocabulary Scales by JC Raven, JH Court and J. Raven; Section2; Coloured Progressive Matrices.* Oxford Psychologist Press, 1995.

[56] J. Raven *et al.*, "Raven progressive matrices," in *Handbook of nonverbal assessment*, Springer, 2003, pp. 223–237.

[57] J. Raven, J. Rust, F. Chan, and X. Zhou, *Raven's 2 progressive matrices, clinical edition (raven's 2)*, 2018.

[58] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The cave: Audio visual experience automatic virtual environment," *Commun. ACM*, vol. 35, no. 6, pp. 64–72, 1992, ISSN: 0001-0782.

[59] S. Choi, K. Jung, and S. Do Noh, "Virtual reality applications in manufacturing industries: Past research, present findings, and future directions," *Concurrent Engineering*, vol. 23, no. 1, p. 56, 2015.

[60] A. A. Rizzo, T. Bowerly, J. G. Buckwalter, D. Klimchuk, R. Mitura, and T. D. Parsons, "A virtual reality scenario for all seasons: The virtual classroom," *Cns Spectrums*, vol. 11, no. 1, pp. 35–44, 2009.

[61] A. Poole and L. J. Ball, "Eye tracking in hci and usability research," *Encyclopedia of human computer interaction*, vol. 1, pp. 211–219, 2006.

[62] A. T. Duchowski and A. T. Duchowski, *Eye tracking methodology: Theory and practice.* Springer, 2017.

[63] J. Hartwig, A. Kretschmer-Trendowicz, J. Helmert, M. Jung, and S. Pannasch, "Revealing the dynamics of prospective memory processes in children with eye movements," *International Journal of Psychophysiology*, vol. 160, pp. 38–55, 2021.

[64] V. K. Sharma, L. Murthy, K. Singh Saluja, V. Mollyn, G. Sharma, and P. Biswas, "Webcam controlled robotic arm for persons with ssmi," *Technology and Disability*, no. Preprint, pp. 1–19, 2020.

[65] A. Keshava, A. Aumeistere, K. Izdebski, and P. Konig, "Decoding task from oculomotor behavior in virtual reality," in *ACM Symposium on Eye Tracking Research and Applications*, 2020, pp. 1–5.

[66] W. R. Hendee and P. Wells, *The perception of visual information*, 1993.

[67] P. Majaranta and A. Bulling, "Eye tracking and eye-based human–computer interaction," in *Advances in physiological computing*, Springer, 2014, pp. 39–65.

[68] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, 2000, pp. 71–78.

[69] A. Olsen and R. Matos, "Identifying parameter values for an i-vt fixation filter suitable for handling data sampled with various sampling frequencies," in *proceedings of the symposium on Eye tracking research and applications*, 2012, pp. 317–320.

[70] A. Olsen, "The tobii i-vt fixation filter," *Tobii Technology*, vol. 21, 2012.

[71] P. Langley, "The changing science of machine learning," *Machine Learning*, vol. 82, no. 3, pp. 275–279, 2011.

[72] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, `http://www.deeplearningbook.org`.

[73] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[74] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[75] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[76] K. Cho, B. Van Merriënboer, C. Gulcehre, *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[77] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[78] A. Labach, H. Salehinejad, and S. Valaee, "Survey of dropout methods for deep neural networks," *arXiv preprint arXiv:1904.13310*, 2019.

[79] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[80] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[81] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[83] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[84] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[85] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks,” in *International Conference on Machine Learning*, PMLR, 2016, pp. 1747–1756.

[86] A. v. d. Oord, S. Dieleman, H. Zen, *et al.*, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[87] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[88] D. W. Hansen and P. Majaranta, “Basics of camera-based gaze tracking,” in *Gaze interaction and applications of eye tracking: Advances in assistive technologies*, IGI Global, 2012, pp. 21–26.

[89] M. L. Mele and S. Federici, “Gaze and eye-tracking solutions for psychological research,” *Cognitive processing*, vol. 13, no. 1, pp. 261–265, 2012.

[90] L. Zhang, J. Sturm, D. Cremers, and D. Lee, “Real-time human motion tracking using multiple depth cameras,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 2389–2395.

[91] A. H. Moreira, S. Queirós, J. Fonseca, P. L. Rodrigues, N. F. Rodrigues, and J. L. Vilaça, “Real-time hand tracking for rehabilitation and character animation,” in *2014 IEEE 3nd International Conference on Serious Games and Applications for Health (SeGAH)*, IEEE, 2014, pp. 1–8.

[92] Tobii AB, *Tobii pro vr integration – based on htc vive development kit description*, v.1.7 - en-US, Accessed on: Feb. 13, 2020. [Online]. Available: `https://www.tobiipro.com/siteassets/tobii-pro/product-descriptions/tobii-pro-vr-integration-product-description.pdf/?v=1.7`, Tobii AB.

[93] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[94] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus),” *arXiv preprint arXiv:1606.08415*, 2016.

[95] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.