Universität Zürich[UZH]

**Institut für Computerlinguistik**

# Predicting Reading Fluency with LSTMs

based on Visual Search Eye-Tracking Data

Bachelorarbeit der Philosophischen Fakultät der Universität Zürich

Referent: J. Brasser

Verfasserin:
Jana Mara Hofmann
Matrikelnummer 17-709-361
Malzstrasse 26
8400 Winterthur

December 1, 2023

## Abstract

Numerous studies have demonstrated that eye-tracking data can provide valuable insights into an individual's reading abilities. The patterns of fixations, saccades and regressions offer valuable information on comprehension, fluency and overall reading proficiency. In addition, visual search tasks, which involve scanning a visual field for a target, have been extensively studied in relation to attentional processes and visual cognition. Despite recent advances, the lack of research that integrates eye-tracking data from visual search tasks is a significant limitation to our understanding of how visual search behaviour holistically affects reading ability. This thesis presents the successful implementation of an LSTM-based neural network architecture that achieves superior accuracy in classifying visual search-based eye-tracking compared to a baseline model. However, the performance variations vary widely across different settings, highlighting the need for further investigation and refinement.



## Zusammenfassung

Zahlreiche Studien haben gezeigt, welche Einblicke Eye-Tracking-Daten in die Lesefähigkeiten einer Person geben können. Die Muster von Fixationen, Sakkaden und Regressionen liefern wertvolle Informationen über das Verständnis, den Lesefluss und die allgemeine Lesekompetenz. Auch visuelle Suchaufgaben, bei denen ein Gesichtsfeld nach einem Ziel durchsucht wird, sind in Bezug auf Aufmerksamkeitsprozesse und visuelle Kognition eingehend untersucht worden. Trotz dieser Fortschritte stellt das Fehlen von Forschungsarbeiten, die Eye-Tracking-Daten aus einer visuellen Suchaufgabe mit Lesefähigkeiten kombiniert untersuchen, eine kritische Einschränkung in unserem Verständnis darüber dar, wie das visuelle Suchverhalten zu den allgemeinen Lesefähigkeiten beiträgt. In dieser Arbeit wird die erfolgreiche Implementierung einer LSTM-basierten neuronalen Netzwerkarchitektur vorgestellt, die im Vergleich zu einem Vergleichsmodel eine höhere Genauigkeit bei der Klassifizierung von Blickverfolgungsdaten für die visuelle Suche aufweist. Allerdings variiert die Leistung in den verschiedenen Konfigurationen beträchltlich, was die Notwendigkeit weiterer Untersuchungen und Optimierungen unterstreicht.

# Acknowledgement

I want to thank Jan Brasser for supervising this thesis, having an open ear for all my struggles, and enabling an environment to pitch ideas and get valuable feedback. Thanks to Jens Vogler for helping with any coding problems I ran into and to Maurus Dora, Sarah Last, Eva Stehrenberger, and Niclas Bodemann for proofreading the different stages of my thesis.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| VS | Visual Search |
| LSTM | Long-Short-Term-Memory machine learning architecture |
| ET | Eye-Tracking |
| LRS | Lese-Rechtschreibschwäche, Reading- and Writing disorder |
| SRLT-II | Second edition of the 'Salzburger Lese- und Rechtschreibtests' |
| RT | Reaction Time |
| ADHD | Attention Deficit Hyperactivity Disorder |
| DD | Developmental Dyslexia |
| RNN | Recurrent Neural Network |
| CNN | Convolutional Neural Network |

# 1 Introduction

## 1.1 Motivation

Eye-tracking research has significantly contributed to our understanding of cognitive processes during reading tasks, shedding light on the relationship between eye movements and reading skills. Similarly, investigations into visual search tasks have shed light on the mechanisms involved in searching for specific visual stimuli. However, a noticeable gap exists in the current literature concerning the exploration of eye-tracking data specifically within the context of visual search tasks. While extensive research has delved into eye movements during reading and visual search tasks separately, the intersection of these two domains remains largely unexplored.

Understanding how eye-tracking patterns in a visual search task correlate with reading skills has practical implications. This knowledge can inform the development of tailored interventions for individuals with specific reading difficulties, contributing to the fields of education and cognitive rehabilitation. Furthermore, insights from this research may have implications for the design of educational materials, user interfaces, and accessibility features, enhancing the overall learning experience for individuals with diverse reading abilities.

## 1.2 Research Questions

To explore the correlation between gazing patterns during Visual Search tasks and the reading proficiencies of children, diverse neural network architectures are employed in this thesis. The primary research inquiries guiding this investigation are: Can a neural network architecture based on Long Short-Term Memory (LSTM) effectively predict reading fluency scores measured by the 'Weiterentwicklung des Salzburger Lese-und Rechtschreibtest' (SRTL-II) based on eye-tracking data from visual search tasks? What is the structure of such an architecture, and how accurately can it perform?

To address these questions, I:

- constructed a dataset
- preprocessed the data
- designed different neural network configurations
- implemented the architectures
- implemented hyperparameter tuning, training, and validation procedures
- compared the performance of the different architectures
- and evaluated the best performing models

The dataset comprises data obtained from the ongoing study 'Lesen im Blick' conducted by the Digital Linguistics group at the Department of Computational Linguistics at the University of Zurich as of the date of thesis submission. It encompasses eye-tracking data from a Visual Search task and SRTL-II scores from first and second-grade school children. Various machine learning architectures are constructed, trained, and tested using this dataset. This process involves not only the creation of the architectures but also fine-tuning their hyperparameters, training, and subsequent performance evaluation.

## 1.3 Thesis Structure

This opening chapter delineates the motivation, research questions, and overall thesis structure. Chapter 2 offers a comprehensive survey of existing research, elucidating the role of eye-tracking and the interrelation between visual search tasks and skills associated with reading. Chapter 3 focuses on the data used in this thesis. It includes a section about its collection, its characteristics, and any preprocessing steps applied to ensure its suitability for analysis. In Chapter 4 the problem setting is formalized, providing a foundation for the subsequent sections that detail the methodological approach and results. The methodology chapter 5 is dedicated to explaining the neural model architectures chosen. It details how the architecture is designed to capture the sequential nature of eye-tracking data during visual search tasks. Additionally, the evaluation procedure is outlined, encompassing the metrics used, training methods, and validation strategies employed to assess the model's performance. Chapter 6 presents the results. The best-performing model configurations are showcased and a comprehensive evaluation of the results is presented. The thesis culminates in the conclusion in Chapter 7, where the key contributions of

this research are summarized. The initial research questions are revisited and it is reflected on the insights gained throughout the thesis. Additionally, future research is suggested, and problems are analyzed.

# 2 Related Work

Predicting reading scores using eye-tracking patterns in the context of a visual search task combines insights from eye-tracking research, cognitive psychology, and machine learning techniques. This chapter tries to acknowledge this wide field of research in an appropriate scope for the thesis. Section 2.1 gives a broad overview of the history of eye-tracking research with a deeper insight into the use of eye-tracking data based on reading tasks and machine learning in subsection 2.1.1. Section 2.2 looks at reading-related skills and in particular visual search as a potential predictor of reading abilities.

## 2.1 Eye-Tracking

Eye-tracking technology has applications in various fields, including cognitive science and education. It is used to study cognitive processes, and eye-movement disorders, and inform educational practices, such as understanding reading difficulties and developing reading interventions.

In the mid-20th century, researchers like Yarbus explored eye movements during reading and found that they provided insights into cognitive processes. This laid the groundwork for understanding reading behavior through eye-tracking. In reading research, studies by Rayner; Just and Carpenter and others advanced our understanding of eye movements during reading. Their work profoundly shaped and influenced the field, by establishing the concepts of fixations and saccades, which are still used today. Their focus on the relationship between eye movements and comprehension introduced eye-tracking as a valuable methodology for researching psycho-linguistic questions. Since then Eye-tracking technology has found applications in various fields, including cognitive science and education. My thesis centers around reading related eye-tracking data, therefore the following passages focus on literature relevant to this:

Hutzler and Wimmer [2004] for example showed more fixations, longer first fixations, and gaze duration for short and long German words and more fixations and longer

gaze duration in pseudowords for German dyslexic readers.

### 2.1.1 Eye-Tracking and Prediction of Reading related Abilities

Recent developments involve the use of eye-tracking data for predictive modeling. Researchers have explored the potential of machine learning techniques, including LSTMs and other deep learning approaches, to predict cognitive and reading-related outcomes. They can broadly be categorized into three types:

- predicting general reading abilities [Strandberg et al., 2022]
- predicting text comprehension or reader confusion [Reich et al., 2022; Sims and Conati, 2020]
- predicting neurodevelopmental disorders like ADHD and Dyslexia [Haller et al., 2022; Deng et al., 2023; Szalma and Weiss, 2020; Benfatto Nilsson et al., 2016]

All the referenced studies presented encouraging outcomes. Nevertheless, it is noteworthy that these studies predominantly rely on eye-tracking data acquired during participants' reading activities. The objective of the 'Lesen im Blick' study, within which our eye-tracking data was procured, is to forecast prospective reading challenges in children before they attain reading proficiency. This aim is geared towards facilitating interventions at an earlier stage than conventional standardized diagnostic and testing procedures currently permit.

In light of this overarching goal, the design of the eye-tracking experiment within the 'Lesen im Blick' study necessitates careful consideration. Tasks incorporated into the experiment are tailored to measure prereading skills or skills closely linked to reading. This deliberate selection is made to align with the overarching objective of predicting reading difficulties at an early stage, emphasizing the importance of assessing foundational skills that precede the formal acquisition of reading abilities.

## 2.2 Visual Search and Reading related Skills

Frequently investigated prereading skills encompass a range of cognitive abilities such as rapid automated naming, phonological awareness, working memory, speech production, visual attention, and efficient visual search. For an overview of the influence of sensorimotor and cognitive abilities on reading abilities, see for example: Carroll et al. [2016].

The eye-tracking data utilized in this thesis comes from a visual search task. Existing research underscores that children encountering reading difficulties often manifest distinctive outcomes when engaging in visual search tasks in comparison to their peers. For instance, Ferretti et al. [2008] observed that dyslexic children exhibited prolonged reaction times (RT) in visual search tasks where the target appeared on the right side, distinguishing them from their control group. Moreover, while the control group demonstrated a linear increase in RTs corresponding to a progressively rightward placement of the target, dyslexic children did not exhibit such distinctions. Notably, the study found that the processing speed in a visual search task during kindergarten years served as a reliable predictor of early literacy acquisition. Another study by Cui et al. [2020] reported that Chinese children with developmental dyslexia (DD) and/or attention deficit hyperactivity disorder (ADHD) displayed significantly lower accuracies in a visual search task involving five Chinese characters compared to their counterparts. Children with DD and ADHD also demonstrated notably extended gaze duration and viewing times, along with an increased number of fixations.

### 2.2.1 Visual Search, Machine Learning and Reading Skills

A lot of research has been done regarding the relationship between eye-tracking data based on reading tasks and reading skills, and the relationship between visual search tasks and reading skills. However, no research regarding eye-tracking data of a visual search (VS) task could be found. By bridging the gap between these two domains, this research aims to provide a comprehensive understanding of the intricate interplay between eye movements, VS, and reading skills.

# 3 Data

## 3.1 Data Collection

The data was collected as part of a larger study ('Lesen im Blick'), which was ongoing at the time of submission. The project aims to investigate the connection between eye movements and Dyslexia.

### 3.1.1 Eye-Tracking Visual Search Task

The VS task was the first of the three sub-tasks of the eye-tracking experiment.

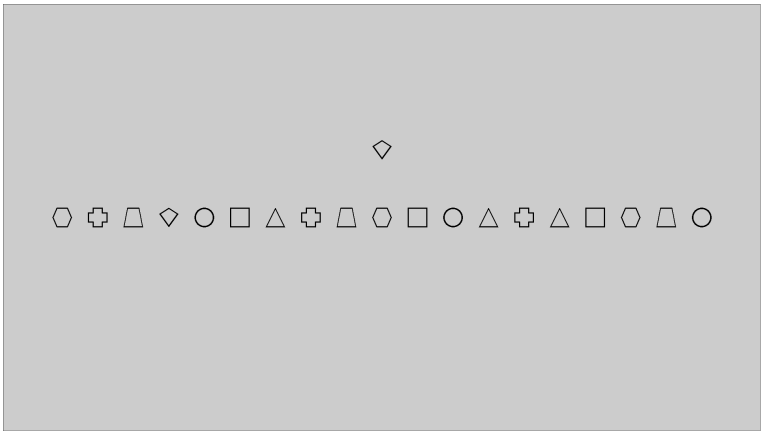
The EyeLink Portable Duo eye tracker was used with the technical specifications found here: SR Research Ltd. [2017] and the study-specific settings:

- Remote, head free to move
- Monocular, stronger eye
- 25 mm lens
- Eye-tracking principle: Pupil with Corneal Reflection
- Pupil Detection Model: Ellipse Fitting
- Calibration: 13 points
- Sampling rate: 1000Hz

The experiments were conducted in rooms designated by the school, and prepared by the experiment conductors. Preparing included: eliminating possible distractors, turning on the light, setting up the eye-tracker and laptop, etc. The camera was positioned on top of the laptop's keyboard, on which the tasks were displayed. The participants were seated in front of the laptop and instructed to move their heads as little as possible. They were approximately placed 60 cm away from the screen and instructed to use a button box, having four buttons in the colors yellow, blue red,

and green. The experiment started with two test trials, where the participants were instructed and corrected by the computer's audio. Theoretically, the test consisted of 2 example trials and 42 real trials. However, the VS task was terminated after 20 minutes regardless of done trials, to prevent exhaustion, resulting in different number of trials for each participant. The VS task has three modes: geometrical shapes, letters, and mirrored letters. Each mode is composed of seven different figures. They are randomly distributed over an array with 19 positions, in which the target figure appears one to three times. Figure 1 depicts how such an exemplary screen could look like.

Figure 1: Visual Search Example Screen



Firstly, the target symbol, also called the stimulus, is displayed. In the example screen, the target symbol is the diamond. The participant is instructed to fixate on it to start the trial. The fixation triggers the appearance of the search array. The participant terminates the trial by pressing a button, the search array disappears only leaving the stimulus on the screen. They then are asked by the computer's audio how many targets they have found. The experiment conductor records the answer. If the answers are in the range of possible answers (one to three) they are recorded accordingly, otherwise, 4 is logged, referencing a wrong answer.

The VS task implemented is inspired and constructed similarly to the VS task introduced in Ferretti et al. [2008]. However, in the cited study the collected data is not eye-tracking based, but reaction times are measured.

### 3.1.2 SRTL-II

The SLRT-II is designed to capture and rate its participants' reading and writing competence. Our participants have only partaken in the reading exercise of

the SLRT-II. The SLRT-II reading exercise is a standardized test to measure and test the ability of synthetic reading and direct word recognition in German and Swiss-German-speaking individuals. Synthetic reading describes the ability to join depicted sounds to form an articulatory unit. It enables one to read independently as it allows one to process and understand new and unknown words. Direct word recognition means the ability to activate a memorized representation of a known written word based on its visual appearance without needing to decipher its sound- or letter-wise. This ability enables efficient reading. The participant read out words and pseudo-words as quickly and accurately as possible. The SLRT-II time limit is set for one minute for each the real word part and the pseudo word part. The pseudo words test for synthetic reading in more proficient readers, who might not need synthetic reading in the real-word setting as they know all non-pseudo words already. The depicted pseudo-words neither have a known letter sequence nor does their articulation correspond to an existing German word. However, their letter sequence and their phonetic structure adhere to German phonological and writing structures Kristina Moll and Karin Landerl [2017, 27-28].

#### 3.1.2.1 Procedure

The SLRT-II was conducted as part of further psychometric tests in the first of two sessions. The experiment was conducted as follows: The experimenter provided the participant with instructions, after which the participant vocalized the exercise words while the experimenter provided additional guidance. The participant then vocalized the test words aloud. The same procedure was followed for the pseudo-words.

For the exercise words they were presented with eight exercise items in two columns on the back of the respective reading sheet as exercises before the paper was turned, the timer was started and they read the test words.

In Figure 2 the initial instruction read to the participants is depicted, and the instructions between exercise and testing are in Figure 3. In short, the instructions include an explanation of the order to read words column-wise. The conductor recorded all wrongly read or skipped words on the protocol sheet. After one minute, the conductor concluded the reading and moved on to either the pseudo-word task or another psychometric test following the completion of the pseudo-word task.

Figure 2: SRLT- II instructions part one

**Instruktion Leseblätter Übungsseite:**

**Übungsseite Wörter für Kinder und Erwachsene:**

*»Du siehst hier Spalten mit einzelnen Wörtern. Lies diese Wörter der Reihe nach von oben nach unten laut vor. Lies, so schnell du kannst, aber ohne Fehler zu machen. Wir üben das jetzt mit diesen Wörtern.«*[3]

**Übungsseite Pseudowörter für Kinder:**

Das Pseudowortlesen wird im gleichen Wortlaut anhand der Übungspseudowörter (Rückseite des Leseblattes Pseudowörter) instruiert und die Pseudowörter folgendermaßen eingeführt:

*»Du siehst hier Spalten mit einzelnen Fantasiewörtern, die sich jemand ausgedacht hat. Diese Wörter gibt es nicht, aber man kann sie trotzdem lesen. Lies diese Fantasiewörter der Reihe nach von oben nach unten laut vor. Lies, so schnell du kannst, aber ohne Fehler zu machen. Wir üben das jetzt mit diesen Fantasiewörtern.«*

Figure 3: SRLT- II instructions part two

**Instruktion Leseblätter Testseite:**

**Leseblatt Wörter für Kinder:**

*»Wenn ich das Blatt umdrehe, siehst du wieder Spalten mit Wörtern. Lies diese Wörter der Reihe nach von oben nach unten laut vor. Lies, so schnell du kannst, aber möglichst ohne Fehler zu machen – du musst nicht das ganze Blatt lesen, sondern nur so lange, bis ich ›stopp‹ sage.«*

#### 3.1.2.2 Scores

Based on the protocol sheets the scores were computed by the number of mistakes multiplied by 100, divided by the number of read words. To be able to compare the reading scores of first and second-graders, the percentage rank of their performance in the SRTL reading task was taken. The percentage rank is normed for the grade and the semester the participants were in when partaking. The confidence level is 0.05. For 82 first graders, in the second semester, the confidence interval is $\pm$ 4.98 with a mean of 15.28 and a standard deviation of 11.54 rightly read words. For 154 second graders, in the second semester, the confidence interval is $\pm$ 7.63 with a mean of 46.43 and a standard deviation of 17.70 correctly read words Kristina Moll and Karin Landerl [2017, 75-77].

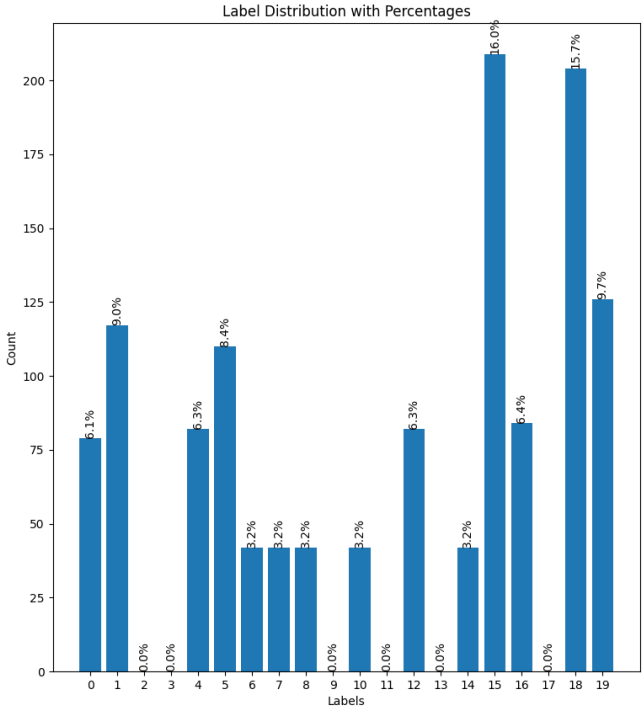
## 3.2 Dataset

The dataset used comprises 33 children in first and second grade. It contains 1302 VS eye-tracking trials with their corresponding SRTL-II score as Labels.

The eye-tracking data is from a VS Task, where the children had to count the occurrence of a displayed target symbol in a line of 19 symbols as fast and as precise as possible. The target symbol occurred one to three times in the sequence.

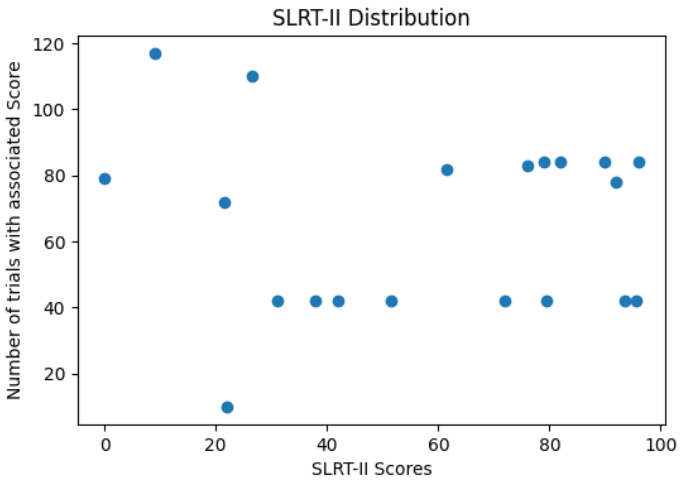
The label assigned to the trials is the participant's achieved SLRT-II percentage rank score normed for their age group. For the classification task, the label is one of 20 classes, where each class represents the span of 5 percentage ranks. Figure 4 shows the occurrence distribution of the labels. The y-axis is the absolute occurrence of trials with the corresponding label on the x-axis. Note that the dataset contains no data with labels 2, 3, 9, 11, 13, and 17 since none of the participants had a corresponding SLRT-II score. It also shows that classes 15 and 18 are overly represented. For the regression task, the scores were taken directly as labels without any further

Figure 4: Class Distribution in the Dataset




transformation. The distribution of the SLRT-II scores are displayed in Figure 5

Figure 5: SLRT-II Score Percentage Rank Distribution in the Dataset




### 3.2.1 Preprocessing

This Section describes what the eye-tracking data used looks like and how it is preprocessed to fit the model's architecture and input requirements. Furthermore, the transformation of the SLRT-II score to a label format is explained.

#### 3.2.1.1 Raw Eye-Tracking Data

With the Dataviewer proprietary software [SR Research Ltd., 2023] only the data for the interest period of the VS task were extracted, meaning the eye-tracking recording for the time between the appearance of the search array and its disappearance. To trigger the appearance of the search array the participant had to fixate on the stimulus symbol (The diamond in the example screen of Figure 1 and the square in Figure 6), and its disappearance by pushing a button, indicating that they found all symbols.

Table 1 lists all gaze measurements used. The selection of relevant gaze measurement was mainly based on findings of other studies, discussed in section 2.2, and on educated guessing since there is no comparable study documented. It includes four different types of measurements: numerical values regarding the current fixation, categorical values regarding the current fixation, and numerical and categorical values regarding the whole trial.

Durations are measured in milliseconds. The current fixation Area position is de-

| Numerical, fixation dependent |
|---|
| Current fixation x-axis coordinate |
| Current fixation y-axis coordinate |
| Current fixation pupilsize |
| Duration of fixation |
| Duration of outgoing saccade |
| Amplitude of outgoing saccade |
| Angle of outgoing saccade in relation to the current fixation |
| Velocity of outgoing saccade |
| Current fixation area position |
| Next fixation area position |

| Categorical, fixation dependent |
|---|
| Current fixation area label |
| Direction of outgoing saccade in relation to the current fixation |
| Target status |

| Numerical, whole trial dependent |
|---|
| Number of total fixations |
| Reaction trial time |
| Number of targets |

| Categorical, whole trial dependent |
|---|
| Experiment condition |
| Screen configuration |

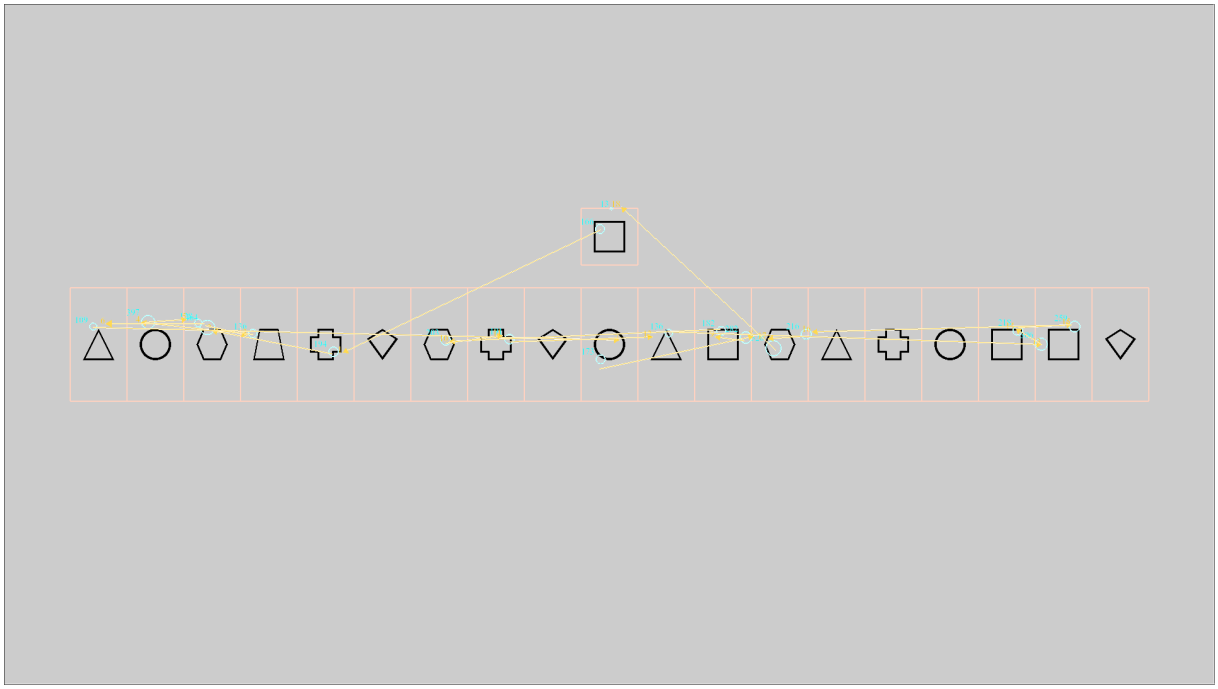
Table 1: Gaze Measurements

noted by the ID of the position on the array, whereas the current area label encodes the symbol gazed at.

Target status encodes, if the current fixation is on a target symbol, on the stimulus (the exemplary symbol the participant needs to search for) or if it is neither on a target nor on the stimulus.

In Figure 6 some measurements are visualised. The orange boxes mark the interest areas, each with a corresponding position ID and a label. The position ID encodes where on the screen the interest area is (target or position 1 to 19) and the label which symbol it displays. The blue circles denote fixations, where the size of the

Figure 6: VS Screen with Eye-Tracking Measurements



circle indicates their duration, and the yellow lines encode saccades, with start and end points as well as arrows visualising their directions.

#### 3.2.1.2 Dataset to Input Tensor

To transform the eye-tracking data into a suitable format for machine learning the following steps were taken: Normalization was applied to numeric features across the entire dataset (see Table 1 for the exhaustive list of features). The normalisation was performed by the standard scaler of the sklearn package (see Pedregosa et al. [2011] for details). This step standardized the numeric data to a common scale, enabling more consistent and meaningful comparisons across these variables.
In addition to normalization, categorical features were one-hot encoded to convert them into a binary format suitable for machine learning algorithms. For example, the feature 'direction of next saccade' can have the values: up, down, left, right, or NaN. The corresponding one-hot encoding for the value 'left' would therefore look like: [...0,0,1,0,0, ...] Where [ ] denotes the whole tensors and ... denotes other entries not related to the feature 'direction of next saccade'. The columns that underwent one-hot encoding include the direction of the next saccade, the label of the current fixation area label, the target status (stimulus, target, non-target, off-search

array), and the experimental condition (letter, mirrored letter, geometrical) as well as the trial screen configuration. These transformations facilitated the integration of categorical information and screen-specific details into the analysis, allowing for a comprehensive exploration of the eye-tracking data in the context of the research objectives. None values were filled with zeros.

#### 3.2.1.3 SLRT-II Score to Label

The SLRT-II scores referred to in this thesis are percentage ranks and are in a format unsuitable for machine learning. They include different ranges, greater than and smaller than symbols, and their ranges are not uniform. To generate a suitable format a custom label transformer function was employed. It takes a score as input and returns a float. First, it checks if the score contains a 'greater than' (>) or 'smaller than' (<) symbol. If this symbol is present, the score is returned without it, as all those scores represent either the lowest or the highest possible label score. If the score contains a hyphen (-), signifying a range, it is split into two values, and the average of these values is computed and returned as a float.

Following these initial transformations, an additional step is taken to convert the transformed score into a one-hot encoded format. The score is firstly divided by 5, and the result is used to determine the appropriate position in a 20-element tensor filled with zeros. The position corresponding to the label's value is set to 1, effectively one-hot encoding the label. This final one-hot encoded tensor is then used as the true label for training the model.

These transformation steps are split to enable the possibility of changing the label implementation. It would be possible to change up the classification task for example to a binary classification, or to change up the number of classes to predict.

For this thesis, in the first setting a number of 20 classes were implemented to still approach the large range of scores but also take into consideration, that the scores between the age norm tables do not have a one-to-one correspondence. In a second setting where the prediction is seen as a regression problem, the labels were kept unchanged after the initial transformation to directly predict the score.

# 4 Problem Setting

The task of predicting the SRTL-II score based on an eye gaze sequence on one visual search task can be formalized in different ways; I chose to try two different approaches. In the first approach, the problem is viewed as a classification task, where the model tries to predict one of 20 classes associated with the reading score. The second approach interprets the problem as a regression task, where the model tries to predict the reading score directly.

Additionally, I implemented two variations of the base model, implementing three different versions in total. In the first version, the model was implemented in its base form as detailed in section 4.1.1, providing the model with no screen information at all. In the second version detailed in 4.1.1.2, the screen configuration is added to each input tensor of the fixation sequence, giving it the most influence on the model's learning behavior. In the third version explained in 4.1.1.3, the model has an additional attention layer put ahead to process the screen information of the visual search task separately, which is then fed into the LSTM.

## 4.1 Formalisation

The following sections detail the formalisation of the different variations.

The goal is either to train a multi-class classifier $g_\theta$, where $\hat{y}_i$ is the predicted class label, or interpret the prediction as a regression problem, where $\hat{y}_i$ is the predicted target label $y_i$ with a value between 0 - 100 for participant $i$.

### 4.1.1 Model

For all my models the LSTM layer of the PyTorch library was used; having the following mathematical formalisation based on the Sak et al. [2014, 2]:
It "...computes a mapping from an input sequence -$x = (x_1, \ldots, x_T)$ to an output sequence $y = (y_1, \ldots, y_T)$ by calculating the network unit activations using the

following equations iteratively from $t = 1$ to $T$:

$$\begin{aligned} i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \\ f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \\ o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \\ m_t &= o_t \odot h(c_t) \\ y_t &= W_{ym}m_t + b_y \end{aligned} \tag{4.1}$$

Where the $W$ terms denote weight matrices (e.g., $W_{ix}$ is the matrix of weights from the input gate to the input), the $b$ terms denote bias tensors ($b_i$ is the input gate bias tensor), $\sigma$ is the logistic sigmoid function, and $i$, $f$, $o$, and $c$ are respectively the input gate, forget gate, output gate, and cell activation tensors, all of which are the same size as the cell output activation tensor $m$. $\odot$ represents the element-wise product of the tensors and $g$ and $h$ are the cell input and cell output activation functions, generally *tanh*."

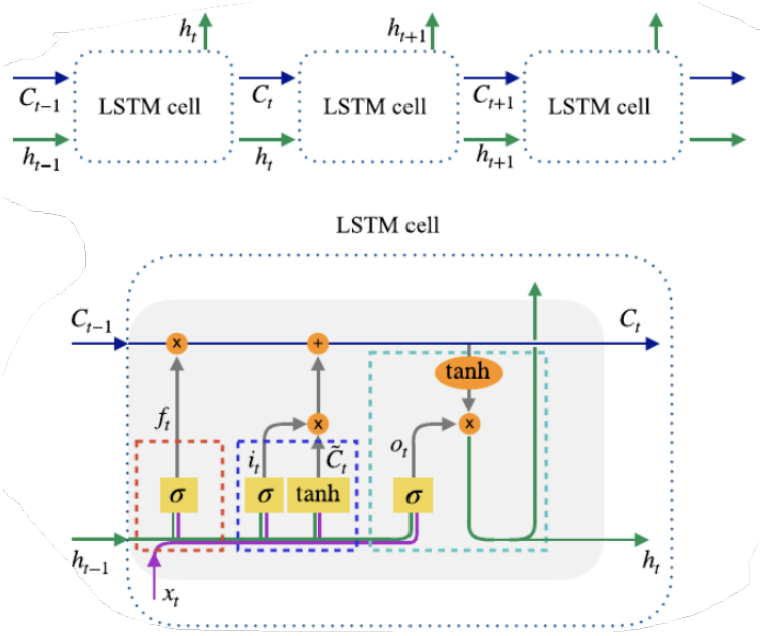
To improve the performance of the architecture the LSTM was implemented bidirectionally.

Figure 7, taken from Siciliano et al. [2021, 5] is a visual representation of the structure of such LSTM cells. On top multiple connected LSTM Cells are depicted, below the 'inner' working of one such cell. The blue line represents the cell state $C_t$, the green line the cell output $h_t$, cell input $x_t$ is represented by the purple line, the forget gate by the dashed red rectangle, the input gate by the dashed blue rectangle, the output gate by dashed green rectangle, the activation functions by yellow rectangles and other operators by orange circles. The output gates are the gray lines.

#### 4.1.1.1 Base LSTM Model

The base form of the proposed models is a bidirectional LSTM, which takes as input a tensor $r_{ij}$ of visual search screen $j$, gazed at by participant $i$ and predicts label $y_i$, where the tensors $r_{i1}, \ldots, r_{iT}$ represent the input sequence tensor $x$ in the formalisation of the LSTM. The last tensor $y_{iT}$ of the output sequence $y = (y_{i1}, \ldots, y_{iT})$ is connected to a linear layer with the output size of the label tensor (20 for the implementation of the proposed classifier model, or one for the proposed regression model). The resulting tensor either encodes the model's predicted probabilities for each label class or the predicted score. In the case, where no screen information is passed to the model, the formalisation is used as explained in this subsection.

Figure 7: LSTM cells




#### 4.1.1.2 Base LSTM Model with Screen Information

To include screen information, each input tensor $r_{ij}$ was expanded by the one hot encoded screen configuration. The input tensor is then fed into one or multiple connected LSTM layers, which are connected to a linear layer, projecting it to either the classification size of 20 or to one resulting value for the regression model.

#### 4.1.1.3 LSTM Model with Attention Layer

The attention layer is adjusted from the attention mechanism described in Vaswani et al. [2017]. This model takes an additional input tensor $u_j$, where the symbol sequences of the screen are one-hot-encoded. This tensor is then split into 20 sub-tensors (one tensor for each of the nineteen positions plus one target position). This tensor sequence of 20 tensors is then fed to an attention layer. Its attention mechanism is defined using linear transformations. Three linear transformations are employed to map the input data to query, key, and value tensors: The key tensor is used to compare against the query tensor, and the value tensor holds the actual information one wants to retrieve based on the query's attention. A score tensor is calculated by performing a dot product between the query and key tensors. It is then scaled to control the magnitude of the scores. The resulting scores reflect

the similarity or relevance between the query and key tensors for each element in the input data. The softmax function is then applied to the scores to obtain the attention weights. The softmax operation normalizes the scores to produce values between 0 and 1, where higher values indicate higher attention or importance. The context tensor for the current batch is computed by taking a weighted sum of the value tensors, where the attention weights determine the weights for each element in the value tensor. This means that more attention is given to more relevant elements based on the query-key similarity. This resulting context tensor is then flattened into a 1D tensor and adjusted to the right dimension to represent the batch size, to conform to the subsequent layers. This tensor is then used as the initialisation of the $h_0$ and $c_0$ tensors, which are normally comprised of zeros.

### 4.1.2 Data

To investigate the task of predicting the SRTL-II score based on an eye gaze sequence on one visual search task screen the used data can be formalized as a set $D = \{(W_{11}, y_1), \ldots, (W_{N,M}, y_N)\}$, where $W_{ij} = \langle w_{ij1}, \ldots, w_{ijK} \rangle$ is a sequence of reading measure tensors for each fixation $k \in 1, \ldots, K_j$ obtained from subject $i$ looking at VS screen $j$, where $N$ is the number of participants, $M$ is the number of VS screens processed by each of the participants, and $K_j$, is the number of fixations registered while looking at screen $j$. Each reading measure tensor consists of $R$ gaze measures, i.e., $w_{ijk} = (r_{ijk1}, \ldots, r_{ijkR})$ and possibly an encoded representation of screen $j$. See Table 1 for the exhaustive list of gaze measurements used.
The target label $y_i$ has two different formalisations: When interpreting the prediction of the SRTL-II score as a classification task ($a$), the categorical target label $y_i$ represents one of 20 classes for participant $i$. The goal is then to train a multi-class classifier $g_\theta$, where $\hat{y}_i$ is the predicted class label. In this case, $\hat{y}_i$ can take one of the 20 possible values, corresponding to the predicted class, based on the model's output. Interpreting the prediction as a regression problem ($b$), the true target label $y_i$ is a value between 0 - 100 for participant $i$. The goal is then to train a model $g_\theta$, where $\hat{y}_i$ is the predicted value. In this case, the predicted target $\hat{y}_i$ can have any rational value theoretically, but the model should learn to assign values between 0-100.

# 5 Method

In this chapter, the choices of model architectures with their input configurations are explained, as well as how their hyperparameters were tuned and what the evaluation procedure looks like[1].

## 5.1 Model Architecture

The use of the Long Short-Term Memory (LSTM) architecture was driven by the idea of encapsulating the sequential nature in the VS task. The engagement in a visual search task necessitates a consecutive and evolving interaction with an array of symbolic information displayed on a screen. This interaction is comprised of a succession of distinct ocular events, including fixations, saccades, and blinks. LSTMs, as a specialized recurrent neural network (RNN) variant, are adept at modeling and predicting sequences, rendering them well-suited for this task. Their key distinguishing feature lies in their capacity to retain and process temporal dependencies and long-range interactions within sequential data [Sak et al., 2014], allowing for the effective modeling of the intricate dynamics associated with eye-tracking patterns [Haller et al., 2022] in the context of visual search. This architectural choice facilitates the preservation of context and the extraction of meaningful information from the ordered sequences of ocular events, ultimately enhancing the model's capacity to predict the ensuing reading score based on these patterns[2].

To acknowledge different ideas about how the screen configuration influences the eye-tracking pattern of the participant and the model predictive power based on it, I came up with three different variations of the base model and its input:

1. not processing the screen information at all, based on the idea that the model can learn from patterns in the data independent of the screen configuration

[1] The whole code implementing the methodology was written in Python and can be found on Github.

[2] An other popular neural network architecture choice regarding processing sequential data are Convolutional Neural Networks for example used in Deng et al. [2023].

2. adding the screen information to each input, following the observable differences in eye movements depending on the number of targets, the experiment condition, how similar neighboring symbols are, etc.

3. combining the two approaches by providing the model with some information about the screen but only once in the course of a fixation sequence, mimicking the idea of gathering information about the screen as a whole first before processing the individual symbols, sequentially.

For the first setting, the model was implemented in its base form as detailed in section 4.1.1.1. In the second setting, the screen configuration is added to each input of the fixation sequence, giving it the most influence on the model's learning behavior (see section 4.1.1.2. In the third setting, the model has an additional attention layer put ahead to process the screen information of the visual search task separately, which is then fed into the LSTM part as initial values for the hidden states (section 4.1.1.3).

Based on their performance in the tuning process detailed in section 5.1.1, only the best-performing model was evaluated.

### 5.1.1 Hyperparameter Tuning

The Ray framework, as detailed by Liaw et al. [2018], was employed for hyperparameter tuning. This framework utilizes a random combination approach, assembling hyperparameter settings from the respective values[3] outlined in Table 2. The model undergoes training for a predetermined number of epochs or until no further improvement is discernible. The tuning process employs the 'Asynchronous Hyper Band scheduler', a scheduling algorithm introduced by Li et al. [2018]. It is an algorithm specifically designed for hyperparameter tuning of neural networks. It involves the concurrent execution of multiple configurations of hyperparameters, optimizing the efficiency of the tuning process. It employs an asynchronous early-stopping approach based on its successive halving algorithm. Its idea is to allocate resources dynamically, adapting to the performance of different configurations as they progress through the tuning procedure. By asynchronously managing the allocation of resources and exploiting parallelisation, it significantly enhances efficacy and expedites the convergence of hyperparameter optimization compared to traditional methods.

Figure 8 shows how such tuning results look like.

[3]Embedding Size is only used in models with attention layer

| Hyperparameter | Possible values |
|---|---|
| Batchsize | 2, 4, 8, 16, 32 |
| Learning rate | 0.001 - 0.1 |
| Hidden layersize | 64, 128, 254, 512, 1024 |
| Number of stacked LSTM cells | 1, 2, 4, 8 |
| Embbedding size | 32, 64, 128, 254 |

Table 2: Possible Hyperparameters for Tuning

In the training process the optimizer Adagrad [Duchi et al., 2011] was used. As loss function of the classifier, the Cross-Entropy-Loss and for the Regression task Huber-Loss [Zhu et al., 2008] from Pytorch were used.

## 5.2 Evaluation Procedure

For the evaluation phase, the models configured with the most effective hyperparameter settings, as determined through the tuning process, were utilized. The evaluation employed a **10-fold random cross-validation** and a **10-fold group cross-validation** implemented by Scikit Pedregosa et al. [2011]. The dataset was partitioned into ten subsets, and the model underwent training and testing iteratively ten times. During each iteration, a different subset was designated as the test set, while the remaining subsets served as training sets. Training phase had 100 epochs.
In the **random** validation, the test and training set were split randomly by trial indices. In the **group** validation, groupings were established based on participant IDs, ensuring that the test set comprised only participants unseen by the model until that point.
Subsequently, the obtained results were compared against baseline results. A T-Test, implemented from SciPy Virtanen et al. [2020], was conducted on the results to ascertain their statistical significance. The null hypothesis is, that the baseline results have a higher mean in regards to their underlying distribution, as my model results, meaning the baseline model performs equally or better than my models regarding their accuracy in case of the classifier and the mean square error in case of the regression model. A significance level of 0.05 is chosen.

Figure 8: Hyperparameter Tuning Results of LSTM with Screen Information

| hidden_dim | embedding_dim | lr | batch_size | num_layers | iter | total time (s) | loss | accuracy |
|---|---|---|---|---|---|---|---|---|
| 128 | 16 | 0.00246119 | 16 | 1 | 1 | 166.538 | 2.95384 | 0.160399 |
| 64 | 32 | 0.0190787 | 32 | 4 | 1 | 168.802 | 2.92393 | 0.156562 |
| 64 | 32 | 0.0309254 | 16 | 1 | 10 | 1817.22 | 2.48243 | 0.213354 |
| 512 | 32 | 0.0193707 | 16 | 4 | 2 | 566.972 | 2.84521 | 0.158097 |
| 512 | 16 | 0.0930351 | 4 | 2 | 1 | 234.663 | 5.51792 | 0.161167 |
| 64 | 32 | 0.0289411 | 16 | 4 | 2 | 344.426 | 2.84924 | 0.158097 |
| 64 | 64 | 0.00128455 | 16 | 1 | 1 | 164.73 | 2.96626 | 0.105142 |
| 512 | 32 | 0.00113621 | 4 | 2 | 1 | 242.879 | 3.20234 | 0.161167 |
| 64 | 256 | 0.00346269 | 16 | 4 | 1 | 185.632 | 2.95349 | 0.160399 |
| 1024 | 256 | 0.00343532 | 8 | 4 | 1 | 794.456 | 3.01684 | 0.161167 |
| 128 | 128 | 0.0268366 | 2 | 1 | 1 | 205.088 | 3.79528 | 0.161934 |
| 128 | 256 | 0.00246625 | 16 | 4 | 1 | 216.222 | 2.95549 | 0.095165 |
| 64 | 16 | 0.0158653 | 2 | 4 | 1 | 252.298 | 3.70047 | 0.159632 |
| 512 | 128 | 0.00178169 | 2 | 1 | 1 | 272.319 | 3.54566 | 0.0974674 |
| 1024 | 256 | 0.0315946 | 4 | 4 | 1 | 997.756 | 4.41502 | 0.161167 |
| 128 | 256 | 0.0714226 | 32 | 2 | 4 | 792.372 | 2.63022 | 0.163469 |
| 256 | 32 | 0.0547053 | 16 | 4 | 4 | 917.658 | 2.65596 | 0.158097 |
| 128 | 128 | 0.0692248 | 8 | 1 | 10 | 1758.9 | 2.20403 | 0.270146 |
| 512 | 128 | 0.0462282 | 32 | 4 | 2 | 566.981 | 2.80451 | 0.160399 |
| 1024 | 256 | 0.00182948 | 16 | 1 | 1 | 248.361 | 2.98748 | 0.158097 |
| 256 | 32 | 0.00604568 | 8 | 1 | 2 | 350.272 | 2.91706 | 0.161167 |
| 128 | 64 | 0.00141792 | 4 | 4 | 2 | 414.404 | 2.93331 | 0.111282 |
| 64 | 16 | 0.00576596 | 8 | 1 | 2 | 351.35 | 2.83785 | 0.176516 |
| 128 | 16 | 0.00174367 | 32 | 4 | 1 | 177.586 | 2.96893 | 0.0943975 |
| 64 | 128 | 0.0309684 | 8 | 4 | 1 | 189.777 | 3.00551 | 0.161167 |
| 512 | 256 | 0.0366868 | 2 | 2 | 1 | 303.726 | 4.34585 | 0.159632 |
| 512 | 256 | 0.00102239 | 4 | 2 | 1 | 277.321 | 3.16506 | 0.161167 |
| 128 | 16 | 0.0231219 | 8 | 2 | 1 | 210.67 | 2.98444 | 0.161167 |
| 1024 | 256 | 0.0109268 | 8 | 1 | 1 | 284.284 | 3.05738 | 0.161167 |
| 1024 | 32 | 0.00471505 | 4 | 1 | 1 | 301.301 | 3.47943 | 0.161167 |

### 5.2.1 Reference Method

To facilitate the interpretation of the models' performances, baseline models are introduced to serve as benchmarks for evaluating the efficacy of the more intricate models and assessing their performance relative to a fundamental predictive strategy. In the *classification task*, a baseline model was implemented, which consistently predicted the majority class of the current cross-validation folds training set. For the *regression task*, a baseline was established by utilizing the median of the true scores within the current cross-validation folds training set. Those straightforward approaches establish a baseline against which the performance of more complex models can be measured and help understand their effectiveness in comparison to a basic predictive strategy.

# 6 Results and Discussion

## 6.1 Results

This section presents a detailed examination of the outcomes derived from the implemented models. It encompasses an analysis of classification and regression results, providing insights into the models' predictive capabilities concerning reading scores. The discussion includes an exploration of the impact of different configurations, the differences observed in cross-validation settings, and an evaluation of the models' efficacy compared to baseline strategies.

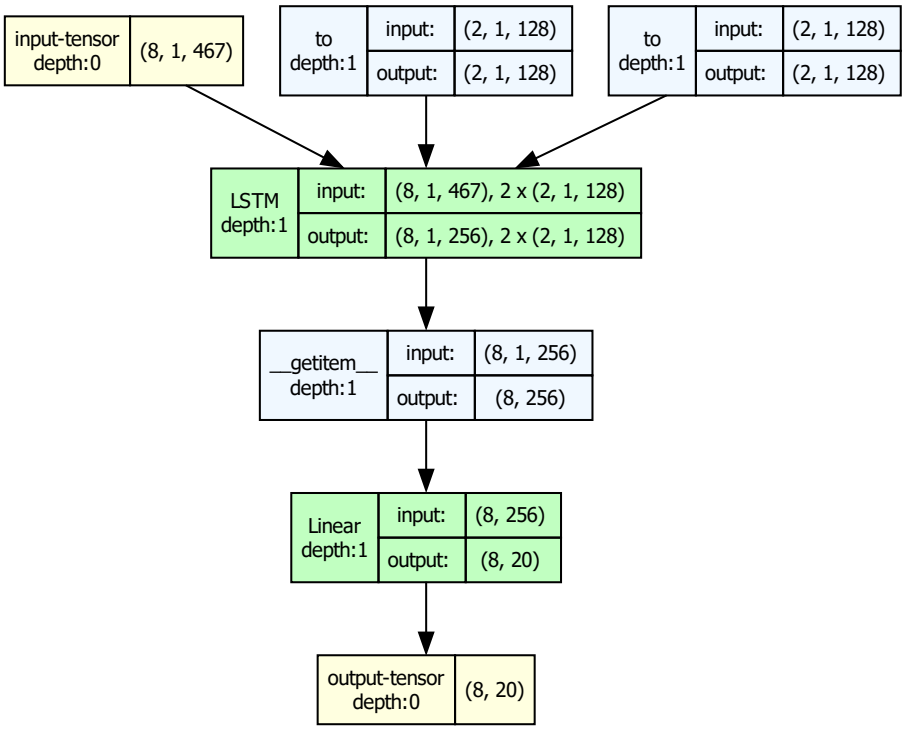
### 6.1.1 Best-performing Model Classification Task

Following the outcomes of the hyperparameter tuning process, as depicted in Figure 8 introduced in Section 5.1.1, the LSTM model with Screen information was configured with the following parameters: a singular LSTM layer featuring a hidden size of 128, an optimizer employing a learning rate of 0.0692, and batches comprising 8 trial sequences.

Figure 9 presents a visual representation[1] of the configured model. The yellow input tensor encompasses one batch of 8 fixation vectors, each containing 467 entries encoding measurements detailed in Section 4.1 and the current screen arrangement. The gray boxes labeled 'to depth:1' illustrate the dimensions of the $h_0$ and $c_0$ tensors. The green box labeled 'LSTM depth:1' visually depicts the transformation from a tensor of size (8, 1, 467) to a hidden size tensor of (8, 1, 256). Notably, due to the bidirectional nature of the model, the effective hidden size is twice the tuned hidden size of 128. The subsequent gray box illustrates the removal of the batch dimension, as it holds no additional information. The succeeding green box labeled 'linear depth:1' illustrates the linear transformation from the hidden size tensor to the output tensor, resulting in a reduction from (8, 256) to (8, 20).

[1]The visualisation was generated using the torchview package available at: https://github.com/mert-kurttutan/torchview

Figure 9: Best LSTM Model Classifier Configuration




#### 6.1.1.1 10-Fold Random Cross-Validation

In the context of the cross-validation, where the random split of test and training data was employed, the outcomes prove promising, revealing an average Baseline accuracy of 13.75 percent and an average model accuracy of 17.03 percent. The specific results for each fold are provided in Table 3. A one-sided T-test, assuming the model's underlying sample distribution is not greater, yields a P-value of 0.04, satisfying the criterion $p < 0.05$. This statistical significance indicates that the null hypothesis, asserting that the baseline model performs better or equivalently, can be rejected. The corresponding T-value is -1.80.

#### 6.1.1.2 10-Fold Group Cross-Validation

In the case of the 10-Fold group cross-validation, where the training and test data are segregated by participants, both models exhibit generally low accuracy. The

| | Random setting | | Group setting | |
|---|---|---|---|---|
| Fold | Baseline | Model | Baseline | Model |
| 1 | 17.19 | 17.19 | 0.0 | 0.83 |
| 2 | 20.31 | 21.88 | 0.0 | 3.33 |
| 3 | 10.16 | 12.50 | 33.33 | 31.67 |
| 4 | 14.06 | 22.06 | 0.0 | 0.0 |
| 5 | 17.97 | 17.97 | 0.0 | 1.67 |
| 6 | 8.59 | 9.38 | 0.0 | 0.83 |
| 7 | 11.72 | 19.53 | 0.0 | 0.0 |
| 8 | 12.50 | 14.06 | 0.0 | 0.0 |
| 9 | 10.16 | 20.31 | 0.0 | 1.39 |
| 10 | 14.84 | 14.84 | 0.0 | 0.0 |
| Average | 13.75 | 17.03 | 3.3 | 3.97 |

Table 3: 10 Fold Cross-Validation Results Classifier as Percentage

average accuracy is 3.97 percent for the proposed model and 3.3 percent for the baseline model. This suboptimal performance is attributed to the uneven distribution of reading score labels, resulting in scenarios where only one participant carries a particular label. When such a participant is excluded from the training set, the model is inclined to assign a very low probability to this label class, leading to infrequent predictions. Similarly, the baseline model predicts the majority class of the training set of the current fold, resulting in zero percent accuracy if the majority class label is absent from the test set. However, when the majority class label is present, as exemplified in fold 3, the baseline model performs better than the proposed model, correctly predicting a third of the true class label. Consequently, the t-test results suggest an inconclusive determination of whether the proposed model outperforms the baseline, as evidenced by a T-value of -0.14 and a P-value of 0.44, failing to satisfy the criterion $P < 0.05$.

### 6.1.2 Best-performing Model Regression

Moving on to the best-performing model for regression, the hyperparameter tuning process outlined in Section 5.1.1 yielded the configurations displayed in Figure 10.

The optimal parameters include a hidden size of 128, a batch size of 1, eight con-

Figure 10: Hyperparameter Tuning of Regression Model

| hidden_dim | embedding_dim | lr | batch_size | num_layers | iter | total time (s) | loss |
|---|---|---|---|---|---|---|---|
| 256 | 16 | 0.00346472 | 1 | 2 | 10 | 1697.71 | 28.3845 |
| 64 | 128 | 0.0302312 | 2 | 16 | 1 | 262.121 | 41.4551 |
| 256 | 32 | 0.00317621 | 2 | 4 | 2 | 423.88 | 42.8439 |
| 64 | 16 | 0.00263846 | 2 | 16 | 1 | 257.095 | 49.4952 |
| 64 | 64 | 0.0191644 | 1 | 4 | 10 | 1996.68 | 31.6701 |
| 256 | 64 | 0.0148521 | 4 | 2 | 1 | 168.14 | 49.0304 |
| 512 | 64 | 0.0604644 | 2 | 8 | 8 | 2300.27 | 43.0847 |
| 256 | 32 | 0.00360961 | 2 | 1 | 10 | 1646.29 | 41.1076 |
| 128 | 16 | 0.0316074 | 4 | 1 | 10 | 1515.35 | 47.0112 |
| 64 | 16 | 0.00164727 | 2 | 8 | 1 | 205.954 | 50.9661 |
| 256 | 32 | 0.0113798 | 4 | 16 | 1 | 291.878 | 48.7094 |
| 128 | 16 | 0.00627702 | 1 | 8 | 10 | 2203.18 | 28.2993 |
| 128 | 32 | 0.00237223 | 2 | 8 | 4 | 787.43 | 44.5858 |
| 64 | 16 | 0.0403588 | 4 | 16 | 4 | 794.038 | 47.299 |
| 256 | 128 | 0.00499053 | 4 | 4 | 1 | 177.294 | 49.6488 |
| 128 | 64 | 0.00106906 | 2 | 16 | 1 | 245.397 | 49.8772 |
| 64 | 16 | 0.0226588 | 4 | 2 | 1 | 157.895 | 49.1467 |
| 64 | 64 | 0.0354286 | 4 | 2 | 2 | 315.774 | 47.9339 |
| 128 | 128 | 0.0830624 | 4 | 4 | 1 | 160.993 | 48.9983 |
| 256 | 128 | 0.00533005 | 2 | 8 | 10 | 2384.21 | 41.5395 |
| 64 | 128 | 0.00332372 | 1 | 8 | 10 | 2274.06 | 31.5194 |
| 256 | 32 | 0.0516458 | 4 | 16 | 2 | 616.304 | 49.2319 |
| 64 | 32 | 0.00106371 | 2 | 16 | 1 | 291.248 | 51.725 |
| 64 | 16 | 0.0138084 | 4 | 8 | 1 | 179.765 | 50.1014 |
| 64 | 64 | 0.00166275 | 2 | 2 | 10 | 1606.64 | 46.2265 |
| 64 | 32 | 0.00121858 | 4 | 2 | 10 | 1520.92 | 52.9095 |
| 64 | 64 | 0.00688394 | 1 | 16 | 2 | 615.416 | 31.602 |
| 256 | 64 | 0.0499829 | 1 | 2 | 1 | 189.467 | 35.1605 |
| 64 | 32 | 0.028597 | 1 | 8 | 2 | 449.902 | 28.7591 |

nected LSTM layers, and a learning rate of 0.00627.

#### 6.1.2.1 Random Cross-Validation Regression

Evaluation based on mean square error indicates a suboptimal performance for the SRTL-Scorer, predicting, on average, 35.51 score points off from the true score ranks [2] in the group evaluation setting, whereas the baseline model was 35.38 score points off. Table 4 depicts the results of the random cross-validation setting, where for each fold the mean square error is computed. The results were rounded to integers for better readability.

Model and Baseline have very similar results for each fold of the cross-validation. Unsurprisingly, the T-test results in a T-value of 0.0001 and a P-value of 0.9999, not

[2] This number is the result of taking the square root of each mean square error, summing them up and dividing it by the number of folds

| | Random Setting | | Group Setting | |
|---|---|---|---|---|
| Fold | Model | Baseline | Model | Baseline |
| 1 | 1407 | 1396 | 1060 | 1048 |
| 2 | 1440 | 1441 | 834 | 721 |
| 3 | 1353 | 1355 | 323 | 374 |
| 4 | 1170 | 1172 | 1419 | 1418 |
| 5 | 1086 | 1096 | 636 | 583 |
| 6 | 1386 | 1369 | 3104 | 3053 |
| 7 | 1156 | 1120 | 411 | 534 |
| 8 | 1157 | 1132 | 3594 | 3608 |
| 9 | 1177 | 1193 | 884 | 919 |
| 10 | 1402 | 1430 | 2014 | 2004 |
| Average | 1273 | 1273 | 1426 | 1438 |

Table 4: 10-Fold Cross-Validation Regression Results Mean Square Error

satisfying the criterion $P < 0.05$, indicating that there is a high chance that both models perform equally or that the baseline model even outperforms the proposed model, since the null-hypothesis can not be rejected.

#### 6.1.2.2 Group Cross-Validation Regression

The results in the group Setting displayed in 4 were similar to the ones of the random setting. One can observe a slightly better performance in the random setting for both the proposed and the Baseline Model compared to the group setting. However, both models performed almost the same compared to each other, leading to a T-Value of -0.0229 and a P-Value of 0.98 for the group setting not satisfying the criterion $P < 0.05$.

## 6.2 Other Models

### 6.2.1 LSTM with Attention layer

During the hyperparameter tuning process for the LSTM model without screen information and incorporating an additional attention layer, as elaborated in Section 4.1.1.3, it became evident that this architectural modification conferred no

discernible advantages over the baseline model. Its performance exhibited a noticeable decline in terms of accuracy, as illustrated in Figure 11 compared to the LSTM Model given the screen configuration for each fixation step. The optimal

Figure 11: Hyperparameter Tuning of Model with Attention Layer

| hidden_dim | embedding_dim | lr | batch_size | num_layers | iter | total time (s) | loss | accuracy |
|---|---|---|---|---|---|---|---|---|
| 64 | 256 | 0.00272921 | 4 | 4 | 10 | 1517.3 | 2.8628 | 0.111282 |
| 512 | 32 | 0.00517165 | 8 | 4 | 4 | 2422.25 | 3.02007 | 0.161167 |
| 1024 | 64 | 0.0762166 | 32 | 4 | 2 | 2213.28 | 3.29796 | 0.160399 |
| 512 | 16 | 0.076063 | 8 | 2 | 1 | 342.379 | 3.92806 | 0.161167 |
| 1024 | 128 | 0.0123291 | 8 | 4 | 2 | 3671.2 | 3.04134 | 0.161167 |
| 256 | 32 | 0.00579557 | 4 | 1 | 10 | 1267.5 | 2.82434 | 0.164236 |
| 512 | 16 | 0.0581345 | 8 | 4 | 1 | 546.336 | 3.67193 | 0.161167 |
| 256 | 32 | 0.024499 | 4 | 4 | 1 | 220.927 | 3.33499 | 0.161167 |
| 512 | 256 | 0.0023861 | 2 | 2 | 1 | 727.619 | 4.50563 | 0.159632 |
| 128 | 256 | 0.0113754 | 4 | 2 | 1 | 152.525 | 3.36999 | 0.161167 |
| 1024 | 16 | 0.00417212 | 2 | 1 | 1 | 1112.98 | 4.62361 | 0.159632 |
| 256 | 32 | 0.0637468 | 8 | 1 | 10 | 1475.86 | 2.62441 | 0.179586 |
| 64 | 256 | 0.00201863 | 4 | 4 | 8 | 1327.73 | 2.86926 | 0.0890253 |
| 1024 | 256 | 0.00399186 | 32 | 4 | 2 | 1920.95 | 2.94926 | 0.0867229 |
| 64 | 64 | 0.00194199 | 2 | 4 | 10 | 1860.44 | 2.83582 | 0.0990023 |
| 512 | 256 | 0.0749724 | 2 | 2 | 1 | 510.804 | 5.79157 | 0.159632 |
| 64 | 32 | 0.0175884 | 32 | 4 | 10 | 1195.83 | 2.78072 | 0.162701 |
| 128 | 32 | 0.099178 | 4 | 1 | 1 | 105.1 | 3.49269 | 0.161167 |
| 512 | 256 | 0.0061213 | 32 | 1 | 2 | 368.406 | 2.9425 | 0.160399 |
| 512 | 16 | 0.00869675 | 2 | 4 | 1 | 934.481 | 5.25539 | 0.159632 |
| 256 | 256 | 0.00411221 | 4 | 1 | 1 | 152.544 | 3.29955 | 0.161167 |
| 1024 | 128 | 0.0013811 | 2 | 1 | 1 | 876.717 | 4.51238 | 0.159632 |
| 128 | 128 | 0.00196143 | 8 | 4 | 2 | 318.395 | 2.94594 | 0.100537 |
| 1024 | 256 | 0.0543789 | 4 | 1 | 1 | 575.394 | 5.61765 | 0.161167 |
| 128 | 32 | 0.0089053 | 2 | 4 | 1 | 240.136 | 4.15804 | 0.159632 |
| 64 | 128 | 0.0628391 | 32 | 4 | 10 | 863.298 | 2.5867 | 0.160399 |
| 512 | 128 | 0.0048776 | 2 | 1 | 1 | 312.619 | 4.52923 | 0.159632 |
| 512 | 256 | 0.00201126 | 4 | 1 | 2 | 402.712 | 3.28785 | 0.161167 |
| 128 | 16 | 0.00390219 | 4 | 1 | 10 | 731.722 | 2.74669 | 0.175748 |
| 256 | 64 | 0.0729895 | 2 | 4 | 1 | 199.303 | 4.77945 | 0.159632 |

configuration, characterized by a hidden size of 256, embedding size of 32, batch size of 8, a single LSTM layer, and a learning rate of 0.064, achieved only an accuracy of 17.96 percent. Given the observed performance drop in comparison to the baseline model, particularly when splitting the dataset based on participants, no further investigation went into the performance of this model.

### 6.2.2 LSTM without Screen Information

Similar results (refer to Figure 12 for details) were observed when entirely excluding screen configuration information from the input in the LSTM model, as outlined in Section 4.1.1.1. Subsequently, the performance of this model was not subjected to further exploration.

Figure 12: Results of Hyperparameter Tuning for LSTM without Screen Information

| hidden_dim | embedding_dim | lr | batch_size | num_layers | iter | total time (s) | loss | accuracy |
|---|---|---|---|---|---|---|---|---|
| 512 | 64 | 0.00109767 | 16 | 1 | 10 | 889.025 | 2.92946 | 0.160399 |
| 512 | 32 | 0.0224557 | 1 | 2 | 1 | 138.747 | 6.83574 | 0.156562 |
| 64 | 128 | 0.00119059 | 1 | 2 | 10 | 1062.5 | 2.7473 | 0.18419 |
| 512 | 16 | 0.0419678 | 2 | 2 | 1 | 270.239 | 4.79197 | 0.159632 |
| 256 | 64 | 0.0150891 | 16 | 4 | 1 | 92.2926 | 2.96662 | 0.158097 |
| 128 | 128 | 0.0312913 | 16 | 1 | 2 | 154.551 | 2.85717 | 0.158097 |
| 512 | 32 | 0.00528854 | 16 | 2 | 1 | 126.061 | 2.98018 | 0.0414428 |
| 64 | 16 | 0.00285518 | 2 | 8 | 1 | 111.267 | 3.06388 | 0.159632 |
| 512 | 32 | 0.0537813 | 1 | 4 | 1 | 239.242 | 8.96308 | 0.156562 |
| 128 | 128 | 0.00223183 | 4 | 8 | 1 | 100.836 | 3.00938 | 0.0920952 |
| 128 | 64 | 0.00779289 | 1 | 8 | 1 | 130.941 | 5.64165 | 0.156562 |
| 64 | 64 | 0.00524499 | 16 | 2 | 2 | 143.145 | 2.93227 | 0.158097 |
| 128 | 128 | 0.00109062 | 2 | 8 | 4 | 473.544 | 2.87108 | 0.159632 |
| 128 | 16 | 0.0048191 | 16 | 16 | 2 | 203.811 | 2.95304 | 0.158097 |
| 64 | 16 | 0.0894269 | 1 | 2 | 1 | 92.4462 | 6.92351 | 0.156562 |
| 64 | 16 | 0.00232582 | 8 | 8 | 2 | 169.578 | 2.93021 | 0.161167 |
| 64 | 64 | 0.00231995 | 16 | 16 | 2 | 190.893 | 2.93211 | 0.160399 |
| 512 | 16 | 0.00381612 | 4 | 1 | 1 | 132.616 | 3.11615 | 0.161167 |
| 128 | 16 | 0.00518506 | 8 | 4 | 1 | 86.271 | 2.97584 | 0.0498849 |
| 256 | 64 | 0.0861171 | 16 | 8 | 10 | 1311.3 | 2.62017 | 0.158097 |
| 256 | 64 | 0.015367 | 2 | 4 | 1 | 239.363 | 4.18497 | 0.159632 |
| 64 | 64 | 0.0422911 | 4 | 16 | 1 | 104.133 | 3.65281 | 0.161167 |
| 256 | 16 | 0.0175502 | 4 | 1 | 1 | 92.1508 | 3.14376 | 0.161167 |
| 64 | 128 | 0.0228115 | 16 | 2 | 10 | 760.774 | 2.70543 | 0.158097 |
| 64 | 32 | 0.0230659 | 4 | 16 | 1 | 112.612 | 3.22939 | 0.161167 |
| 128 | 32 | 0.00121721 | 8 | 1 | 2 | 148.955 | 2.9205 | 0.160399 |
| 512 | 64 | 0.0153398 | 16 | 4 | 4 | 562.339 | 2.82466 | 0.158097 |
| 64 | 32 | 0.00272755 | 16 | 16 | 2 | 186.343 | 2.92959 | 0.160399 |
| 128 | 16 | 0.0010847 | 1 | 2 | 1 | 102.432 | 3.26015 | 0.156562 |

## 6.3 Discussion

This section discusses key aspects related to the performance and implications of the developed models. It aims to address the reasons behind divergent outcomes observed in random and group Cross-Validation settings, the crucial role played by screen configuration information in model performance, and potential avenues for addressing challenges encountered in this thesis. By analyzing these elements, potential directions are identified for future research and refinement.

### 6.3.1 Random vs. Group Cross-Validation of Classifier

Comparing the outcomes of random and group Cross-Validation settings provides valuable insights. The model demonstrates significantly better accuracy over the baseline when subsets are randomly segregated. However, when data is split by participants, the model's performance diminishes, raising questions about its generalisability. Potential reasons for this discrepancy include the model potentially learning participant-specific patterns rather than predicting reading scores. Factors such as the model architecture, input data appropriateness, and dataset size limitations may contribute to this phenomenon. Challenges related to dataset distribution and socio-demographics of participants should be considered for future research., as well as using a more suitable validation process.

### 6.3.2 Regression Model

The regression model's performance was below expectations in both the random and the group settings. Its prediction ability was practically the same, even a bit worse in this specific setting than the baseline model, which always predicts the median of the scores in the test set of the current cross-validation fold. Despite a loss of around 28 during hyperparameter tuning, the model struggled to approach this value in practice. The selection of the Huber-Loss loss function may have impacted this difference because one of its key traits is being less sensitive to outliers [Zhu et al., 2008, 1639]. This means that its output may appear superior to that of a standard mean square error used to assess the model's performance. Further investigation is required to determine whether the model's configuration is inadequate or if other factors are influencing its performance.

### 6.3.3 Screen Information

During the hyperparameter tuning phase, a notable observation is, that the LSTM architecture with one-hot-encoded screen information added to each input vector of the sequence outperforms other models. Table 5 displays the models and their highest achieved accuracy during hyperparameter tuning. These results suggest the significance of screen information, with the model emphasizing this information performing the best. Although not further explored in this thesis, the observed trend highlights the potential importance of screen configuration data.

| Model variant | Highest accuracy (percent) |
|---|---|
| Without screen Information | 16.12 |
| With attention layer | 17.57 |
| With screen information | 27.01 |

Table 5: Achieved Accuracy during Hyperparameter Tuning for Classification

### 6.3.4 Possible Solutions

Various strategies can be considered to address the challenges encountered. The following passage addresses distinct aspects, ranging from fine-tuning the classifier's feedback mechanisms to more substantial changes in model architecture and dataset characteristics:

For example, one could refine the classification task: This involves adjustments such as altering the number of classes or adopting a binary classification approach, for instance, distinguishing between SRLT-II scores lower or higher than the dataset's median.

An alternative involves preserving the ordinal nature of classes. implementing an adjusted accuracy measurement tailored to maintain the ordinal relationships between classes can potentially enhance classifier performance.

Further enhancement may be achieved by modifying the model architecture with the attention layer. One specific adjustment is to relocate screen configuration information to the end of the computation sequence, aiming to provide the model with a more robust computational weight. The aspect of the screen configuration information might be fruitful to experiment with in general, such as future research could involve experimenting with different input variations. Exploring alternative placements within the model structure may yield insights into optimizing its impact

on prediction outcomes.

Another approach might be to consider entirely different architectures. For example, investigating Convolutional Neural Network architectures is a promising avenue, given their documented superiority in some settings in sequential prediction tasks compared to LSTMs (see for example Siciliano et al. [2021]; Deng et al. [2023]).

And lastly, in my opinion, the most promising and easiest solution might be increasing the participant dataset size. A larger dataset should enhance the model's generalisation abilities and contribute to a better performance.

# 7 Conclusion

In this thesis, I was able to implement an LSTM-based neural network architecture that can classify visual search-based eye-tracking significantly more accurately than the baseline model. However, it also shows that further investigation is needed since it only outperforms the baseline in one setting explored in this thesis, as discussed in Chapter 6.

Nonetheless, this thesis has addressed a significant research gap in the understanding of the intricate interplay between eye movements, visual search, and reading skills. As highlighted in Chapter 2, this investigation might not only contribute to advancing the comprehension of cognitive processes but also might hold promise for practical applications that can benefit individuals with diverse levels of reading proficiency.

The outcomes of hyperparameter tuning, as detailed in Section 5.1.1, underscore the importance of incorporating screen configuration information into models utilized for predicting visual search tasks.

This finding suggests that future models should be equipped with such information to enhance learning outcomes. This insight could prove instrumental in refining existing models and developing new ones, thereby improving the efficacy of predictive systems in visual search scenarios.

In Section 6.3.4, potential approaches to address challenges encountered in this thesis were discussed. Strategies such as augmenting the dataset, manipulating screen information in alternative manners, and adjusting the classification task were explored. These considerations provide a foundation for future investigations aimed at refining methodologies and overcoming limitations in similar research endeavors.

Moreover, the developed code framework, a byproduct of this work, serves as a valuable resource that can be adapted and extended for future research endeavors, particularly those related to the 'Lesen im Blick' data.

Future research endeavors could involve further refinement and adaptation of the proposed model. Testing the architecture on additional datasets, including those

from different participant demographics, such as adults or a more diverse population, since it would enhance the generalisability of the findings.

Additionally, extending the scope to incorporate prediction tasks for other languages would contribute to establishing a more universal relationship between eye movement patterns and reading fluency, independent of language constraints.

In conclusion, this thesis not only addresses existing gaps in research but also lays the groundwork for future investigations, offering valuable insights and tools for advancing the understanding of cognitive processes related to visual search and reading skills.

# Glossary

The explanations of eye-tracking related terms stem from: SR Research Website other explanations are from Googles Machine Learning Glossary

**Accuracy** The number of correct classification predictions divided by the total number of predictions

**Baseline** A model used as a reference point for comparing how well another model (typically, a more complex one) is performing. [...] For a particular problem, the baseline helps model developers quantify the minimal expected performance that a new model must achieve for the new model to be useful.

**Bidirectional** A term used to describe a system that evaluates both preceding and following input of a target section of a sequence. In contrast, a unidirectional system only evaluates the preceding input of a target section of the sequence.

**Categorical Data** Features having a specific set of possible values [...]. Also sometimes called discrete features.

**Numerical Data** Features represented as integers or real-valued numbers [...] sometimes called continuous features.

**One-hot Encoding** Representing categorical data as a vector in which: One element is set to 1 and all other elements are set to 0. One-hot encoding is commonly used to represent strings or identifiers that have a finite set of possible values. [...] Representing a feature as numerical data is an alternative to one-hot encoding.[...] With numeric encoding, a model would interpret the raw numbers mathematically and would try to train on those numbers, even if their is no mathematical relationship present.

**LSTM Long -Short-Term-Memory** A type of cell in a recurrent neural network used to process sequences of data in applications such as handwriting recognition, machine translation, and image captioning. LSTMs address the vanishing gradient problem that occurs when training RNNs due to long data sequences by maintaining history in an internal memory state based on new

input and context from previous cells in the RNN.

**RNN** **Recurrent Neural Network**: A neural network that is intentionally run multiple times, where parts of each run feed into the next run. Specifically, hidden layers from the previous run provide part of the input to the same hidden layer in the next run. Recurrent neural networks are particularly useful for evaluating sequences, so that the hidden layers can learn from previous runs of the neural network on earlier parts of the sequence. [...]

**Saccade** The term saccade [...] was first used to describe eye movements by Javal in the 1880s. It refers to the very rapid, conjugate (both eyes do the same thing) eye movements we make when re-orienting the foveal region to a new spatial location. We typically make around 3 saccades each second. [...] as far as eye-tracking is concerned, we are generally assumed to be "effectively blind" during saccades. [...] saccades can be a very rich source of information, and common metrics include their **latency, amplitude, direction and peak velocity**. Amplitude and velocity are related by what is known as the "main sequence" – larger saccades have a higher peak velocity.

**Fixation** Saccades are typically preceded and followed by a fixation – the term used to describe the period of relative stability during which visual information is processed. The rules and algorithms that researchers (and eye-tracking software) use to determine whether the eye is in a fixation or in a saccade can be complex [...]. Fixations are typically 200-300ms but can be much shorter or much longer, and average fixation duration depends to some extent on the context – e.g. fixations are typically somewhat shorter during reading than when viewing scenes. [...] Fixations are typically described by two key metrics: their **location** (which is defined as the average x and y locations of the samples they contain) and their duration. In standard visualization approaches, fixations are represented by circles, with the circle's center at the average x,y location and its diameter reflecting the duration.

# References

M. Benfatto Nilsson, G. Öqvist Seimyr, J. Ygge, T. Pansell, A. Rydberg, and C. Jacobson. Screening for Dyslexia Using Eye Tracking during Reading. *PLOS ONE*, 11(12):e0165508, Sept. 2016. ISSN 1932-6203. doi: 10.1371/journal.pone.0165508. URL `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0165508`. Publisher: Public Library of Science.

J. M. Carroll, J. Solity, and L. R. Shapiro. Predicting dyslexia using prereading skills: the role of sensorimotor and cognitive abilities. *Journal of Child Psychology and Psychiatry*, 57(6):750–758, 2016. ISSN 1469-7610. doi: 10.1111/jcpp.12488. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/jcpp.12488`. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jcpp.12488.

X. Cui, J. Wang, Y. Chang, M. Su, H. T. Sherman, Z. Wu, Y. Wang, and W. Zhou. Visual Search in Chinese Children With Attention Deficit/Hyperactivity Disorder and Comorbid Developmental Dyslexia: Evidence for Pathogenesis From Eye Movements. *Frontiers in Psychology*, 11, 2020. ISSN 1664-1078. URL `https://www.frontiersin.org/articles/10.3389/fpsyg.2020.00880`.

S. Deng, P. Prasse, D. R. Reich, S. Dziemian, M. Stegenwallner-Schütz, D. Krakowczyk, S. Makowski, N. Langer, T. Scheffer, and L. A. Jäger. Detection of ADHD Based on Eye Movements During Natural Viewing. In M.-R. Amini, S. Canu, A. Fischer, T. Guns, P. Kralj Novak, and G. Tsoumakas, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 13718, pages 403–418. Springer Nature Switzerland, Cham, 2023. ISBN 978-3-031-26421-4 978-3-031-26422-1. doi: 10.1007/978-3-031-26422-1_25. URL `https://link.springer.com/10.1007/978-3-031-26422-1_25`. Series Title: Lecture Notes in Computer Science.

J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*,

12(61):2121–2159, 2011. ISSN 1533-7928. URL http://jmlr.org/papers/v12/duchi11a.html.

G. Ferretti, S. Mazzotti, and D. Brizzolara. Visual Scanning and Reading Ability in Normal and Dyslexic Children. *Behavioural Neurology*, 19(1-2):87–92, 2008. ISSN 0953-4180, 1875-8584. doi: 10.1155/2008/564561. URL http://www.hindawi.com/journals/bn/2008/564561/.

P. Haller, A. Säuberli, S. Kiener, J. Pan, M. Yan, and L. Jäger. Eye-tracking based classification of Mandarin Chinese readers with and without dyslexia using neural sequence models. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 111–118, Abu Dhabi, United Arab Emirates (Virtual), Dec. 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.tsar-1.10.

F. Hutzler and H. Wimmer. Eye movements of dyslexic children when reading in a regular orthography. *Brain and Language*, 89(1):235–242, Apr. 2004. ISSN 0093-934X. doi: 10.1016/S0093-934X(03)00401-2. URL https://www.sciencedirect.com/science/article/pii/S0093934X03004012.

M. A. Just and P. A. Carpenter. A Theory of Reading: From Eye Fixations to Comprehension. *Psychological Review*, 87(4):329–354, 1980.

Kristina Moll and Karin Landerl. *SLRT-II Lese- und Rechtschreibtest*. Hogrefe, Bern, 2., korrigierte edition, 2017.

L. Li, K. Jamieson, A. Rostamizadeh, K. Gonina, M. Hardt, B. Recht, and A. Talwalkar. Massively Parallel Hyperparameter Tuning. Feb. 2018. URL https://openreview.net/forum?id=S1Y7OOlRZ.

R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A Research Platform for Distributed Model Selection and Training, July 2018. URL http://arxiv.org/abs/1807.05118. arXiv:1807.05118 [cs, stat].

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85): 2825–2830, 2011. ISSN 1533-7928. URL http://jmlr.org/papers/v12/pedregosa11a.html.

K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998. URL

https://psycnet.apa.org/journals/bul/124/3/372/. Publisher: American Psychological Association.

D. R. Reich, P. Prasse, C. Tschirner, P. Haller, F. Goldhammer, and L. A. Jäger. Inferring Native and Non-Native Human Reading Comprehension and Subjective Text Difficulty from Scanpaths in Reading. In *2022 Symposium on Eye Tracking Research and Applications*, ETRA '22, pages 1–8, New York, NY, USA, June 2022. Association for Computing Machinery. ISBN 978-1-4503-9252-5. doi: 10.1145/3517031.3529639. URL https://dl.acm.org/doi/10.1145/3517031.3529639.

H. Sak, A. Senior, and F. Beaufays. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition, Feb. 2014. URL http://arxiv.org/abs/1402.1128. arXiv:1402.1128 [cs, stat].

F. Siciliano, G. Consolini, R. Tozzi, M. Gentili, F. Giannattasio, and P. De Michelis. Forecasting SYM-H Index: A Comparison Between Long Short-Term Memory and Convolutional Neural Networks. *Space Weather*, 19, Feb. 2021. doi: 10.1029/2020SW002589.

S. D. Sims and C. Conati. A Neural Architecture for Detecting User Confusion in Eye-tracking Data. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pages 15–23, Virtual Event Netherlands, Oct. 2020. ACM. ISBN 978-1-4503-7581-8. doi: 10.1145/3382507.3418828. URL https://dl.acm.org/doi/10.1145/3382507.3418828.

SR Research Ltd. Portable Duo Technical Specifications, July 2017. URL https://www.sr-research.com/wp-content/uploads/2017/07/portable-duo-specifications.pdf.

SR Research Ltd. EyeLink Data Viewer, 2023.

A. Strandberg, M. Nilsson, P. Östberg, and G. Öqvist Seimyr. Eye Movements during Reading and their Relationship to Reading Assessment Outcomes in Swedish Elementary School Children. *Journal of Eye Movement Research*, 15(4): 10.16910/jemr.15.4.3, Oct. 2022. ISSN 1995-8692. doi: 10.16910/jemr.15.4.3. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10205180/.

J. Szalma and B. Weiss. Data-Driven Classification of Dyslexia Using Eye-Movement Correlates of Natural Reading. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–4, Stuttgart Germany, June 2020. ACM. ISBN 978-1-4503-7134-6. doi: 10.1145/3379156.3391379. URL https://dl.acm.org/doi/10.1145/3379156.3391379.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. Technical Report arXiv:1706.03762, arXiv, Dec. 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs] type: article.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. Van Der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. Van Mulbregt, SciPy 1.0 Contributors, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G.-L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. De Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baeza. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, Mar. 2020. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-019-0686-2. URL https://www.nature.com/articles/s41592-019-0686-2.

A. L. Yarbus. Eye Movements During Perception of Complex Objects. In A. L. Yarbus, editor, *Eye Movements and Vision*, pages 171–211. Springer US, Boston, MA, 1967. ISBN 978-1-4899-5379-7. doi: 10.1007/978-1-4899-5379-7_8. URL https://doi.org/10.1007/978-1-4899-5379-7_8.

J. Zhu, S. C. H. Hoi, and M. R.-T. Lyu. Robust Regularized Kernel Regression. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38 (6):1639–1644, Dec. 2008. ISSN 1941-0492. doi: 10.1109/TSMCB.2008.927279. URL https://ieeexplore.ieee.org/document/4669534. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics).

# Lebenslauf

### Persönliche Angaben

Jana Mara Hofmann
Malzstrasse 26
8400 Winterthur
janamara.hofmann@uzh.ch

### Schulbildung

| | |
|---|---|
| 2017- 2019 | Bachelor-Studium Germanistik an der Universität Zürich |
| seit 2019 | Bachelor-Studium Computerlinguistik und Sprachtechnologie an der Universität Zürich |

### Berufliche und nebenberufliche Tätigkeiten

| | |
|---|---|
| Feb. - Juli 2023 | Praktikum UZH; 'Lesen im Blick' Studie |
| Sep. - Dez 2023 | Wissenschaftliche Mitarbeiterin 'Lesen im Blick' Studie |

# A List of most important Python Packages used in my Code:

- Data
  - Pickles
  - Pandas
- Models
  - `PyTorch`
- Preprocessing
  - `Skicit learn`
- Training
  - `Ray`
  - `Skicit Learn`
- Evaluation
  - `Torchmetrics`
  - `Scipy`

# B Declaration of Independence

**Use of Generative Models/ Github Copilot**

The code used in this thesis was written with the support of Github Copilot. The support included automatic completion of code lines and code snippet suggestions. The thesis writing process included correction suggestions from Grammarly and ChatGPT. ChatGPT was also consulted to improve the idiomaticity of the text.

# Declaration of Independent Authorship

## Original work

I expressly declare that the written work I submitted to the University of Zurich in the spring/autumn semester of 2023 with the title

Predicting Reading Fluency with LSTMs based on Visual Search Eye-Tracking Data

is an original work written by myself, in my own words, and without unauthorized assistance. If it is a work by several authors, I confirm that the relevant parts of the work are correctly and clearly marked and can be clearly assigned to the respective author.

I also confirm that the work has not been submitted in whole or in part to receive credit for another module at the University of Zurich or another educational institution, nor will it be submitted in the future.

## Use of sources

I expressly declare that I have identified all references to external sources (including tables, graphics etc.) contained in the above work as such. In particular, I confirm that, without exception and to the best of my knowledge, I have indicated the authorship both for verbatim statements (citations) and for statements by other authors reproduced in my own words (paraphrases).

## Use of text generation models

I expressly declare that I have not only identified existing external sources, but also any automatically generated text that is contained in the above work. I have used the same citation style as if the text had been generated by a human to indicate the source of the automatically generated text. If the contribution of text generation models cannot be linked to specific text passages (see the associated guidelines), I have included a chapter describing the contributions of the text generation model. I acknowledge that no explicit citation is necessary where text generation models are merely used correctively (to improve grammar or idiomaticity of my own words).

## Sanctions

I acknowledge that a thesis that is used to acquire credit and proves to be plagiarism with the meaning of the document *Erläuterung des Begriffs „Plagiat“* leads to a grade

deduction in minor cases, a grade 1 (one) in more severe cases, without the possibility of revision, and in very severe cases can have the corresponding legal and disciplinary consequences according to §§ 7ff of the “Disziplinarordnung der Universität Zürich” and § 36 of the “Rahmenordnung für das Studium in den Bachelor- und Master-Studiengängen der Philosophischen Fakultät der Universität Zürich”.

I confirm with my signature that this information is correct:

Name: Hofmann First Name: Jana Mara

Matriculation number: 17-709-361

Date: 1.12.2023 Signature: