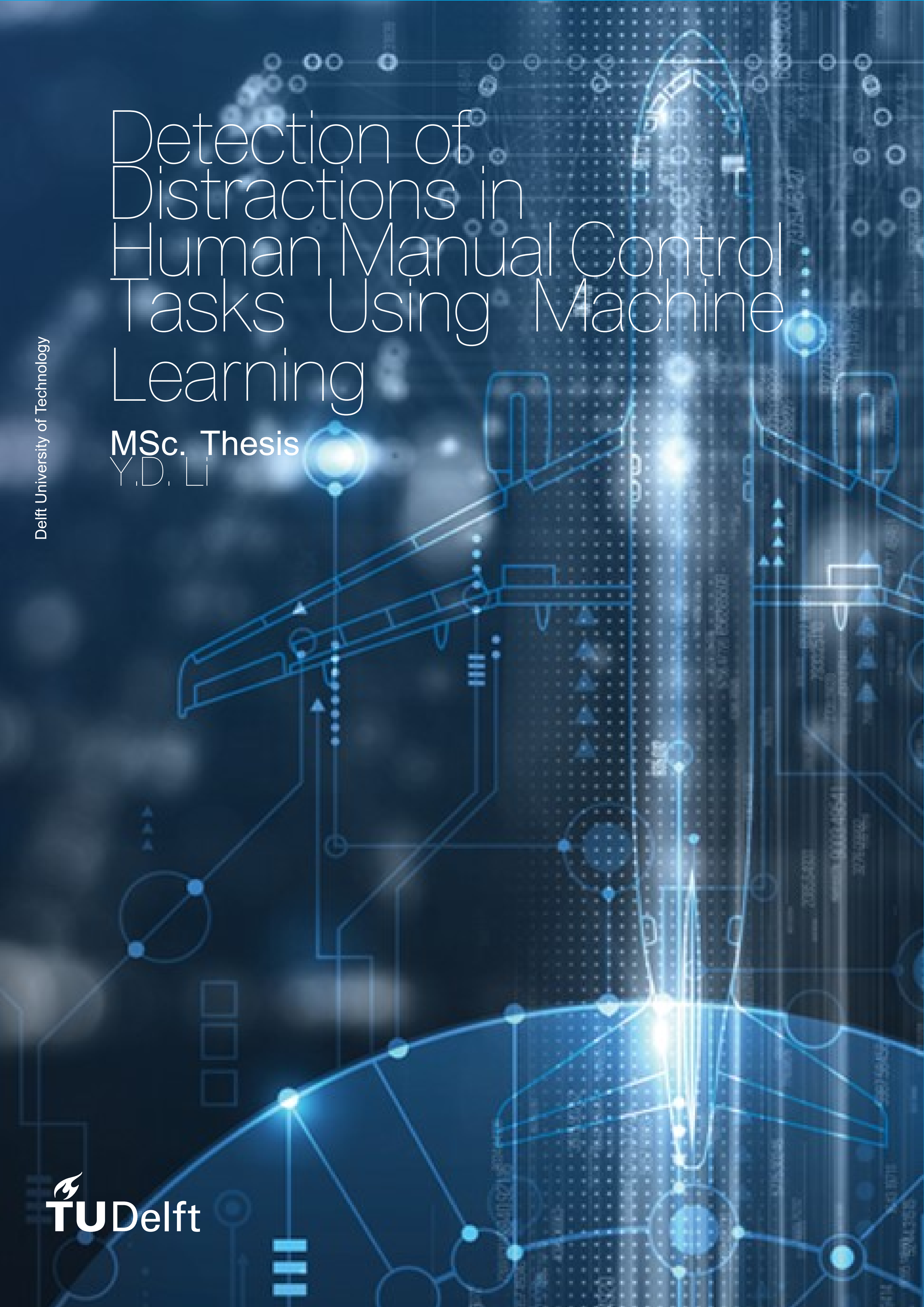


Detection of Distractions in Human Manual Control Tasks Using Machine Learning

MSc. Thesis

Y.D. Li

Delft University of Technology

Detection of Distractions in Human Manual Control Tasks Using Machine Learning

MSc. Thesis

by

Y.D. Li

to obtain the degree of
Master of Science in Aerospace Engineering
at the Delft University of Technology.

Student number:	4872487	
Project duration:	September 20, 2022 – November 1, 2023	
Thesis committee:	Prof. dr. ir. M. Mulder,	TU Delft, supervisor
	Dr. Ir. D. M. Pool,	TU Delft, supervisor
	Ir. T. J. Mulder,	TU Delft, additional member
	Dr. Y. Vardar,	TU Delft, external member

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Cover image obtained from: (https://www.flightsafetyaustralia.com/2020/05/pick-a-number-big-data-artificial-intelligence-and-aviation/)

Preface

This MSc. thesis is written to conclude my Master of Science in Aerospace Engineering at the Delft University of Technology. The research project is about detecting distractions in manual control tasks by applying machine learning techniques to time-series data. This thesis consists of: 1) a scientific paper, 2) appendices to the scientific paper, and 3) a preliminary report.

Looking back on the past year, I really enjoyed working on this challenging topic. Not only did I learn more about human behaviour in manual control tasks, but I also got the opportunity to work with AI (Artificial Intelligence), neural networks in this case. I would like to thank my supervisors Max and Daan for their guidance and support during the entirety of the project. Who were always supportive and ready to help (especially with the eye tracker which caused a lot of headaches).

Others who I want to thank include Rene, who helped in resolving software problems with the eye-tracker. The technical engineers, Andries, Ferdinand, and Harold for helping with preparing equipment for the experiment set-up. And all the participants who participated in the experiment. Not to forget, all the amazing people I have met during the years I have studied in Delft and of course my family.

This is where I conclude my time as a MSc. student and look forward to all the other exciting things what is to come.

Y.D. Li
Delft, November 2023

Contents

List of Figures

List of Tables

List of Symbols

The next list describes several symbols and abbreviations that will be later used within the body of the document

Latin Symbols

e	tracking error
f_d	distrubance function
f_t	forcing function
t	time
u	pilot control output
x	controlled element output

Greek Symbols

τ_p	preview time

Abbreviations

ACMT	Auditory Continuous Memory Task
API	Application Programming Interface
AR	Autoregressive Model
AUC	Area Under Curve
CE	Controlled Element
CED	Controlled Element Dynamics
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CTT	Critical Tracking Task
DI	Double Integrator
DR-ce	(*Data*) Preview, Continuous, Easy Distractions
DR-ch	(*Data*) Preview, Continuous, Hard Distractions
DR-n	(*Data)* Preview (no distractions)
DR-pe	(*Data*) Preview, Prompted, Easy Distractions
DR-ph	(*Data*) Preview, Prompted, Hard Distractions
DS-ce	(*Data*) Pursuit, Continuous, Easy Distractions
DS-ch	(*Data*) Pursuit, Continuous, Hard Distractions
DS-n	(*Data)* Pursuit (no distractions)
DS-pe	(*Data*) Pursuit, Prompted, Easy Distractions

DS-ph	(*Data*) Pursuit, Prompted, Hard Distractions
DUECA	Delft University Environment for Communication and Activation
GPU	Graphics Processing Unit
HASTE	Human Machine Interface and the Safety of Traffic in Europe
HC	Human Controller
HIVE-COTE	Hierarchical Vote Collective of Transformation-based Ensembles
HMI	Human-Machine Interaction
HO	Human Operator
IVIS	In-Vehicle Information System
LCD	Liquid-Crystal Display
ML	Machine Learning
MR-c	(*Model*) Preview, trained on Combined (easy & hard) Distractions
MR-e	(*Model*) Preview, trained on Easy Distractions
MR-h	(*Model*) Preview, trained on Hard Distractions
MRT	Multiple Resource Theorem
MRT	Multiple Resource Theory
MS-c	(*Model*) Pursuit, trained on Combined (easy & hard) Distractions
MS-e	(*Model*) Pursuit, trained on Easy Distractions
MS-h	(*Model*) Pursuit, trained on Hard Distractions
NN	Neural Network
OGRE	Object-Oriented Graphics Rendering Engine
PyGOD	Python Library for Graph Outlier Detection
PyOD	Python Outlier Detection
RMS	Root Mean Square
ROC	Receiver Operating Characteristics
SI	Single Integrator
SURT	Surrogate Visual Research Task
SVM	Support Vector Machine
TI	Thermal Imaging
TODS	Time-series Outlier Detection System
TSC	Time-Series Classification
TTC	Time to Collision
VGG	Visual Geometry Group
WandB	Weights & Biases

I

Scientific Paper

Detecting Distractions in Human Manual Control Tasks Using Machine Learning

Y. D. Li, *Author*, D. M. Pool, *Supervisor*, and M. Mulder, *Supervisor*

***Abstract*—Technological devices are ubiquitous, think of for example smartphones and in-vehicle information systems. Both can contribute towards distracted driving where the visual field of the human controller is shifted away from the primary control task. In this paper a neural network model is trained using the InceptionTime architecture and used to detect distractions in pursuit and preview tracking tasks. For this purpose an experiment has been designed to collect data in which participants are distracted using a visual distraction called the Surrogate Reference Task (SuRT). It was found that distractions are easier to detect in tracking tasks with pursuit displays instead of preview displays. This is because in preview displays the future target trajectory is shown to the human controller, resulting in a lower tracking error compared to pursuit displays. Apart from the tracking error, the InceptionTime neural network was also trained using the time-series data of the control input and system output. Important characteristic of distracted data found were a reduced control input and higher tracking errors, which may have helped in detecting distractions. The classification models were able to predict data samples correctly with an accuracy of 80.78% and 61.66% in pursuit and preview tracking tasks with distractions, respectively. Lastly, individualised models showed better performance when compared to 'one-size-fits-all' models. Results show clear opportunities for applying neural network models in real-time to detect distractions for increasing safety in human operated machines.**

***Index Terms*—Cybernetics, manual control, tracking task, detecting distractions, eye tracker, machine learning.**

NOMENCLATURE

API	Application Programming Interface
AUC	Area Under Curve
CE	Controlled Element
DR-ce	(*Data*) Preview, Continuous, Easy Distractions
DR-ch	(*Data*) Preview, Continuous, Hard Distractions
DR-n	(*Data*) Preview (no distractions)
DR-pe	(*Data*) Preview, Prompted, Easy Distractions
DR-ph	(*Data*) Preview, Prompted, Hard Distractions
DS-ce	(*Data*) Pursuit, Continuous, Easy Distractions
DS-ch	(*Data*) Pursuit, Continuous, Hard Distractions
DS-n	(*Data*) Pursuit (no distractions)
DS-pe	(*Data*) Pursuit, Prompted, Easy Distractions
DS-ph	(*Data*) Pursuit, Prompted, Hard Distractions
DUECA	Delft University Environment for Communication and Activation
GPU	Graphics Processing Unit
HC	Human Controller
HMI	Human-Machine Interaction
IVIS	In-vehicle information system
LCD	Liquid Crystal Display
ML	Machine Learning
MRT	Multiple Resource Theorem
MR-c	(*Model*) Preview, trained on Combined (easy & hard) Distractions
MR-e	(*Model*) Preview, trained on Easy Distractions
MR-h	(*Model*) Preview, trained on Hard Distractions
MS-c	(*Model*) Pursuit, trained on Combined (easy & hard) Distractions
MS-e	(*Model*) Pursuit, trained on Easy Distractions
MS-h	(*Model*) Pursuit, trained on Hard Distractions
NN	Neural Network
RMS	Root Mean Square
ROC	Receiver Operating Characteristics
SuRT	Surrogate Reference Task

I. INTRODUCTION

TRAVEL by passenger vehicles is the deadliest transportation method on a per-mile basis compared to air, rail, and bus travel [5]. Cars are also more accessible to the general public where the road safety depends on the individual traffic behaviour of users. People might over-speed, violate traffic rules, fail to understand signs or are simply not paying attention. Distracted driving is one of the main risk factors in road accidents[1], and is often related to use of smartphones and in-vehicle information systems (IVIS). Both contribute towards distracted driving since people take their eyes off the road and shift their attention elsewhere. Creating a tool that can objectively and non-intrusively detect when people are distracted may help in contributing towards a safer road environment and reduce the number of traffic accidents.

For this purpose, the tracking task is used to investigate whether it is possible to detect distractions in manual control tasks. To be certain distractions can be detected in real-life use cases such as driving a car or flying an aircraft, it should be demonstrated that it works in simpler tasks first.

Literature about detecting distractions in manual control tasks mainly focuses on driver distraction. This includes defining what a distraction is, methods to monitor distractions, and mitigation techniques [17, 18, 19]. Distractions can be in the form of a manual, visual or cognitive distraction [2], while detection methods can vary from vision to sensor-based approaches or a combination of both [10]. The focus of this paper is put on detecting visual distractions using a sensor-based approach.

[1] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

The sensor-based approach can make use of already existing sensors in vehicles, and if necessary small adjustments could be made to these. Using the data collected from the sensors, a ‘driving performance’ profile is created from which a neural network should be able to predict whether the driver is distracted or not. Ersal et al. used only the pedal position data and a Support Vector Machine model to calculate the probability of the driver being distracted [8]. Other common driving performance parameters that were used in previous research were related to steering wheel parameters, speed and lane-offset [3, 8, 13, 14, 18, 19, 24].

Concerning the tracking task, successful application of machine learning models for classification tasks has only been achieved in recent years. This included classifying human pilot skill [9] and human behaviour with various display types [21] in tracking tasks. This means that there is a possibility of deploying ML models for the purpose of detecting distractions in tracking tasks. Nokhai et al. have already carried out an experiment in which participants were presented with a mild visual and cognitive distraction whilst doing the tracking task [15]. Results have shown that training accuracies of up to 81.6% could be achieved.

The objective is to be able to detect when people are distracted in manual control tasks. For this purpose, the classical tracking task is used with a secondary task, the Surrogate Reference Task (SuRT), as distraction. Data collected from experiments are used to train a Neural Network, using the InceptionTime architecture, to detect distractions.

In the experiment, subjects will do tracking runs with both a pursuit and preview display in three different tracking conditions: 1) no distractions, 2) continuous distractions, and 3) prompted distractions. Data obtained from tracking conditions with no distractions and continuous distractions will be used to train the neural network as normal and distracted data, respectively. Runs with prompted distractions will instead be used to test the neural network models that have been trained.

Furthermore, two difficulty levels were used for the distractions by changing the size of the target circle in the SuRT. This was done to investigate how the neural network performed based on various tracking performances. It is hypothesized that hard distractions are easier to detect than easy distractions.

Model performance will be compared between the two different display types, and it is investigated whether a ‘one-size-fits-all’ or individualized model would be more optimised to detect distractions for different individuals. It is hypothesized that distractions are easier to detect in case of pursuit displays since tracking errors would be larger without having the preview time present in preview displays. And an individualized model may be more favourable since it will be only trained based on the tracking data of the individual and will not be influenced by data of others.

The structure of this paper is as follows. In section II the experimental set up and conditions are explained, followed by the methodology in section III. The results are presented in section IV. Finally, a discussion and conclusion is given in section V and section VI, respectively.

II. Experiment

The experiment described in this section has been approved by the HREC (Human Research Ethics Committee) of Delft University of Technology. Application titled: *Measuring Distractions in Human Manual Control Data, #3010*

A. Apparatus

The experiment was conducted in the Human-Machine Interaction (HMI) Laboratory located at the Technical University of Delft, Faculty of Aerospace Engineering. See Figure 1.

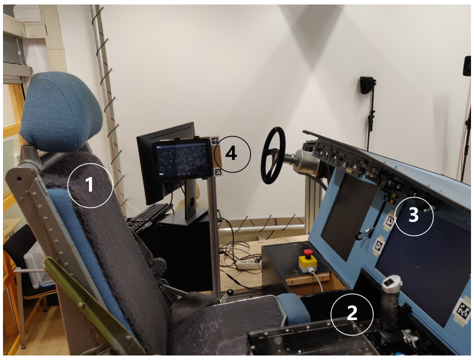


Fig. 1. Experiment set up in the HMI Lab. The participant will be sitting on the right (blue) seat and controls the side-stick.

The participants were seated on the right-hand ‘aircraft’ side (blue seat in Figure 1) from which the participant had to complete several tracking tasks with a Surrogate Reference Task (SuRT) serving as a distraction. The equipment used in this experiment were [1]:

- A fully adjustable aircraft seat, from a Breguet Atlantique, installed on the right-hand side. ①
- A control-loaded hydraulic side stick, with ±30° excursion in roll and ±22° excursion in pitch ②.
- An 18” LCD panel with a 1280 x 1024 pixel resolution as the primary control task display. ③
- An Android tablet on which the SuRT app, developed by the German Aerospace Center [6], was installed. ④
- An eye-tracker from Pupil-labs, Pupil Core headset variant [11], see Figure 2.

The head-worn eye tracker from Pupil Labs Core[2], see Figure 2, was used to record eye and head movements of the participants. This was done in order to determine objectively when a participant was distracted,

Figure 3 shows a schematic of the experiment set up including distances from the subject to the screens of the primary and secondary tasks, which is 74 cm for both. When the subject is doing one of the presented tasks, the other task is outside the human field of view of 150° [20].

B. Tracking Tasks

For the experiment, pursuit and preview tracking tasks were used as the primary task. The tracking tasks were performed

[2] https://docs.pupil-labs.com/core/

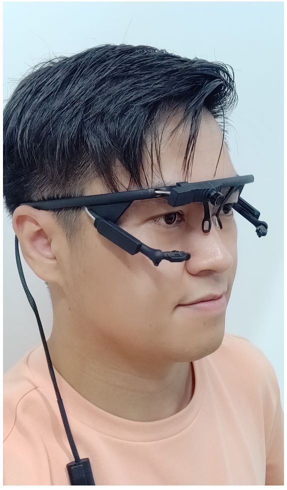


Fig. 2. Pupil Labs head-worn eye tracker.

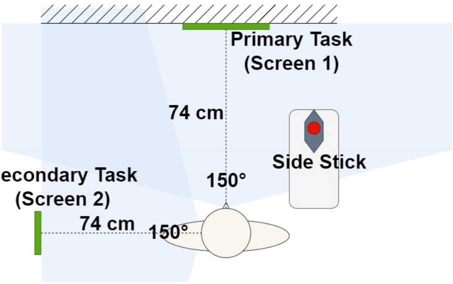


Fig. 3. Schematic of the experiment set up with dimensions.

on the head-down LCD display in front of the participant. An example of how the tracking task is displayed can be seen in Figure 4. The objective here is to minimise the tracking error, $e(t)$. For a preview/pursuit display the tracking error is the distance between symbols representing the controlled element (CE) output and the target. The CE can be controlled with the side-stick on the right-hand side of the seat. The tracking tasks simulated single integrator controlled dynamics (rate control).

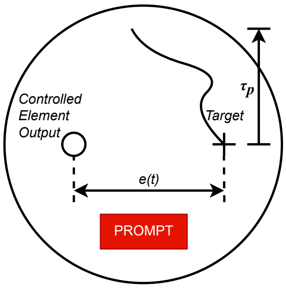


Fig. 4. Schematic representation of the pursuit and preview tracking display. The goal is to steer the state (circle) to the target (plus). The preview trajectory is either present or not.

The tracking task is a closed-loop manual control task composed of 5 signals as a function of time:

- $f_t(t)$, target signal.
- $f_d(t)$, disturbance signal.
- $e(t)$, error signal.
- $u(t)$, HC input signal.
- $x(t)$, output signal.

The relation between the tracking signals is shown in Figure 5.

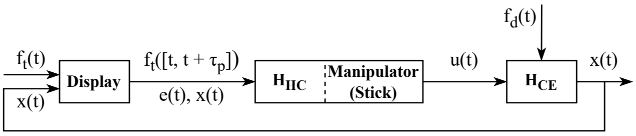


Fig. 5. Block diagram of a closed-loop manual control tasks

In the experiment, the preview and pursuit display types were used. The difference between these display types is the preview time, τ_p, present in preview displays which shows the future trajectory of the target signal, see Figure 4. In real-life manual control tasks the Human Controller (HC) often has access to preview information [7], the future trajectory that should be followed. Examples are driving a car or controlling an aircraft on approach.

However, this preview information could be limited in some conditions. Think of driving in the dark with only the headlights turned on or when the view is hampered due to fog. With limited preview information, it is more important to be not distracted and to focus on the trajectory ahead and it's surroundings. Since in a dynamic environment changes can happen at any moment, and the reaction time is shorter compared to normal conditions.

Thus, the preview display type was used to collect data in a so-called 'normal' condition where the view of a HC would not be obstructed. And the pursuit display type was used to collect data in conditions with no visibility ($\tau_p = 0s$). The choice for choosing these two display types was to mimick the extremes of tracking conditions with or without preview time. It was expected that distractions would have a larger effect on the tracking performance of HCs in pursuit conditions since the future trajectory is not known and there is few opportunity to mitigate the effects caused by distractions.

The tracking task used in this experiment was based on earlier experiments designed by van der El et al. [7]. In the experiment by van der El, multisines were used to create the forcing functions that represents the target signal, f_t. A total of 5 forcing function variations were available to choose from for a tracking run. And a disturbance function, $f_d(t)$ was present on the controlled element. The parameters used in this experiment are:

- Single integrator dynamics (rate control) for the Controlled Element (CE)
  - Gain, K_{CE} = 1.50
- Forcing function settings
  - Bandwidth (rad/s) = 2.5
  - Target gain, K_{ft} = 1.00
  - Disturbance gain, K_{fd} = 1.00
- Stick settings
  - Stick gain, K_{stick} = 10.00

C. Secondary Task

The secondary task in the experiment served as a distraction for the participant whilst performing the primary tracking task. This secondary task was the so-called ‘Surrogate Reference Task’ (SuRT) of which an example is shown in Figure 6. The objective is to find and select the target circle (largest circle) in the midst of a set of smaller circles called ‘distractors’. After selecting the target circle, a new SuRT screen is generated, placing the target circle and ‘distractors’ randomly at new positions. The SuRT was shown on a tablet positioned at eye-height of the participant, at a 90° angle from the LCD screen, held in place with a adjustable tablet holder, as can be seen in Figure 1. Placing the tablet this way required the participant to deliberately look away from the primary task screen. The reason for choosing the SuRT as the secondary task is based on the Multiple Resource Theorem (MRT) by Wickens [23]. This theorem states that the performance of a task worsens when dimensions are shared between concurrent tasks. Since the tracking task is a spatial demanding task, a secondary visual task would cause the strongest interference.

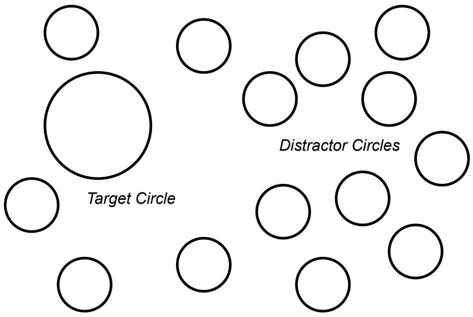


Fig. 6. Example of the SuRT used as a secondary task (The largest circle is the ‘target’ circle, whereas the smaller circles are called ‘distractor’ circles).

For the experiment it was decided to use two variations of the SuRT, an easy and hard variation. In a previous experiment conducted by Nokhai et al. [15] in which participants were also distracted whilst doing the tracking task, it was concluded that the neural network had difficulties in detecting distracted data based on the secondary task used in that experiment. For the preview and pursuit display type data, training accuracies of only 72.1% and 81.6% were reached, respectively. A possible explanation was that the distraction used during this experiment by Nokhai et al. was not visually demanding enough. Participants had to say what color was shown on a projector screen located at 90° angle with respect to the primary screen. The data collected from the two different SuRT difficulties were used to analyse the limitations of the neural network being able to detect distractions in manual control data.

In both easy and hard variation of the SuRT, a total of 49 ‘distractor’ circles and 1 target circle were present. The variation in difficulty was implemented by adjusting the diameter of the ‘distractor’ circles. In easy difficulty, the ‘distractor’ circles had a diameter of 4 mm. And in hard difficulty, the ‘distractor’ circles had a diameter of 7 mm. The diameter of the target circle remained unchanged, namely 8 mm. The design choices for the circle diameters were based on the values used in the experiments conducted by Petzoldt et al. [16].

D. Variation in Distractions

In each tracking run, the participant’s tracking input was measured for 128 seconds, where the first 8 seconds composed of the run-in time needed to stabilize control of the system. Tracking data of participants in normal tracking conditions were collected (without distractions) for both preview and pursuit display types. For tracking runs involving distractions, two variations were used as can be seen in Figures 7a and 7b.

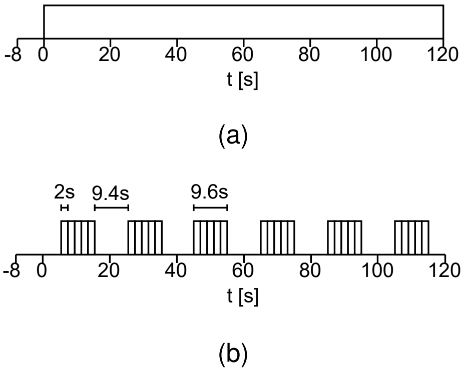


Fig. 7. Tracking run with distractions showing type 1 with continuous distractions in 7a and type 2 with prompted distractions in 7b.

1) Continuous Distractions: In the first trail type, Figure 7a, participants were tasked with completing the secondary task continuously, whilst doing the tracking task. The participants were instructed that the primary and secondary tasks were equally important. The tracking task performance was calculated after each run based on the root-mean-square-error (RMSE). And the performance of the secondary task could be tracked by a predefined scoring system. Each time the participant finds and selects the target circle, a score of 100 points were awarded. If the subject managed to find two subsequent circles within 3 s, an additional 100 points were awarded on top of the previous awarded point. Thus, a streak of 4 subsequent correctly selected circles resulted 1,000 points (100 + 200 + 300 + 400 = 1,000). Whenever a subject selected a ‘distractor’ circle, 100 points were deducted from the total score and the points streak was reset back to 100 points. During the trial type 1 tracking runs a target score of 2,000 had been set which participants had to reach.

2) Prompted Distractions: Tracking run trial type 2 consisted of 6 distractions per run. Figure 7b shows that the distractions occurred in different periods of 9.6 seconds. In these 9.6 second periods, distractions occurred randomly in one of the five, 2 second time slots which were equally spaced apart. The minimum and maximum time between two consecutive prompts were, thus 9.4 s and 24.6 s, respectively. During the tracking run a prompt would appear on the screen of the primary task, see Figure 4, with the text ‘Please find the large circle!’. This meant that the subject had to do the secondary task.

3) Comparison Between Distraction Types: Data of tracking run trial type 2 consist of both normal and distracted tracking data. The same could be said for trial type 1, since the participant also had to focus on the tracking task. However, whether the time in-between secondary task completions was sufficient enough for the participant to reach a steady tracking state cannot be said with confidence. The time of separation between distractions in tracking run trial type 2 does allow for a return to steady state tracking conditions.

The neural network models were trained using the data collected from normal tracking conditions (no distractions) and trial type 1 tracking runs. The performance of these neural network models were then tested using tracking data collected for trial type 2 tracking runs. A tracking run containing a mix of normal and distracted data, trial type 2, is a more realistic representation of how a person could be visually distracted when driving a car compared to being distracted continuously for a longer period of time. A visual comparison between the two different trial types can be seen in Figure 8.

In Figure 8, the left column containing subfigures show the forcing function and controlled element output (8a), control input (8c), and eye tracker data for detecting screen 1 or 2 (8e) for trial type 1. The same is shown for trial type 2 in the right column. There is a clear distinction between the amount of distracted data that can be obtained by using trial type 1 compared to trail type 2 as shown in 8e and 8f, respectively.

E. Execution

1) Participants: A total of 10 students and employees from the faculty of Aerospace Engineering at Delft University of Technology participated in the experiment. All participants were right-handed, their age range varied from 20 to 55+ years old. The participants were only required to select an age range and not mention their exact age due to privacy reasons. Furthermore, participant either had good vision or corrected vision in order to follow the target signal and find the target circle appropriately in the primary and secondary tasks, respectively.

2) Experiment Design: The complete experiment consisted of 10 conditions, 5 per display type, see Table I, with the following definitions for the letters shown in Table I:

TABLE I
COMBINATIONS OF EXPERIMENT TRACKING CONDITIONS.

Tracking conditions			
Preview	PR, -, -	PR, C, E	PR, C, H
		PR, P, E	PR, P, H
Pursuit	PS, -, -	PS, C, E	PS, C, H
		PS, P, E	PS, P, H

- **tracking display type**: preview (PR) or pursuit (PS) layout,
- **secondary task difficulty**: easy (E) or hard (H) variation,
- **variation in distractions**: continuous (C) or prompted (P) distractions.

Due to the amount of conditions, the experiment had been split up in two sessions. One session consisted of conditions with only a pursuit display and the other session would then only consist of runs with a preview display. In order to balance the order in which the subjects would do the different tracking conditions in, 2 Latin squares had been created. One for pursuit sessions and the other for preview sessions. These Latin Squares can be found in Appendix A.

3) Experiment Procedure: At the start of the experiment sessions, participants were briefed on the experiment procedure and tasks that had to be performed. Once the participant was seated in the chair, the eye tracker would be worn on the head. The front camera used to see where the subject is looking was then calibrated to capture the entire screen of the primary and secondary tasks.

Each sessions consisted of 30 tracking runs, of which the first 5 runs consisted of training runs. This was done to familiarise the subjects with the 5 conditions they would do during the session. And every tracking condition was done in blocks of 5 runs after each other, with each block taking approximately 12 minutes. The order in which the conditions were done was based on the Latin square. It should be noted that each run took 128 s to complete, and the first tracking run of each condition was considered to be a training run as well. Thus 4 runs per tracking condition were considered as usable data. Furthermore, the tracking and SuRT score, whenever applicable were noted down after each run. A break of 10 minutes was scheduled half way into the experiment, thus after 15 runs, and participants could ask for extra breaks in between blocks if needed. In total, each experiment session took between 1.5 to 2 hours.

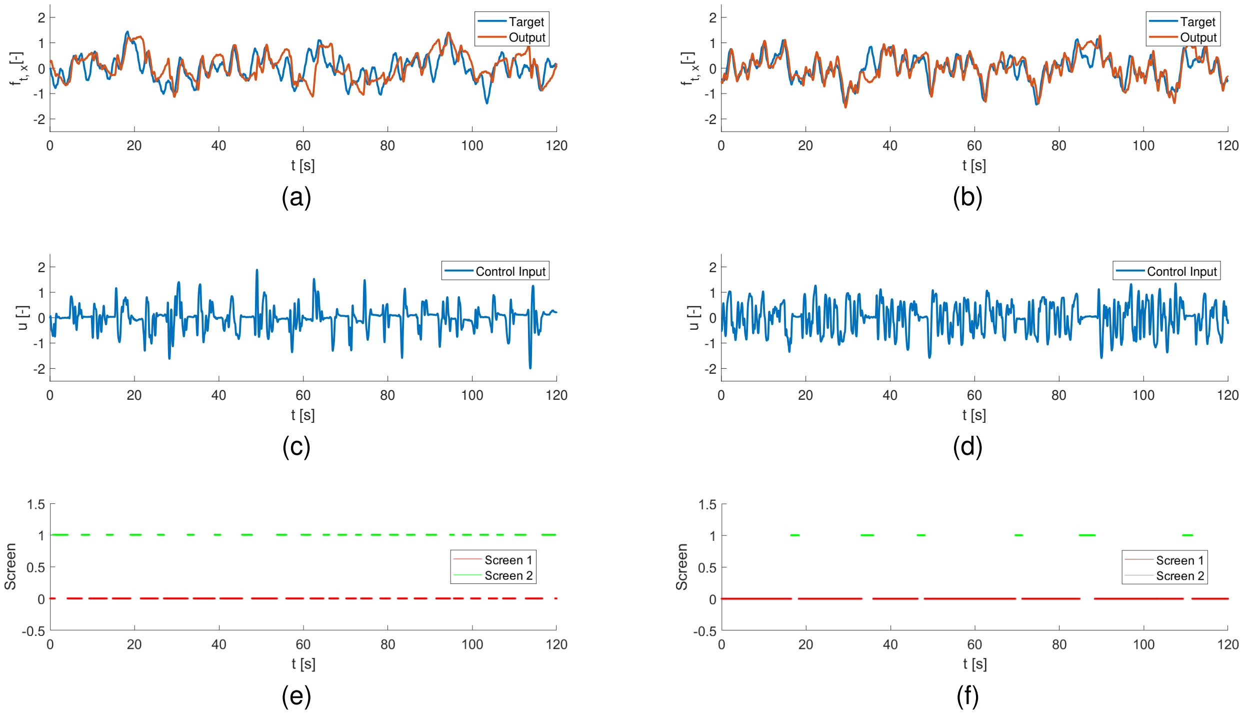
III. Data Processing

This section explains what kind of data had been collected from the experiment and how they have been processed. Furthermore, it includes parameters used for training the Neural Networks (NN) and data analysis methods used to obtain the results.

A. Data Collection

The data collected from the primary tracking tasks consisted of time series data, sampled at a rate of 100 Hz. The eye-tracker provided data about the head position of the participants based on a surface tracker plugin provided by Pupil Labs. The surface tracker uses AprilTags to define planar surfaces in the experimental environment. In this case tags from the tag36h11 family were used to define two surfaces, one for the primary tracking screen and one for the secondary task tablet (4 tags per surface, 1 placed on each corner). Thus, an objective measurement of the screen where participants were looking, was obtained from the frontal camera detecting the tag of the corresponding surface. An example of what kind of data the Pupil Labs surface tracker collected, was shown in Figure 8e and Figure 8f.

The data collected using the surface tracker plugin of the eye tracker were called 'Screen 1', and 'Screen 2'. Green in Figure 8e and Figure 8f means that a tag was detected at a certain time instant for screen 2, for periods highlighted in red the a tag corresponding to screen 1 was detected. Using the



(a) (b) (c) (d) (e) (f)

Fig. 8. Comparison between tracking runs with continuous (left) and prompted (right) distractions.

Pupil Core Network API, real-time eye tracker data could be accessed and synchronised with the tracking task data in the DUECA simulation software used for the experiment.

B. Data Labelling

Based on the eye-tracker data, it could be determined when the participant was distracted by the secondary task. A participant was considered to be distracted during periods in which the camera detected one of the AprilTags used to define the surface of the tablet, since this meant that the participant was engaged in the secondary task. Whenever the frontal camera was not able to detect a tag, the data were considered as distracted since this could happen when the participant was turning their head between screens.

Before the segmented parts of distracted data in a complete tracking run could be used for training NN models, they had to be labelled first. Depending on how long a participant was distracted, these segments could vary from as little as 0.3 s to 6 s. However, a problem arose in deciding how to label training samples as normal or distracted data. Take for example the experiment data collected by Nokhai et al.[15] and NN training parameters used by Verkerk et al. [21]. The samples had a window size of 1.5 s with an overlap of 0.75 s between samples as shown in Figure 9. A sample consists of time-series data, which is sampled at 100 Hz. This means that each sample contains 150 data points for every time signal.

Figure 9 shows that some samples contain only normal or distracted data, highlighted in grey and green, respectively. Red coloured samples contain both normal and distracted data. For this problem, it was decided to label samples with only distracted data (green) and a mix of both distracted and normal data (red) as ‘distracted’. Samples with only normal data (grey) were labelled as ‘non-distracted’. Within the red colored samples, the head of the participant began to turn, shifting the field of view away from or back to the primary tracking task. This meant that the participant did not receive any new information of the tracking task, and could therefore be considered as distracted.

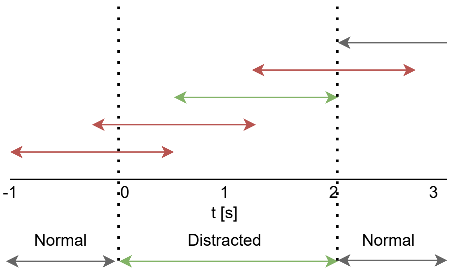


Fig. 9. Consideration in labelling time series samples as distracted or normal.

C. Data Processing

The data samples used to train the NN consist of the following tracking signals:

- $e(t)$, error signal.
- $\dot{e}(t)$, error signal derivative.
- $u(t)$, HC input signal.
- $\dot{u}(t)$, HC input signal derivative.
- $x(t)$, output signal.
- $\dot{x}(t)$, output signal derivative.

Various combinations of these six signals were considered by Verkerk et al. in classifying human control behaviour in tracking tasks with different display types using the InceptionTime NN [21]. The highest model accuracy of 95% was reached by using all signals, $e(t)$, $\dot{e}(t)$, $u(t)$, $\dot{u}(t)$, $x(t)$, and $\dot{x}(t)$ [21].

Furthermore, the data samples are also normalised per run such that the samples are scaled appropriately [22]. This way the magnitudes of each signal in the tracking data could not affect the learning process of the neural net. For example, the tracking error present in distracted data is generally greater compared to normal tracking data. Normalisation prevents the NN from learning the magnitude of error signals.

After the data had been prepared, all samples used to train the neural network were randomly allocated to a training set (80%) or validation set (20%) for training and validation, respectively. The validation set was used to see how well the model performed on the training data. Eventually, the models were tested using test data from prompted tracking runs. The next step was determining the hyperparameters for training the NN. Based on prior work of Verkerk et al. and Kiselev et al., the hyperparameters used to train the neural network are shown in Table II [21, 12]. The hyperparameters had been optimised to classify different display types in tracking tasks and have not been optimised for this particular classification task.

TABLE II
CONFIGURATION OF THE HYPERPARAMETERS USED FOR TRAINING THE NEURAL NETWORK [21, 12].

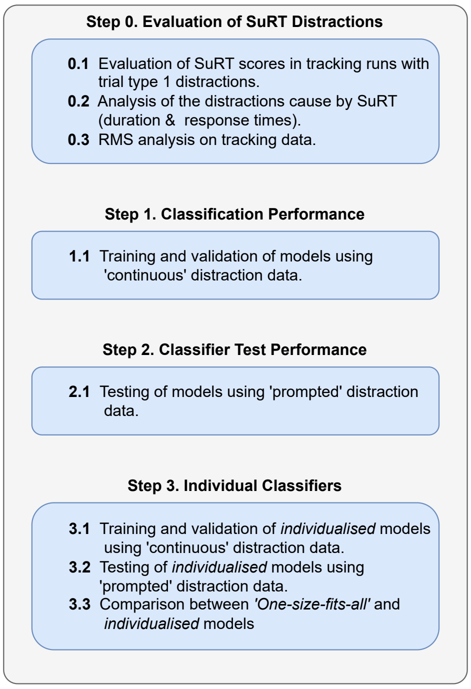
Parameter	Value
Batch Size	64
Epochs	25
Bottleneck	No
Kernel Size	64
Number of Filters	24
Max. Learning Rate	0.00275
Use Residual Connections	No
Weight Decay	0.05
Batch Normalization	No

The models were trained on a P5000 GPU provided by Paperspace[3] through a virtual machine. And the implementation of the neural networks were set-up in Python (version 3.9) using the packages tsai (version 0.3.7), fastai (version 2.7.12), PyTorch (version 1.12.0), pathlib (version 1.0.1). Furthermore, wandb (version 0.15.11) [4] was used to save and export trained models.

D. Data Analysis

Using the data collected from the experiment, and after processing and labelling the data, the steps in Figure 10 were taken to analyse the data.

The steps in Figure 10 include evaluating the distractions caused by SuRT and the performance of tracking tasks based on the RMS of tracking error and control input. The neural network models are analysed on performance based on the training and validation data. These models are then tested using tracking runs with prompted distractions and evaluated on it's ability to detect distractions. At last, individualised models are also trained and compared against the performance of 'One-size-fits-all' models.



Step 0. Evaluation of SuRT Distractions

0.1 Evaluation of SuRT scores in tracking runs with trial type 1 distractions.
0.2 Analysis of the distractions cause by SuRT (duration & response times).
0.3 RMS analysis on tracking data.

Step 1. Classification Performance

1.1 Training and validation of models using 'continuous' distraction data.

Step 2. Classifier Test Performance

2.1 Testing of models using 'prompted' distraction data.

Step 3. Individual Classifiers

3.1 Training and validation of *individualised* models using 'continuous' distraction data.
3.2 Testing of *individualised* models using 'prompted' distraction data.
3.3 Comparison between *'One-size-fits-all'* and *individualised* models

Fig. 10. Steps taken in the data analysis process.

1) Evaluation of Distractions: During tracking runs with trial type 1 (continuous) distractions, participants are asked to reach a SuRT score of 2,000 points. The SuRT score can be seen as a metric of how distracted a person was during the run. Someone with a high SuRT score means that more distracted data of this person are available to train the classifiers with, which may have an influence on how well the model performs in detecting distractions.

Furthermore, the reaction time to a prompt and the time it takes to complete distraction are calculated for tracking runs with trial type 2 (prompted) distractions. This is done to see whether the preview time has an influence on the reactions time. And it can become clear whether there are differences between the easy and hard difficulty of the secondary task.

Lastly, an analysis was done on tracking signals of 'normal' and 'distracted' data collected from tracking runs using prompted distractions. The root mean square of the tracking error and control input signals were calculated for entire runs and individual samples. The RMS tells something about the tracking performance of HCs in general and how distractions affected them.

2) Neural Network Models: A total of 6 different NN models were trained for further analysis.

- **Pursuit**
  1) **MS-e**: trained using PS tracking data in normal conditions and with E distractions.

[3] https://www.paperspace.com/

2) **MS-h**: trained using PS tracking data in normal conditions and with H distractions.
3) **MS-c**: trained using PS tracking data in normal conditions and with E and H distractions.

- **Preview**
  1) **MR-e**: trained using PR tracking data in normal conditions and with E distractions.
  2) **MR-h**: trained using PR tracking data in normal conditions and with H distractions.
  3) **MR-c**: trained using PR tracking data in normal conditions and with E and H distractions.

With a total of 10 participants, the amount of samples used available to train the NNs are presented in Table III:

TABLE III
NUMBER OF SAMPLES PER DISPLAY TYPE, NORMAL AND DISTRACTED (EASY AND HARD) TRACKING DATA.

Display	Normal	Distracted (easy)	Distracted (hard)
Pursuit	6,320	6,320	6,320
(after filtering)	6,318	3,694	4,058
Preview	6,320	6,320	6,320
(after filtering)	6,320	3,212	4,314

It should be noted that the number of samples for distracted data (6320), mentioned in Table III is the maximum possible amount of distracted samples. The tracking runs from which distracted data is obtained also contain normal tracking data. During the labelling process, data labelled as 'normal' were filtered out and not used for training the neural network models. The amount of data samples for pursuit display in normal tracking condition happens to be 6,318. This may be the result of a subject looking at the screen of the secondary task by accident.

3) Metrics: The performance of the models was assessed based on the model accuracy using the validation data set (from trial type 1 distractions) after training. The accuracy is defined as follows:

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

where T and F denote true and false classification of samples for positives (P, non-distracted) and negatives (N, distracted), respectively. Each model type is trained 10 times in order to see how well the neural network is able to classify the training data over multiple training runs. Furthermore, ROC (Receiver Operating Characteristic) curves are plotted for all models using test data. The test data are obtained from the prompted tracking runs in which the subjects are only distracted six times at random moments during a run. The ROC shows how well each model performs on test data and the prediction accuracy of each model for every subject is noted.

Predictions made by the model were based on the decision threshold of 0.5 and the probability, ranging from 0 to 1, of a sample belonging to a certain class. In this research, a probability close to 1 means that the sample was considered to belong to the class 'distracted', whereas a probability of 0 indicated it belongs to a 'normal' sample.

The decision threshold of a model determines what the minimum probability of a prediction should be in order for it to be a 'distracted' sample. For a two class classification problem, the default threshold was set at 0.5. Thus, a sample with prediction probability of 0.6 would mean that it was classified as 'distracted'. A low decision threshold generally results in a lot of false positives. High threshold values on the other hand reduce the number of false positives, but may result in fewer true positives. The threshold is a hyperparameter that can be optimised using ROC curves. This will however not be investigated in this paper and is left for further research.

Finally, individualised models are also trained using data of 1 specific subject. These models are used to make a comparison between 'one-size-fits-all' models based on the classification accuracy of data from prompted tracking runs. This is done in order to see whether a individualised model would be preferred for further research.

From the data analysis the following findings are expected:

- A reduced control input during distractions.
- A better accuracy in classifying distractions for models trained on pursuit data.
- Models trained with hard distractions perform better than models trained with easy distractions.
- Individualised models perform better than 'One-size-fits-all' models.

IV. RESULTS

A. *SuRT Distractions*

The SuRT was used as a secondary task in tracking tasks where participants were continuously distracted or only at prompted moments. In order to see how well the SuRT is able to distract participants, the SuRT scores, response time, time to complete the secondary task and tracking performance based on root means square (RMS) error (e) and input (u) are presented.

1) SuRT score: Subjects were asked to reach a score of 2,000 or higher in the SuRT task in tracking conditions with trial type 1 distractions. Thus, one subject could have been distracted for a longer period of time compared to another subject by doing the SuRT task more often. The obtained SuRT scores have been averaged over the 4 tracking runs per condition and are shown in Table IV.

TABLE IV
AVERAGE SURT SCORES OF EVERY SUBJECT FOR TRACKING CONDITIONS WITH CONTINUOUS DISTRACTIONS.

	DS-ce	**DS-ch**	**DR-ce**	**DR-ch**
Subject	Score	Score	Score	Score
1	6,175	3,650	3,425	2,550
2	2,975	**1,875**	2,950	2,150
3	2,650	2,350	3,100	2,600
4	2,050	**1,900**	2,500	**1,925**
5	204,600	2,925	2,150	2,575
6	2,425	2,050	2,675	2,175
7	22,275	2,400	2,900	2,200
8	2,875	2,050	6,175	2,200
9	2,325	**1,675**	2,225	**1,475**
10	17,450	2,825	7,700	2,100
Median	2,925	2,200	2,925	2,187.5

Table IV shows that only 3 subjects were not able to reach a score of 2000, subjects 2, 4, and 9. These cases all happened to be during tracking runs with hard distractions. From the SuRT scores it can be seen that most subjects took a lot of time finding the target circle in the SuRT task in hard conditions. Namely, the scores for DS-ce and DR-ce are significantly higher, take for example the scores of subjects 1, 8, and 10.

2) SuRT Distractions: Apart from the SuRT scores, the difference in secondary task difficulty can also be found in the time it takes for subjects to find the target circle. For this purpose, the duration of the secondary task and response time of all subjects have been computed using data collected in tracking runs with prompted conditions. The response time and duration can be seen in Figure 11 and Figure 12, respectively.

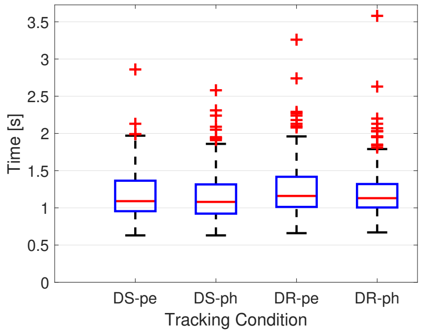


Fig. 11. Response time of subject reacting to the prompt asking to complete the secondary task.

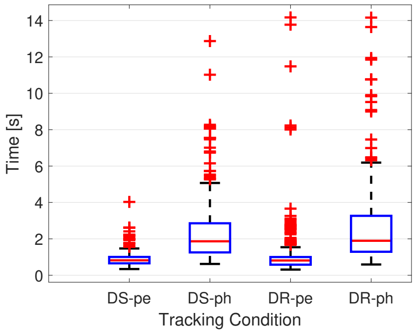


Fig. 12. The duration of subjects to complete the secondary task.

Figure 11 shows that participants took on average 1.1 s to respond to the prompt and look at the tablet on which the secondary task was presented. The time difference between tracking runs using a pursuit and preview display is negligible.

In Figure 12 a large difference can be seen between the time it takes to find the target circle in easy or hard mode. It took on average 0.82 s to complete the easy difficulty secondary task, whereas for the hard difficulty the average time was 1.75 s. The time for subjects to complete the easy and hard secondary task varied from 0.3 s to 4.0 s and 0.6 s to 14 s, respectively. For a more quantitative comparison, the percentage of distractions with a duration lower than 2 s is presented in Table V, since the average time to complete the hard distraction is 1.88 s.

TABLE V
NUMBER OF COMPLETED SECONDARY TASKS WITH A DURATION LOWER THAN 2 S.

	DS-pe	DS-ph	DR-pe	DR-ph
Nr. Distractions	231	133	216	127
	96.3%	55.4%	90.0%	52.9%

3) Tracking performance: The RMS error and input were computed for complete tracking runs in all tracking conditions. The RMS error and input show how well a subject performed the tracking run and the amount of control input needed to follow the target signal. Since the models were trained using data from the tracking conditions DS-n, DS-ce, DS-ch for pursuit, and DR-n, DR-ce, DR-ch for preview, these data sets were analysed first. Starting with the RMS error, the results for pursuit and preview display data are plotted in Figure 13 and Figure 14, respectively.

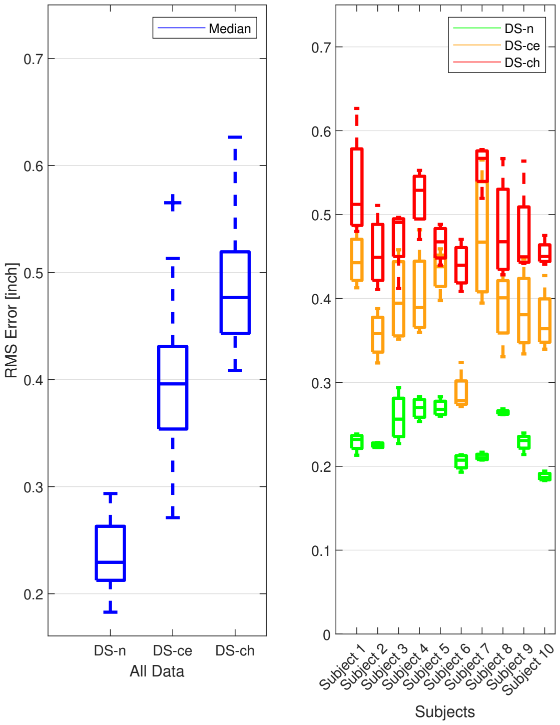


Fig. 13. RMS error of tracking runs performed by all 10 subjects in DS-n, DS-ce, and DS-ch conditions.

4) RMS error: Figure 13 to 15 show the RMS(e) and RMS(u) for all subjects together (left plots, N=40) and each individual subject (right plot, N=4). Figure 13 shows that the RMS error in PS tracking conditions (without distraction) is the lowest. When a secondary task was introduced, the tracking performance became worse as can be concluded from an increase in RMS error for all subjects. The DS-n RMS error was in between 0.2 and 0.3 inch, for DS-ce and DS-ch the RMS error increased to values above 0.3 and 0.4 inch, respectively.

In Figure 14 the RMS error for preview data are shown,

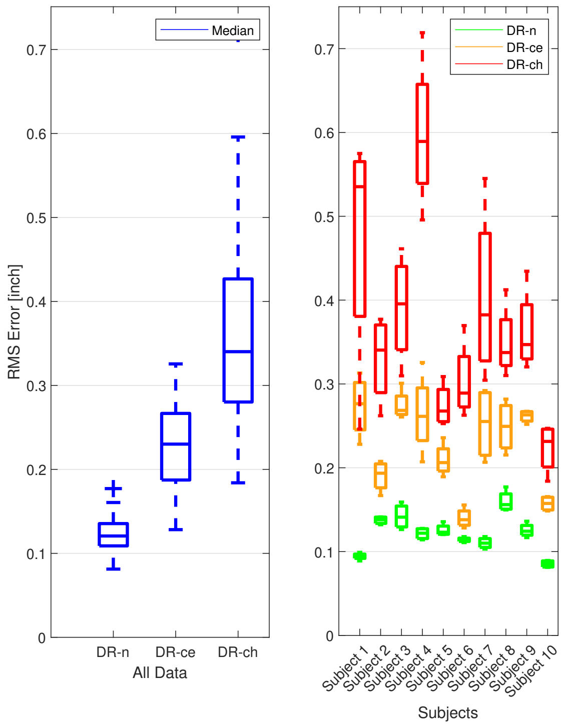


Fig. 14. Root mean square (RMS) error of tracking runs performed by all 10 subjects in DR-n, DR-ce, and DR-ch conditions.

the average RMS error was smaller compared to pursuit data. The tracking performance in normal conditions (DR-n) stayed below 0.17 inch, whereas when subjects were tracking with a hard distraction (DR-ch) the RMS error could reach 0.7 inch with the lowest being 0.18 inch. Tracking performance with an easy distraction (DR-ce) showed a lower spread in RMS error, namely between 0.33 and 0.13 inch.

5) RMS input: The RMS input data obtained from pursuit tracking tasks can be found in Figure 15. When looking at the data in Figure 15, there is no clear general trend. For subjects 3, 5, 9, the RMS input seems to decrease with increasing difficulty of the distraction. The same could be said for subjects 1 and 2, however the RMS input in tracking runs with a hard distraction was higher compared to when there was a easy distraction. On the other hand, the RMS input increases with increasing secondary task difficulty for subjects 6 and 8. And subjects 4, 7, and 10 appear to have a nearly constant RMS input regardless of any distraction.

The RMS input for PR data in Figure 16 show a value around 0.45 inch for tracking runs with preview only. However, when comparing the data with distractions (DR-ce and DR-ch) no correlation is found between the RMS input and difficulty of the distraction. Data from subject 9 show a great decrease in RMS input whereas it increases for subject 4 with increasing distraction difficulty.

B. Classification Performance

The neural networks were trained using the samples collected from normal and continuous distracted conditions of all

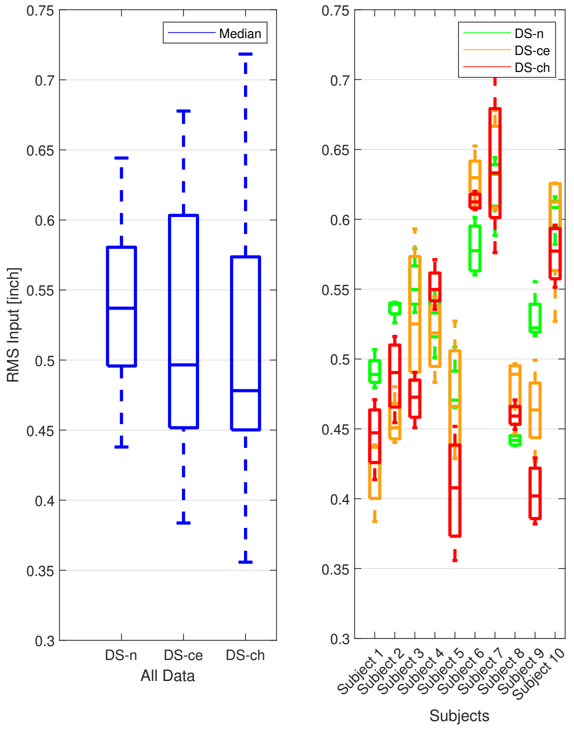


Fig. 15. RMS input of tracking runs performed by all 10 subjects in DS-n, DS-ce, and DS-ch conditions.

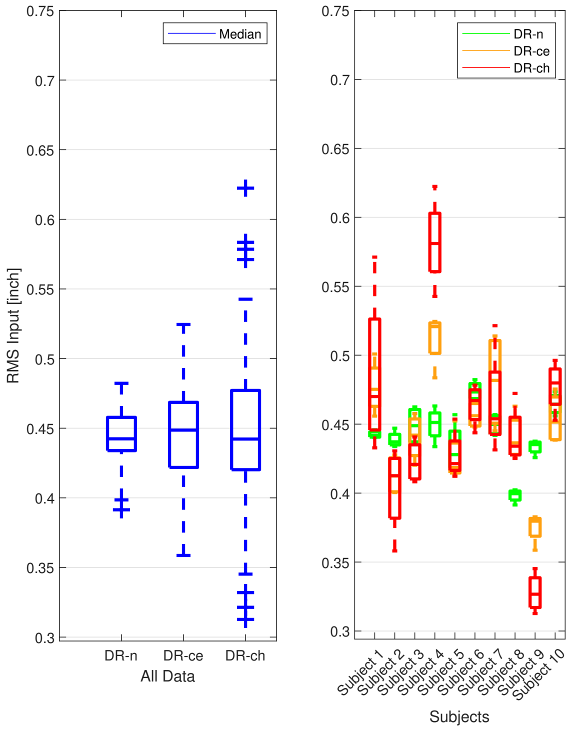


Fig. 16. RMS input of tracking runs performed by all 10 subjects in DR-n, DR-ce, and DR-ch conditions.

subjects. Data from the two different tracking displays, pursuit or preview, were used separately. Regarding the difficulty level of the distraction, the neural networks were trained on solely easy or hard difficulty data as well as both data types together. For each model, 10 training runs have been completed of which the validation accuracies are presented in Table VI.

TABLE VI
VALIDATION ACCURACY OF ALL 6 MODELS SHOWING THE MAXIMUM, MINIMUM, AND AVERAGE ACCURACY OF 10 TRAINING RUNS.

Model	MS-e	MS-h	MS-c
Max. Accuracy	99.24%	100.0%	99.49%
Min. Accuracy	98.37%	99.81%	99.06%
Average Accuracy	98.74%	99.93%	99.27%
Model	**MR-e**	**MR-h**	**MR-c**
Max. Accuracy	96.13%	99.82%	97.80%
Min. Accuracy	94.50%	99.66%	96.50%
Average Accuracy	95.71%	99.77%	97.27%

From Table VI it can be seen that all trained models have a high validation accuracy, well above 94%, using the labelling method described in subsection III-B. It can be seen that models trained using pursuit data achieve a higher validation accuracy compared to the respective counterparts trained using preview data. The same can be said about models trained using hard or easy distracted tracking data: both the MS-h and MR-h models have a higher accuracy than the MS-e and MR-e models. The accuracy of MS-c and MR-c models is inbetween the accuracy of MS-e/MS-h and MR-e/MR-h models, respectively. In order to see how well the MS-c and MR-c models are able to make predictions, a breakdown of the samples used to train the MS-c and MR-c models is given in Table VII.

TABLE VII
BREAKDOWN OF THE TRAINING DATA FOR TRAINING THE MS-C AND MR-C MODELS.

Data	Total Samples	True	False	Accuracy
DS-n	6,318	6,300	18	99.72%
DS-ce	3,694	3,680	14	99.62%
DS-ch	4,058	4,058	0	100.0%
DR-n	6,320	6,290	30	99.53%
DR-ce	3,212	3,167	45	98.60%
DR-ch	4,314	4,312	2	99.95%

Table VII shows the number of samples per data type. DS-n/DR-n are samples from normal tracking data, whereas the acronyms in Table VII ending in ce or ch are samples from distracted tracking data caused by an easy or hard distraction, respectively. It can be seen that when training a model with both easy and hard distracted data, samples with a hard distraction are easier to predict correctly compared to samples with an easy distraction. However, the difference in accuracy very is small, 0.38 and 1.35 percentage points for pursuit and preview, respectively.

C. Classifier Test Performance

In order to test the accuracy of the six models that were trained in detecting distractions, data samples of the prompted tracking runs were used. In prompted runs, subjects were asked to complete the secondary task at random time instances during the entire 120 s of a single tracking run.

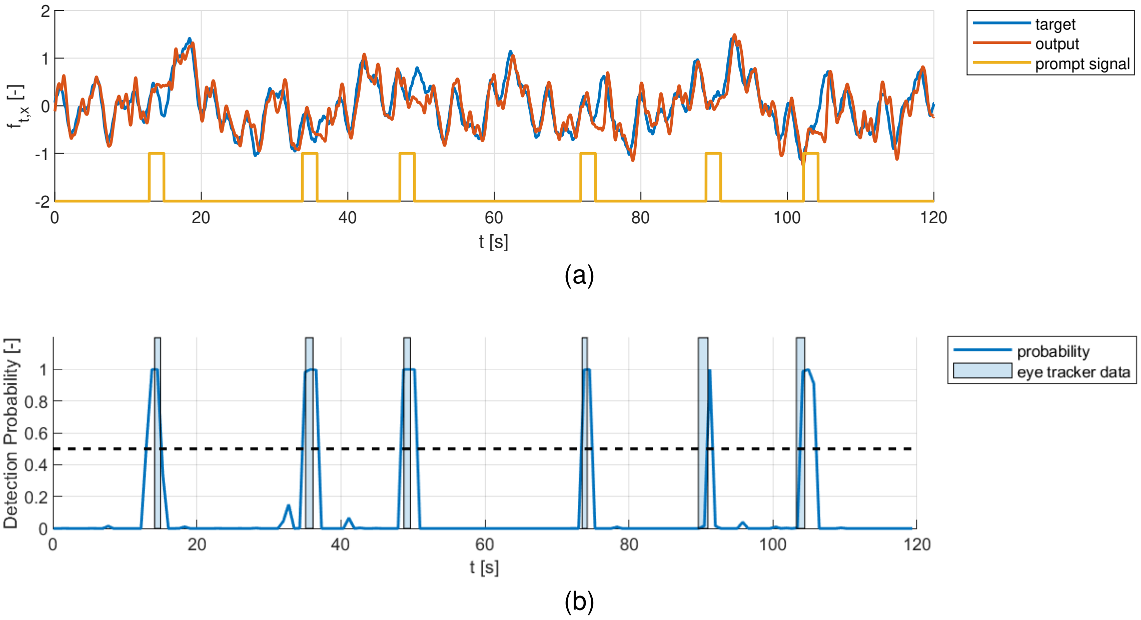
1) Detection Probability: An example of such prompted run can be seen in the top graph of Figure 17. This tracking run was performed by Subject 3 with prompted easy distractions. The blue and red lines represent the forcing function (target) and output (controlled element), respectively, The yellow pulse functions represent the times when the participant was asked to complete the secondary task. An observation that can be made by comparing the forcing function, output, and prompt signals is an increase in tracking error right after or during the prompt when the subject is distracted. The goal for the models is to detect these occurrences in a tracking run and label the data as distracted.

The bottom graph of Figure 17 shows the detection probability produced by the PSE model for a run with easy distractions. The moments in time when the subject is looking at the secondary screen, recorded by the eye tracker, are highlighted in blue. After the prompt has appeared, an increase in detection probability can be seen which also corresponds with when the subject is not looking at the tracking task. Looking at the forcing function and output, deteriorated tracking performance of the subject when distracted is also visible.

During the experiment sessions, every subject completed five tracking runs in each condition of which only data from the last four runs were used. In order to analyze how well the PSE model is able to detect easy distractions of subject 3, the detection probabilities produced by the PSE model for all prompted easy runs have been plotted in Figure 18. The light blue coloured rectangle visible in Figure 18 are the time periods in which the subject is asked to complete the secondary task, see Figure 7. The overall trends of the detection probability curves are as expected. The detection probability increases to 1 within or closely after each coloured rectangle and stays mostly 0 outside of these periods. For a decision threshold of 0.5, the total accuracy of the MS-e model for predicting the samples of subject 3 in Figure 18 is 93.99%.

Having shown the results of the MS-e model for detecting easy distractions, the same has been done for the MS-h model which detects hard distractions. The forcing function, system output and detection probability of Subject 10, Run 2 for the prompted hard conditions (DS-ph) can be seen in Figure 19. Figure 19 shows that the MS-h model is not able to detect all 6 prompted distractions in this particular run. And it has trouble in classifying non-distracted data as normal. By comparing the easy distractions in Figure 17 with the hard distractions in Figure 19, a clear difference can be seen in the duration of each distraction. Hard distractions take on average longer to complete as found in Figure 12.

The accuracy of all other models for classifying samples from prompted runs with easy or hard distractions can be found in Figure 20. Figure 20 shows that not all models were able to reach high accuracies when classifying samples obtained from prompted tracking runs. Furthermore, all models show inconsistency in accuracy when comparing results of different subjects. For example, the MR-e model is able to classify samples of subjects 1, 6, 8, and 10 with accuracies above 80%, whilst the accuracy for the other subjects is below



(a)

(b)

Fig. 17. Forcing function (target), system output 17a, and detection probability 17b of a prompted (P) tracking run (subject 3, run 1) with easy distraction (PSE model).

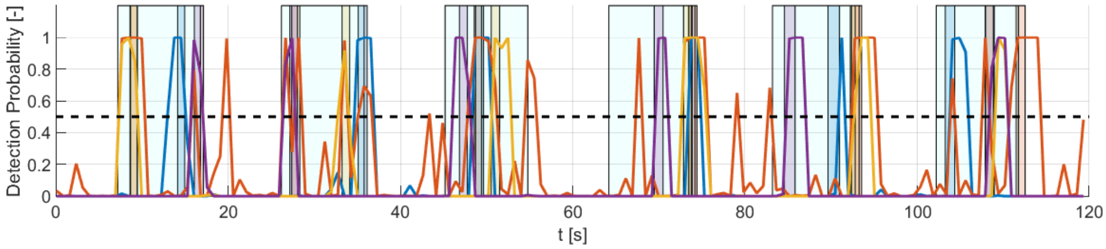
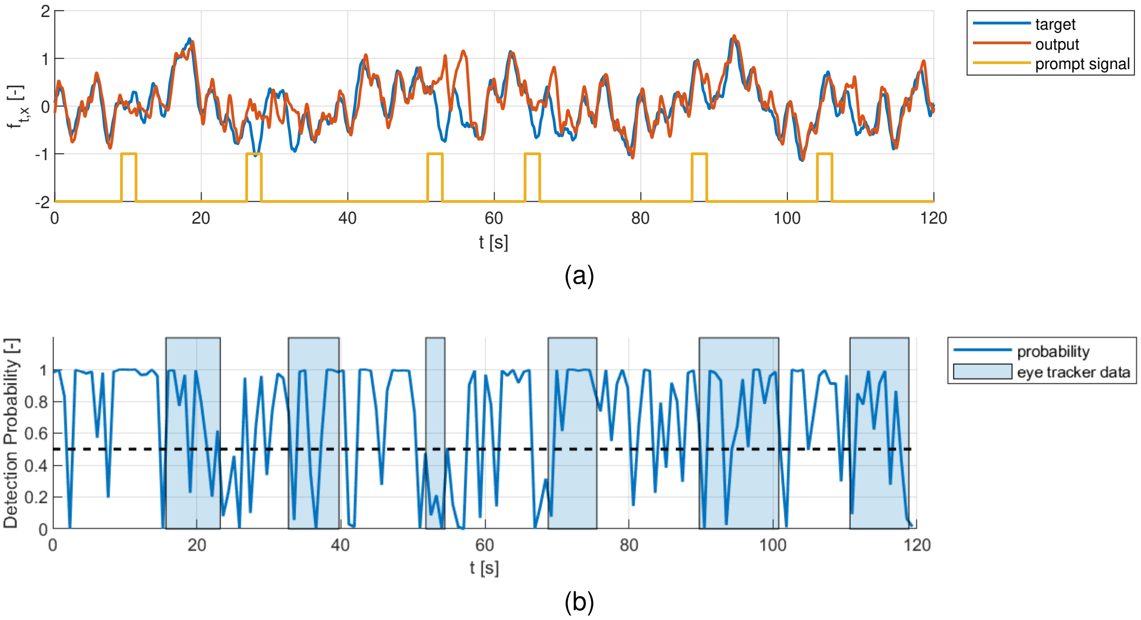


Fig. 18. Detection probability of all prompted (P) tracking runs performed by subject 3 with easy distractions (MS-e model). Decision threshold = 0.5.



(a)

(b)

Fig. 19. Forcing function (target), system output 19a, and detection probability 19b of a prompted (P) tracking run (subject 10, run 2) with hard distraction (MS-h model).

60%. The models are able to classify prompted runs with easy distractions much better compared to prompted runs with hard distractions. Furthermore, the pursuit models (green and red) show an overall better performance compared to preview models (blue and orange). The average prediction accuracy of each model shown in Figure 20 is presented in Table VIII. The low prediction accuracy of some models could suggest that personalised models may be needed to increase performance.

TABLE VIII
AVERAGE MODEL PREDICTION ACCURACY AND THE CORRESPONDING DATA USED.

Model	Data	Accuracy	Model	Data	Accuracy
MS-e	DS-pe	80.78%	MR-e	DR-pe	57.36%
MS-h	DS-ph	48.09%	MR-h	DR-ph	33.78%
MS-c	DS-pe	80.78%	MR-c	DR-pe	61.66%
	DS-ph	41.82%		DR-ph	29.15%

2) Receiver Operating Characteristic (ROC): The classification threshold for determining whether a sample was from normal or distracted data, was set to 0.5. This threshold value was used to produce the results of Figure 18 and Figure 20. In order to see if the classification accuracy of the models would improve by choosing a different threshold, ROC (Receiver Operating Characteristic) curves of all models were generated. A ROC curve also shows how well a model is able to predict in a classification problem. The ROC curves of the pursuit and preview models can be seen in Figure 21 and Figure 22, respectively.

The corresponding AUC (Area Under the Curve of the Receiver Operating Characteristic) values of all ROC shown in Figure 21 and Figure 22 are presented in Table IX.

Figure 21 shows that the MS-e model has the best performance (AUC = 0.8688), followed by the MS-c model when tested on easy distraction data only (AUC = 0.8509). The worst performance can be found for models trained to detect hard distractions. The ROC curves for the preview models in Figure 22 on the other hand show little to no difference when compared to each other. Overall, a clear observation that can be made is that pursuit models have a better performance than preview models. The AUC of all pursuit models are higher than the highest preview model (MR-e) AUC score of 0.6405.

TABLE IX
AUC OF ALL MODELS AND THE CORRESPONDING DATA USED FOR PREDICTIONS.

Model	Data	AUC	Model	Data	AUC
MS-e	DS-pe	0.8688	MR-e	DR-pe	0.6405
	DS-ph	0.7885		DR-ph	0.6308
MS-h	DS-ph	0.6758	MR-h	DR-ph	0.6225
MS-c	DS-pe	0.8509	MR-c	DR-pe	0.6436
	DS-ph	0.7452		DR-ph	0.6105
	DS-pc	0.7897		DR-pc	0.6452

The ROC curves and AUC values of all models have been have plotted separately, together with the ROC curves of each subject data set separately on the respective models. The results can be found in Appendix F.

3) Cross-validation: The ROC curves of the 6 classification models have been plotted on data obtained from prompted runs with hard distractions, which can be found in Figure 23. Figure 23 shows that the MS-e and MR-e models perform the best for pursuit and preview data with hard distractions even though both models are trained with data of easy distractions. The AUC corresponding to the two curves are presented in Table IX. The AUC score for the MS-e model with predictions made on DS-ph data is 0.7885, which is 9.24% lower than using DS-pe data. For the MR-e model, the AUC score is 0.6308, being almost equivalent the result using DR-pe data.

D. Individualised vs. 'One-size-fits-all' Models

Based on the results found in Figure 20, it would also be interesting to see how well subject-specific neural network models would perform. These models are trained and tested solely on tracking data of the specific subject. The validation accuracy of the individual models can be found in Appendix B. Note that the prediction accuracy for models trained with hard distractions are left out for clarity and because of the unsatisfactory performance in Figure 20, see Appendix D. The prediction accuracy of the individual pursuit and preview models can be found in Figure 24 and Figure 25, respectively.

The dashed lines in Figure 24 and Figure 25 represent the model accuracies seen in Figure 20, thus the prediction accuracy of the 'one-size-fits-all' MS-e or MR-e model tested on data of each individual subject. The circles represent the prediction accuracy of subject specific models.

Figure 24 shows that the subject-specific models do not always perform better compared to the general models. This is the case for subjects 1, 7, and 10, where the MS-e and MS-c (easy) models perform worse. For preview models in Figure 25, the subject specific models only perform significantly better for subjects 2 and 3. For other subjects a drop or no change in performance can be seen. When comparing the averaged accuracy of all subject, a decrease in accuracy is found for the MR-e and MR-c (easy) models.

The average model prediction accuracy for both the 'one-size-fits-all' and personalised models are reported in Table X and Table XI for pursuit and preview, respectively. The tables show that individual models perform better than 'one-size-fits-all' models on average, except for MS-c/MR-c models when predicting DS-pe/DR-pe data, respectively.

TABLE X
AVERAGE MODEL PREDICTION ACCURACY OF 'ONE-SIZE-FITS-ALL' AND PERSONALISED PURSUIT MODELS.

Model	Data	'One-size-fits-all'	Personalised	Δ Difference
MS-e	DS-pe	80.78%	83.50%	2.72%
MS-h	DS-ph	48.09%	71.06%	22.97%
MS-c	DS-pe	80.78%	79.02%	-1.76%
	DS-ph	41.82%	61.63%	19.81%

V. Discussion & Recommendations

The main research objective of this paper is to train and test a neural network model that is able to detect when a human controller is distracted when performing a tracking task. The classifier uses the InceptionTime neural network architecture with a time window of 1.5 s. Training data consisted only

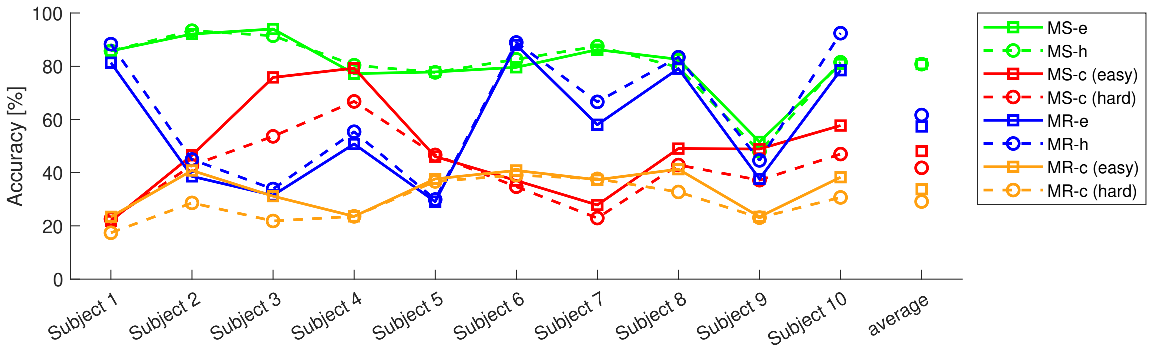


Fig. 20. Accuracy of all 6 models when predicting samples from prompted runs.

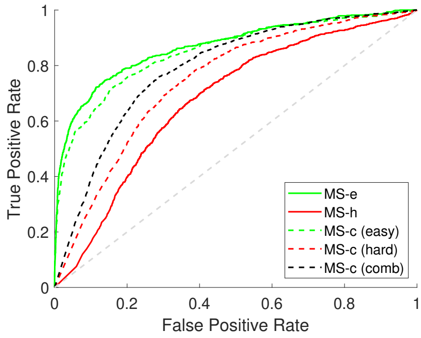


Fig. 21. ROC curves of all pursuit models generated with corresponding easy/hard distraction samples.

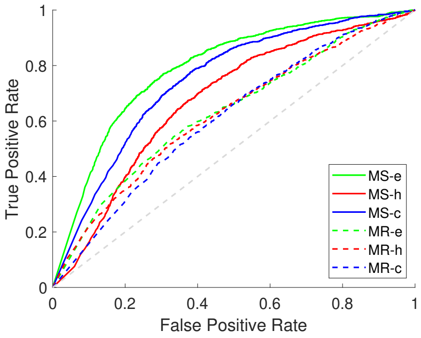


Fig. 23. ROC curves of all models generated on data from prompted runs with hard distractions

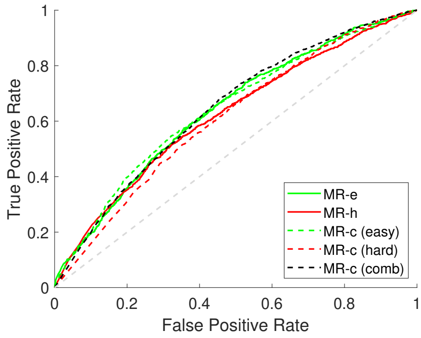


Fig. 22. ROC curves of all preview models generated with corresponding easy/hard distraction samples.

TABLE XI
AVERAGE MODEL PREDICTION ACCURACY OF 'ONE-SIZE-FITS-ALL' AND PERSONALISED PREVIEW MODELS.

Model	Data	'One-size-fits-all'	Personalised	Δ Difference
MR-e	DR-pe	57.36%	61.57%	4.21%
MR-h	DR-ph	33.78%	44.03%	10.25%
MR-c	DR-pe	61.66%	55.30%	-6.36%
	DR-ph	29.15%	34.29%	5.14%

of the system output, x, control input, u, the tracking error, e, and their respective derivatives. A total of 6 'one-size-fits-all' classifier models have been trained and were able to produce good results. The highest validation accuracy was 100.0% while the lowest was 94.50%. When the classifiers are evaluated on test data, the accuracy dropped to 80.78% and with the lowest being 29.15%. In the following paragraphs, the results from section IV will be interpreted in order to point out challenges to which possible resolutions are given. Additionally, limitations of this research and recommendations for future work are given.

In this research a total of 6 'one-size-fits-all' neural network models have been trained to detect distractions. The validation accuracy of all models indicate that there is a distinct difference between the normal and distracted tracking data. Even so, between the data of easy and hard distractions, both of which are used the train the MS-c and MR-c models. Average validation accuracies of 95.71% and 97.27% are reached for the MS-c and MR-c model, respectively. This means that using data of both easy and hard distractions, high validation accuracies can be obtained.

Furthermore, the validation accuracy also shows that continuously distracting subjects is a viable way of collecting a large number of samples of distracted data. For tracking runs with easy or hard distractions, 54.62% and 66.23% of the total amount of samples could be labelled as distracted, respectively. Tracking runs with continuous distractions allow participant to decide for themselves when and how many times they would like to complete the secondary task. However, a minimum score should be set to encourage subject to do the secondary

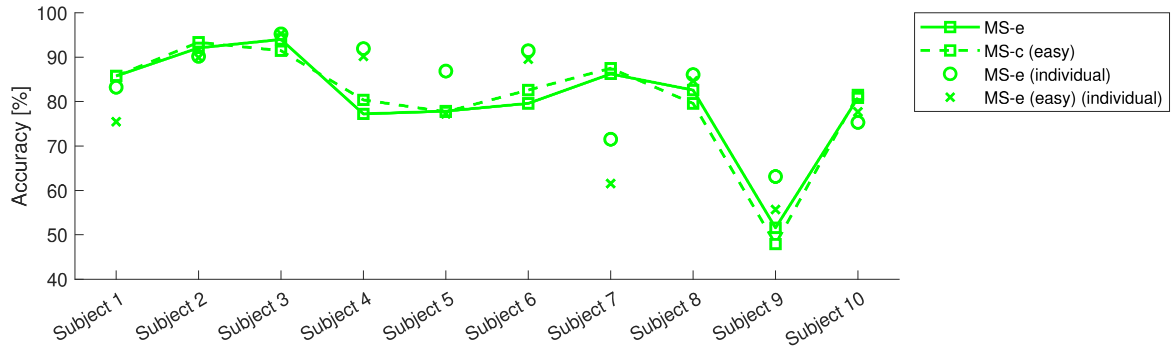


Fig. 24. Accuracy of pursuit models when predicting samples from prompted runs with easy distractions.

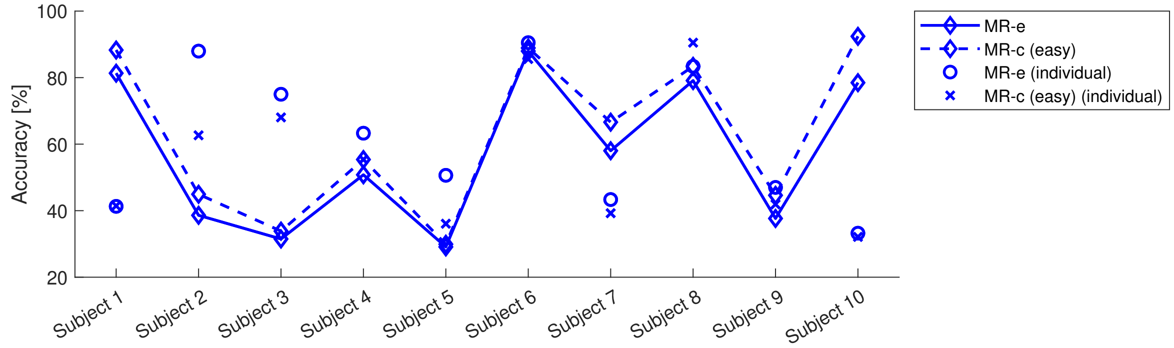


Fig. 25. Accuracy of preview models when predicting samples from prompted runs with easy distractions.

task. The difficulty of doing both the primary and secondary task may discourage subjects to do the secondary task, as informally mentioned by subject 2 during the experiment sessions. When distractions in tracking runs are prompted only a set amount of data samples can be collected. The amount of prompts could be increased, but depending on the time between prompts and difficulty of the secondary task, the time needed to complete it might not be enough. All in all, distracting participants continuously is a better way of collecting samples with distracted data compared to distracting participants a fixed amount of times.

Tracking runs with prompted distractions can be used to test the classification models that have been trained using the data from normal and continuous distracted tracking data. In this research six prompts were spaced apart from each other in such a way that subjects would have sufficient time to return in a steady-state tracking condition. An example of how a prompted run can be used in detecting distractions with a neural network was shown in Figure 17. This example shows the capabilities and promising results of how neural networks could be applied. When the subject was prompted and doing the secondary task, the neural network could detect this, and returned a detection probability of 1. Thus, continuous and prompted distractions are similar to each other, since a model trained on continous distractions is able to detect prompted distractions.

After analysing the prompted runs for all model types per subject (see Figure 20), insightful results were found. Overall, models trained with pursuit data perform, on average, 39.4% better compared to preview models. The reason behind this is that the preview display in preview tasks allows the subject to see the future trajectory of the target. The preview display explicitly shows how the target will behave when doing the secondary distraction. This is an advantage which is missing in pursuit displays, leading to lower tracking errors with the preview display. It should be noted that the tracking error also depends on the preview time, τ_p, shown to the subject [7]. The main finding is in line with the hypothesis that it is easier to detect distractions in pursuit data compared to preview data.

Furthermore, the models performed better when detecting data samples of easy distractions as opposed to hard distractions. It was expected that hard distractions would be easier to detect than easy distractions. Following the results from Figure 20 this is not the case. A possible explanation for why models trained with hard distractions have a worse performance in detecting distractions could be that there is a large variation in tracking performance. When a subject is distracted with a hard distraction, it could take up to 6 s to find the target circle (ignoring outliers). This means that large variations in tracking data for hard distractions is possible, which was also seen in the RMS error and input analysis. Hard distracted data span over a larger error or input range. Therefore, normal samples could be seen as a distraction, resulting in a lower prediction accuracy as shown in Figure 19.

Looking at the ROC curves in Figure 23, a model trained with data of easy distractions only has a better performance in classifying normal samples from distracted samples. This could be caused by the large variation in tracking performance. Thus, samples from hard distractions can be problematic in training classifiers for detecting distractions, as normal data can be classified as distracted data.

Another reason for the bad performance of the models for detecting hard distractions may be the result of the labelling method. In this research, samples were labelled as distracted if the samples contained distracted data. Thus, a sample containing a mix of non-distracted and distracted data would also be labelled as distracted. This method of labelling seemed to be viable due to the high validation accuracies achieved by all trained classification models. It may be insightful to investigate whether using samples with only distracted data can improve the test accuracy of models.

Whether data were considered as distracted was based on the eye tracker data which had information about which screen a participant was looking at. In the experiments, the eye tracker only kept track of which screen was seen using the frontal camera. The accuracy could be improved by also taking into account the gaze of the subject by using the pupil cameras. However, due to technical problems this was not taken into consideration. In the end, using only the frontal camera the data was accurate to tell when someone was distracted.

The configuration of hyperparameters were chosen based on the research conducted by Verkerk et al.[21] and Kiselev et al.[12] on the implementation of the InceptionTime architecture for classifying time-series data of tracking tasks. Since the problem in this research is about classifying non-distracted and distracted data in tracking tasks, the same hyperparameters were used. The validation accuracies show that this was a viable choice.

The test accuracies compared to the validation accuracies were lower, the best results were 80.78% and 61.66% for pursuit and preview, respectively. The AUC values showed that the accuracy for some models could be improved, especially the models trained to detect hard distractions. The ROC can help in finding the optimal decision thresholds for classifying samples. Thus, it is recommended to investigate what the optimal decision threshold would be for theses models.

However, the models do not perform equally well between different subjects. A reason for this could be that the classifier models are not subject-specific. Meaning, the current models have been trained with data of all subjects together instead training an individual-specific model for each subject, using only the respective subject's data. Take for example the MR-e model in Figure 20, the test accuracy also show fluctuating performance between subjects in classifying samples. The model is able to reach accuracies higher than 80% for subjects 1, 6, 8, and 10. The accuracy for the other subjects is worse, the lowest being 29.11% (Subject 5), which is below chance level for a 2-class classifier. A 'one-size-for-all' model is thus not the a good method to achieve high classification accuracies for all subjects.

Subject-specific models have also been trained and tested in order to investigate whether subject-specific or general models perform better. Results from comparing pursuit models show that for 6 out of 10 participants, an personalised model works better. This is true for 7 out of 10 participants in case of preview models. By personalising the neural network models, the classification accuracy increases in general for both pursuit and preview data.

To conclude, the detection of distractions in tracking tasks using machine learning is feasible, since high validation and test accuracies have been achieved. It is proven that it is easier to detect distractions when a pursuit display shown compared to preview displays. A reduced control input is given to the control stick when subjects are distracted. However, samples caused by easy distractions are easier to detect than ones caused by hard distractions. And individualised models perform better compared to 'one-size-fits-all' models.

The ultimate goal of this research is to improve the safety of human-operated vehicles, whether it is in the sky or on the ground, by detecting distractions in control tasks. This paper is a first step towards this direction in which experiments have been performed, for a primary tracking task and a secondary SuRT task as a distraction.

Recommendations for further research on this topic is to test the classification model in real-time. Since this would be the only way to apply neural networks to detect distractions. Possible mitigation strategies could also be considered in case a detection is detected in order to warn or help the human operator to regain control.

VI. Conclusion

The purpose of this research is to detect distractions in human manual control tasks using machine learning and to investigate how to best train classifiers (individualised or 'one-size-fits-all' models). The InceptionTime neural network architecture was trained and tested using collected experiment data (10 participants) from tracking tasks with pursuit and preview displays. Distractions are easier to detect in case of pursuit displays compared to preview displays with test accuracies of 80.78% and 61.66%, respectively. When humans are distracted the control input is reduced since the primary task is neglected. And distractions caused by easy secondary tasks are easier to detect than ones caused by hard tasks.

The success of being able to detect distractions in tracking tasks using machine learning shows promise towards real-time use cases.

References

[1] E.H.H. Thung A. Muis F.N. Postema. *HMI Lab*. URL: https://cs.lr.tudelft.nl/facilities/hmi-lab/.

[2] National Highway Traffic Safety Administration. "Overview of the National Highway Traffic Safety Administration's Driver Distraction Program (DOT HS 811 299)". In: *The Indian Medical Gazette* (2010), p. 36.

[3] Andrei Aksjonov et al. "Detection and Evaluation of Driver Distraction Using Machine Learning and Fuzzy Logic". In: *IEEE Transactions on Intelligent Transportation Systems* 20.6 (2019), pp. 2048–2059. DOI: 10.1109/TITS.2018.2857222.

[4] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: https://www.wandb.com/.

[5] National Safety Council. *Deaths by Transportation Mode*. 2022. URL: https://injuryfacts.nsc.org/home-and-community/safety-topics/deaths-by-transportation-mode/.

[6] DLR Institute of Transportation Systems. *SuRT - Mobile*. Version 1.0 (3). 2015. URL: https://apkcombo.com/surt-mobile/de.lapoehn.surt/.

[7] Kasper van der El et al. "An Empirical Human Controller Model for Preview Tracking Tasks". In: *IEEE Transactions on Cybernetics* 46.11 (2016), pp. 2609–2621. DOI: 10.1109/TCYB.2015.2482984.

[8] Tulga Ersal et al. "Model-Based Analysis and Classification of Driver Distraction Under Secondary Tasks". In: *IEEE Transactions on Intelligent Transportation Systems* 11.3 (2010), pp. 692–701. DOI: 10.1109/TITS.2010.2049741.

[9] M.J.L de Jong. "Classifying Human Pilot Skill Level Using Deep Artificial Neural Networks". In: *MSc. Thesis* (2021). URL: https://repository.tudelft.nl/islandora/object/uuid%3Af09da3e9-3220-46c4-9a3e-e066c9fb3ea0.

[10] Alexey Kashevnik et al. "Driver Distraction Detection Methods: A Literature Review and Framework". In: *IEEE Access* 9 (2021), pp. 60063–60076. DOI: 10.1109/ACCESS.2021.3073599.

[11] Moritz Kassner, William Patera, and Andreas Bulling. "Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction". In: *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '14 Adjunct. Seattle, Washington: ACM, 2014, pp. 1151–1160. ISBN: 978-1-4503-3047-3. DOI: 10.1145/2638728.2641695. URL: http://doi.acm.org/10.1145/2638728.2641695.

[12] Alexander Kiselev et al. "Deep Neural Networks for Classifying the Response of Human Controllers by Display Types". In: *Honours Track Project Report, Faculty of Aerospace Engineering (TU Delft)* (2022).

[13] Zhaojian Li et al. "Visual-Manual Distraction Detection Using Driving Performance Indicators With Naturalistic Driving Data". In: *IEEE Transactions on Intelligent Transportation Systems* 19.8 (2018), pp. 2528–2535. DOI: 10.1109/TITS.2017.2754467.

[14] Kotaro Nakano and Basabi Chakraborty. "Real-Time Distraction Detection from Driving Data Based Personal Driving Model Using Deep Learning". In: *International Journal of Intelligent Transportation Systems Research* 20.1 (2022), pp. 238–251. DOI: 10.1007/s13177-021-00288-9.

[15] Aryan Nokhai, Daan M. Pool, and Max Mulder. "Classification of Distraction in Preview". In: *Honours Track Project Report, Faculty of Aerospace Engineering (TU Delft)* (2023).

[16] Tibor Petzoldt, Hanna Otto, and Josef Krems. "The Critical Tracking Task: A Potentially Useful Method to Assess Driver Distraction?" In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 56 (June 2014). DOI: 10.1177/0018720813501864.

[17] Michael A Regan, John D Lee, and Kristie Young. *Driver distraction: Theory, effects, and mitigation*. CRC press, 2008.

[18] Fabio Tango and Marco Botta. "Real-Time Detection System of Driver Distraction Using Machine Learning". In: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), pp. 894–905. DOI: 10.1109/TITS.2013.2247760.

[19] Kari Torkkola, N. Massey, and C. Wood. "Driver inattention detection through intelligent analysis of readily available sensors". In: Nov. 2004, pp. 326–331. ISBN: 0-7803-8500-4. DOI: 10.1109/ITSC.2004.1398919.

[20] Harry Moss Traquair. "An Introduction to Clinical Perimetry". In: *The Indian Medical Gazette* 79 (1943), pp. 46–46. URL: https://api.semanticscholar.org/CorpusID:30072329.

[21] Gert-Jan H.A. Verkerk. "Classifying Human Manual Control Behaviour in Tracking Tasks with Various Display Types Using the Inception Time CNN". In: *MSc. Thesis* (2021). URL: https://repository.tudelft.nl/islandora/object/uuid%3A74e19a71-01b9-4ce5-b684-15709771f1f7.

[22] Rogier Versteeg, Daan M. Pool, and Max Mulder. "Classifying Human Manual Control Behaviour using LSTM Recurrent Neural Networks". In: *IEEE Transactions on Human-Machine Systems* (2023).

[23] Christopher Wickens. "Multiple resources and performance prediction". In: *Theoretical Issues in Ergonomic Science* 3 (Jan. 2002), pp. 159–177. DOI: 10.1080/14639220210123806.

[24] Lora Yekhshatyan and John D. Lee. "Changes in the Correlation Between Eye and Steering Movements Indicate Driver Distraction". In: *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2013), pp. 136–145. DOI: 10.1109/TITS.2012.2208223.

II

Appendices to Scientific Paper

A

Experiment Documents

In this appendix the experiment documents are presented, these are:

- *Experiment Briefing*: Before the experiment session, participants were briefed on the experiment. This includes the objective, set-up, procedure as well as the rights participants have for participating in the experiment.
- *Experiment Consent Form*: Participants were asked at the start of the first experiment session to fill in the '*Experiment Consent Form*'
- *Latin Square*: For the experiment, data collected from 10 participants were used. In each session, five tracking conditions were done with either a pursuit or preview display. In order to balance the order in which the tracking conditions were completed, two Latin squares were created.

Experiment Briefing

Measuring distractions in human manual control tasks

Thank you for your participation. This experiment is part of a MSc thesis research project that aims to measure the effects of distraction in manual tracking tasks. The experiment is performed in the Human-Machine Interaction Laboratory (HMILab) at TU Delft's Faculty of Aerospace Engineering. This briefing will give an overview of the experiment and explains what is expected from the participants. Please read this document carefully. Should any questions or comments remain, always feel free to discuss these with the researcher conducting the experiment.

Experiment Objective

An ability to monitor manual control behavior is essential for improving safety in human-controlled vehicles. This experiment is meant to collect data on human tracking behavior under visual distractions.

Experiment Set-up

The "HMILab" (Fig. 1), a fixed-base simulator set-up at TU Delft's Faculty of Aerospace Engineering, is used to investigate interaction between human operators and controlled elements. You are asked to take place in the right chair, where you can control the side-stick with your right hand. On the head-down display in front of you, you will look at one of the tracking display types (Fig. 2), with the objective to minimize the tracking error, e(t). For a preview/pursuit display (Fig. 2) the tracking error is the distance between the controlled element output and the target. On a touchscreen tablet positioned on your left-hand side, a "Surrogate Reference Task" will be presented where your goal is to find and select the target circle (largest circle) (Fig. 3) in the midst of smaller circles called "distractors". During your tracking runs, you will be prompted on your tracking display to complete this secondary task. Several runs are collected, and per run, your tracking input will be measured for 128 seconds, where the first 8 seconds compose the run-in time needed to calibrate the identification software. Throughout the experiment, your eye and head movements will be recorded with a head-worn eye tracker (Pupil Labs Core, see Fig. 4) to objectively detect when distraction occurs.

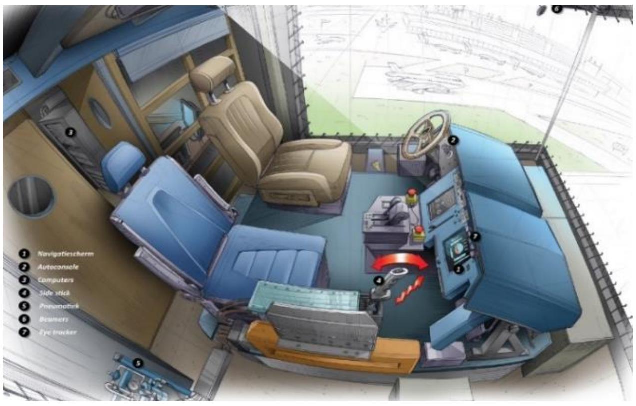


Figure 1: Illustration of HMI Lab. The participant will be sitting on the right (blue) seat and controls the side-stick. The participant will have to look at a tablet on the left.

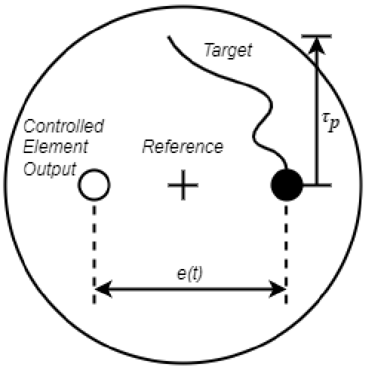


Figure 2: Sketch of the HMI Lab pursuit/preview tracking display.

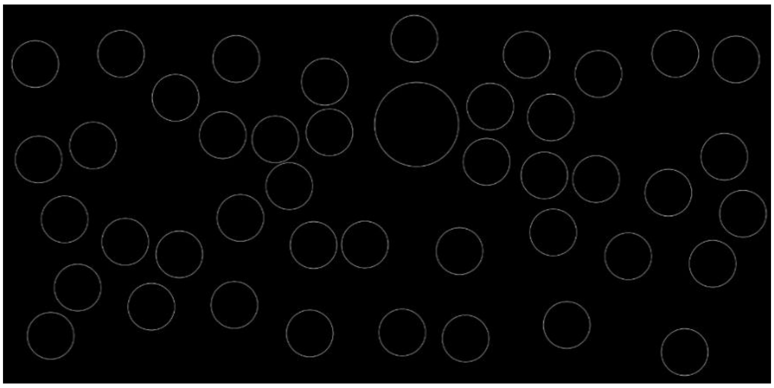


Figure 3: Example of the "Surrogate Reference Task " display on the left hand side. The target circle and distractors are randomly repositioned after selecting the target circle.

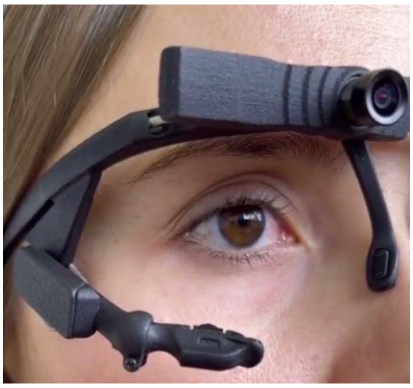


Figure 4: Pupil Labs head-worn eye tracker. Source: https://docs.pupil-labs.com/core/

Experiment procedure

During the experiment, you are tasked with making the controlled element state follow the target in case of a pursuit/preview display by giving tracking input to the side-stick. These tasks always include a *single integrator controlled element* (rate control). During the experiments, you will be prompted to do the secondary task on the tablet on your left-hand side. All tracking runs collect performance in a specific score, which will be communicated to you by the researcher. You will be asked to complete a number of repeated tracking runs and the experimenter will notify you when sufficient data has been collected. Periodically, you will be asked to take a short (i.e., 15 minute) break to avoid fatigue. Should more breaks be required, you can request them at any moment. Prior to data collection, a short period of time will be needed for calibration of the eye tracker. Conducting the full experiment takes approximately 2-3 hours.

Your Rights & Consent

Experiment participation is voluntary. Should you feel uncomfortable, you can decide to stop your participation at any time. By participating in the experiment you agree that the collected data may be published. Your personal data will remain confidential and anonymous, only the researcher can link the collected data to a specific participant. To ensure you understand and comply with the conditions of the experiment, you will be asked to sign an informed consent form.

Contact information researcher: David Li y.d.li@student.tudelft.nl	Contact information research supervisor Dr. ir. Daan M. Pool d.m.pool@tudelft.nl

Thank you again for participating!

Experiment Consent Form

Measuring distraction in human manual control tasks

I hereby confirm, by ticking the box, that:

1. I volunteer to participate in the experiment conducted by the researcher (**David Li**), under supervision of **dr.ir. Daan Pool,** from the Faculty of Aerospace Engineering of TU Delft. I understand that my participation in this experiment is voluntary and that I may withdraw (“opt-out”) from the study at any time, for any reason. □

2. I have read the briefing document and I understand the experiment instructions, and have had all remaining questions answered to my satisfaction. □

3. I understand that taking part in the experiment involves performing manual tracking tasks in the HMILab simulator at TU Delft with an additional side-task shown on a “distractor” display. I understand that only the pseudonimized recorded time traces of the tracking tasks, eye tracking data, and side-task performance are saved and used for data analysis. □

4. I confirm that the researcher has provided me with detailed safety and operational instructions for the HMILab simulator (simulator setup, electro-hydraulic side stick, emergency procedures) used in the experiment. Furthermore, I understand the researcher’s instructions for guaranteeing the experiment’s compliance with current COVID-19 guidelines, and that this experiment shall at all times follow these guidelines. □

5. I understand that the researcher will not identify me by name in any reports or publications that will result from this experiment, and that my confidentiality as a participant in this study will remain secure. Specifically, I understand that any demographic information I provide (gender, handedness, age range, ***see next page***) will only be used for reference and always presented in aggregate form in scientific publications. □

6. I understand that this research study has been reviewed and approved by the TU Delft Human Research Ethics Committee (HREC). To report any problems regarding my participation in the experiment, I know I can contact the researchers using the contact information below. □

My Signature

Date

My Printed Name

Signature of researcher

Contact information researcher:
David Li
y.d.li@student.tudelft.nl

Contact information research supervisor:
dr. ir. Daan Pool
d.m.pool@tudelft.nl

Participant Demographic Information

Measuring distraction in human manual control tasks

Age range:

○ 18 – 19
○ 20 – 24
○ 25 – 29
○ 30 – 34
○ 35 – 39
○ 40 – 44
○ 45 – 49
○ 50 – 55
○ 55+

Handedness:

○ Left handed
○ Right handed
○ Ambidextrous

Gender: __________

Participant number: __________
(filled out by the researcher)

Contact information researcher:	Contact information research supervisor:
David Li	dr. ir. Daan Pool
y.d.li@student.tudelft.nl	d.m.pool@tudelft.nl

PreviewDistractionExp

Experiment Conditions:

PS (Pursuit Display)
PR (Preview Display)

N (Normal Tracking Conditions)
D (Distracted Tracking Conditions)

C (Continuous Distractions)
P (Prompted Distractions)

E (Easy Distractions)
H (Hard Distractions)

Time Tracking Run			128	[s]
Time Btw Runs			16	[s]
Nr Runs per Condition			**Duration**	
PS/PR, N, -, -	5 runs		12	[min]
PS/PR, D,C, E	5 runs		12	[min]
PS/PR, D,C, H	5 runs		12	[min]
PS/PR, D,P, E	5 runs		12	[min]
PS/PR, D,P, H	5 runs		12	[min]
Total	25 runs		60	[min]

Participant Nr	**Session 1**					**Session 2**				
1	PS, D, C, H	PS, D, C, E	PS, N, -, -	PS, D, P, H	PS, D, P, E	PR, D, P, H	PR, D, P, E	PR, N, -, -	PR, D, C, E	PR, D, C, H
2	PR, D, P, E	PR, N, -, -	PR, D, C, H	PR, D, P, H	PR, D, C, E	PS, D, C, E	PS, D, P, H	PS, D, C, H	PS, D, P, E	PS, N, -, -
3	PS, N, -, -	PS, D, C, H	PS, D, P, E	PS, D, C, E	PS, D, P, H	PR, D, C, H	PR, D, C, E	PR, D, P, H	PR, N, -, -	PR, D, P, E
4	PR, D, C, E	PR, D, P, H	PR, D, P, E	PR, D, C, H	PR, N, -, -	PS, D, P, H	PS, D, P, E	PS, D, C, E	PS, N, -, -	PS, D, C, H
5	PS, D, P, E	PS, N, -, -	PS, D, P, H	PS, D, C, H	PS, D, C, E	PR, N, -, -	PR, D, C, H	PR, D, C, E	PR, D, P, E	PR, D, P, H
6	PR, N, -, -	PR, D, C, H	PR, D, C, E	PR, D, P, E	PR, D, P, H	PS, D, P, E	PS, N, -, -	PS, D, P, H	PS, D, C, H	PS, D, C, E
7	PS, D, P, H	PS, D, P, E	PS, D, C, E	PS, N, -, -	PS, D, C, H	PR, D, C, E	PR, D, P, H	PR, D, P, E	PR, D, C, H	PR, N, -, -
8	PR, D, C, H	PR, D, C, E	PR, D, P, H	PR, N, -, -	PR, D, P, E	PS, N, -, -	PS, D, C, H	PS, D, P, E	PS, D, C, E	PS, D, P, H
9	PS, D, C, E	PS, D, P, H	PS, D, C, H	PS, D, P, E	PS, N, -, -	PR, D, P, E	PR, N, -, -	PR, D, C, H	PR, D, P, H	PR, D, C, E
10	PR, D, P, H	PR, D, P, E	PR, N, -, -	PR, D, C, E	PR, D, C, H	PS, D, C, H	PS, D, C, E	PS, N, -, -	PS, D, P, H	PS, D, P, E

B

Validation Accuracy

The validation accuracy of all ‘one-size-fits-all’ and subject specific models are shown in this appendix. Each boxplot is based on the validation accuracy of 10 separate training runs (N = 10).

B.1. ‘One-size-fits-all’ Models

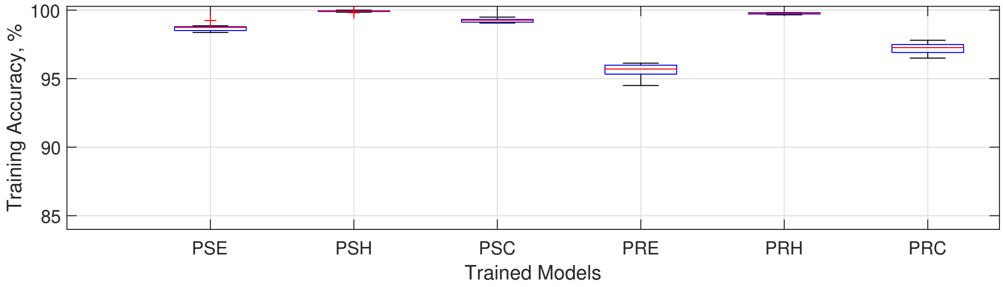


Figure B.1: Validation accuracy of all ‘one-size-fits-all’ models.

B.2. Subject Specific Models

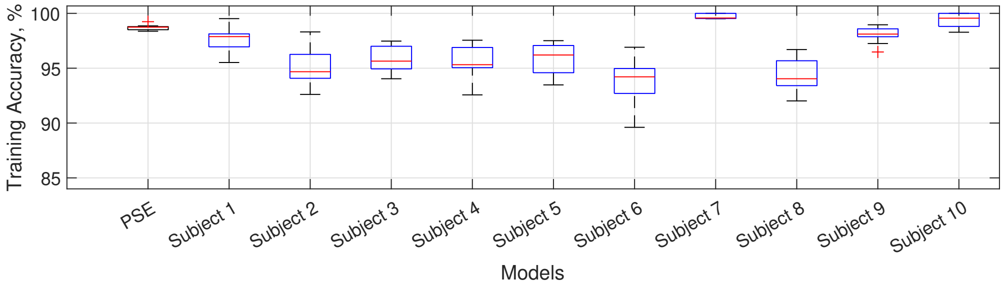


Figure B.2: Validation accuracy of MS-e ‘one-size-fits-all’ and subject specific models.

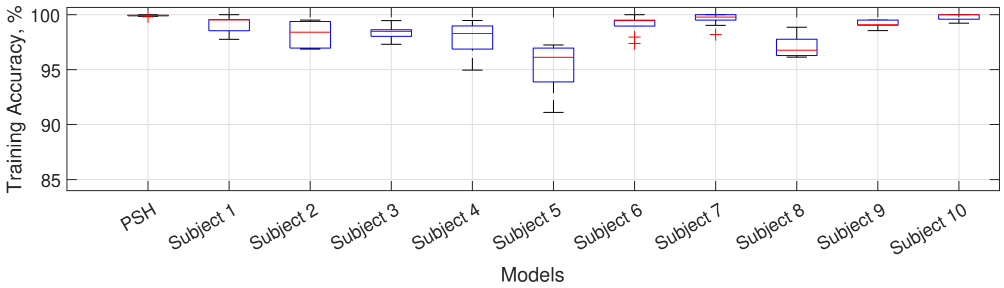


Figure B.3: Validation accuracy of MS-h 'one-size-fits-all' and subject specific models.

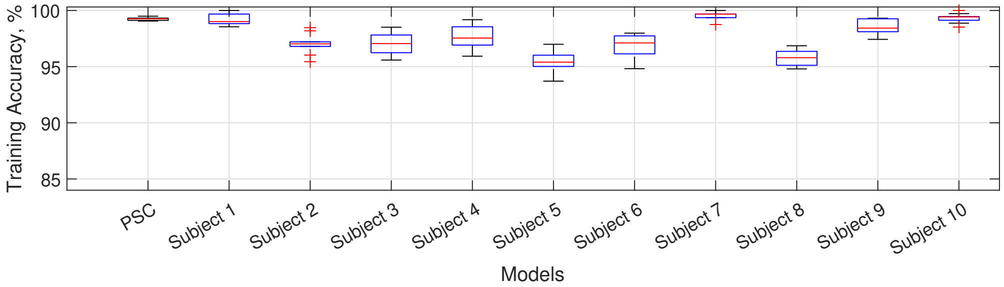


Figure B.4: Validation accuracy of MS-c 'one-size-fits-all' and subject specific models.

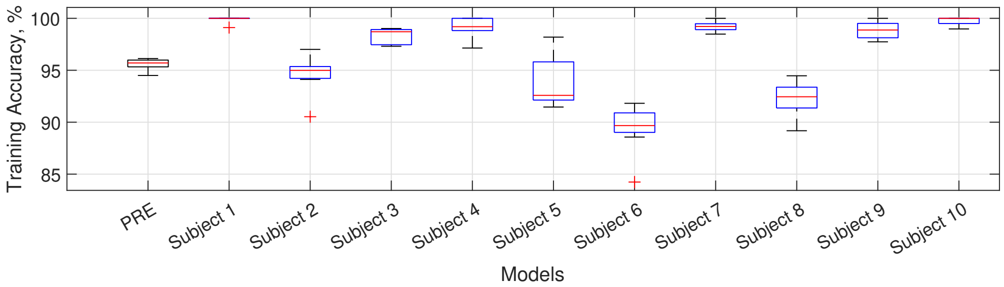


Figure B.5: Validation accuracy of MR-e 'one-size-fits-all' and subject specific models.

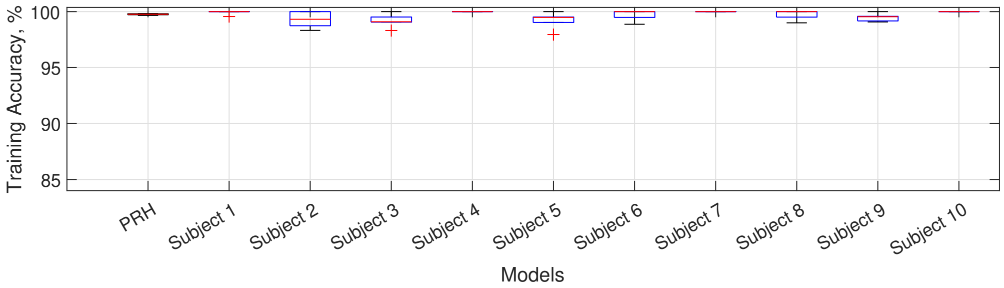


Figure B.6: Validation accuracy of MR-h 'one-size-fits-all' and subject specific models.

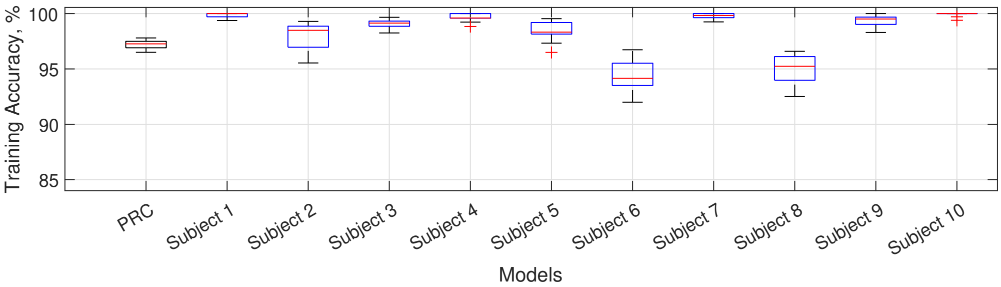


Figure B.7: Validation accuracy of MR-c ‘one-size-fits-all’ and subject specific models.

C
Confusion Matrix

The confusion matrices for ‘one-size-fits-all’ and subject-specific models are shown in this appendix.

C.1. ‘One-size-fits-all’ Models

Table C.1: Confusion matrix for MS-e ‘one-size-fits-all’ model.

Total Accuracy		**Predicted**	
98.76%		Normal	Distracted
Actual	Normal	98.58%	1.42%
		12322	178
	Distracted	1.06%	98.94%
		77	7184

Table C.2: Confusion matrix for MS-h ‘one-size-fits-all’ model.

Total Accuracy		**Predicted**	
99.94%		Normal	Distracted
Actual	Normal	99.87%	0.13%
		12639	16
	Distracted	0.00%	100.0%
		0	8129

Table C.3: Confusion matrix for MS-c ‘one-size-fits-all’ model.

Total Accuracy		**Predicted**	
99.21%		Normal	Distracted
Actual	Normal	98.78%	1.22%
		12447	154
	Distracted	0.35%	99.65%
		54	15239

Table C.4: Confusion matrix for MR-e ‘one-size-fits-all’ model.

Total Accuracy		**Predicted**	
95.26%		Normal	Distracted
Actual	Normal	96.27%	3.73%
		12289	476
	Distracted	5.74%	94.26%
		365	5992

Table C.5: Confusion matrix for MR-h ‘one-size-fits-all’ model.

Total Accuracy		**Predicted**	
99.79%		Normal	Distracted
Actual	Normal	99.64%	0.36%
		12594	45
	Distracted	0.06%	99.94%
		5	8563

Table C.6: Confusion matrix for MR-c ‘one-size-fits-all’ model.

Total Accuracy		**Predicted**	
97.12%		Normal	Distracted
Actual	Normal	96.09%	3.91%
		12186	496
	Distracted	1.85%	98.15%
		277	14672

C.2. subject-specific Models (MS-e)

Table C.7: Confusion matrix for MS-e subject-specific model (Subject 1).

Total Accuracy		Predicted	
97.70%		Normal	Distracted
Actual	Normal	97.71% 1195	2.29% 28
	Distracted	2.31% 20	97.69% 846

Table C.8: Confusion matrix for MS-e subject-specific model (Subject 2).

Total Accuracy		Predicted	
93.77%		Normal	Distracted
Actual	Normal	87.34% 1206	2.66% 33
	Distracted	9.79% 52	90.21% 479

Table C.9: Confusion matrix for MS-e subject-specific model (Subject 3).

Total Accuracy		Predicted	
94.87%		Normal	Distracted
Actual	Normal	87.99% 1268	2.01% 26
	Distracted	8.26% 55	91.74% 611

Table C.10: Confusion matrix for MS-e subject-specific model (Subject 4).

Total Accuracy		Predicted	
93.23%		Normal	Distracted
Actual	Normal	98.11% 1249	1.89% 24
	Distracted	11.66% 57	88.34% 432

Table C.11: Confusion matrix for MS-e subject-specific model (Subject 5).

Total Accuracy		Predicted	
95.81%		Normal	Distracted
Actual	Normal	86.28% 1243	3.72% 48
	Distracted	4.67% 55	95.33% 1123

Table C.12: Confusion matrix for MS-e subject-specific model (Subject 6).

Total Accuracy		Predicted	
90.66%		Normal	Distracted
Actual	Normal	97.24% 1164	2.76% 33
	Distracted	15.92% 71	84.08% 375

Table C.13: Confusion matrix for MS-e subject-specific model (Subject 7).

Total Accuracy		Predicted	
99.71%		Normal	Distracted
Actual	Normal	99.84% 1244	0.16% 2
	Distracted	0.42% 4	99.58% 953

Table C.14: Confusion matrix for MS-e subject-specific model (Subject 8).

Total Accuracy		Predicted	
93.31%		Normal	Distracted
Actual	Normal	96.23% 1200	3.77% 47
	Distracted	9.62% 58	90.38% 545

Table C.15: Confusion matrix for MS-e subject-specific model (Subject 9).

Total Accuracy		Predicted	
97.91%		Normal	Distracted
Actual	Normal	98.53% 1340	1.47% 20
	Distracted	2.71% 16	97.29% 575

Table C.16: Confusion matrix for MS-e subject-specific model (Subject 10).

Total Accuracy		Predicted	
99.33%		Normal	Distracted
Actual	Normal	99.51% 1224	0.49% 6
	Distracted	0.85% 9	99.15% 1047

C.3. subject-specific Models (MS-h)

Table C.17: Confusion matrix for MS-h subject-specific model (Subject 1).

Total Accuracy		Predicted	
99.03%		Normal	Distracted
Actual	Normal	99.35%	0.65%
		1218	8
	Distracted	1.30%	98.70%
		12	914

Table C.18: Confusion matrix for MS-h subject-specific model (Subject 2).

Total Accuracy		Predicted	
97.95%		Normal	Distracted
Actual	Normal	98.54%	1.46%
		1279	19
	Distracted	2.63%	97.37%
		16	593

Table C.19: Confusion matrix for MS-h subject-specific model (Subject 3).

Total Accuracy		Predicted	
98.18%		Normal	Distracted
Actual	Normal	99.28%	0.72%
		1233	9
	Distracted	2.92%	97.08%
		25	831

Table C.20: Confusion matrix for MS-h subject-specific model (Subject 4).

Total Accuracy		Predicted	
97.37%		Normal	Distracted
Actual	Normal	99.12%	0.88%
		1240	11
	Distracted	4.37%	95.63%
		30	656

Table C.21: Confusion matrix for MS-h subject-specific model (Subject 5).

Total Accuracy		Predicted	
95.18%		Normal	Distracted
Actual	Normal	95.35%	4.65%
		1250	61
	Distracted	4.98%	95.02%
		45	859

Table C.22: Confusion matrix for MS-h subject-specific model (Subject 6).

Total Accuracy		Predicted	
99.06%		Normal	Distracted
Actual	Normal	99.20%	0.80%
		1235	10
	Distracted	1.09%	98.91%
		7	638

Table C.23: Confusion matrix for MS-h subject-specific model (Subject 7).

Total Accuracy		Predicted	
99.57%		Normal	Distracted
Actual	Normal	99.75%	0.25%
		1208	3
	Distracted	0.61%	99.39%
		6	979

Table C.24: Confusion matrix for MS-h subject-specific model (Subject 8).

Total Accuracy		Predicted	
96.75%		Normal	Distracted
Actual	Normal	97.92%	2.08%
		1223	26
	Distracted	4.41%	95.59%
		30	650

Table C.25: Confusion matrix for MS-h subject-specific model (Subject 9).

Total Accuracy		Predicted	
99.21%		Normal	Distracted
Actual	Normal	98.88%	1.12%
		1241	14
	Distracted	0.45%	99.55%
		4	876

Table C.26: Confusion matrix for MS-h subject-specific model (Subject 10).

Total Accuracy		Predicted	
99.80%		Normal	Distracted
Actual	Normal	99.61%	0.39%
		1275	5
	Distracted	0.00%	100.0%
		0	1172

C.4. subject-specific Models (MS-c)

Table C.27: Confusion matrix for MS-c subject-specific model (Subject 1).

Total Accuracy		Predicted	
99.06%		Normal	Distracted
Actual	Normal	98.51% 1187	1.49% 18
	Distracted	0.38% 7	99.62% 1812

Table C.28: Confusion matrix for MS-c subject-specific model (Subject 2).

Total Accuracy		Predicted	
97.05%		Normal	Distracted
Actual	Normal	97.04% 1246	2.96% 38
	Distracted	2.95% 36	97.05% 1186

Table C.29: Confusion matrix for MS-c subject-specific model (Subject 3).

Total Accuracy		Predicted	
97.15%		Normal	Distracted
Actual	Normal	97.73% 1208	2.27% 28
	Distracted	3.44% 51	96.56% 1430

Table C.30: Confusion matrix for MS-c subject-specific model (Subject 4).

Total Accuracy		Predicted	
97.54%		Normal	Distracted
Actual	Normal	98.06% 1210	1.94% 24
	Distracted	2.97% 35	97.03% 1144

Table C.31: Confusion matrix for MS-c subject-specific model (Subject 5).

Total Accuracy		Predicted	
94.90%		Normal	Distracted
Actual	Normal	92.50% 1160	7.50% 94
	Distracted	2.69% 55	97.31% 1986

Table C.32: Confusion matrix for MS-c subject-specific model (Subject 6).

Total Accuracy		Predicted	
96.87%		Normal	Distracted
Actual	Normal	97.51% 1291	2.49% 33
	Distracted	3.78% 42	96.22% 1070

Table C.33: Confusion matrix for MS-c subject-specific model (Subject 7).

Total Accuracy		Predicted	
99.52%		Normal	Distracted
Actual	Normal	99.35% 1227	0.65% 8
	Distracted	0.31% 6	99.69% 1923

Table C.34: Confusion matrix for MS-c subject-specific model (Subject 8).

Total Accuracy		Predicted	
95.75%		Normal	Distracted
Actual	Normal	95.04% 1168	4.96% 61
	Distracted	3.55% 44	96.45% 1197

Table C.35: Confusion matrix for MS-c subject-specific model (Subject 9).

Total Accuracy		Predicted	
98.46%		Normal	Distracted
Actual	Normal	97.53% 1226	2.74% 31
	Distracted	0.61% 9	99.39% 1461

Table C.36: Confusion matrix for MS-c subject-specific model (Subject 10).

Total Accuracy		Predicted	
99.21%		Normal	Distracted
Actual	Normal	98.70% 1287	1.30% 17
	Distracted	0.27% 6	99.73% 2187

C.5. subject-specific Models (MR-e)

Table C.37: Confusion matrix for MR-e subject-specific model (Subject 1).

Total Accuracy		Predicted	
99.93%		Normal	Distracted
Actual	Normal	99.85% 1363	0.15% 2
	Distracted	0.00% 0	100.0% 785

Table C.38: Confusion matrix for MR-e subject-specific model (Subject 2).

Total Accuracy		Predicted	
92.67%		Normal	Distracted
Actual	Normal	96.95% 1238	3.05% 39
	Distracted	11.60% 53	88.40% 404

Table C.39: Confusion matrix for MR-e subject-specific model (Subject 3).

Total Accuracy		Predicted	
98.40%		Normal	Distracted
Actual	Normal	98.32% 1228	1.68% 21
	Distracted	1.53% 11	98.47% 709

Table C.40: Confusion matrix for MR-e subject-specific model (Subject 4).

Total Accuracy		Predicted	
98.92%		Normal	Distracted
Actual	Normal	99.53% 1262	0.47% 6
	Distracted	1.69% 10	98.31% 580

Table C.41: Confusion matrix for MR-e subject-specific model (Subject 5).

Total Accuracy		Predicted	
92.59%		Normal	Distracted
Actual	Normal	94.99% 1176	5.01% 62
	Distracted	9.82% 49	90.18% 450

Table C.42: Confusion matrix for MR-e subject-specific model (Subject 6).

Total Accuracy		Predicted	
84.27%		Normal	Distracted
Actual	Normal	96.06% 1171	3.94% 48
	Distracted	27.52% 131	72.48% 345

Table C.43: Confusion matrix for MR-e subject-specific model (Subject 7).

Total Accuracy		Predicted	
99.14%		Normal	Distracted
Actual	Normal	99.43% 1228	0.57% 7
	Distracted	1.16% 7	98.84% 598

Table C.44: Confusion matrix for MR-e subject-specific model (Subject 8).

Total Accuracy		Predicted	
92.28%		Normal	Distracted
Actual	Normal	91.88% 1176	8.13% 104
	Distracted	7.32% 50	92.68% 633

Table C.45: Confusion matrix for MR-e subject-specific model (Subject 9).

Total Accuracy		Predicted	
98.85%		Normal	Distracted
Actual	Normal	98.82% 1255	1.18% 15
	Distracted	1.12% 10	98.88% 883

Table C.46: Confusion matrix for MR-e subject-specific model (Subject 10).

Total Accuracy		Predicted	
99.78%		Normal	Distracted
Actual	Normal	99.69% 1302	0.31% 4
	Distracted	0.13% 1	99.87% 783

C.6. subject-specific Models (MR-h)

Table C.47: Confusion matrix for MR-h subject-specific model (Subject 1).

Total Accuracy		Predicted	
99.95%		Normal	Distracted
Actual	Normal	100.0% 1322	0.00% 0
	Distracted	0.10% 1	99.90% 981

Table C.48: Confusion matrix for MR-h subject-specific model (Subject 2).

Total Accuracy		Predicted	
99.29%		Normal	Distracted
Actual	Normal	99.26% 1214	0.74% 9
	Distracted	0.68% 7	99.32% 1025

Table C.49: Confusion matrix for MR-h subject-specific model (Subject 3).

Total Accuracy		Predicted	
99.25%		Normal	Distracted
Actual	Normal	99.19% 1222	0.81% 10
	Distracted	0.69% 6	99.31% 858

Table C.50: Confusion matrix for MR-h subject-specific model (Subject 4).

Total Accuracy		Predicted	
100.0%		Normal	Distracted
Actual	Normal	100.0% 1293	0.00% 0
	Distracted	0.00% 0	100.0% 730

Table C.51: Confusion matrix for MR-h subject-specific model (Subject 5).

Total Accuracy		Predicted	
99.41%		Normal	Distracted
Actual	Normal	98.81% 1247	1.19% 15
	Distracted	83.50% 0	71.06% 709

Table C.52: Confusion matrix for MR-h subject-specific model (Subject 6).

Total Accuracy		Predicted	
99.73%		Normal	Distracted
Actual	Normal	99.83% 1162	0.17% 2
	Distracted	0.37% 3	99.63% 801

Table C.53: Confusion matrix for MR-h subject-specific model (Subject 7).

Total Accuracy		Predicted	
100.0%		Normal	Distracted
Actual	Normal	100.0% 1255	0.00% 0
	Distracted	0.00% 0	100.0% 855

Table C.54: Confusion matrix for MR-h subject-specific model (Subject 8).

Total Accuracy		Predicted	
99.77%		Normal	Distracted
Actual	Normal	99.70% 1310	0.30% 4
	Distracted	0.15% 1	99.85% 660

Table C.55: Confusion matrix for MR-h subject-specific model (Subject 9).

Total Accuracy		Predicted	
99.54%		Normal	Distracted
Actual	Normal	99.30% 1273	0.70% 9
	Distracted	0.21% 2	99.79% 938

Table C.56: Confusion matrix for MR-h subject-specific model (Subject 10).

Total Accuracy		Predicted	
100.0%		Normal	Distracted
Actual	Normal	100.0% 1335	0.00% 0
	Distracted	0.00% 0	100.0% 1100

C.7. subject-specific Models (MR-c)

Table C.57: Confusion matrix for MR-c subject-specific model (Subject 1).

Total Accuracy		**Predicted**	
99.87%		Normal	Distracted
Actual	Normal	99.85%	0.15%
		1292	2
	Distracted	0.11%	99.89%
		2	1870

Table C.58: Confusion matrix for MR-c subject-specific model (Subject 2).

Total Accuracy		**Predicted**	
97.82%		Normal	Distracted
Actual	Normal	97.07%	2.93%
		1224	37
	Distracted	1.42%	98.58%
		21	1461

Table C.59: Confusion matrix for MR-c subject-specific model (Subject 3).

Total Accuracy		**Predicted**	
98.96%		Normal	Distracted
Actual	Normal	98.04%	1.96%
		1200	24
	Distracted	0.13%	99.87%
		2	1595

Table C.60: Confusion matrix for MR-c subject-specific model (Subject 4).

Total Accuracy		**Predicted**	
99.60%		Normal	Distracted
Actual	Normal	99.36%	0.64%
		1233	8
	Distracted	0.16%	99.84%
		2	1251

Table C.61: Confusion matrix for MR-c subject-specific model (Subject 5).

Total Accuracy		**Predicted**	
98.37%		Normal	Distracted
Actual	Normal	97.69%	2.31%
		1186	28
	Distracted	0.96%	99.04%
		11	1135

Table C.62: Confusion matrix for MR-c subject-specific model (Subject 6).

Total Accuracy		**Predicted**	
94.32%		Normal	Distracted
Actual	Normal	95.97%	4.03%
		1215	51
	Distracted	7.34%	92.66%
		89	1124

Table C.63: Confusion matrix for MR-c subject-specific model (Subject 7).

Total Accuracy		**Predicted**	
99.73%		Normal	Distracted
Actual	Normal	99.52%	0.48%
		1254	6
	Distracted	0.07%	99.93%
		1	1474

Table C.64: Confusion matrix for MR-c subject-specific model (Subject 8).

Total Accuracy		**Predicted**	
94.77%		Normal	Distracted
Actual	Normal	91.10%	8.90%
		1105	108
	Distracted	1.55%	98.45%
		21	1330

Table C.65: Confusion matrix for MR-c subject-specific model (Subject 9).

Total Accuracy		**Predicted**	
99.21%		Normal	Distracted
Actual	Normal	98.59%	1.41%
		1256	18
	Distracted	0.16%	99.84%
		3	1822

Table C.66: Confusion matrix for MR-c subject-specific model (Subject 10).

Total Accuracy		**Predicted**	
99.89%		Normal	Distracted
Actual	Normal	99.77%	0.23%
		1320	3
	Distracted	0.00%	100.0%
		0	1946

D
Performance 'One-size-fits-all' vs Individualised Model

The test accuracy of the pursuit and preview models are shown in Figure D.1 and Figure D.2, respectively. In these figures the test accuracy of both 'one-size-fits-all' and individualised models are shown.

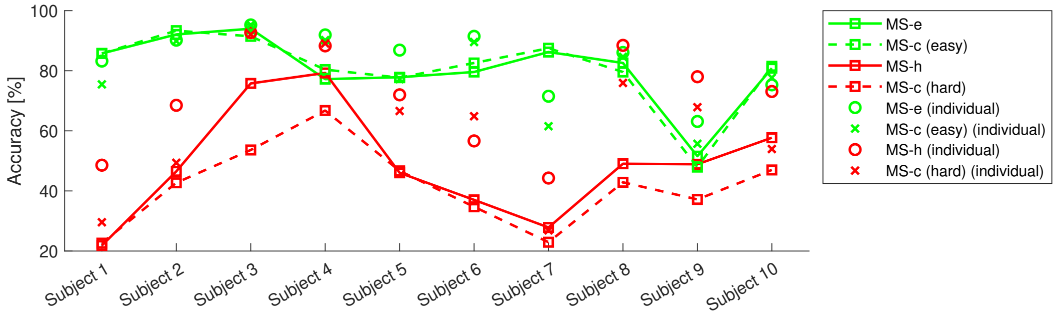


Figure D.1: Accuracy of all pursuit models when predicting samples from prompted runs.

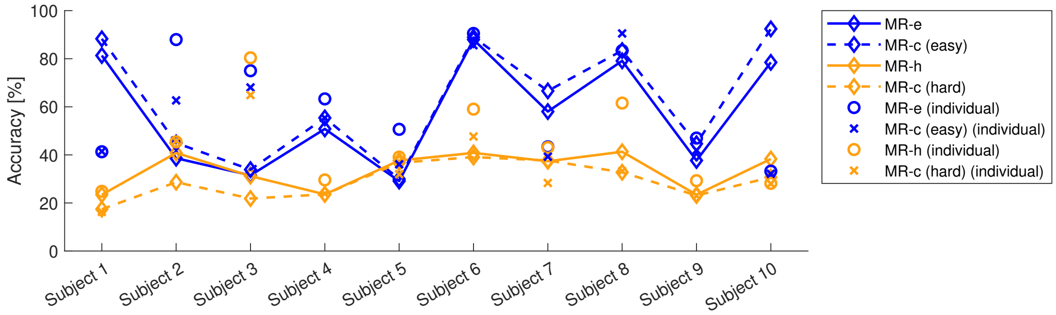


Figure D.2: Accuracy of all preview models when predicting samples from prompted runs.

E

Test Accuracy

In this appendix the accuracy of all models are presented tested on data collected from runs with prompted distractions. Models trained on detecting easy distractions are tested on its performance to detect easy distraction. The same logic holds for models used to detect hard distractions. And models trained on both easy and hard distractions are tested to detect easy and hard distractions separately.

E.1. MS-e

Table E.1: Average test accuracy of each subject for 'one-size-fits-all' MS-e model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	3	93.99%	530	20	64	18
2	2	92.09%	521	40	61	10
3	7	86.23%	486	71	59	16
4	1	85.76%	470	83	72	7
5	8	82.59%	470	93	52	17
6	10	80.85%	450	81	61	40
7	6	79.59%	463	104	40	25
8	4	77.22%	422	131	66	13
9	5	77.85%	437	125	55	15
10	9	51.58%	252	302	74	4

Table E.2: Average test accuracy of each subject for subject-specific MS-e models.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	3	95.25%	540	10	62	20
2	4	91.93%	519	34	62	17
3	6	91.46%	535	32	43	22
4	2	90.19%	508	53	62	9
5	5	86.87%	502	60	47	23
6	8	86.08%	494	69	50	19
7	1	83.23%	453	100	73	6
8	10	75.32%	381	150	95	6
9	7	71.52%	387	170	65	10
10	9	63.13%	328	226	71	7

E.2. MS-h

Table E.3: Average test accuracy of each subject for ‘one-size-fits-all’ MS-h model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	4	79.27%	412	106	89	25
2	3	75.79%	384	134	95	19
3	10	57.71%	162	172	145	53
4	8	49.05%	223	316	87	6
5	9	48.89%	202	305	107	18
6	2	46.52%	189	320	105	18
7	5	46.04%	209	326	82	15
8	6	37.03%	142	380	92	18
9	7	27.85%	39	451	137	5
10	1	21.84%	14	491	124	3

Table E.4: Average test accuracy of each subject for subject-specific MS-h models.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	3	92.72%	491	27	95	19
2	8	88.45%	482	57	77	16
3	4	88.29%	483	35	75	39
4	9	78.01%	391	116	102	23
5	10	73.10%	286	148	176	22
6	5	71.99%	378	157	77	20
7	2	68.51%	322	187	111	12
8	6	56.65%	255	267	103	7
9	1	48.58%	186	319	121	6
10	7	44.30%	144	346	136	6

E.3. MS-c (easy distraction)

Table E.5: Average test accuracy on easy distractions of each subject for ‘one-size-fits-all’ MS-c model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	2	93.35%	530	31	60	11
2	3	91.46%	526	24	52	30
3	7	87.50%	494	63	59	16
4	1	85.76%	474	79	68	11
5	6	82.59%	484	83	38	27
6	10	81.49%	466	65	49	52
7	4	80.38%	441	112	67	12
8	8	79.59%	453	110	50	19
9	5	77.69%	441	121	50	20
10	9	47.94%	230	324	73	5

Table E.6: Average test accuracy on easy distractions of each subject for subject-specific MS-c models.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	3	95.25%	539	11	63	19
2	4	90.19%	506	47	64	15
3	6	89.56%	523	44	43	22
4	8	84.65%	482	81	53	16
5	2	82.91%	462	99	62	9
6	10	77.69%	403	128	88	13
7	5	77.22%	438	124	50	20
8	1	75.47%	402	151	75	4
9	7	61.55%	322	235	67	8
10	9	55.70%	283	271	69	9

E.4. MS-c (hard distractions)

Table E.7: Average test accuracy on hard distractions of each subject for ‘one-size-fits-all’ MS-c model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	4	66.77%	321	197	101	13
2	3	53.64%	228	290	111	3
3	10	46.99%	125	309	172	26
4	5	46.68%	212	323	83	14
5	8	42.88%	180	359	91	2
6	2	42.72%	154	355	116	7
7	9	37.18%	114	393	121	4
8	6	34.73%	112	400	104	6
9	7	22.94%	6	484	139	3
10	1	22.63%	16	489	127	0

Table E.8: Average test accuracy on hard distractions of each subject for subject-specific MS-c models.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	3	92.09%	476	42	106	8
2	4	89.08%	470	48	93	21
3	8	75.95%	393	146	87	6
4	9	67.88%	313	194	116	9
5	5	66.61%	338	197	83	14
6	6	64.87%	308	214	102	8
7	10	53.96%	156	278	185	13
8	2	49.37%	196	313	116	7
9	1	29.59%	63	442	124	3
10	7	26.90%	29	461	141	1

E.5. MR-e

Table E.9: Average test accuracy of each subject for ‘one-size-fits-all’ MR-e model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	1	81.33%	456	98	58	20
2	6	87.97%	534	34	22	42
3	8	79.11%	479	78	21	54
4	10	78.48%	467	53	29	83
5	7	59.02%	320	236	53	23
6	4	50.79%	256	290	65	21
7	2	38.61%	192	376	52	12
8	9	37.66%	171	385	67	9
9	3	31.49%	120	428	79	5
10	5	29.11%	101	424	83	24

Table E.10: Average test accuracy of each subject for subject-specific MR-e models.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	6	90.51%	533	35	39	25
2	2	87.97%	514	54	42	22
3	8	83.39%	515	42	12	63
4	3	75.00%	417	131	57	27
5	4	63.29%	335	211	65	21
6	5	50.63%	240	285	80	27
7	9	46.99%	236	320	61	15
8	7	43.35%	210	346	64	12
9	1	41.30%	194	360	67	11
10	10	33.23%	122	398	88	24

E.6. MR-h

Table E.11: Average test accuracy of each subject for ‘one-size-fits-all’ MR-h model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	8	41.30%	168	357	93	14
2	2	40.82%	173	345	85	29
3	6	40.82%	151	351	107	23
4	10	38.29%	90	365	152	25
5	5	37.66%	144	373	94	21
6	7	37.34%	134	371	102	25
7	3	31.17%	88	431	109	4
8	4	23.58%	0	483	149	0
9	1	23.42%	55	479	93	5
10	9	23.42%	4	484	144	0

Table E.12: Average test accuracy of each subject for subject-specific MR-h model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	3	80.38%	436	83	72	41
2	8	61.55%	332	193	57	50
3	6	59.02%	269	233	104	26
4	2	45.41%	202	316	85	29
5	7	43.04%	159	346	113	14
6	4	29.59%	47	436	140	9
7	9	29.27%	50	438	135	9
8	5	39.08%	146	371	101	14
9	10	28.16%	1	454	177	0
10	1	24.84%	61	473	96	2

E.7. MR-c (easy distractions)

Table E.13: Average test accuracy on easy distractions of each subject for 'one-size-fits-all' MR-c model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	10	92.41%	491	29	93	19
2	6	88.92%	542	26	20	44
3	1	88.29%	506	48	52	26
4	8	83.39%	513	44	14	61
5	7	66.61%	371	185	50	26
6	4	55.38%	286	260	64	22
7	2	44.94%	230	338	54	10
8	9	44.62%	217	339	65	11
9	3	33.86%	138	410	76	8
10	5	29.91%	113	412	76	31

Table E.14: Average test accuracy on easy distractions of each subject for subject-specific MR-c model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	8	90.51%	508	49	64	11
2	6	85.60%	507	61	34	30
3	3	68.04%	362	186	68	16
4	2	62.66%	341	227	55	9
5	4	55.38%	288	258	62	24
6	9	41.93%	196	360	69	7
7	1	41.46%	190	364	72	6
8	7	39.24%	186	370	62	14
9	5	36.08%	136	389	92	15
10	10	32.12%	116	404	87	25

E.8. MR-c (hard distractions)

Table E.15: Average test accuracy on hard distractions of each subject for ‘one-size-fits-all’ MR-c model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	6	39.08%	134	368	113	17
2	7	37.66%	125	380	113	14
3	5	36.71%	132	385	100	15
4	8	32.75%	108	417	99	8
5	10	30.70%	19	436	175	2
6	2	28.64%	75	443	106	8
7	4	23.58%	0	483	149	0
8	9	23.10%	2	486	144	0
9	3	21.84%	25	494	113	0
10	1	17.41%	14	520	96	2

Table E.16: Average test accuracy on hard distractions of each subject for subject-specific MR-c model.

Rank	Subject	Accuracy	TP	FP	TN	FN
1	3	64.87%	317	202	93	20
2	6	47.63%	186	316	115	15
3	2	45.73%	189	329	100	14
4	5	31.96%	97	420	105	10
5	8	33.23%	122	403	88	19
6	7	28.32%	59	446	120	7
7	10	28.01%	0	455	177	0
8	4	23.89%	2	481	149	0
9	9	22.94%	1	487	144	0
10	1	16.30%	5	529	98	0

F

Receiver Operation Characteristics Curve

The Receiver Operation Characteristics (ROC) curve of the 6 classification models based on data of individual subjects are plotted in this appendix. Each figure is produced using 1 classification model, for example the MS-e model in Figure F.1. The colored lines show the ROC curves of individual subjects whereas the black line corresponds to data of all subjects.

F.1. Pursuit Models

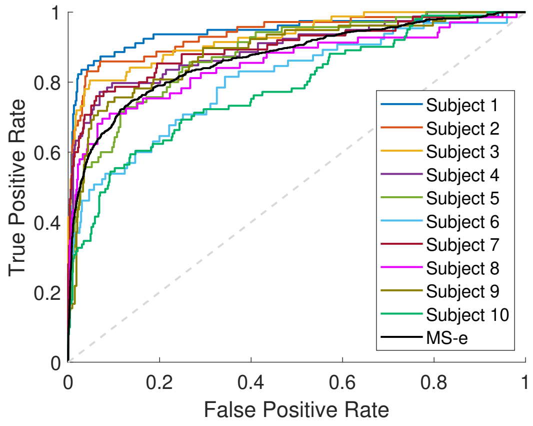


Figure F.1: ROC of the MS-e model and individual subjects for predicting easy distracted data.

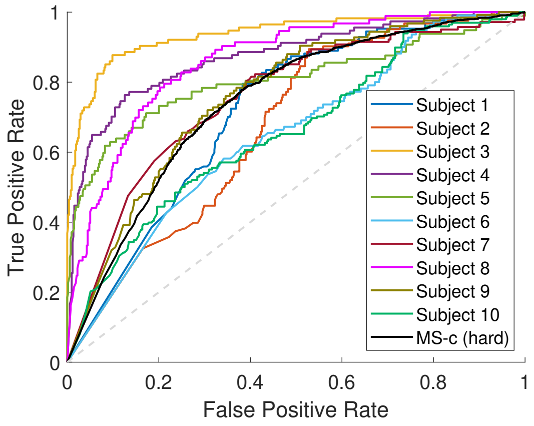


Figure F.4: ROC of the MS-c model and individual subjects for predicting hard distracted data.

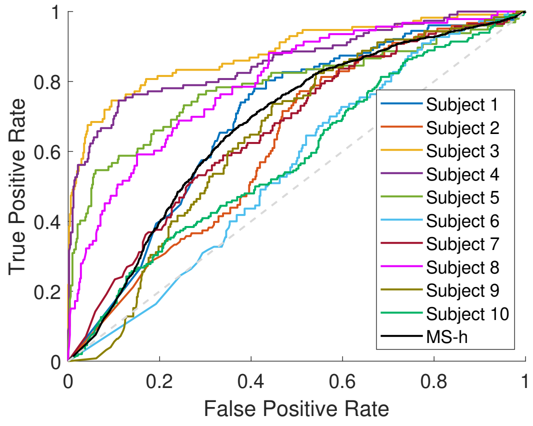


Figure F.2: ROC of the MS-h model and individual subjects for predicting hard distracted data.

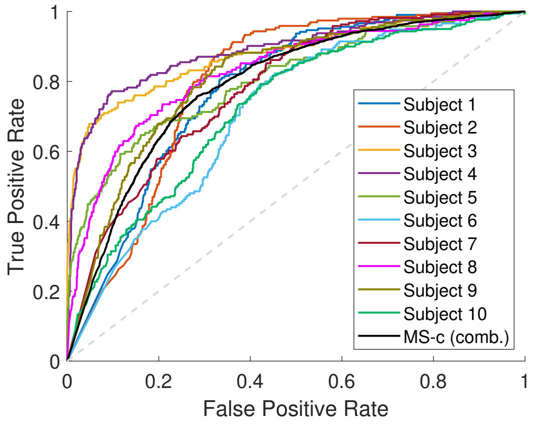


Figure F.5: ROC of the MS-c model and individual subjects for predicting easy and hard distracted data.

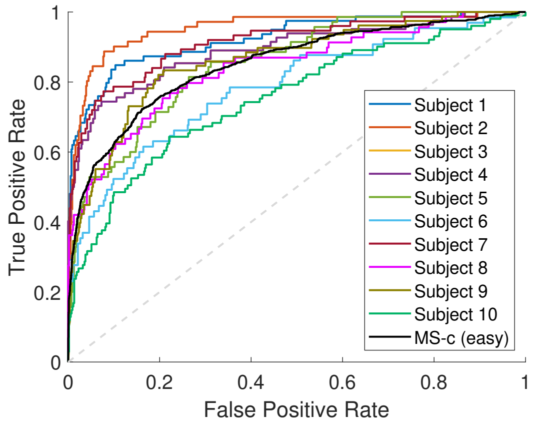


Figure F.3: ROC of the MS-c model and individual subjects for predicting easy distracted data.

F.2. Preview Models

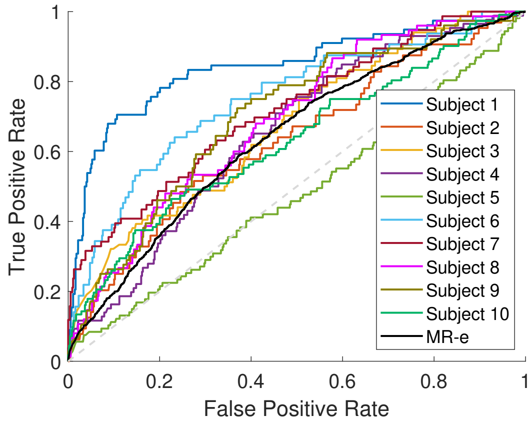


Figure F.6: ROC of the MR-e model and individual subjects for predicting easy distracted data.

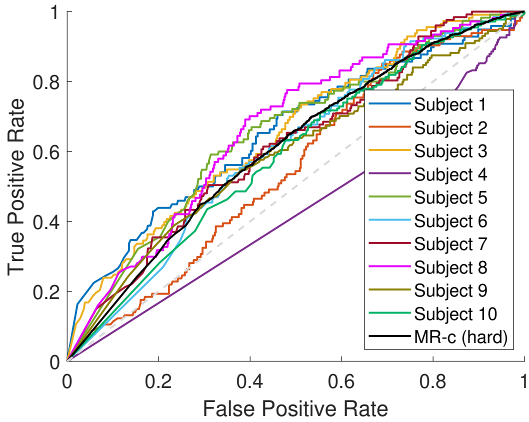


Figure F.9: ROC of the MR-c model and individual subjects for predicting hard distracted data.

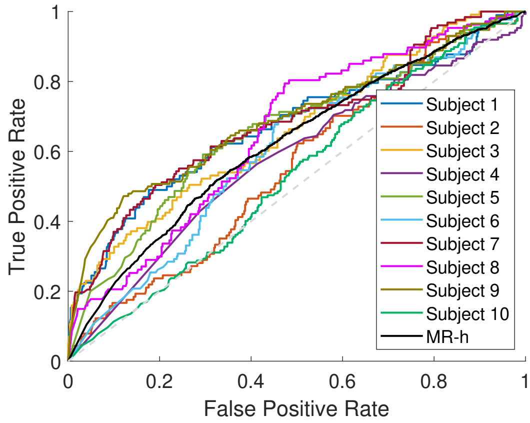


Figure F.7: ROC of the MR-h model and individual subjects for predicting hard distracted data.

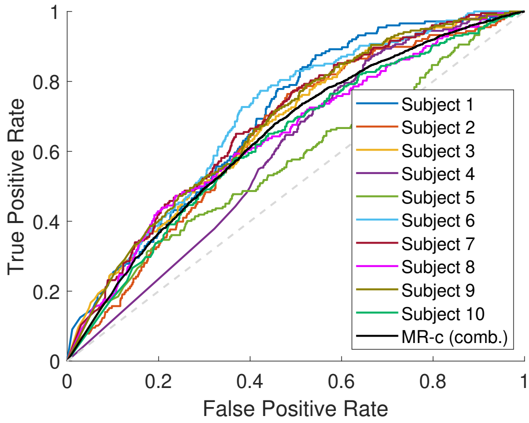


Figure F.10: ROC of the MR-c model and individual subjects for predicting easy and hard distracted data.

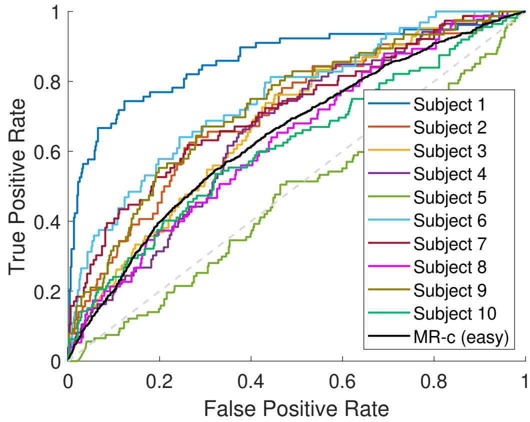


Figure F.8: ROC of the MR-c model and individual subjects for predicting easy distracted data.

G

RMS Error and Input (Samples)

The data from tracking runs with prompted distractions are analysed on tracking performance. Now the RMS error and input of every sample with distracted data has been plotted of every subject separately. For completeness the RMS error or input of the training data are also shown.

G.1. RMS error of samples

The RMS error for normal and distracted pursuit data are presented in Figure G.1 and Figure G.2, respectively. The results for normal and distracted preview data can be found in Figure G.3 and Figure G.4, respectively.

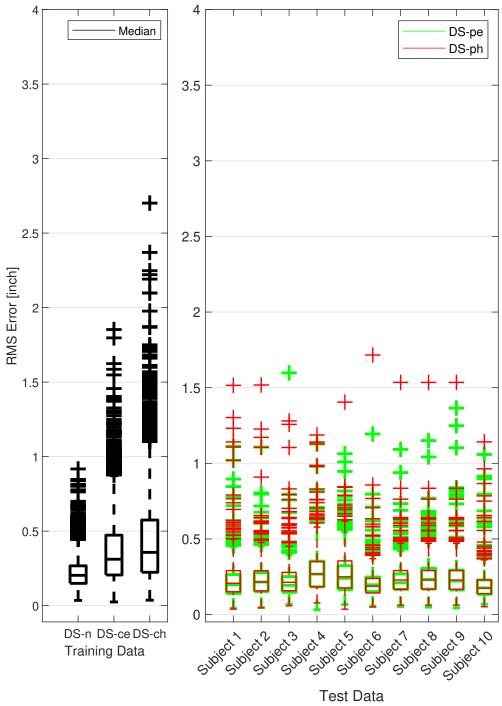


Figure G.1: RMS error of individual normal samples from pursuit tracking runs with prompted distractions.

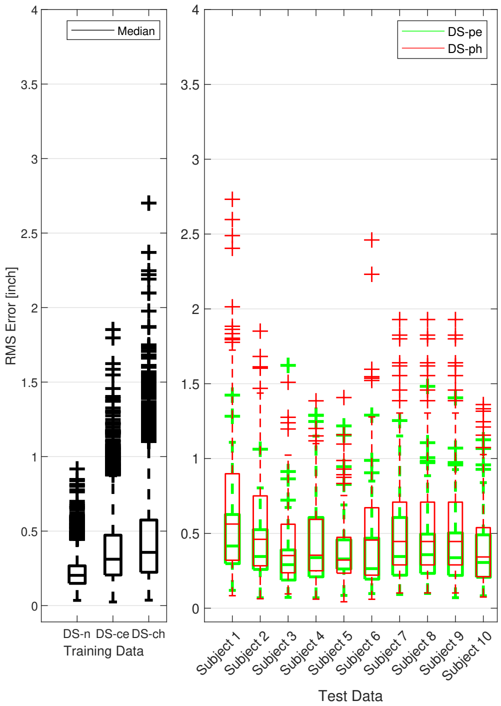


Figure G.2: RMS error of individual distracted samples from pursuit tracking runs with prompted distractions.

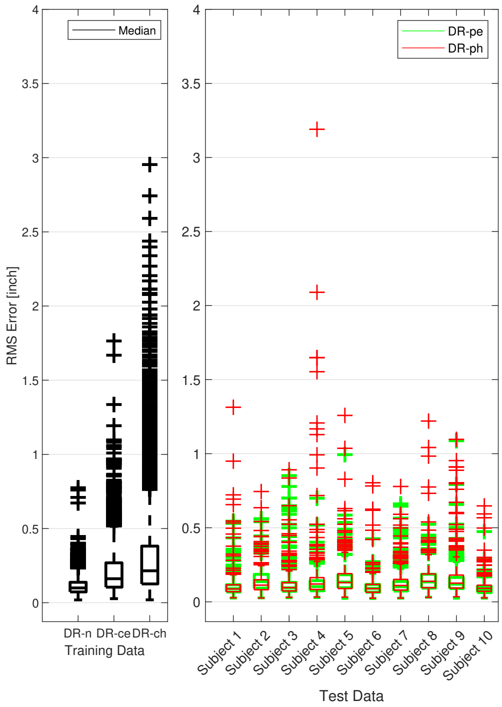


Figure G.3: RMS error of individual normal samples from preview tracking runs with prompted distractions.

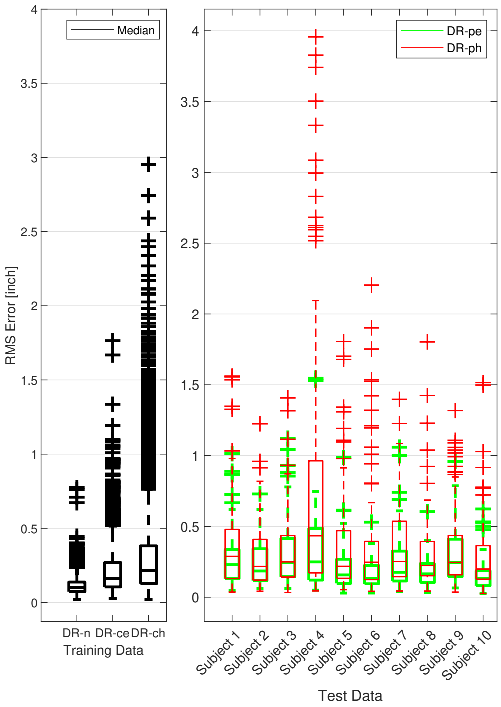


Figure G.4: RMS error of individual distracted samples from preview tracking runs with prompted distractions.

The distracted samples in Figure G.2 show in general a higher RMS error (0.15 on average) compared to samples from PSN data. This can also be concluded by comparing Figure G.1 with Figure G.2. The RMS error for normal samples in the test data is about the same as expected with a some outliers. Samples obtained from hard distractions show a large variation in RMS error, ranging from 0.1 to 2.0 inches. When it comes to samples from easy distractions the range is approximately 0.1 to 1.2 inch, thus overlapping with the data of hard distractions. However, comparing the median of each boxplot for every subject, the RMS error of hard distractions is still higher on average than for easy distractions. The same could be said for the results in Figure G.4 where preview data are shown.

G.2. RMS input of samples

The RMS input of the both normal and distracted samples for pursuit and preview data are plotted in Figure G.5, Figure G.6, Figure G.3, and Figure G.4.

The RMS input in Figure G.5 of DS-n, DS-ce, and DS-ch data show a decrease in input given by the human controller on average for distracted samples. When comparing the RMS input of normal data in Figure G.5 with the RMS input of distracted data in Figure G.6, a clear distinction in control input can be found.

In preview tracking tasks, the RMS input of DR-ce and DR-ch data are similar, as can be seen in Figure G.7. And by comparing the data in Figure G.7 and Figure G.8, the input when distracted is in general still lower compared to normal tracking data.

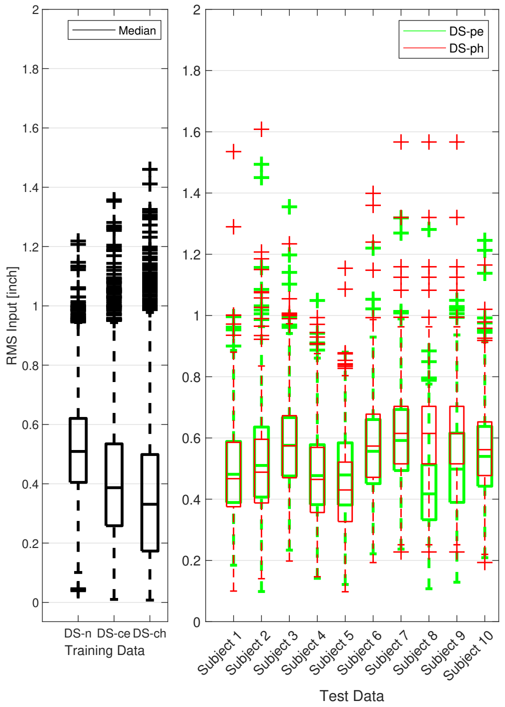


Figure G.5: RMS input of individual normal samples from pursuit tracking runs with prompted distractions.

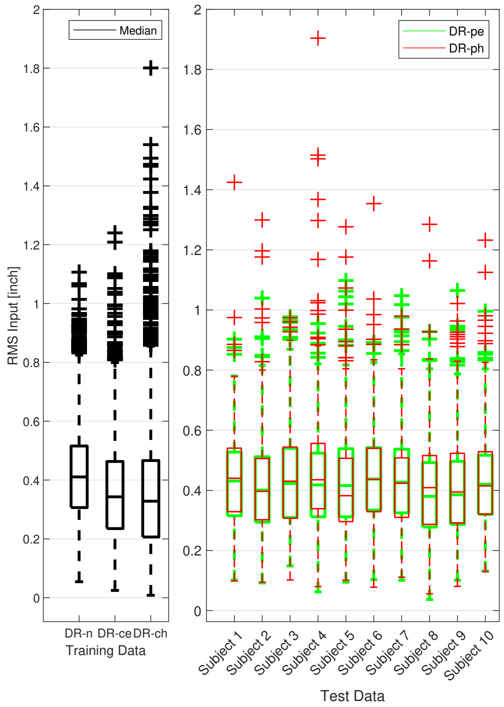


Figure G.7: RMS input of individual normal samples from preview tracking runs with prompted distractions.

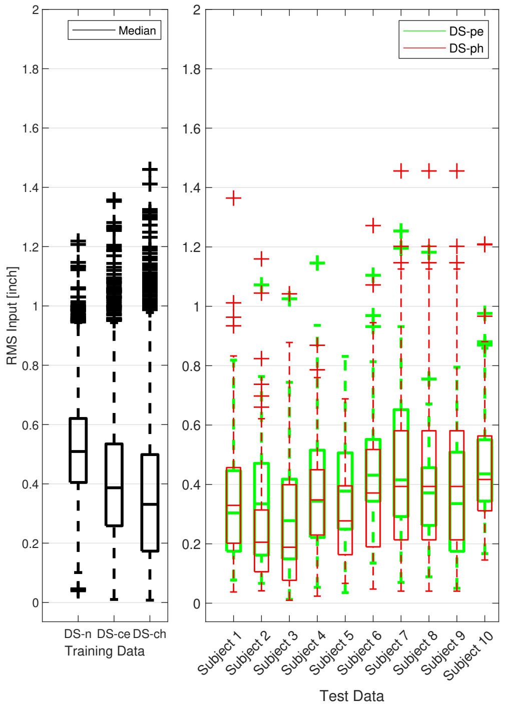


Figure G.6: RMS input of individual distracted samples from pursuit tracking runs with prompted distractions.

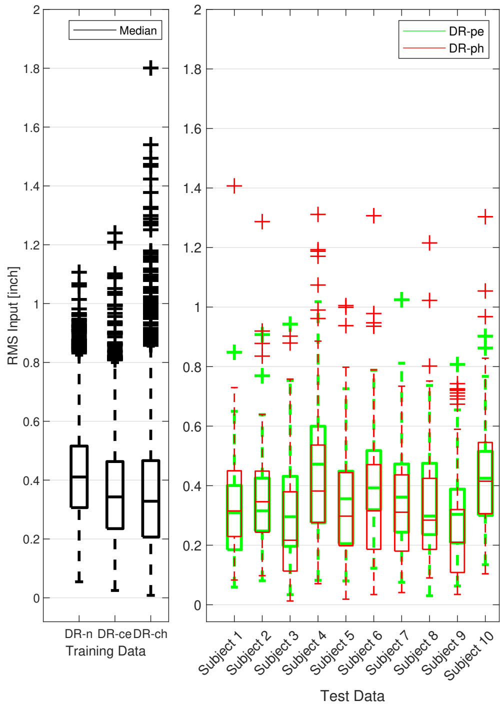


Figure G.8: RMS input of individual distracted samples from preview tracking runs with prompted distractions.

Figure G.1, Figure G.2, Figure G.3 and Figure G.4 show the RMS error of individual samples containing normal or distracted data per subject for pursuit or preview tracking tasks. As expected the error increases with increasing difficulty of the secondary task. And a clear distinction can be made in how large the tracking error is between normal and distracted samples for both pursuit and preview displays. The RMS error of the individual subjects are higher than DS-ce, DS-ch, DR-ce, and DR-ch runs because these boxplots also contain tracking data without distractions. When the models were trained, samples without distracted data were filtered out.

The RMS behaviour of the control input are presented in Figure G.5, Figure G.6, Figure G.3 and Figure G.8 for both sample types in pursuit and preview. When subjects are distracted in pursuit, it is more clear that the control input decreases compared to normal tracking conditions. In pursuit the subject does not know the future trajectory of the target signal and will therefore most likely give no control inputs. And for hard distractions the input is less compared to easy distractions, since it would take more time to find the target circle. In preview, the decrease in control input is less prominent. Subjects 4 and 10 give at times more control input, whereas this is lower or does not change for other subjects.

After analysing and interpreting the RMS error and input of the tracking data it is clear that subjects have different tracking strategies and performance.

H

Detection Probability

H.1. Subject 3, Run 1 (DS-ph)

The forcing function, system output and detection probability of Subject 3, Run 1 for the prompted hard conditions (DS-ph) can be seen in Figure H.1 and Figure H.2. The figures show that the MS-h model is able to detect all 6 prompted distractions in this particular run. The model is also able to detect a distraction despite of the small tracking error at approximately $t = 17s$.

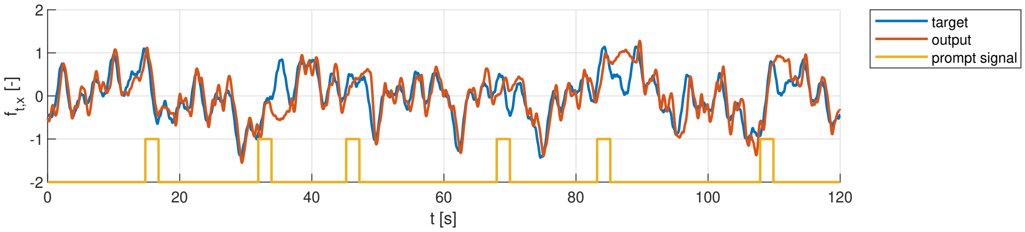


Figure H.1: Forcing function (target) and system output of a prompted (P) tracking run (subject 3, run 1) with hard distraction (MS-h model)

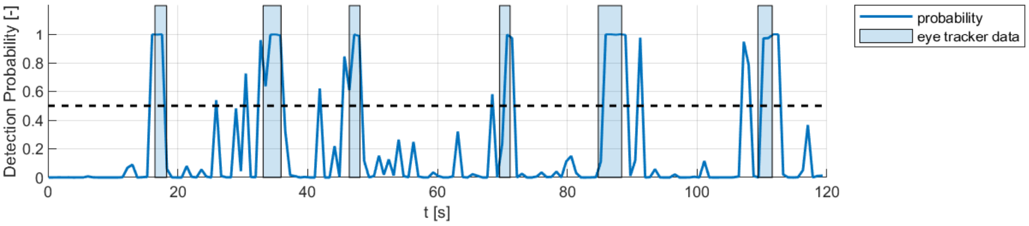


Figure H.2: Detection probability of a prompted (P) tracking run (subject 3, run 1) with hard distraction (MS-h model)

H.2. Cumulative Detection Probability

Figure H.3 and Figure H.4 show the averaged cumulative detection probability (N = 40) for pursuit and preview models, respectively. The highlighted blue areas are periods in which a prompt is shown at a random instant to the subject. Thus, it is not possible to reach an average detection probability of 1 since a prompt could appear at the start or at the end of a highlighted area in different tracking runs. For a good performing model it is expected that the average detection probability would be higher in the highlighted areas compared to periods in which no prompts would be shown.

The best performing model based on the test accuracy was the MS-e model and in Figure H.3 peaks and valleys can be seen alternating with the highlighted and non-highlighted areas. For bad performing models such as MR-h in Figure H.4, the average detection probability is nearly 1 along the entire tracking runs. This means that the model classifies the majority of both non-distracted and distracted samples as distracted.

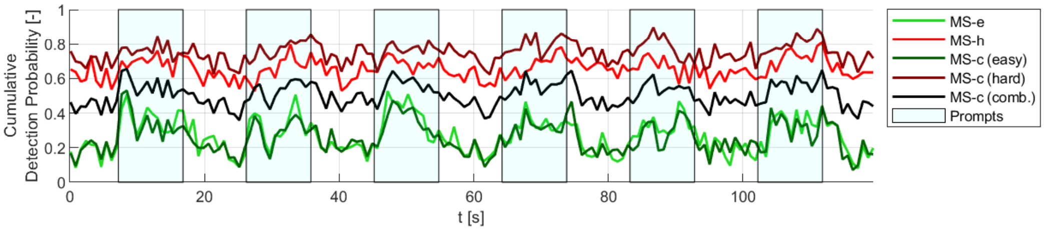


Figure H.3: Averaged cumulative detection probability for tracking runs with prompted distractions.

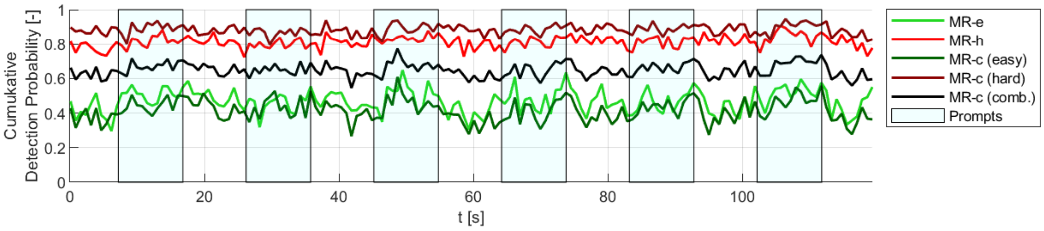


Figure H.4: Averaged cumulative detection probability for tracking runs with prompted distractions.

III

Preliminary Report (Already Graded)

1
Introduction

Travel by passenger vehicles is by far the deadliest transportation method on a per-mile basis in the United States [6]. Comparing this to air, rail, and bus travel, much lower death rates are found as can be seen in Figure 1.1. In the European Union similar results are found in the period of 2001-2002. The number of deaths per 100 million person kilometres for passenger vehicles is 0.7, for busses, railroad passengers, and scheduled airlines the numbers are 0.07, 0.035, and 0.025, respectively [5]. Cars are more accessible to the general public where the road safety depends on the individual traffic behaviour of users. People might over-speed, violate traffic rules more easily, fail to understand signs or are simply not paying attention. Distracted driving is one of the main risk factors in road accidents[1] for which a step towards a feasible mitigation solution is created in this report.

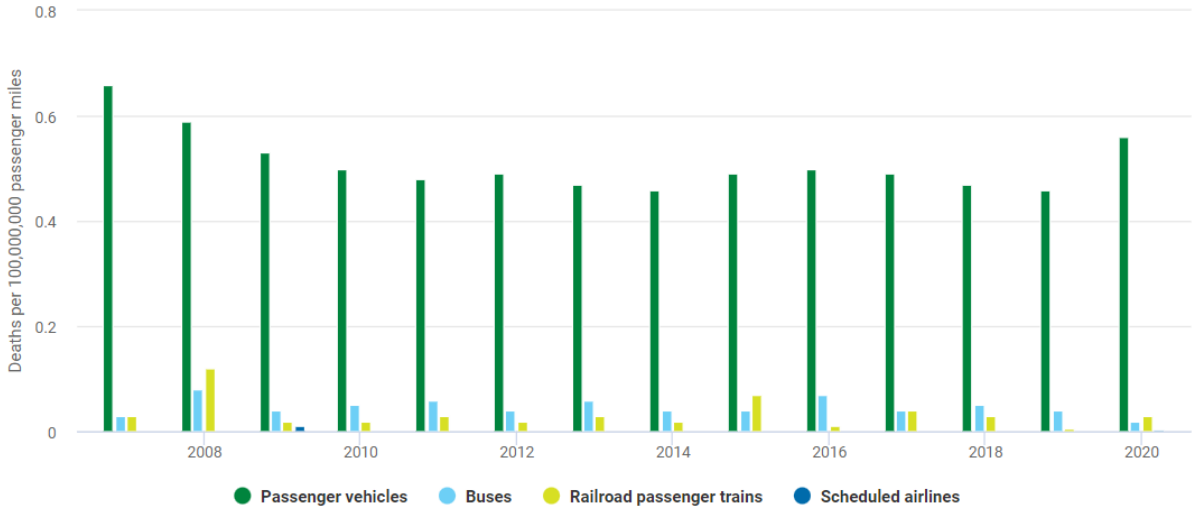


Figure 1.1: Passenger death rates in the United States between 2007 and 2022. (Deaths per 100,000,000 passenger miles)[6]

The goal of the research presented is to be able to detect when people are distracted in control tasks. In this day and age technological devices are everywhere, think of smartphones and in-vehicle information systems. Both are a contributor towards distracted driving since people have to take their eyes of the road and shift attention to somewhere else [50]. Creating a tool that can detect when people are distracted may help in contributing towards a safer road environment and reduce the number of traffic accidents [46]. Furthermore, such a tool can make decisions easier when determining who is liable in accidents in case there is hard evidence in the form of for example camera footage.

The purpose of this report is to gain insights in developing a tool for detecting distractions in human pilot tracking tasks using machine learning. Research was conducted after the successful application of

[1] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

machine learning models in this research area. This included classification models for human pilot skill [16] and human behaviour in tracking tasks with various display types [59]. On the basis of a literature study on topics such as driver distraction, task demand, and anomaly detection techniques, decisions are made on the approach of answering the main research question.

The structure of this report is as follows, Appendix 2 defines a definition of distraction based on related work such as research on driver distraction. In Appendix 3 methods for detecting distractions are discussed, covering both vision based and sensor based approaches using machine learning techniques. In order to detect distraction in manual tracking tasks, data of distracted people should be generated first. Therefore, Appendix 4 investigates how for instance secondary tasks can distract people. Furthermore, challenges in, and aspects of anomaly detection are elaborated upon in Appendix 5. Preliminary simulations are conducted in Appendix 6 together with a discussion of results, followed by the research plan in Appendix 7. Lastly, Appendix 8 holds the conclusion on the preliminary report regarding detecting distractions in human pilot tracking tasks.

2 Distractions

The definition of distraction given by Cambridge is “something that prevents someone from giving their attention to something else” [1]. A distraction could for example be a push notification alerting you to check your phone whilst working or when someone is talking to you during lectures when you only want to pay attention to the professor. There are many more examples that one could think of, but in the light of the current research problem, the focus is put on detecting distractions in manual control tasks. section 2.1 defines what distraction is in the field of research and section 2.2 presents chapter takeaways.

2.1. Definition of Distraction

A general concern for road safety are distractions and inattention of drivers. With the increase of cell phone usage by drivers and in-vehicle touchscreens, distractions are becoming a growing issue in many countries [41]. Studies have found that simulated driving tasks are compromised by tasks intended to replicate phone conversations, whether using hand-held or hands-free phones, and may be further compromised by the physical distraction of handling phones [33].

This has also led to more research on the topic of detecting distractions, mainly concerning road users such as car drivers. Examples of this can be found in the paper by Tango et al. [53] where real-time detection systems of driver distraction using machine learning are investigated due to the increase of in-vehicle information systems (IVIS). Furthermore, Torkkola et al. [54] explored options in detecting driver inattention by letting participants in the experiment perform secondary tasks.

In some published articles the word “inattention” is also used as a synonym for distraction even though there is a small difference in meaning. Inattention is defined as “failure to give attention” by Cambridge [2]. An example of inattention would be failing to pay attention to the driving task, perhaps due to fatigue. A difference between the definition of distraction and inattention is that there is a competing task removing your attention from the main tasks. Thus, distraction can be considered to be inattention, but not the opposite. It is therefore important to note the definition of the authors with respect to distraction and inattention in order to determine whether the research is important to ones own research problem.

Regan et al. has collected a sample of 14 definitions for driver distraction in which distraction is considered as a subset of inattention [45]. A few examples are:

- Diversion of attention from the driving task that is compelled by an activity or event inside the vehicle [55].
- A shift in attention away from stimuli that is critical for safe driving toward stimuli that are not related to safe driving [51].

[1] https://dictionary.cambridge.org/dictionary/english/distraction

[2] https://dictionary.cambridge.org/dictionary/english/inattention

- Any activity that takes the attention of a driver away from the task of driving [44].

In most definitions five elements of distraction are present, *sources, location of sources, intentionality, process* and *outcome* [45]. These are listed in Table 2.1 with examples for each element.

Table 2.1: Common Elements of Distraction Definitions and Examples of Each Element [45].

Source	**Location of Source**	**Intentionality**	**Process**	**Outcome**
Object	Internal activity (e.g., daydreaming)	Compelled by Source	Disturbance of control	Delayed response
Person	Inside Vehicle	Driver's choice	Diversion of attention	Degraded longitudinal and lateral control
Event	Outside Vehicle		Misallocation of attention	Diminished situation awareness
Activity				Degraded decision making
				Increased crash risk

In order for someone to be distracted there must be one or more *sources*, this could be a person talking in the car or some activity happening outside of the vehicle. For the general definition of distraction the *location* of the source does not matter. Furthermore, the driver could be compelled by a source in which he has to divert attention to it or the driver may willingly choose to be distracted, such as texting unnecessarily whilst driving. When a driver is distracted, there is normally a process tied to distributing the attention. This process could be a diversion or misallocation of attention. The outcome of being distracted mentioned in various definitions are for example a slower reaction time or a worse performance. However, defining specific outcomes are open to doubt since it may be subject to a particular combination of measures or events [45].

2.2. Chapter Takeaways

Due to an increase of mobile devices and in-vehicle information systems in cars, the risk of being distracted whilst driving has increased. This poses a danger to others and has led to more road traffic incidents. Extensive research has been performed on driver distraction, however in literature distraction has been given various definitions. As presented in Table 2.1, most definitions include the following elements:

- Source: something that distracts the driver.
- Location: the location of the source.
- Intentionality: why was the driver distracted?
- Process: what happens during the distraction.
- Outcome: the result of the distraction.

In this research, the subject will be distracted deliberately in order to collect data of human operators doing a control task. When designing the experiment, the various elements found in definition of distractions can help in forming a good task objective and experimental set-up.

3

Detecting Distractions

The ultimate goal is to be able to detect when a human operator is distracted during manual control tasks. When a driver is distracted, it creates a dangerous environment for other road users. This is unwanted and therefore a tool which can detect distractions can notify the driver when he is distracted. This chapter will present previous related work to detecting distractions of car drivers. section 3.1 presents current vision and sensor-based approaches, section 3.2 introduces information about variables and displays related to the tracking task. Finally, chapter takeaways are presented in section 3.3.

3.1. Current Approaches

As mentioned in Appendix 2, driver distraction is one main contributor to car accidents. In order to reduce such accidents, attempts have been made to detect driver distractions. This can be in the form of manual, visual or cognitive distraction and detection methods can vary from vision to sensor-based approaches or a combination of both [17]. In other research done about this topic, the distractions are generally provoked by introducing a secondary task besides the primary driving task [60][32][11][28]. A more detailed discussion is presented in Appendix 4. Whereas the current section discusses vision and sensor-based driver distraction detection methods in subsection 3.1.1 and subsection 3.1.2, respectively.

3.1.1. Vision-Based Approach

Vision-based detection methods generally require additional equipment such as video cameras compared to sensor-based approaches. A study by Miwata et al. for example detects distracted driving behaviour based on the driver's hand movements using computer vision [37]. But vision is not only limited to cameras, Thermal Imaging (TI) systems can also be used. The skin temperature can be measured using TI in the supraorbital region to determine participants mental activities. When participants were performing secondary tasks, considerable skin temperature increase could be observed due to altered blood supply to the supraorbital regions [60].

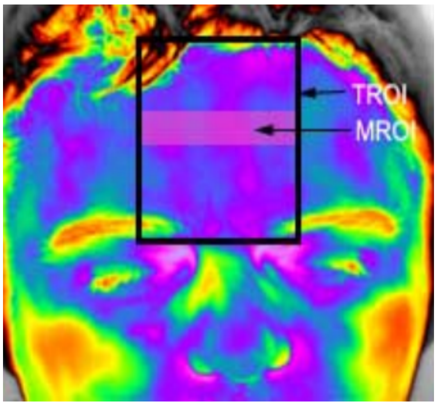


Figure 3.1: The supraorbital signal was extracted from the mean thermal footprint of the pink colored region [60].

On each thermal clip the region of interest is set such that the entire supraorbital region is taken into account as can be seen from Figure 3.1. Using this small area the mean temperature is computed, thus from a 2D thermal image a 1D signal is obtained. Due to noise and imperfections in the tracking process, a Fast Fourier Transform is applied to reduce the noise [57].

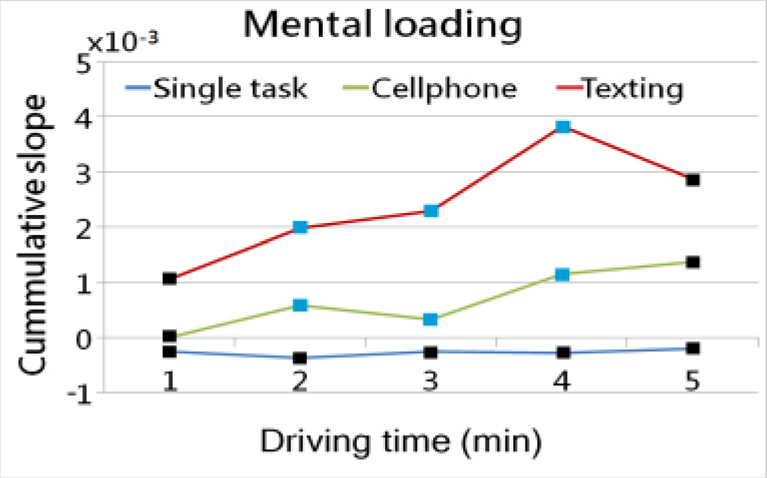


Figure 3.2: The graph illustrates the mental loading of the mean participant for the various segments. The black colored datapoints represent single tasking. The blue colored data-points represent dual tasking [60].

The results of the research presented in Figure 3.2 show that TI can distinguish between cognitive and visual distractions. The secondary tasks performed during the experiment were a talking task and texting task, both using cellphones, for simulating cognitive and visual distraction, respectively. It is clear that visual distractions cause a higher mental loading compared to cognitive distractions, since one has to look away from the road. This result can also be found in the driving performance measured during the experiment [60].

Another visual approach in detecting driver distraction is using eye-trackers (or head-trackers), which seems to be the most popular method [10][36][39][18][26][28][25][64]. Masood et al. uses Convolutional Neural Networks (CNN) to classify 10 classes of different driver behaviour, which are the most common activities which lead to distraction while driving [32]. With 1 class being labelled as safe driving and the remaining 9 as distracted driving. Sample images for each class are shown in Figure 3.3 to Figure 3.12.

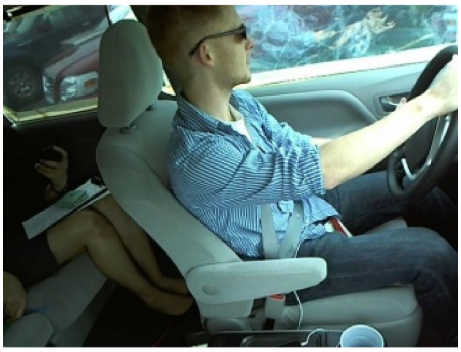


Figure 3.3: Safe driving.

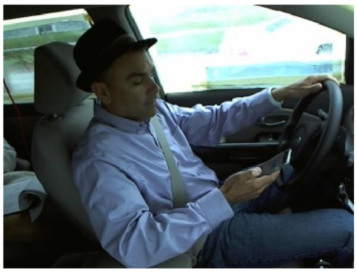


Figure 3.4: Texting using right hand.

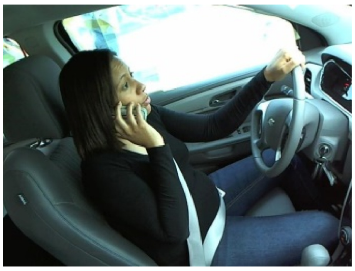


Figure 3.5: Calling using right hand.

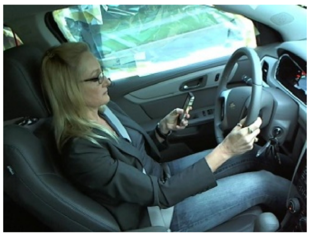


Figure 3.6: Texting using left hand.

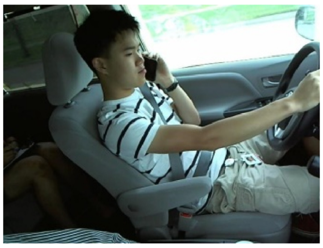


Figure 3.7: Calling using left hand.

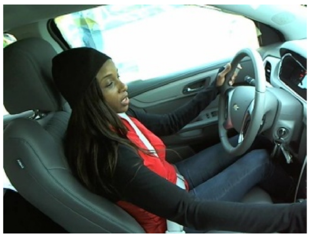


Figure 3.8: Operating the radio.

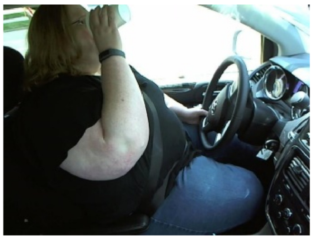


Figure 3.9: Drinking while driving.

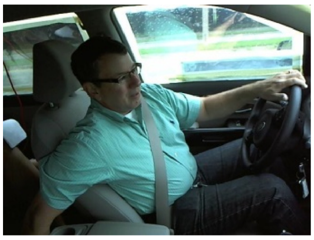


Figure 3.10: Reaching behind.

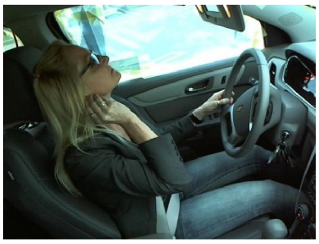


Figure 3.11: Doing makeup while driving.

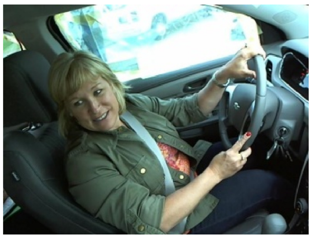


Figure 3.12: Talking to passenger.

The images of distracted people vary from actions such as talking on the phone in Figure 3.5 to putting on makeup in Figure 3.11. By defining classes which are connected to action also helps in detecting the cause of distraction. It is reported that the CNN models, VGG16 and VGG19, can classify the 10 classes with an accuracy of 99% [32]. However, it is difficult to know whether are all possible options of distracted driving behaviour is captured in the 9 most common classes for distracted driving.

Apart from only taking into account eye-movements, Liu et al. also captured the head movement of drivers to classify driver states: attentive or distracted [28]. The eye and head movement data consist of the number of blinks, blink frequency $[Hz]$ and duration $[s]$ as well as the head position $[m]$ and rotation in Euler angles $[rad]$. The head position is measured relative to a World Coordinate, where the origin is fixed at the middle point between two cameras of the eye tracker [28]. A semi-supervised approach is used to reduce labeling costs, and reaches an accuracy of 97.2%.

3.1.2. Sensor-Based Approach

The Sensor-based approach can make use of already existing sensors in the car, if necessary small adjustments could be made to these. The goal is to create a driving performance profile from which a neural network should be able to predict whether the driver is distracted or not.

Starting with a simple example, Ersal et al. [11] developed a model-based analysis for detecting driver distraction using only the pedal position and secondary tasks. It has the potential to use baseline driving characteristics to predict distracted behaviour. Participants were asked to drive in a simulator while they performed a visual secondary task in which they had to match icons. The actual pedal position and predicted pedal position were compared to each other in order to obtain the residuals of pedal positions as can be seen in Figure 3.13. The data is then used to train a Support Vector Machine (SVM) model which is able to produce a probability of distraction shown in Figure 3.14.

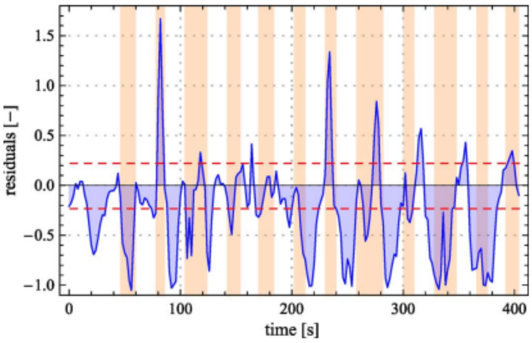


Figure 3.13: Residuals of pedal position while driving with secondary tasks. The shaded rectangles are the time windows from assignment to completion of the task. The dashed lines mark the 2σ interval [11].

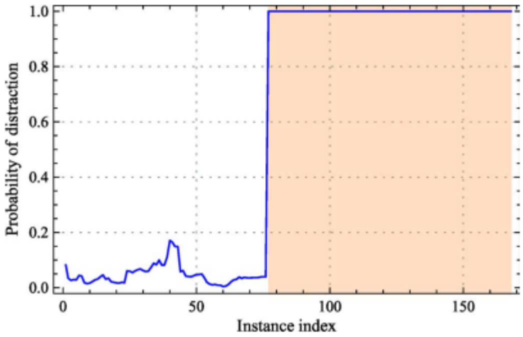


Figure 3.14: Probability of distraction for test instances, as predicted by the SVM. The shaded region indicates the instances from driving with secondary tasks, whereas the instances in the clear region are from normal driving [11].

The trained model is applied to the pedal position of each participant's experiment data from which an classification accuracy is obtained. Figure 3.15 shows the accuracy obtained in [11] for 16 participants. It is noticeable that some participants have low accuracy numbers, this means that for some participants it is difficult to differentiate between normal and distracted behaviour. Therefore, the accuracy could also be used as a metric to define how difficult a task is. A general model which detects driver distraction might thus be difficult to develop. However, more sensors and driving parameters can be used and personalised models can be created.

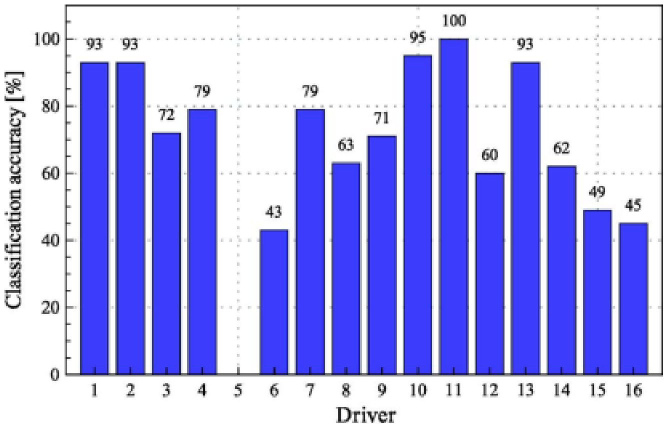


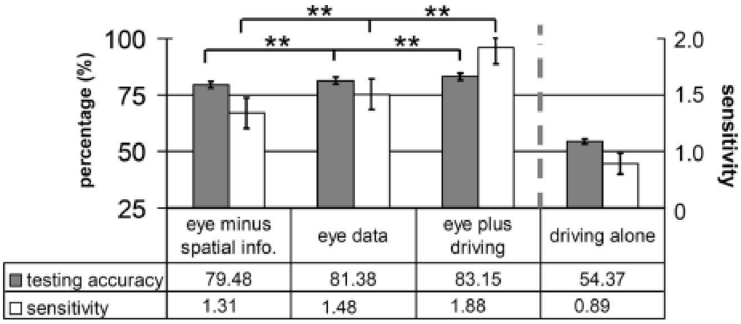
Figure 3.15: Classification accuracy across drivers using the moving average and standard deviation of residuals [11].

Other sensor features that can be used are for example the steering wheel of the car, speed and lane keeping performance. Table 3.1 shows common driving performance parameters that are used in research about detecting driver distractions. The most common feature is the steering wheel from which different steering characteristics such as the steering wheel angle, steering rate, or steering entropy. The steering entropy measures how consistent or random the steering wheel angle is in a certain condition compared to baseline driving [20]. This can be calculated from a time-series history of steering angle data [42]. Li et al. [23] studied distraction detection techniques using multiple features as presented in Table 3.1 and was able to reach an overall accuracy of 95% using SVM. A better result compared to the findings of Ersal et al. which only used the pedal position. Other researchers in Table 3.1 using multiple features were also able to produce high accuracy values.

Table 3.1: Accuracy and common features used in driver distraction detection following a sensor based approach.

	Pedal Position	Steering Wheel	Range/TTC	Speed	Lane Offset	Accuracy
Ersal et al. [11]	x					73.1 %
Li et al. [23]		x	x	x		95.0 %
Tango et al. [53]	x	x	x	x	x	95.3%
Aksjonov et al. [3]				x	x	86.5 %
Torkkola et al. [54]	x	x			x	92.7%
Nakano et al. [38]	x	x		x		86.0%

Lastly, Liang et al. combines eye-movement tracking with the driving performance of drivers when they are distracted cognitively [24]. The driving features taken are the standard deviation of the steering wheel position and lane position as well as the mean of the steering error. The eye data includes features such as pursuit duration, distance, direction and speed, the blinking frequency and eye fixation position and duration. Figure 3.16 shows the results of SVM models. Driving data alone demonstrates worse accuracy when compared to any combination including eye data.



	eye minus spatial info.	eye data	eye plus driving	driving alone
■ testing accuracy	79.48	81.38	83.15	54.37
□ sensitivity	1.31	1.48	1.88	0.89

Figure 3.16: SVM testing accuracy and sensitivity for the feature combinations [25].

3.2. Tracking Task

For detecting distractions in manual control tasks, a tracking task could be performed in conjunction with a secondary task [11]. A tracking task is a control task in which a human operator (HO) tracks a target with the controlled element (CE). The target, $f_t(t)$ is tracked by generating a control input $u(t)$ while the CE is perturbed by disturbance $f_d(t)$ [7]. The goal is to minimise the tracking error, $e(t)$:

$$e(t) = f_t(t) - x(t), \tag{3.1}$$

where $x(t)$ is the CE output.

The classical tracking task can be displayed on a screen with 3 different possible layouts, a compensatory, pursuit and preview layout. The pursuit and preview display show the target signal whilst the compensatory display only shows a reference point. The difference between the pursuit and preview display is the look ahead time of, τ_p.

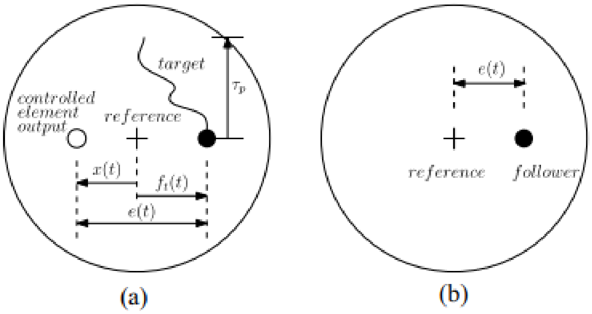


Figure 3.17: Layouts of pursuit/preview (a) and compensatory (b) displays [7].

Prior work done on tracking tasks involving machine learning models are for example classifying human pilot skill levels [16] and classifying human control behaviour based on display types [59][19]. In these studies various combinations of variables have been considered. Results show that for different research purposes, different combinations of variables reach the highest obtained accuracy. For classifying pilot skill levels the variables $u, \dot{u}, e$, and $\dot{e}$ were used, reaching an accuracy of 92% using a ResNet CNN architecture [16]. Whereas for the other research question the variables $u, \dot{u}, e, \dot{e}, x$, and $\dot{x}$ were used. The highest accuracy turned out to be 93%, however a different neural network architecture had been used, namely InceptionTime [59]. The use of different NN architectures could be an alternate explanation for the use of different variables.

3.3. Chapter Takeaways

Researchers are developing methods to detect distractions in an attempt to reduce traffic accidents caused by distracted drivers. Various approaches have been studied, varying from vision to sensor-based approaches.

Vision-based approaches typically require additional equipment such as eye-trackers or thermal imaging cameras. These methods can be costly and can form an obstruction for drivers. Besides these downsides, promising results have been achieved with neural network models reaching accuracies of up to 99%.

Sensor-based approaches on the other hand can make use of readily available sensors in the car such as the steering wheel or pedals. In most cases multiple sensor features have to be used to achieve good results. Furthermore, combining of sensor features and eye-trackers result in a higher accuracy.

The most straightforward method in this research is a sensor-based approach, since the goal is to detect distractions in a manual control task. The difference between driving a car and the tracking task is the importance of being aware of the surrounding environment where a vision-based approach may be interesting to track head or eye movements. However, in a tracking task the focus is on the screen showing the target and the controlled element. Furthermore, a secondary task (distraction) has to be implemented in to create distracted data in the primary task. The sensor based approach can make use of different features such as the error e, input u and output x collected from the tracking task.

4

Task Demand

In order to produce data with distracted people that can be used in developing a tool used to detect distractions, people need to be distracted first. This can be done by introducing a secondary task next to the primary task, which in this project is the tracking task. For this purpose, literature is gathered about task load. Information about resource allocation is presented in section 4.1. Examples of secondary tasks in other research are given in section 4.2. Furthermore, information related to the tracking task and chapter takeaways can be found in section 4.3 and section 4.4, respectively.

4.1. Multiple Resource Theory

Multitasking can increase a person's workload depending on the nature of the task. Resources have to be dedicated to each task in for example cognitive or visual terms. Therefore, Wickens created a 4 dimensional model called the Multiple Resource Theory (MRT), shown in Figure 4.1, used to determine the performance of humans when resources need to be shared between concurrent tasks [62]. The 4 dimensions are:

1. Processing stages (perception/cognition and response activity),

2. Processing Codes (spatial or verbal activity),

3. Perceptual Modalities (visual or auditory inputs), and

4. Visual Processing (focal or ambient vision).

According to the model, the execution of a task worsens when dimensions are shared between concurrent tasks. For example, a visual-perception and auditory-cognitive task do not have any common dimensions, leading to little interference. However, a visual-perception and auditory-perception task do share 1 dimension and are thus prone to greater interference.

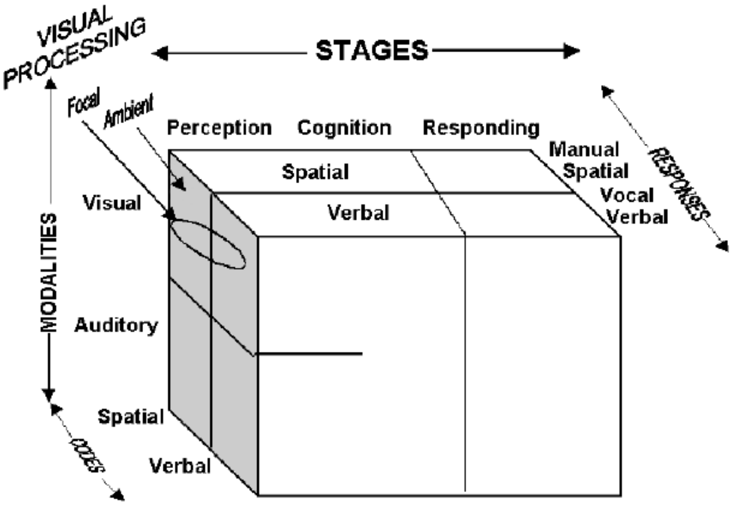


Figure 4.1: Dimensional representation of multiple resources [62].

In the *stages of processing* the interference depends on human perception/cognition or the response a human decides to take and how to execute it. The MRT considers perceptual and cognitive activity to share the same resources, whilst response activity has its own resources available. The reason is that during multitasking, speech and motor activity (responses) are often controlled in the frontal regions of the brain (in particular, the central sulcus) while perceptual and language comprehension tend to be undertaken in the posterior section of the central sulcus [29]. Furthermore, the prediction is made that the interference is high between resource demanding perceptual/cognitive tasks involving working memory to store information [61]. For example performing the following two tasks at the same time: understanding speech while rehearsing a speech.

The *processing codes* refer to the spatial and verbal activity that are separate and different from the stages of processing, perception, cognition and responses. The separation of resources could be seen as an explanation for low task interference when giving manual and verbal responses simultaneously. Typical spatial or verbal activities are for example a tracking task or speaking, respectively.

Modalities are considered to be the inputs of the system, how humans perceive a task. This can either be perceived visually or auditory. Once again, cross-modal time sharing such as visual-auditory task cause more interference compared to intra-modalities such as two visual tasks [29]. The actions considered to be a visual or auditory modality are for example looking at the environment or listening to instructions, respectively.

Finally, *visual processing* is the last dimension in the MRT model. There are two visual processing types, focal and ambient vision. Focal vision is used for capturing details, object/pattern recognition and other high acuity tasks. The human central visual field, in which we can see sharp (i.e., foveal vision), is very small compared to the peripheral vision which is very large. peripheral vision is linked to ambient vision, which helps in getting a sense of orientation in the environment [29].

4.2. Secondary Tasks

A notable cause of distractions during driving are secondary tasks, these can vary from reading a message on the phone to talking to the person in the passengers seat. The introduction of a new task leads to re-allocation of resources from the primary driving task to the new secondary task. This can cause an overall degraded performance or adaptive behaviour in completing both tasks. Kimura et al. studied the driver's attentional resource allocation to visual, cognitive and response processing by measuring physiological measures [18]. A slalom course with four driving conditions, speed (fast & slow) and path width (narrow & wide) are introduced to the participant. The results mainly showed changes in resources allocated to cognitive and visual processing for the different conditions with little change in response processing. Research performed on detecting driver distraction also includes a secondary task mainly focused on either introducing a cognitive or visual load.

4.2.1. Cognitive Tasks

Secondary cognitive tasks in driving experiments are used to induce a mental load on the driver. This can be done in many ways and for a variety of reasons. For example, measuring the driving performance or to detect and prevent distractions. The HASTE (Human Machine Interface and the Safety of Traffic in Europe) project explored the relationships between task loads and risks in the context of safety critical driving scenarios by employing secondary tasks [47].

The cognitive task used in the HASTE project was based on the visual Continuous Memory Task mentioned by Veltman [58]. Since the task initially required visual stimulation, thus inducing visual loads, it has been modified to an auditory task. The new task called Auditory Continuous Memory Task (ACMT) would only create an additional cognitive load on the participants [13] [10]. The goal is to remember the amount of times a target has been heard in a sequence of non-target sounds. An example sequence is shown in Figure 4.2 where the letters A, B, C, and D represent target sounds and X is considered to be a non-target sound.

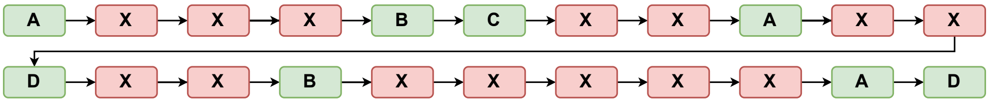


Figure 4.2: Auditory Continuous Memory Task example sequence with 4 target sounds A, B, C, and D.

It should be noted that participants would have to keep track of multiple counts, since each target sound is counted separately. The correct result from Figure 4.2 would therefore be A = 3, B = 2, C = 1, and D = 2. Furthermore, the difficulty of the task can be adjusted by increasing or decreasing the number of target sounds.

Another cognitive load task is the auditory-nonverbal version of the n-back task [39] [40]. A sequence of numbers between 0 and 9 are played through an audio device. The goal is to detect the instance at which the same number is played n numbers ago. An example of the task is shown in Figure 4.3.

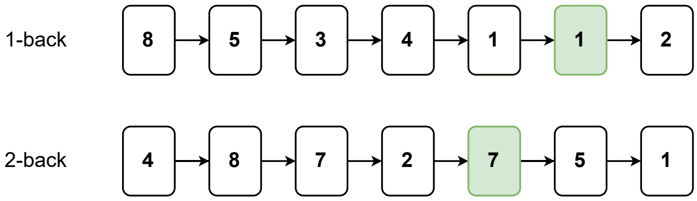


Figure 4.3: Examples of the n-back task, showing variations of n = 1 (top) and n = 2 (bottom).

In case of the 1-back task, the participant should notify the experiment instructor when the highlighted number 1 has been heard, since the previous number was also 1. For the 2-back task this is the case for the highlighted number 7, since 2 numbers ago number 7 was heard.

Other simple cognitive tasks include having a conversation with someone on the phone [60] or answering math questions which can vary in difficulty [34].

4.2.2. Visual Tasks

Secondary visual tasks are meant to shift someones attention away from, for example the primary driving task, in a visual manner. In current distraction detection approaches eye-trackers are often used to help determining whether someone is distracted as explained in subsection 3.1.1. Furthermore, eye-trackers can also be used for labelling or confirmation purposes. Since driving is mainly a visual task where it is important to observe the dynamic environment, it is easy to employ a secondary visual task to take someone's attention away from the road and to distract people.

In the HASTE project secondary visual tasks were also employed to study the effects increased of visual load on driving behaviour [13]. The task design is based on Treisman's feature integration theory [56] which states that the speed at which a visual target is identified within a display is affected by its visual similarity to other objects in the display [10]. When multiple visual characteristics, such as shape and colour, have to be distinguished from each other the reaction time increases. Therefore, the difficulty of the task can be altered by implementing multiple shapes, colours, orientation, etc.

The visual task in the HASTE project is the arrow task in which the participant has to determine whether an arrow pointing upwards was present in a matrix filled with arrows. An example can be seen in Figure 4.4.

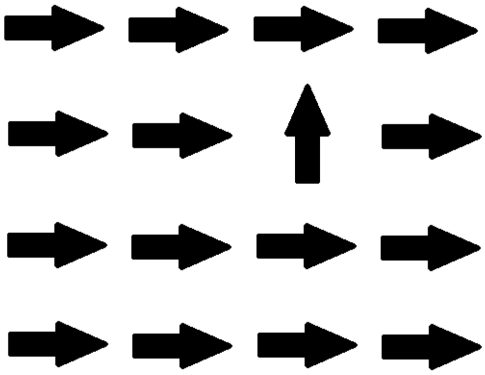


Figure 4.4: The visual arrow task.

Ersal et al. used a visual search task in which the participant had to locate and match several pairs of "scouts" and "targets" which appeared randomly on a touchscreen next to the driving task [11]. An example is shown in Figure 4.5 where three pairs are shown.

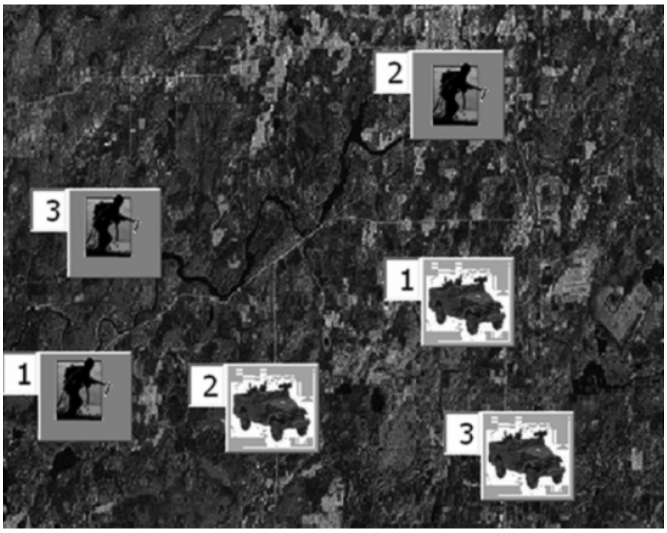


Figure 4.5: "scouts" (vehicle) and "targets" (person) secondary visual task.

The participants had to match the correctly numbered "scout" and "target" pictures by selecting them on a screen. Thus, a participant would first select scout 1 and target 1, after which a button is pressed to complete the assignments. This was repeated for the remaining pictures [11]. The "scout & target" task is a find and match exercise of which the difficulty could be varied by changing the icons and omitting the numbering of icons. Icons with similar shapes would be more difficult to match.

Other visual task include sending messages on the phone whilst driving [60] or reading numbers from a screen [36]. Another popular task is a surrogate visual research task (SURT), Figure 4.7, which mimics the in-vehicle display system (IVIS) [53]. Most modern cars have a IVIS on which the driver can navigate to a destination, turn on the radio or use some other functionality. The IVIS is therefore also considered as a possibility of distraction [30][25][2][43]. These tasks are for example a simple IVIS menu navigation task, inserting a destination in the navigation menu or finding a certain radio station.

4.3. Tracking Task

With a variety of possible secondary tasks which can be employed next to the primary tracking task, the next step is to find the most suitable task in order to distract people. From the viewpoint of the Multiple Resource Theory the tracking task can be placed as any other task into the 4 dimensional representation of multiple resources shown in Figure 4.1. The tracking task requires the participant to track a target with the controlled element. This would require foveal vision for vision processing, since the human operator needs sharp vision to minimize the tracking error. The modalities of the task can therefore also be considered to be visual, since no auditory inputs are necessary to complete the tracking task. Furthermore, the tracking task is a spatial activity requiring motor responses for moving the CE.

From the MRT analysis it can be seen that secondary visual tasks would share more of the same resources needed to do a tracking task compared to secondary cognitive tasks. Petzoldt et al. [43] investigated the possibilities of measuring driver distraction using the so called critical tracking task (CTT) shown in Figure 4.6 as the primary task. The CTT is similar to the compensatory tracking task since only a target and reference point is shown with the goal of minimising the error.

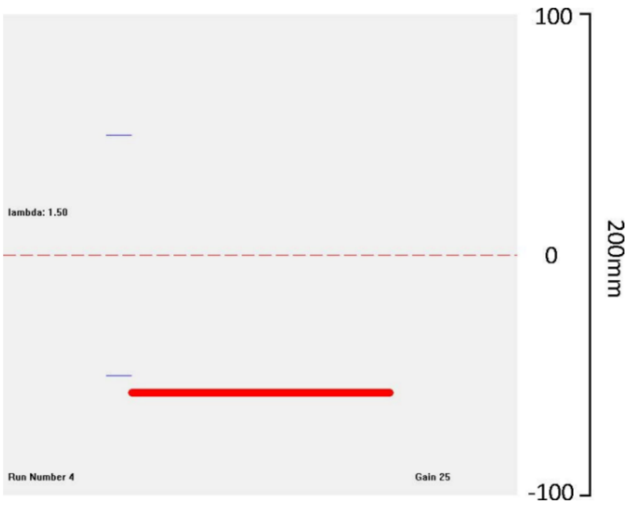


Figure 4.6: Critical tracking task (CTT); example screen [43].

Both cognitive and visual secondary tasks were used to distract participants in [43]. The cognitive task is a simple counting task where participants are asked to count forwards in steps of either 2 or 5 from 212 or 45, respectively. A more difficult version required participant to count backwards in steps of 6 from 831, or in steps of 7 from 581. The visual task is performed on a second screen on which the participant has to find the target object in the midst of “distractors”. An example of the task can be seen in Figure 4.7.

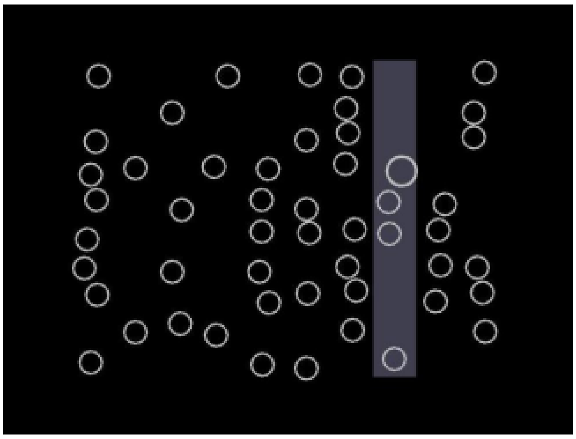


Figure 4.7: Surrogate reference task (SuRT); example screen[43].

The target has a different size compared to the distractors, namely 9 mm and 8 mm in diameter respectively. The difficulty could be varied by minimizing the size difference between the target and other objects.

The performance of the participants doing the tracking task was measured by the mean deviation in millimeters. For the experiment, 5 different variations and a total of 10 runs were performed (2 runs per variation). The variations were a baseline run with no secondary tasks and 2 runs including either a cognitive or visual task with a easy or hard difficulty. The result of the experiment is shown in Figure 4.8 with a total of 24 participants.

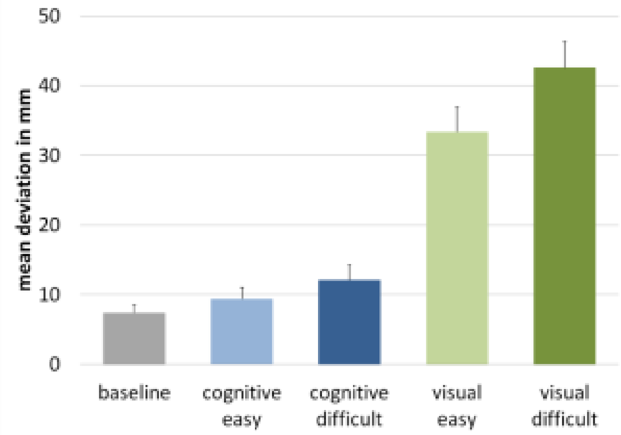


Figure 4.8: Mean deviation from centre position when cognitive or visual secondary tasks are performed in comparison with the baseline experiment [43].

Figure 4.8 shows a mean deviation of 7 mm for the baseline runs, meaning that participants will still show deviations in the critical tracking task even when there are no distractions. When comparing the baseline to runs with secondary cognitive tasks a slight increase of approximately 2 to 5 mm in mean deviation can be seen depending on the task difficulty. Runs with the secondary visual task show a much larger increase in mean deviation with differences of 25 to 35 mm compared to the baseline. An explanation of this result can be given based on the multiple resource theory by Wickens. Since both the CTT and visual search task require the same resources for visual perception, the information can not be processed simultaneously resulting in a worse performance of the primary task. The cognitive task shared little resources with the CTT resulting in little interference between both tasks as can be concluded from the results.

4.4. Chapter Takeaways

When a person is able to focus on a singular task, instead of multitasking, the results will show a better performance. This is because when people have to do two or more tasks at the same time, resources that process information have to be shared between the tasks. Wickens' Multiple Resource Theory (MRT) discusses this, in general terms tasks which require the same processing resources for example visual perception will cause greater interference compared to tasks that require different resources such as visual perception and auditory perception. The higher the interference the more difficult it is for humans to perform both tasks well.

Regarding experiments in task demand two main types of secondary tasks are used, these are cognitive and visual tasks. Examples of cognitive tasks are memory or counting tasks. Visual tasks consist of finding irregularities in patterns (search tasks) and texting.

Based on the MRT, a secondary task can be designed for the experiment in this research that will distract participants sufficiently. The tracking task is a task that requires visual perception, precision and a manual response. According to the MRT and previous research it is evident that cognitive tasks have little impact on tracking results. Visual tasks on the other hand are a better tool for distracting people since the human operator has to take in visual information from both tasks.

Furthermore, it could be of interest to investigate to what extent a neural network is able to detect distractions. The visual secondary task can be modified to change the difficulty in order to research this question.

5 Anomaly Detection

The aforementioned work uses time series classification models in which various time series signals are labelled [11][53][54][22]. A neural network is trained to recognize the different patterns and structures of the labelled signals and to classify them into the appropriate classification classes. However, due to the nature of the research problem and a lack of readily available time-series human control data of distracted pilots, a different approach is considered. Namely, anomaly detection. Anomaly detection in machine learning aims at finding unexpected or off-nominal events in data streams, commonly referred to as anomalous events [48]. section 5.1 discusses current challenges in anomaly detection techniques whereas section 5.2 discusses interesting aspects to this matter. section 5.4 holds the takeaways of this chapter.

5.1. Challenges in Anomaly Detection

The main principle in anomaly detection is to have a model trained to recognize normal behaviour and to identify the irregularities. Even though the approach is simple, in general there are still challenges which have to be tackled depending on the problem itself. Some challenges are for example:

1. **Definition of Normal**: It is difficult to determine whether every aspect that is deemed to be normal is covered in the data used to train the neural network. Abnormal behaviour that shows little to no difference compared to what is considered as normal, may result in false positives or false negatives.

2. **Evolving Behaviour**: Depending on the area of application, normal behaviour may change over time. In this case the model should be adaptable and change which the current trends. In the past where the norm was to shop in physical stores, a large group of consumers has shifted to shopping online as new technological advancements are made through the years [31]. To accommodate these changes, stores had to devise new strategies in order to adapt.

Data related challenges could be present in the limited availability of labelled data, growth in data size resulting in an increase for the need of computing resources, or noisy data making it more difficult to distinguish between normal and abnormal behaviour.

5.2. Aspects of Anomaly Detection

Every research problem is unique and involves a wide variety of aspects that have to be taken into account. Aspects that are important for an anomaly detection problem could be in the nature of the input data, anomaly types, detection techniques and the output.

5.2.1. Input Data

The input data can be referred to as for example data points, vectors, or observations taken at an instance. When grouped together they describe a certain system. These instances are described by attributes such as dimension or characteristics which can be of the type binary, categorical or continuous. A data instance can consist of only one attribute or multiple attributes, which are called univariate or multivariate, respectively [52].

The attributes play the biggest role in determining which anomaly detection techniques should be used [4]. For example, statistical techniques may require different statistical models depending whether the input data is continuous or categorical. Anomaly detection techniques can be based on point data, in which it is assumed that there is no relationship between the data instances. However, in most cases there is a certain relationship, examples are sequence data such as time-series data in pilot control tasks, spatial data where neighbouring instances affect each other in ecological data and graph data when different vertices are connected with each other.

5.2.2. Anomaly Types

A different aspect is the type of anomaly which should be detected in the problem, a total of three anomaly types can be identified. These are:

1. *Point Anomalies*: The simplest form of an anomaly is considered to be a point example. This is visualised in Figure 5.1 where points A and B are point anomalies since they are different from the normal data points. Point anomalies can for example be applied in detecting suspicious bank transactions. For simplicity reasons only the amount spent is taken into account, when a person suddenly spends a lot more than usual it can be considered a point anomaly.

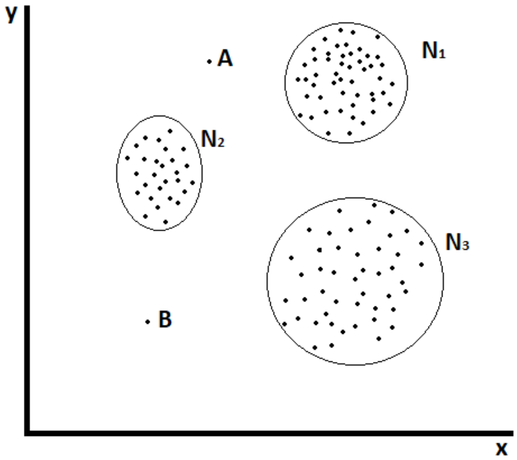


Figure 5.1: Data instances including point anomalies in 2D space.

2. *Contextual Anomalies*: Contextual anomalies only apply to a certain context, thus an anomaly in context A may be considered as normal in context B, also known as a conditional anomaly [49]. It is important to formulate the context of the problem when trying to detect anomalies, which is explained by Figure 5.2.

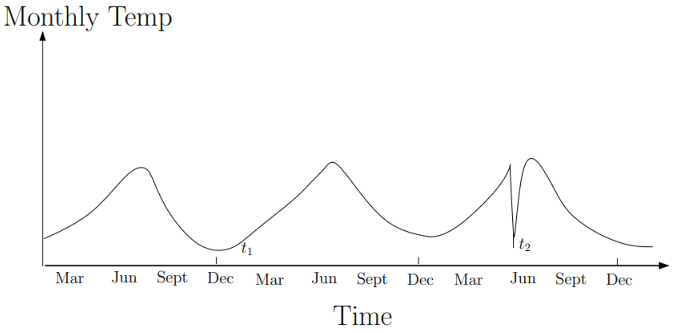


Figure 5.2: Contextual anomaly t_2 in a temperature time series. Note that the temperature at time t_1 is same as that at time t_2 but occurs in a different context and hence is not considered as an anomaly [4].

A data point can be defined by contextual attributes and behavioural attributes. A contextual attribute is used to determine the context of that point, for example the time stamp in time-series data. Or the longitude and latitude of a location in a spatial data set. Furthermore, behaviour attributes are non-contextual, which is for example the average temperature at a location when considering a spatial data set.

3. *Collective Anomalies*: If a group of data points behaves abnormal compared to the rest of the data set it is called a collective anomaly. An example can be seen in Figure 5.3 showing an electrocardiogram of an human. The collective anomaly is highlighted in red, since the particular value is present for a much longer time than expected.

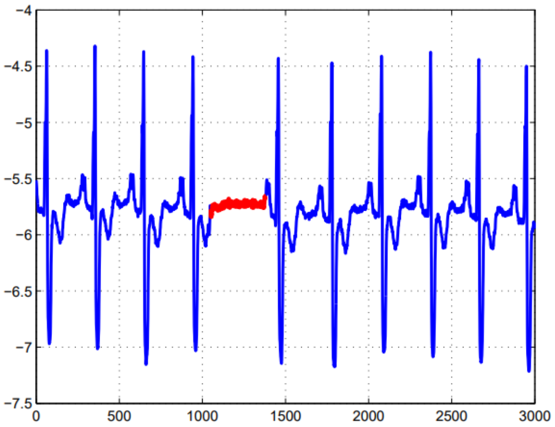


Figure 5.3: Collective anomaly in an electrocardiogram due to an atrial premature contraction [12].

5.2.3. Anomaly Detection Techniques

Detecting anomalies can be performed in several ways, it varies from supervised to unsupervised methods wherein the difference lies in data labelling.

Supervised anomaly detection uses labelled data for both the normal and abnormal instances. The model is trained to classify data into two classes, from which it should be able to predict whether the newly presented data correspond to the normal or anomaly class. This approach is shown in Figure 5.4. The model is trained to classify 3 classes in the visualisation. The training data consists of three classes as represented by the 3 different labels (colors). A model with an accuracy of 100% is able to classify all 3 classes perfectly. When using the test data, which is the same as the unlabelled training data (gray points), the same result should be reported.

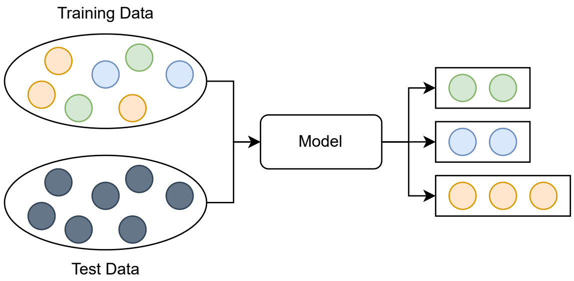


Figure 5.4: Supervised anomaly detection. Example of a 3 class classification problem.

Semisupervised anomaly detection techniques are useful in cases when there is a limited amount of anomalous data. The training and test datasets only include data from normal cases (no anomalies), thus any deviations from the learned model will be seen as an anomaly. This one-class classification approach are more widely applicable to supervised techniques as it is more difficult to capture all possible anomalous data and to label them. The semisupervised approach is visualised in Figure 5.5. It can be seen that first only normal data (white) is used for training the model. When the model is tested with test data, containing both normal and anomalous data, it is able to recognize what is considered to be normal (classified as green by the model), marking unfamiliar data as anomalies (classified as red by the model).

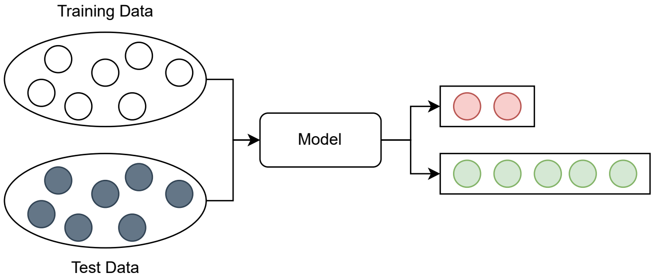


Figure 5.5: Semisupervised anomaly detection.

Unsupervised anomaly detection does not include any data labelling. Thus it is not known what the normal or abnormal state is. In order to detect anomalies, the assumption is made that normal data appear more frequent or grouped together compared to anomalies. Figure 5.6 shows an example of unsupervised models where unlabeled data is used. When the unlabeled data is fed through the model, it looks at the data distribution and separates data instances into two groups, normal data (green) and anomalies (red).

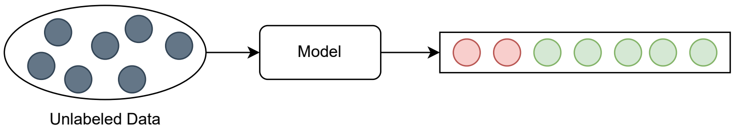


Figure 5.6: Unsupervised anomaly detection.

5.2.4. Ouput

The last aspect in anomaly detection is to determine how to present the abnormalities in a meaningful way. The two most commons ways to do so are giving scores or using labels [4].

1. *Scores*: The scoring methods assign each data instance with a score. And depending on the scale used it can be determined whether the data is considered to be anomalous. To filter out the particular data instances, a cut-off threshold can be set.

2. *Labels*: This technique assigns labels to every data instance as either ‘normal’ or ‘anomalous’. This binary approach classifies the data instances based on the mentioned scores.

5.3. Anomaly Detection Models

For the purpose of anomaly detection, several algorithms have already been developed of which most are readily available in python libraries. These libraries contain algorithm for anomaly detection in point data (PyOD) [14], time-series (TODS) [21], and graph data (PyGOD) [27].

Han et al. compared several anomaly detection models on 57 data sets by reporting the area under curve Receiver operating characteristic (AUCROC) as shown in Figure 5.7 [14].

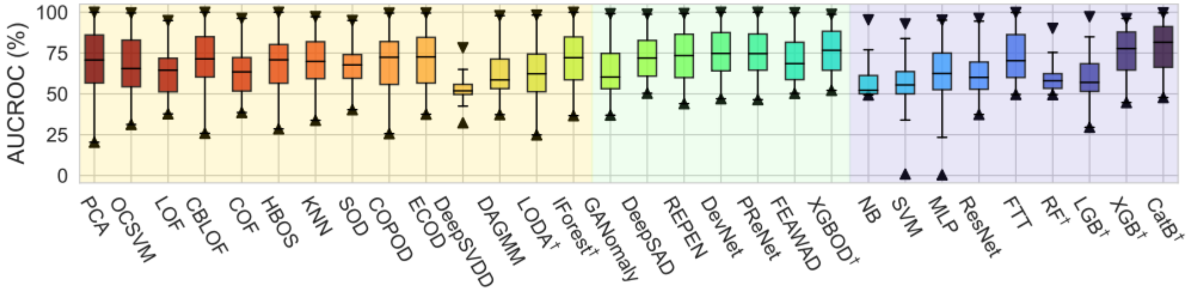


Figure 5.7: Boxplot of AUCROC (@1% labeled anomalies) on 57 data sets; un-, semi-, and fully supervised methods are denoted in light yellow, green, and purple, respectively [14].

The models consist of un-, semi-, and fully supervised methods. Figure 5.7 shows that when 1% of the labelled data is considered as an anomaly, the AUCROC for each model is comparatively the same. However, when the percentage of labelled data samples increases, the AUCROC of semi-, and fully supervised methods increase as well. This results is presented in Figure 5.8, where the percentage of labelled anomalies is increased to 10%.

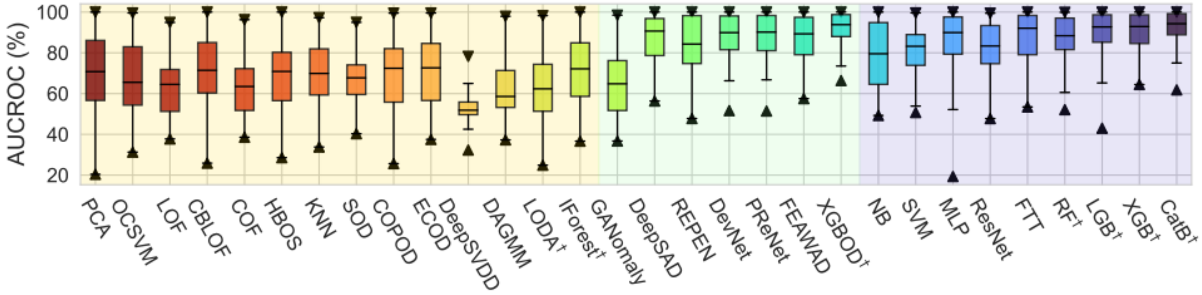


Figure 5.8: Boxplot of AUCROC (@10% labeled anomalies) on 57 data sets; un-, semi-, and fully supervised methods are denoted in light yellow, green, and purple, respectively [14].

5.4. Chapter Takeaways

The idea of anomaly detection is simple: find the abnormality in a large group of normal data. However, there are still many challenges that have to be overcome. It is difficult to train a neural network on what is considered to be normal. This could be due to lack of available data, changing trends through time, or noisy data. A step in the good direction is to analyse the aspects of the problem and to use the appropriate anomaly detection approaches for solving a case.

Important aspects to think of are:

- Input Data: the availability and type of data; Is the data labelled or unlabelled? How much data is available? Does the data contain images or time series data?
- Anomaly Types: there are three main types of anomalies.

  - *point anomalies* are single outliers in a data set.
  - *contextual anomalies* are for example freezing temperatures during summer instead of in winter times.
  - *Collective anomalies* are a group of data points that show abnormal behaviour.

- Anomaly Detection Techniques:
  - *Supervised anomaly detection* only uses labelled data (normal and abnormal data) to train the model.
  - *Semisupervised anomaly detection* uses labelled data of what is considered to be normal to train the model.
  - *Unsupervised anomaly detection* does not use labelled data, it relies on the network itself to recognize anomalies.
- Output: the desired way to indicate an anomaly, can be done using scores or labels.

The problem in detecting distractions lies in the limited amount of data available of distracted people doing the tracking task. Because of the sheer amount of available normal-labelled data, a supervised anomaly detection method is recommended. The anomalies can be considered as collective anomalies since the tracking task generates time series data and little context is needed, compared to the example given in Figure 5.2, to understand what is deemed as normal.

6

Preliminary Simulation

For the final implementation of detecting distractions in manual control data, preliminary simulations have been performed to understand neural networks and its implementations. In section 6.1 the various simulations performed in this chapter are listed. Furthermore, section 6.2, section 6.3, and section 6.4 present the methods and results of the simulations. The tracking error in tracking tasks are analysed in section 6.5. Finally, the chapter takeaways are presented in section 6.6.

6.1. Simulations

The simulations that have been performed, provide a better understanding into the work that has been done by other student and how to solve the main problem. For this purpose the following 3 simulations have been conducted:

1. Classifying the response of pilot control tasks based on display types using deep neural networks [19].

2. An initial set up for detecting anomalies using display type data.

3. Detecting distractions using tracking data with distracted people.

The first simulation is based on prior work done by Kiselev [19] in which the three different display types used in control tasks are classified using a neural network called InceptionTime based on 1.5-second time series data samples. An additional data set had been collected in which the controlled element dynamics had single integrator properties compared to the data from van der El. et al [9]. Using the data set of the first simulation, a new model is trained with only one display type. The goal of the second simulation is for the model to indicate the detection of an anomaly when the other display type data are fed to the trained model. In the third simulation, a model is trained using data with distractions and analyzed on how well it is able to detect distractions.

6.2. Classification Simulation

The tracking task is used as a tool to study how a human operator adapts to changes in task variables whilst keeping the experiment conditions the same as much as possible. A total of four task variables were identified in the pilot-vehicle system by McRuer [35]. These are presented in Figure 6.1

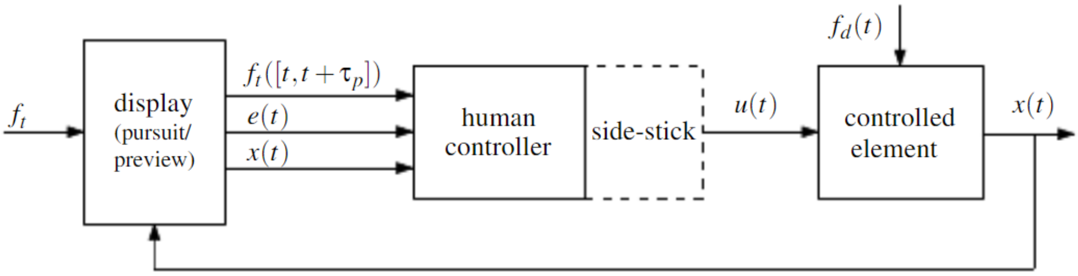


Figure 6.1: Task variables affecting the pilot-vehicle system for a pursuit/preview display [8].

The manipulator can be considered as a passive stick of which the dynamics can be safely ignored. The controlled element represents the dynamics between human control inputs and the state of system that is controlled. The third task variable is the forcing function. Depending on the type of tracking task, a target signal, $f_t(t)$, has to be followed in a following task. This target signal is a sum of sinusoids from which a quasi-random signal can be created. In the other task a disturbance, $f_d(t)$, is forced on the controlled element in a disturbance task. Lastly, the display, discussed in section 3.2 is the final variable which can be changed. The 3 main display types are a compensatory, pursuit and preview display.

The main goal of the first simulation is to classify human control responses by display types. The neural network used for this simulation is called InceptionTime, an ensemble of deep Convolutional Neural Network models [15]. InceptionTime was introduced to improve upon a previous machine learning model called HIVE-COTE for Time Series Classification (TSC) problems to reduce training times. TSC in machine learning deals with the categorization of time series data, which are also produced in tracking tasks.

The data used in the simulation comes from experiments done by van der El et al. [8] and Kiselev [19]. In these tracking task experiments, the three different display types were used with the only difference being the controlled element dynamices (CED). Van der El's data consists of experiments using a double integrator (DI) CED while Kiselev's experiment CED was a single integrator (SI) which is easier to control by human operators (HO). A SI means that the HO controls the velocity of the controlled element, for DI dynamics this would be the acceleration of the controlled element.

From the experiments the following signals were recorded and used as time series data, sampled at a rate of 100 Hz:

- t, time.
- $f_t(t)$, target signal.
- $f_d(t)$, disturbance signal.
- $e(t)$, error signal.
- $u(t)$, HO input signal.
- $x(t)$, output signal.

The neural networks were trained using a machine learning platform called Weights and Biases (WandB)[1]. This tool allows for tracking, comparing, and visualizing ML experiments by importing and initializing it in the Python script. The hyper-parameters used for training the model are shown in Table 6.1. The combination of these parameters were determined by Kiselev and are optimised to obtain the highest accuracy possible for the classification model.

[1] https://wandb.ai/site

Table 6.1: Hyper-parameters used for training neural network.

Parameter	Value
Batch Size	64
Epochs	25
Bottleneck	No
Convolutional Dropout	0.05
Kernel Size	64
No. Convolutional Filter	24
Max. Learning Rate	0.00275
Residual Connection	No
Weight Decay	0.05
Batch Normalization	No

Apart from the hyper-parameters in Table 6.1 the variables, e, u, x, and their derivatives of the time series data are used with time windows of 1.5 s [19]. Furthermore, 80% of the data is used for training and the remaining 20% is used for validation. The training process and final results are visualized in Figure 6.2 showing a steady increase in model accuracy. With increasing steps in the training process, the model is able to recognize the data from different display types better, causing the slow increase in accuracy.

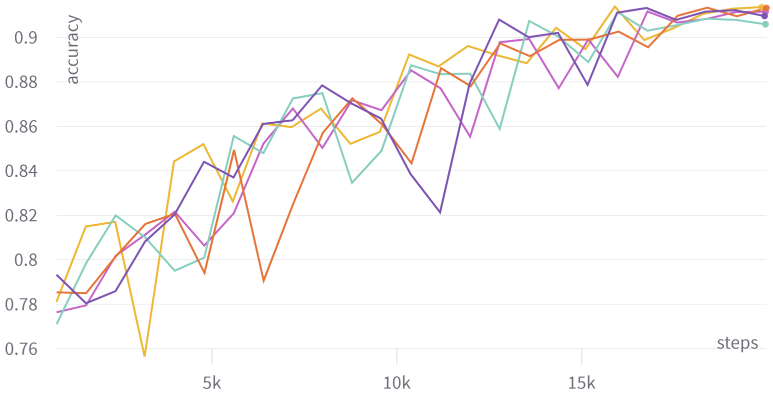


Figure 6.2: Model accuracy of 5 different training runs as a function of steps. Each run consists of 25 epochs.

Figure 6.2 shows a steady increase in model accuracy with increasing steps. Each line represents a single training run using the single integrator data, and each point on a line represents an epoch. Starting from an average accuracy of 0.78, the curves eventually converge to an average of 0.92. Similar results are also obtained for the double integrator data set. To get a better idea of the predictions made by the model, confusion matrices are also created, these are found in Table 6.2 and Table 6.3 for the SI and DI data set, respectively.

Table 6.2: Confusion Matrix of the SI data set (C = Compensatory, P = Pursuit, PR = Preview).

Total Accuracy		Predicted		
91.83%		C	P	PR
Actual	C	93.31% 19999	6.68% 1431	0.01% 2
	P	17.32% 3571	82.63% 17041	0.05% 11
	PR	0.21% 45	0.24% 53	99.55% 21546

Table 6.3: Confusion Matrix of the DI data set (C = Compensatory, P = Pursuit, PR = Preview).

Total Accuracy		Predicted		
95.38%		C	P	PR
Actual	C	99.87% 21168	0.09% 20	0.04% 8
	P	7.13% 1499	89.96% 18902	2.90% 610
	PR	1.73% 372	1.97% 422	96.30% 20657

The confusion matrices show the number of samples predicted labels against the actual label. The confusion matrices show a total accuracy of 91.83% and 95.38% for the SI and DI data set, respectively. This result is in accordance with the findings of Kiselevs work, which this analysis was meant to replicate, and where equivalent accuracies of 91.66% and 95.42%, respectively, were reported [19]. Therefore, a successful replication of Kisilevs work has been achieved.

6.3. Anomaly Detection

In preparation of detecting anomalies in time series data, a neural network using the same parameters as in section 6.2 is trained for two class classification. The two classes are:

1. Preview display data
2. Compensatory/Pursuit display data

When time series data is fed to the model it should be able to give a probability whether it ‘thinks’ the data sample corresponds to preview data. This is done for two cases in which only preview data is fed to the model and when only pursuit data is used. Figure 6.3 shows the probability for classifying data as preview data, in this case the data set used to train the model consisted of 75% preview data and 25% pursuit/compensatory data. The preview time series data used to produce the results in Figure 6.3 is shown in Figure 6.4.

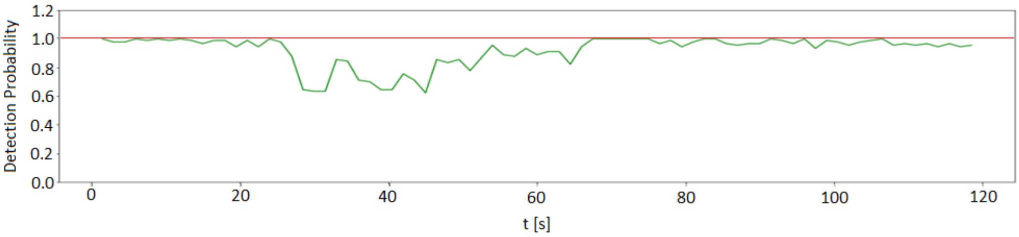


Figure 6.3: Probability of preview data samples classified as preview data for a single run.

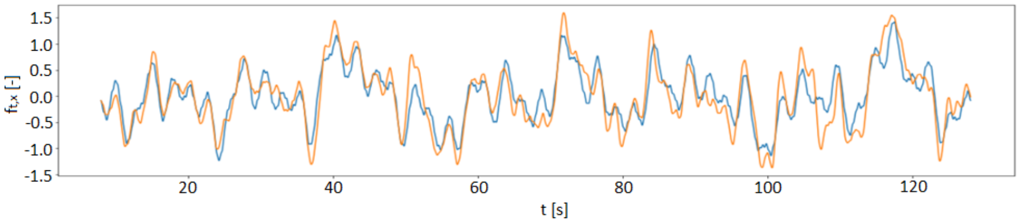


Figure 6.4: Time series data corresponding to a preview display tracking task (blue = output, orange = target). Figure shows 120 seconds of tracking data starting from t = 8 s. t = 8 s corresponding to t = 0 s in Figure 6.3.

The probability shown in Figure 6.3 corresponds with the expected result. For the majority of the time series data shown the probability is close to 1. However, between $t = 25$ s to $t = 65$ s the probability is lower, the lowest value being approximately 0.6. An explanation for this behaviour could be the larger errors found in this particular preview time series data set making it look more like compensatory or pursuit data. However, this is only a speculation and may not be representative to the complete data set since the probabilities of Figure 6.3 was obtained using only data of 1 run.

The simulation described for classifying preview data is also done for a data set containing only pursuit data. The result of this simulation is shown in Figure 6.5 with the corresponding pursuit time series data in Figure 6.6.

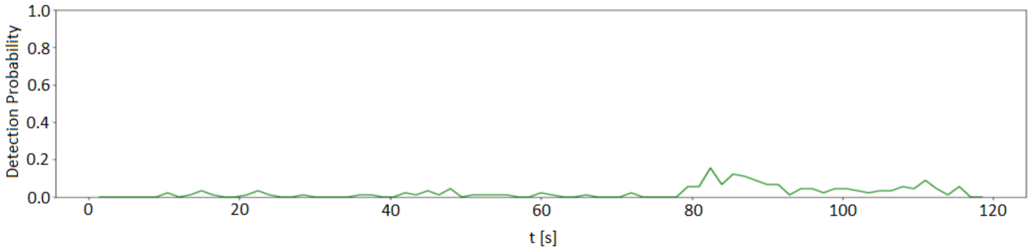


Figure 6.5: Probability of pursuit data samples classified as preview data.

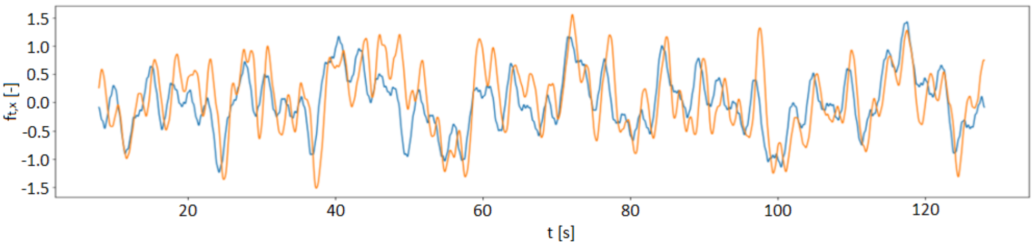


Figure 6.6: Time series data corresponding to a pursuit display tracking task (blue = output, orange = target). Figure shows 120 second of tracking data of starting from t = 8 s corresponding to t = 0 s in Figure 6.5.

Figure 6.5 shows that the probability of the samples correspond to a preview display are near 0 as expected. This result shows that only from $t = 80$ s onwards, the probability increases by a small percentage, staying below a probability of 0.2. The neural network architecture InceptionTime is thus able to distinguish and classify different display types well at an acceptable level of accuracy shown here and by the results of section 6.2.

The next step is to see whether the neural network can differentiate between the preview display data and the other two types of displays when the model is only trained on preview data. By training the data on only one display type, the problem turns from data classification to detecting anomalies. The results of this simulation are shown in Figure 6.7 and Figure 6.8. Apart from the preview data, the model is tested using compensatory and pursuit data, respectively.

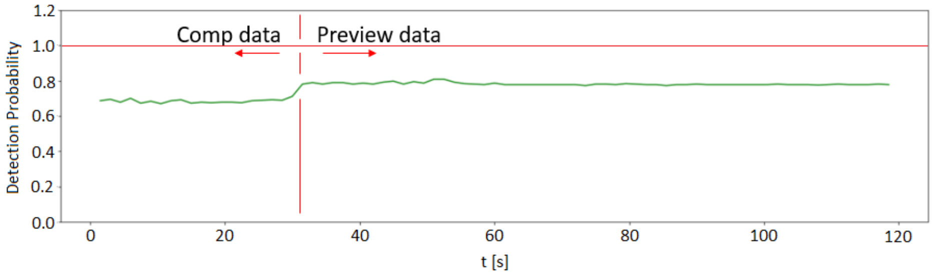


Figure 6.7: Probability of compensatory or preview data samples being classified as preview data, averaged over 180 runs.

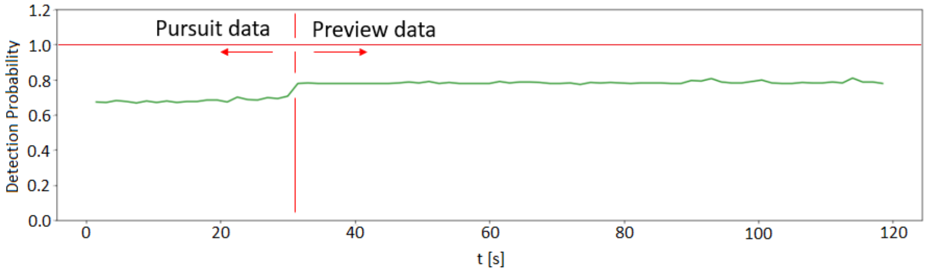


Figure 6.8: Probability of pursuit or preview data samples being classified as preview data, averaged over 180 runs.

Figure 6.7 shows the results where compensatory and preview display data are used. And the result in Figure 6.8 are generated with pursuit and preview data. From both figures it can be noticed that the probability of the samples corresponding to preview data are nearly the same and only differ by an average of 0.1. An explanation for this result is that the model is only trained using one class of data (preview) while it is tasked to classify between two data classes (preview and compensatory/pursuit). The model has therefore little knowledge of how compensatory and pursuit data look like.

In addition to this, tracking data from different display types can show many similarities. Take for example the preview and pursuit data in Figure 6.4 and Figure 6.6. The data of different display types can be seen as clusters as presented in Figure 6.9. And overlapping parts of the different clusters may have contributed to the small difference in probability between preview and other types of data.

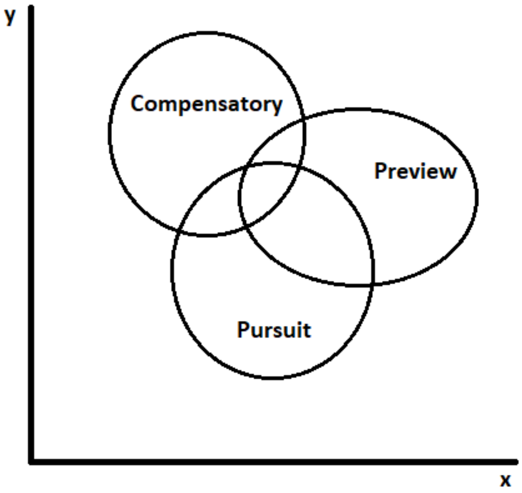


Figure 6.9: Example data clusters of compensatory, pursuit and preview tracking data.

When a model does not have any knowledge of how the different classes may look like in a two class classification task, the classification probability would most likely be split 50:50. In this case, the model has an idea of how preview data looks like and would therefore generate a higher classification probability for preview data. In Figure 6.7 and Figure 6.8, the probability for classifying preview data as preview data is around 0.8 in both figures. When it comes to either compensatory or pursuit data, the probability of it being classified as preview data is approximately 0.7.

A possible solution to this problem is to use a few positive samples (distracted data) when training the anomaly detection model as proposed by Xue et al. [63]. An autoregressive model (AR) with a recurrent network as backbone has been modified with either a margin loss function or an auxiliary classification loss function. The results showed that the normal AR model would only reach an average detection rate of 0.6498. The modified auxiliary and margin model reached averages of 0.8152 and 0.8110, respectively, showing much better results.

6.4. Detecting Distractions

Moving away from the classification problem of display types, the goal now is to detect actual distractions in tracking tasks. Figure 6.10 shows the time series data of the forcing function and output of a tracking task with a preview display. The difference between a normal tracking task is the addition of distractions which are visualized by square pulses with a duration is 2 seconds (6 distractions per run).

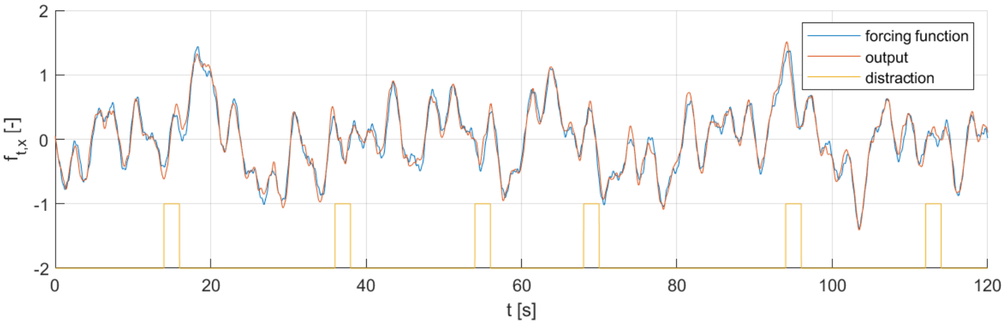


Figure 6.10: Time series data corresponding to a preview display tracking task including distractions.

The data which will be used for the preliminary simulation of detecting distractions is collected by Nokhai[2] and consists of preview and pursuit tracking data including distractions. The secondary task used as a distraction in a tracking run was identifying colours projected onto a screen, positioned at an angle of 90°. An example screen is shown in Figure 6.11

Figure 6.11: Sketch of the “distractor” display showing a red square.

Before the data can be used to train a model, each distraction has to be labelled first. From the experiment performed by Nokhai it is known that each distraction had a duration of 2 seconds. However, for the simulations a time window of 1.5 seconds and data overlap of 0.75 seconds will be used, the same values used by Kiselev [19]. From Figure 6.12 it can be seen that a decision has to be made to determine what is considered as distracted.

[2] reference not available

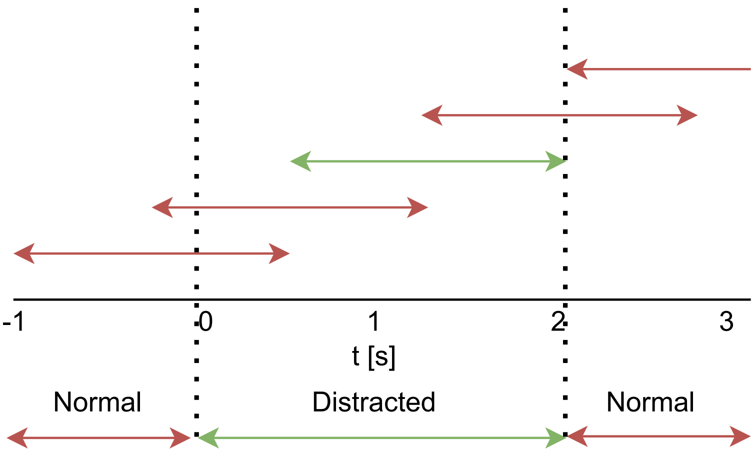


Figure 6.12: Consideration in labelling time series sample as distracted or normal.

Figure 6.12 shows that some samples contain both distracted and normal data, but only 1 sample contains true distracted data (highlighted in green). For the preliminary simulation, it was decided to use only the green sample as distracted data. This way samples labelled as distractions are guaranteed to consist of only distracted tracking data.

Using InceptionTime, two models have been trained using preview and pursuit data separately with the same parameters in Table 6.1. The confusion matrices for the preview model and pursuit model are presented in Table 6.4 and Table 6.5, respectively.

Table 6.4: Confusion matrix of the preview model (N = Normal, D = Distracted).

Total Accuracy		Predicted	
61.27%		N	D
Actual	N	98.92% 19216	1.08% 209
	D	76.39% 576	23.61% 178

Table 6.5: Confusion matrix of the pursuit model (N = Normal, D = Distracted).

Total Accuracy		Predicted	
71.27%		N	D
Actual	N	98.92% 19281	1.08% 211
	D	56.38% 451	43.63% 349

The preview model has an accuracy of 61.3%, whilst the pursuit model has a higher accuracy of 71.27%. Table 6.4 shows that 76.4% of the samples containing distractions are classified as normal, in Table 6.5 this percentage is 56.38%. In a tracking task with a preview display, the participant is able to see a few seconds ahead of time and therefore knows what the trajectory would be, whilst being distracted for approximately a second. The behaviour of the human controller when being distracted in a tracking task might thus be almost the same in normal conditions. For pursuit data the difference in behaviour should be greater compared to preview data, because the human operator does not know the future trajectory of the forcing function.

Furthermore, the limited amount of distracted data compared to the amount of normal used in training may also be a cause for the low accuracies of both models. Each run only consists of 6 distracted samples, in comparison to the 154 normal samples. A solution to this problem is to reduce the number of normal samples used in the training process. In order to test this, two new models are trained using 1 distracted sample for every 4 normal samples. The confusion matrices of the new preview and pursuit model are presented in Table 6.6 and Table 6.7, respectively.

Table 6.6: Confusion matrix of the preview model trained using a 4 to 1 ratio of normal and distracted data (N = Normal, D = Distracted).

Total Accuracy		Predicted	
72.09%		N	D
Actual	N	75.26% 572	24.74% 188
	D	31.07% 298	68.93% 661

Table 6.7: Confusion matrix of the pursuit model trained using a 4 to 1 ratio of normal and distracted data (N = Normal, D = Distracted).

Total Accuracy		Predicted	
81.63%		N	D
Actual	N	84.08% 639	15.92% 121
	D	20.82% 197	79.18% 749

Table 6.6 and Table 6.7 show an improvement for both the preview and pursuit model, with accuracies of 72.1% and 81.6%, respectively. The trained models were hereafter saved and exported to analyse the detection probability of distractions for single runs. Figure 6.13 shows the detection probability of every sample in a single run in blue, the orange bars are the instances (samples) where the distraction takes place.

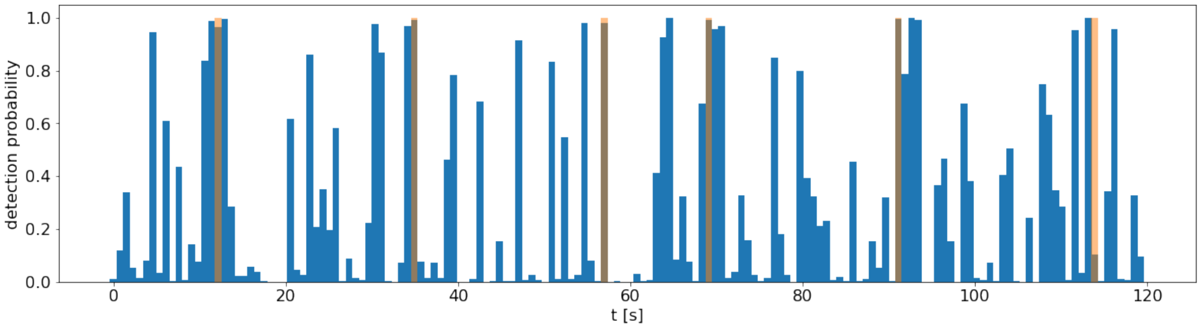


Figure 6.13: Detection probability of distractions for a single preview run highlighting distracted samples (subject 5, run 7).

Similarly to Figure 6.13, Figure 6.14 shows the detection probability of the same single run, preview tracking data (subject 5, run 7). However, the orange bars in Figure 6.14 represent the predicted distracted samples. It can be seen that more samples are classified as distracted by the model compared to the 6 actual distracted samples in each tracking run.

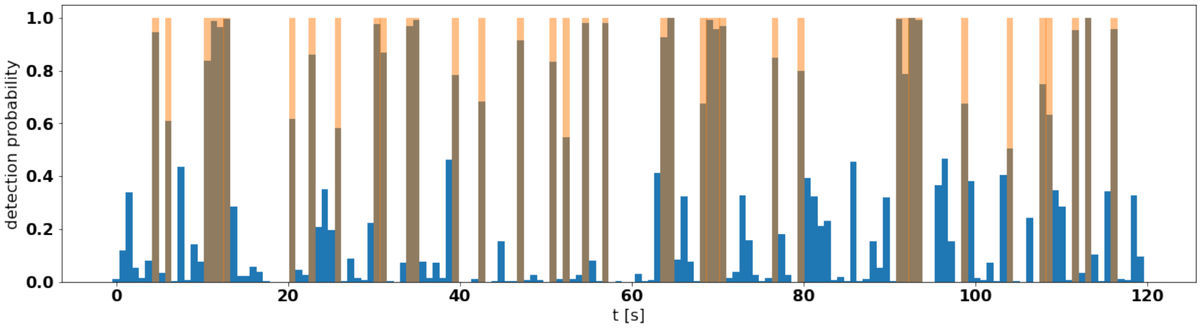


Figure 6.14: Detection probability and classification of distractions for a single preview run (subject 5, run 7).

The same has also been done for pursuit data, the results can be found in Appendix A. The confusion matrices of both single preview and pursuit runs are presented in Table 6.8 and Table 6.9, respectively.

Table 6.8: Confusion matrix of a single preview run (subject 5, run 7).

Total Accuracy		Predicted	
77.85%		N	D
Actual	N	77.63% 118	22.37% 34
	D	16.6% 1	83.33% 5

Table 6.9: Confusion matrix of a single pursuit run (subject 5, run 7).

Total Accuracy		Predicted	
81.01%		N	D
Actual	N	80.26% 122	19.74% 30
	D	0.000% 0	100.0% 6

The confusion matrices show that most distractions in these particular runs can be detected. However, a significant amount of normal samples have been classified as distracted. A possible solution to reduce this number, is to increase the threshold at which a sample is deemed as distracted. Currently, the threshold has been set to 0.5.

In order to test this solution, increments of 0.1 have been taken, starting from the original decision threshold of 0.5 up to 0.9. The value of 0.95 has also been taken into account in this analysis for a better comparison at high threshold values. Figure 6.15 shows the classification of distracted samples at a decision threshold of 0.95 in a single run for subject 5, run 7. The classification results of the other threshold values of this particular run can be found in Appendix A.

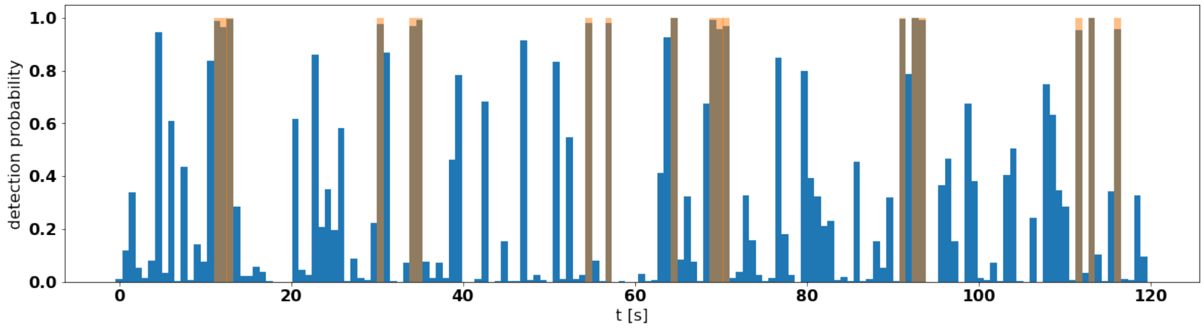


Figure 6.15: Detection probability and classification of distractions for a single preview run with a decision threshold of 0.95 for classifying distracted samples (subject 5, run 7).

By comparing Figure 6.15 with Figure 6.14, a clear distinction can be found in the number of samples that have been classified as distracted. At the higher decision threshold of 0.95, only 18 samples are classified as distracted. To get a better idea of how the classification of samples has changed the confusion matrices in Table 6.10 and Table 6.11 are presented for the preview and pursuit model, respectively. The detection probability and classification results with varying threshold values for the pursuit tracking run can be found in Appendix A

Table 6.10: Confusion matrix of a single preview run with a decision threshold of 0.95 for classifying distracted samples (subject 5, run 7).

Total Accuracy		Predicted	
91.14%		N	D
Actual	N	91.45% 139	8.55% 13
	D	16.6% 1	83.33% 5

Table 6.11: Confusion matrix of a single pursuit run with a decision threshold of 0.95 for classifying distracted samples (subject 5, run 7).

Total Accuracy		Predicted	
89.97%		N	D
Actual	N	89.47% 136	10.53% 16
	D	0.000% 0	100.0% 6

Table 6.10 shows the confusion matrix of the preview model with a decision threshold of 0.95, the only difference with Table 6.8 is the change in classification of actual normal samples. Fewer normal samples have been classified as distracted. The same can be said about the pursuit model in Table 6.11. By increasing the decision threshold, the accuracy of both models have increased to around 90%. 91.14%

and 89.97% for the preview and pursuit model, respectively.

In the interest of determining whether increasing the decision threshold would increase the model accuracy in general, 4 other single tracking runs (2 preview, 2 pursuit) have been randomly selected and tested on the preview or pursuit model. The model accuracy at different decision thresholds between 0.5 and 1 have been plotted in Figure 6.16. For the results of the pursuit model, see Appendix A.

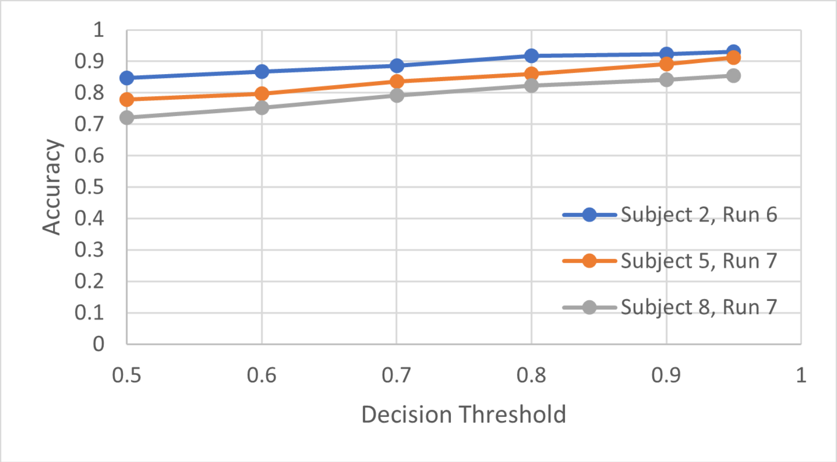


Figure 6.16: Model accuracy with varying decision threshold for classifying distracted samples, preview data.

Figure 6.16 shows the model accuracy of three different preview runs. The general trend is that the accuracy does indeed increase by increasing the threshold. The highest accuracy achieved is 93.04%, corresponding to subject 2, run 6. The same conclusion is also found for the pursuit run, where the highest accuracy is 92.41%.

6.5. Tracking Error

Apart from training machine learning models to detect distractions, it might also be interesting to look at the tracking error in a tracking task. When a participant is distracted, it can be expected that the tracking error would increase since the focus of the participant has been shifted from the primary tracking task to the secondary task.

First, it is important to know when the participant is distracted. From Figure 6.17 it can be seen that 6 distractions occur in a single run within 6 different periods of 10 seconds. In these 10 second periods, distractions occur randomly in 1 of the 5, 2 second slots available.

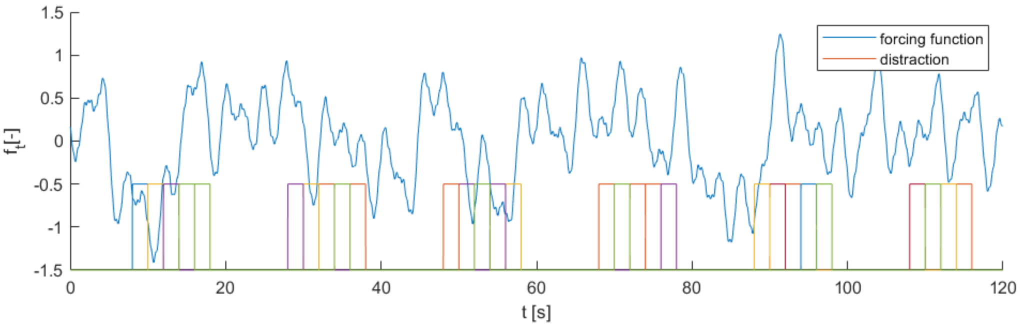


Figure 6.17: visualisation of possible distraction occurrences in a single run.

In order to analyse the tracking error, the absolute error at each and every specific time instance, for example $t = 21$ s, in each preview run has been summed up and averaged. The results of this procedure can be seen in Figure 6.18 and Figure 6.18 for all preview and pursuit runs, respectively.

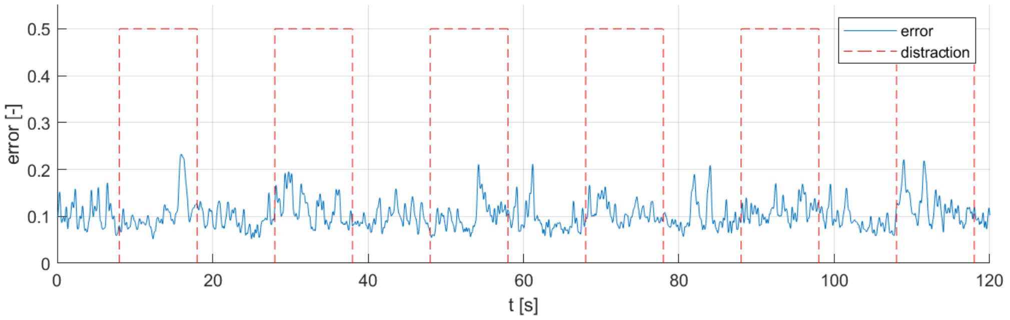


Figure 6.18: Average error of the all individual preview runs.

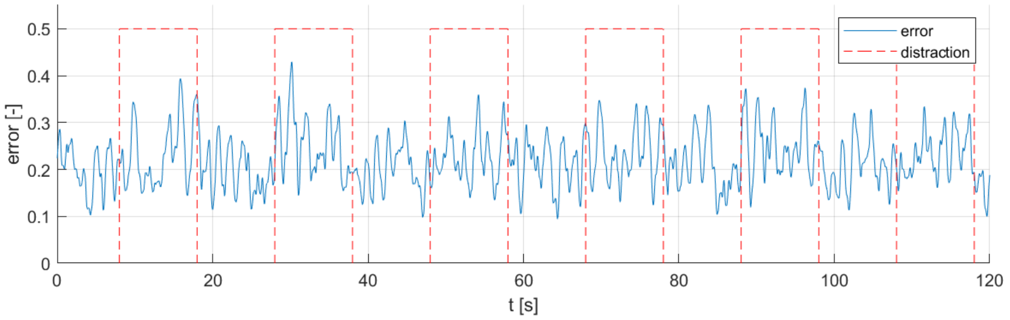
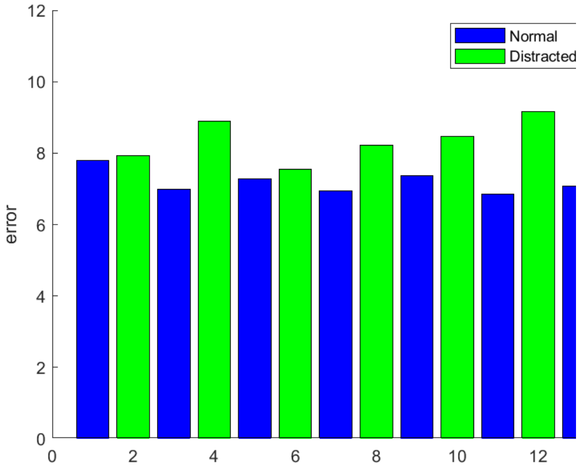


Figure 6.19: Average error of the all individual pursuit runs.

The sections in which distractions occur have been marked with a red border. It can be noted that the average error in preview data are lower compared to pursuit data. Furthermore, the distractions do seem the increase the tracking error. This effect can be better observed from Figure 6.20 and Figure 6.21 for preview and pursuit data, respectively.



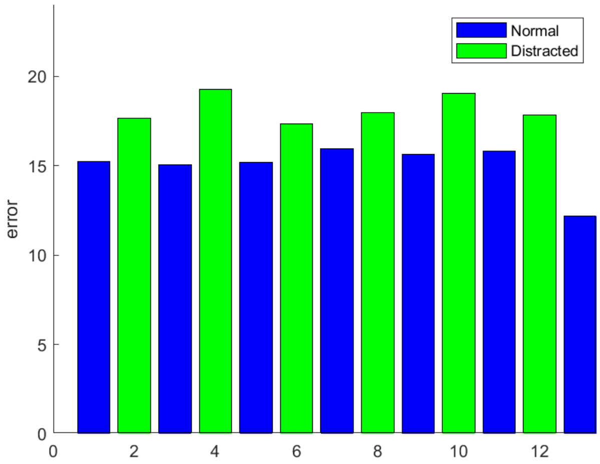


Figure 6.20: Sum of the average error in preview tracking tasks for each section.

Figure 6.21: Sum of the average error in pursuit tracking tasks for each section.

Aside from the distractions causing spikes in the tracking error, the forcing function itself could also contribute to an increase of error. At moments where the forcing function suddenly changes moving direction from left to right or vice versa, the human controller needs to react to this sudden change and will most likely overshoot the target signal. This is also visible in Figure 6.22 and Figure 6.23 showing the average preview and pursuit error, respectively.

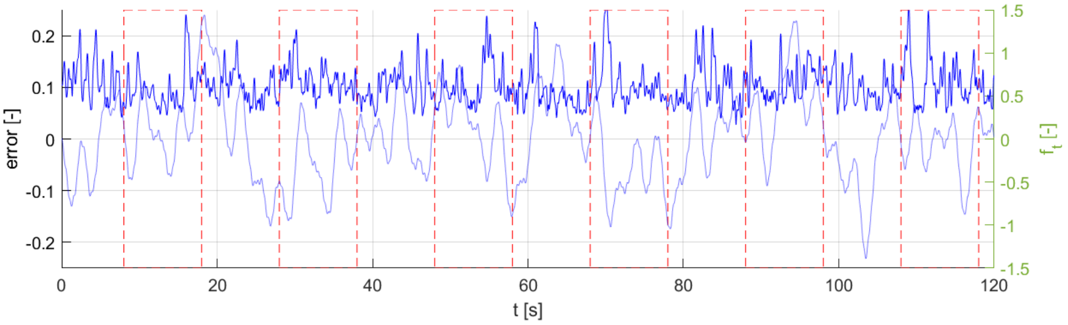


Figure 6.22: Average preview error of forcing function 1 (blue = error, light blue = forcing function).

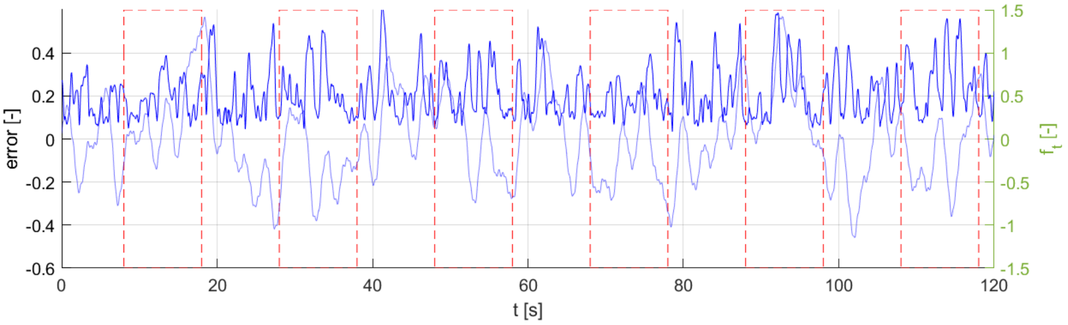


Figure 6.23: Average pursuit error of forcing function 5 (blue = error, light blue = forcing function).

It seems however that this effect could be neglected, since Figure 6.20 and Figure 6.21 already showed that section with possible distractions do have a higher tracking error average compared to normal sections.

6.6. Chapter Takeaways

The preliminary simulations provide new insights on the capabilities and limitations of what is possible in relation with the research problem: detecting distractions in human pilot control tasks using machine learning. InceptionTime, a deep convolutional neural network has been used to train the models.

The first simulation, classification of human control responses based on display types, showed that the InceptionTime architecture can reach high accuracies for both single and double integrator dynamics of the controlled element.

When moving from a 3 class to a 2 class classification problem with preview display and compensatory/pursuit displays the model was still be able to differentiate the different display types. The probability of samples corresponding to preview display, when preview or pursuit data are fed to the NN model were appropriate. This was the case since the model was still trained using 3 different display type data.

However, when the model is only trained with preview data it had a more difficult time in differentiating between the preview display and other display types. This could be caused by different display type data having some overlapping characteristics and most machine learning architectures are developed for image recognition.

Finally, NN models have been trained using tracking data in which subject have been distracted with a secondary task. Results showed that when all of the tracking data are used, the accuracy is lower compared to models trained on data where the ratio of normal and distracted data have a ratio of, for example, 4:1. In addition to this, increasing the decision threshold for classifying samples as distracted increases the accuracy even more. Accuracies of up to 93.04% and 92.41% have been reached for the preview and pursuit model, respectively. Furthermore, the tracking error in the tracking runs have been analysed and shows indeed that distractions do cause a small increase.

Following from the preliminary analysis, it is decided to use the InceptionTime neural network architecture for the final phase of the research. This is the same NN architecture used by Verkerk [59] and Kisilev [19] for classifying time-series data. This data is obtained from experiments involving the classical tracking task, which is also used in this research project.

7 Research Plan

The Preliminary phase of this research project helped gaining a better understanding of the research topic, *Detecting Distractions in Human Manual Control Tasks Using Machine Learning*, and the possible tools that can be used the execution phase.

Following from the preliminary phase it is clear that an experiment has to be designed to collect data of distracted people doing a control task. And how the data will be used to reach the research objective.

7.1. Experimental Design

The research consists of an experimental part using the Human-Machine Interaction (HMI) Laboratory located at the Technical University of Delft, Faculty of Aerospace Engineering. The HMI is a fixed base simulator for cars or aircraft which can also be used for experiments with control tasks of visual perception research [1]. An illustration of the HMI lab can be seen in Figure 7.1, showing the experiment room where the subjects are doing the experiment and the experimenter is controlling the simulation from an observation room. The observation room contains the computers that are used to control the various devices in the lab.

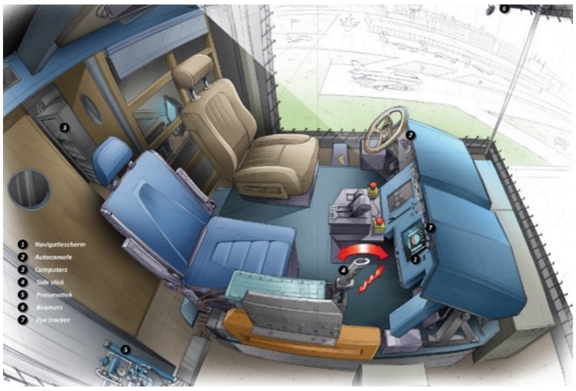


Figure 7.1: Illustration of HMI Lab. The participant will be sitting on the right (blue) seat and controls the side-stick. The participant will have to look at the screen on the left [1].

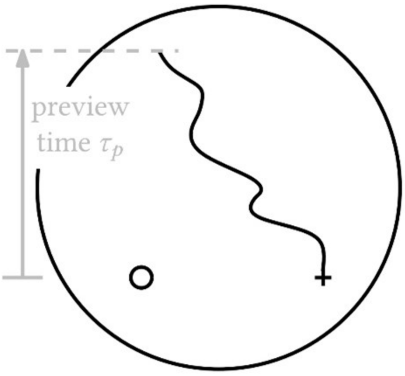


Figure 7.2: Sketch of HMI Lab tracking display [7]. The goal is to steer the state (circle) to the target (plus). The preview trajectory line is either present or not.

The participant will be seated on the 'aircraft side' (blue seat in Figure 7.1) from which the the participant will complete several tracking tasks with a to be determined secondary task used as a distraction. The equipment used to do this perform this experiment are [1]:

- A fully adjustable aircraft seat, from a Breguet Atlantique, installed on the right-hand side.
- A control-loaded hydraulic side stick, with $\pm 30^\circ$ excursion in roll and $\pm 22^\circ$ excursion in pitch.

- Two 18” LCD panels for the instrument displays, one installed in portrait mode, both displays with a 1280 x 1024 pixel resolution.

Furthermore, a visual projection system is available which use a three-sided projection screen. This allows for 180° field of view that can be used for displaying visual distractions. The participant will have to remove its eyes from the primary tracking task and focus on what is displayed on the screens. Options for generating the projected image are using OpenSceneGraph, OGRE or FlightGear image generation [1].

The tracking task is displayed on one of the screens in front of the participant, representative of what is shown in Figure 7.2. The goal is to minimise the error (difference) between the controlled element (circle) and the target (plus). The preview trajectory line shows the trajectory of the target for a certain time, τ_p. Each tracking task run has a duration of 128 seconds of which only 120 seconds of data will be used. The first 8 seconds is considered as a run-in time that allows participants to adjust of focus on the task. Furthermore, the data is collected at a sample rate of 100 Hz.

The following steps are taken to design and do the actual experiment:

1. Design a visual secondary task used to distract subjects, if possible include varying task difficulties.
2. Decide the scope of the control task: controlled element dynamics, display types.
3. Integrate and test the secondary task with the control task.
4. Invite participants to the experiment for data collection.

7.2. Data Processing

The second part includes data processing and training/applying ML models. The program used for data processing is MATLAB since the data is collected in a `.dat` file and the base code to process the data is readily available from previous research [7][19]. The machine learning code is written in Python with imported scientific libraries such as Numpy & Scipy and machine learning libraries like torch, tsai, and fastai. These ML libraries are used to support the InceptionTime neural network architecture properly. Tracking task data consist of multivariate time-series data which is different from the purpose what most ML models are designed for, recognising images [32][60].

Training and documentation of the ML models are done through an online API called Weights and Biases (WandB)[1] which visualized and stores each training cycle. Furthermore, the models are uploaded onto a virtual computer with a GPU provided by Paperspace[2]. This significantly reduces computing times compared to a laptop CPU. The only limitation is the availability of these GPUs as they are not always available. After training the model, the weights in the neural network can be exported and used for various applications.

The steps taken in the data processing phase will be:

1. Collect and process the tracking data by categorising it by display type and secondary task difficulty.
2. Prepare data used to train the ML model (formatting and labelling).
3. Analyse how well the model is able to detect distractions based on secondary task difficulty.

[1] https://wandb.ai/site

[2] https://www.paperspace.com/

8 Conclusion

The goal of this preliminary report is to provide a theoretical foundation for the main research phase, which is creating a tool for detecting distractions in human pilot control tasks using machine learning. A literature study is performed on current approaches in detecting driver distraction and task demand, as well as anomaly detection techniques. Furthermore, results of several preliminary simulation are presented.

First, definitions of distraction are analysed in literature. In general each definition mentions a source, the location of the source, intentionality, process, and outcome. For the tracking task, a fitting definition of distraction with the goal of generating data of distracted people would be:

A visual secondary task presented to the pilot at a certain angle from the display diverting the attention away from the primary tracking task resulting in an increase of error.

In current studies, researchers have approached detection of driver distraction with vision and sensor based methods. Vision-based methods have been able to reach higher accuracies in general. However, for the tracking tasks it is recommended use a sensor based approach. Furthermore, successful application of machine learning models have been developed for classification problems using time-series data obtained from tracking tasks. The main variables used to train these models are the error e, input u, output x, and its respective derivatives.

Producing data of distracted people in tracking tasks for training models will require some sort of secondary task during runs. According to the Multiple Resource Theory and research done on task demand, the ideal secondary task would be vision based.

Data of distracted people can be considered as anomalies since it does not represent normal behaviour during a tracking task. And taking in mind that only limited data of distracted people is available for this purpose. When following a supervised ML approach using data including distracted people, positive results have been obtained by training a model with the InceptionTime architecture. It is important to keep a reasonable ratio between normal and distracted samples in the data set used for training. And improvements in the results are also achieved by increasing the decision threshold for classifying a sample as distracted.

A

Appendix

A.1. Pursuit run (subject 5, run 7)

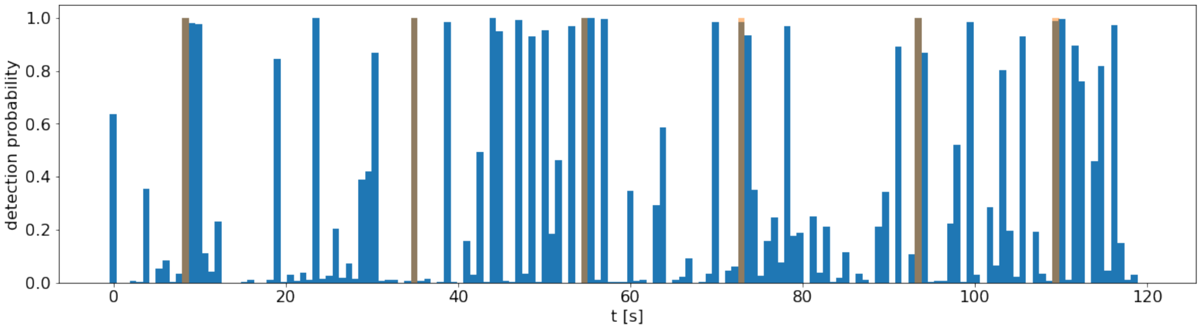


Figure A.1: Detection probability of distractions for a single pursuit run (subject 5, run 7).

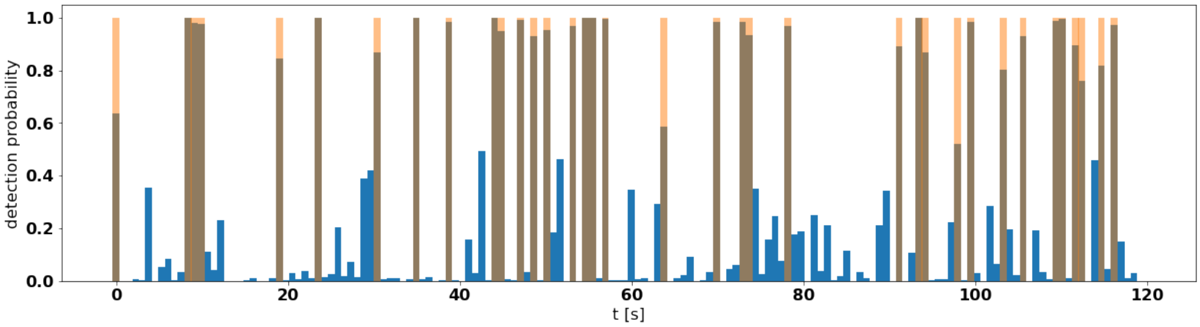


Figure A.2: Detection probability and classification of distractions for a single pursuit run (subject 5, run 7).

A.2. Varying Threshold (preview subject 5, run 7)

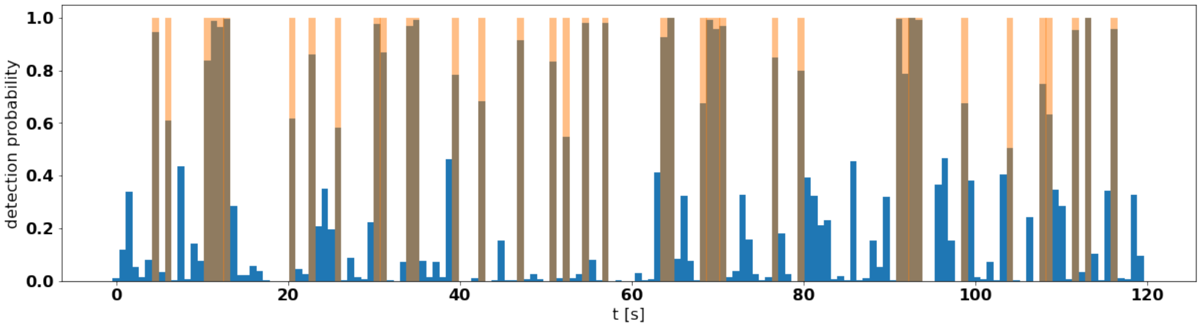


Figure A.3: Detection probability and classification of distractions for a single preview run, decision threshold = 0.5 (subject 5, run 7).

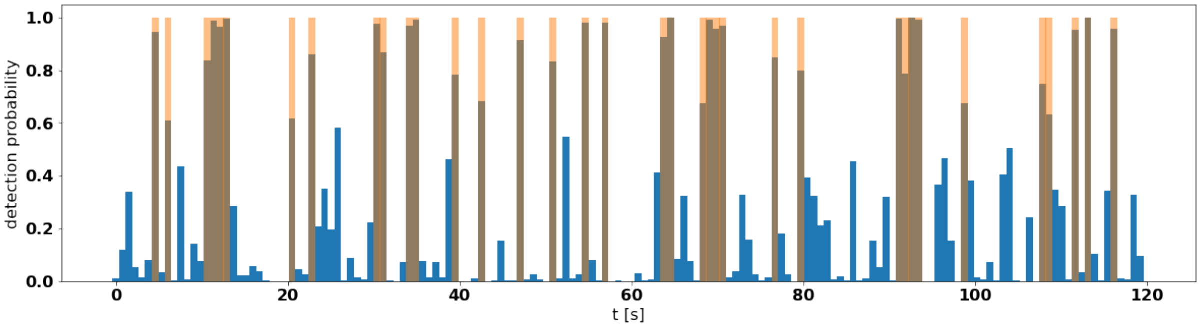


Figure A.4: Detection probability and classification of distractions for a single preview run, decision threshold = 0.6 (subject 5, run 7).

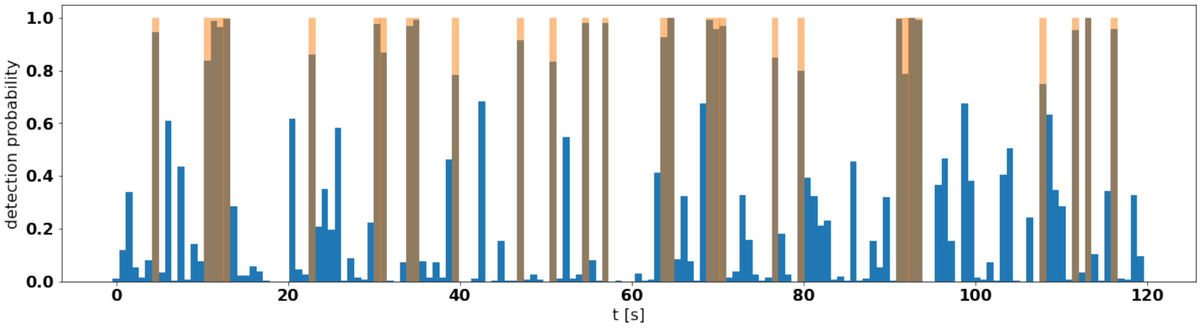


Figure A.5: Detection probability and classification of distractions for a single preview run, decision threshold = 0.7 (subject 5, run 7).

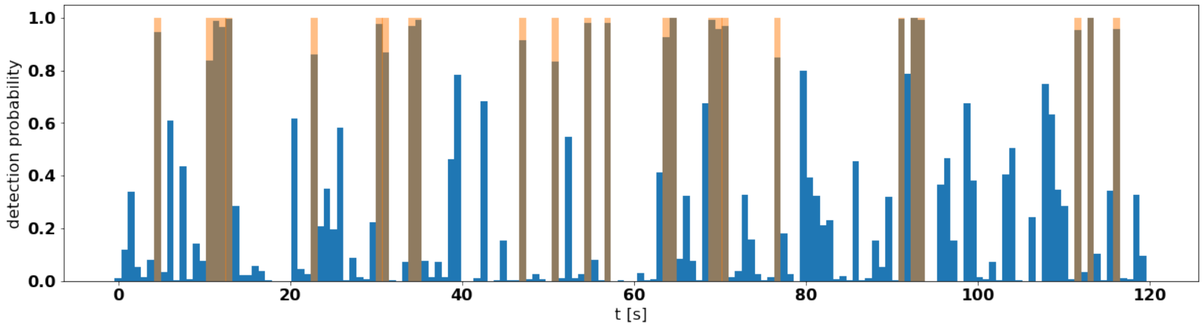


Figure A.6: Detection probability and classification of distractions for a single preview run, decision threshold = 0.8 (subject 5, run 7).

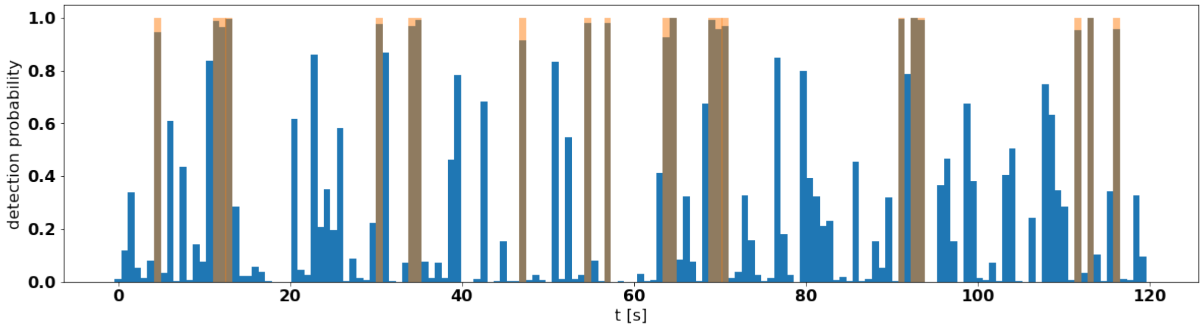


Figure A.7: Detection probability and classification of distractions for a single preview run, decision threshold = 0.9 (subject 5, run 7).

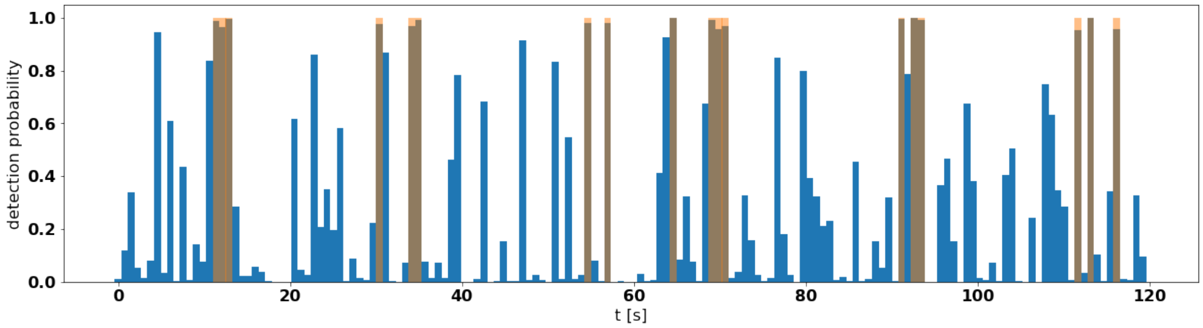


Figure A.8: Detection probability and classification of distractions for a single preview run, decision threshold = 0.95 (subject 5, run 7).

A.3. Varying Threshold (pursuit subject 5, run 7)

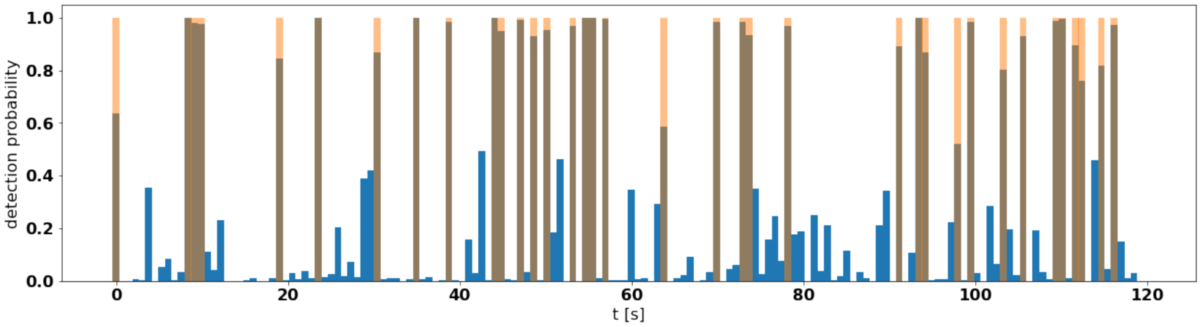


Figure A.9: Detection probability and classification of distractions for a single pursuit run, decision threshold = 0.5 (subject 5, run 7).

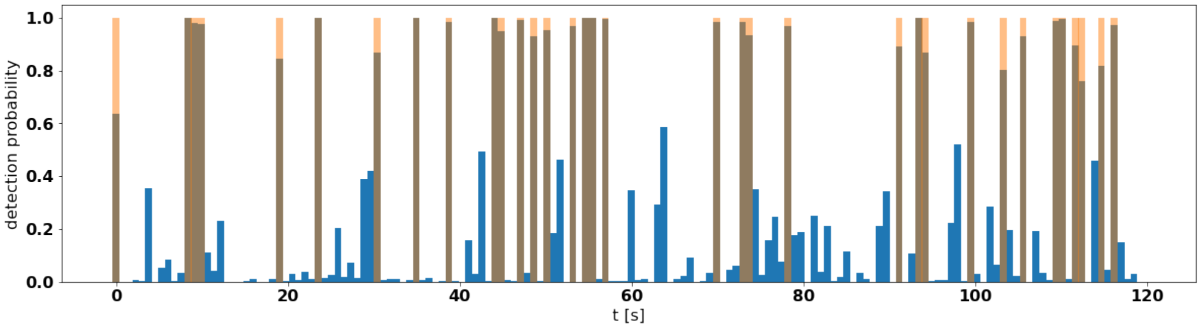


Figure A.10: Detection probability and classification of distractions for a single pursuit run, decision threshold = 0.6 (subject 5, run 7).

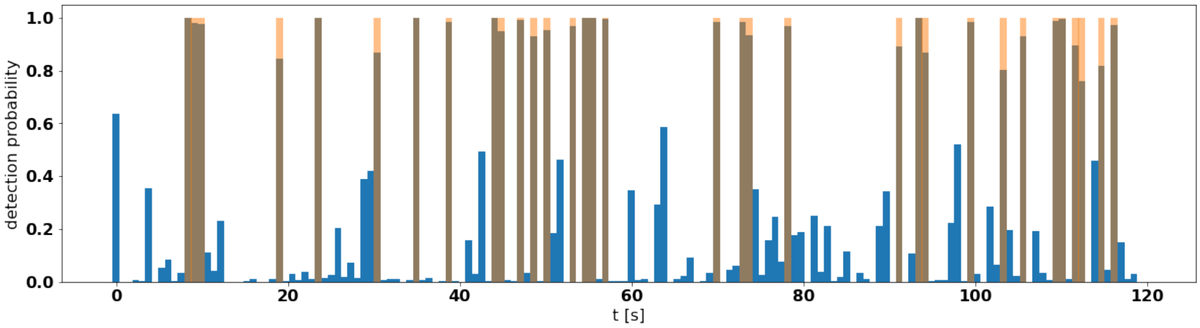


Figure A.11: Detection probability and classification of distractions for a single pursuit run, decision threshold = 0.7 (subject 5, run 7).

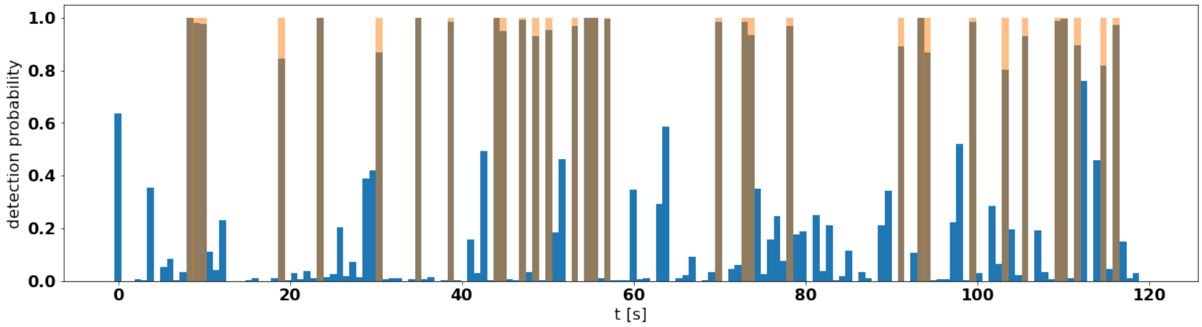


Figure A.12: Detection probability and classification of distractions for a single pursuit run, decision threshold = 0.8 (subject 5, run 7).

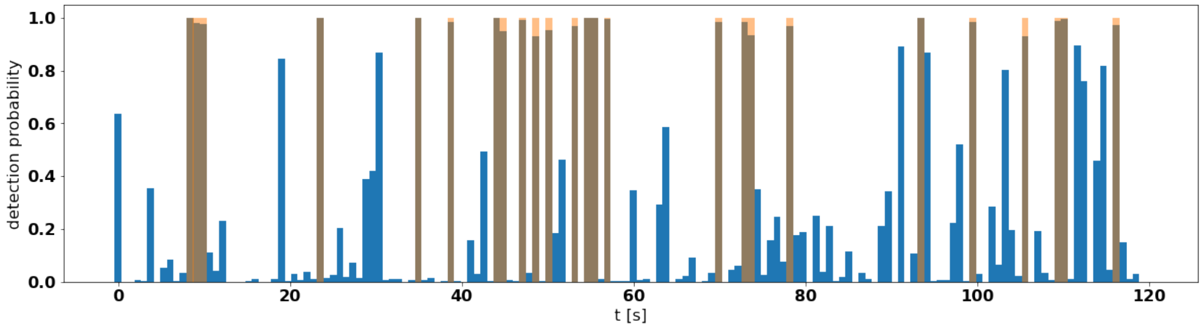


Figure A.13: Detection probability and classification of distractions for a single pursuit run, decision threshold = 0.9 (subject 5, run 7).

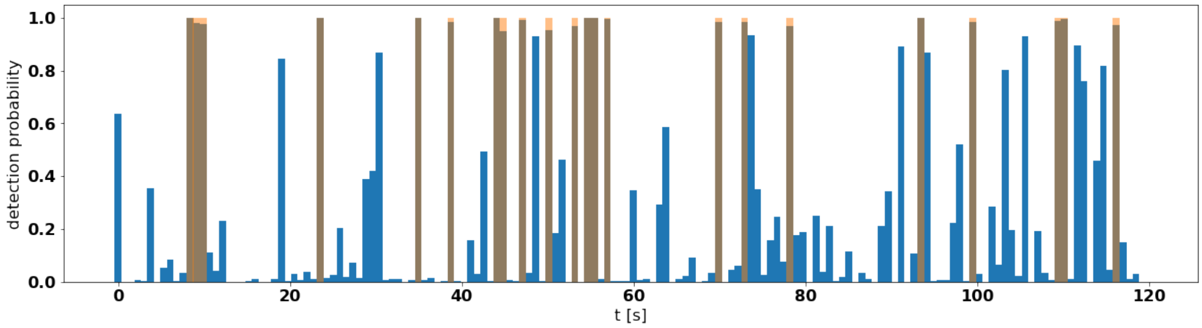


Figure A.14: Detection probability and classification of distractions for a single pursuit run, decision threshold = 0.95 (subject 5, run 7).

A.4. Model Accuracy vs. Decision Threshold (Pursuit)

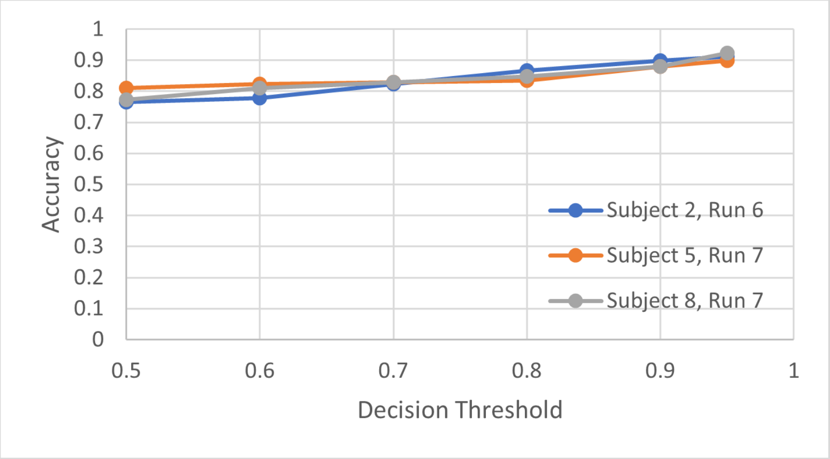


Figure A.15: Model accuracy with varying decision threshold for classifying distracted samples, pursuit data.

Bibliography

[1] E.H.H. Thung A. Muis F.N. Postema. *HMI Lab*. URL: https://cs.lr.tudelft.nl/facilities/hmi-lab/.

[2] Andrei Aksjonov et al. “A Method for Detection and Evaluation of Driver Distraction Induced by In-Vehicle Information Systems”. In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. 2018, pp. 4513–4518. DOI: 10.1109/IECON.2018.8591252.

[3] Andrei Aksjonov et al. “Detection and Evaluation of Driver Distraction Using Machine Learning and Fuzzy Logic”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.6 (2019), pp. 2048–2059. DOI: 10.1109/TITS.2018.2857222.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly Detection: A Survey”. In: *ACM Comput. Surv.* 41 (July 2009). DOI: 10.1145/1541880.1541882.

[5] European Transport Safety Council. *TRANSPORT SAFETY PERFORMANCE IN THE EU A STATISTICAL OVERVIEW*. 2003. URL: https://etsc.eu/wp-content/uploads/2003_transport_safety_stats_eu_overview.pdf.

[6] National Safety Council. *Deaths by Transportation Mode*. 2022. URL: https://injuryfacts.nsc.org/home-and-community/safety-topics/deaths-by-transportation-mode/.

[7] Kasper van der El et al. “An Empirical Human Controller Model for Preview Tracking Tasks”. In: *IEEE Transactions on Cybernetics* 46.11 (2016), pp. 2609–2621. DOI: 10.1109/TCYB.2015.2482984.

[8] Kasper van der El et al. “Effects of Preview on Human Control Behavior in Tracking Tasks With Various Controlled Elements”. In: *IEEE transactions on cybernetics* PP (Apr. 2017). DOI: 10.1109/TCYB.2017.2686335.

[9] Kasper van der El et al. “Effects of Preview Time in Manual Tracking Tasks”. In: *IEEE Transactions on Human-Machine Systems* PP (May 2018), pp. 1–10. DOI: 10.1109/THMS.2018.2834871.

[10] Johan Engström, Emma Johansson, and Joakim Östlund. “Effects of visual and cognitive load in real and simulated motorway driving”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 8.2 (2005). The relationship between distraction and driving performance: towards a test regime for in-vehicle information systems, pp. 97–120. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2005.04.012.

[11] Tulga Ersal et al. “Model-Based Analysis and Classification of Driver Distraction Under Secondary Tasks”. In: *IEEE Transactions on Intelligent Transportation Systems* 11.3 (2010), pp. 692–701. DOI: 10.1109/TITS.2010.2049741.

[12] Ary Goldberger et al. “Components of a new research resource for complex physiologic signals”. In: *PhysioNet* 101 (Jan. 2000).

[13] A. Hamish Jamson and Natasha Merat. “Surrogate in-vehicle information systems and driver behaviour: Effects of visual and cognitive load in simulated rural driving”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 8.2 (2005). The relationship between distraction and driving performance: towards a test regime for in-vehicle information systems, pp. 79–96. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2005.04.002.

[14] Songqiao Han et al. “ADBench: Anomaly Detection Benchmark”. In: *Neural Information Processing Systems (NeurIPS)*. 2022.

[15] Hassan Ismail Fawaz et al. “InceptionTime: Finding AlexNet for Time Series Classification”. In: *Data Mining and Knowledge Discovery* (2020).

[16] M.J.L de Jong. “Classifying Human Pilot Skill Level Using Deep Artificial Neural Networks”. In: *MSc. Thesis* (2021).

[17] Alexey Kashevnik et al. “Driver Distraction Detection Methods: A Literature Review and Framework”. In: *IEEE Access* 9 (2021), pp. 60063–60076. DOI: 10.1109/ACCESS.2021.3073599.

[18] Motohiro Kimura, Kenta Kimura, and Yuji Takeda. “Assessment of driver’s attentional resource allocation to visual, cognitive, and action processing by brain and eye signals”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 86 (2022), pp. 161–177. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2022.02.009. URL: https://www.sciencedirect.com/science/article/pii/S136984782200033X.

[19] Alexander Kiselev et al. “Deep Neural Networks for Classifying the Response of Human Controllers by Display Types”. In: *IEEE on Transactions on Human-Machine Systems* (2022).

[20] George Kountouriotis et al. “Identifying cognitive distraction using steering wheel reversal rates”. In: *Accident Analysis and Prevention* 96 (July 2016). DOI: 10.1016/j.aap.2016.07.032.

[21] Kwei-Herng Lai et al. “TODS: An Automated Time Series Outlier Detection System”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.18 (May 2021), pp. 16060–16062.

[22] Jing Li et al. “Distracted driving caused by voice message apps: A series of experimental studies”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 76 (2021), pp. 1–13. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2020.10.008.

[23] Zhaojian Li et al. “Visual-Manual Distraction Detection Using Driving Performance Indicators With Naturalistic Driving Data”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.8 (2018), pp. 2528–2535. DOI: 10.1109/TITS.2017.2754467.

[24] Yulan Liang and John D. Lee. “A hybrid Bayesian Network approach to detect driver cognitive distraction”. In: *Transportation Research Part C: Emerging Technologies* 38 (2014), pp. 146–155. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2013.10.004.

[25] Yulan Liang, Michelle L. Reyes, and John D. Lee. “Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines”. In: *IEEE Transactions on Intelligent Transportation Systems* 8.2 (2007), pp. 340–350. DOI: 10.1109/TITS.2007.895298.

[26] Yuan Liao et al. “Detection of driver cognitive distraction: An SVM based real-time algorithm and its comparison study in typical driving scenarios”. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 394–399. DOI: 10.1109/IVS.2016.7535416.

[27] Kay Liu et al. “PyGOD: A Python Library for Graph Outlier Detection”. In: *arXiv preprint arXiv:2204.12095* (2022).

[28] Tianchi Liu et al. “Driver Distraction Detection Using Semi-Supervised Machine Learning”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.4 (2016), pp. 1108–1120. DOI: 10.1109/TITS.2015.2496157.

[29] University College London. *Multitasking*. https://wiki.ucl.ac.uk/display/UCLICACS/Multitasking. 2013.

[30] Yanli Ma et al. “Support vector machines for the identification of real-time driving distraction using in-vehicle information systems”. In: *Journal of Transportation Safety & Security* 14.2 (2022), pp. 232–255. DOI: 10.1080/19439962.2020.1774019. eprint: https://doi.org/10.1080/19439962.2020.1774019. URL: https://doi.org/10.1080/19439962.2020.1774019.

[31] Kees Maat and Rob Konings. “Accessibility or Innovation? Store Shopping Trips versus Online Shopping”. In: *Transportation Research Record* 2672.50 (2018), pp. 1–10. DOI: 10.1177/0361198118794044.

[32] Sarfaraz Masood et al. “Detecting distraction of drivers using Convolutional Neural Network”. In: *Pattern Recognition Letters* 139 (2020), pp. 79–85. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2017.12.023.

[33] Anne T. Mccartt, Laurie A. Hellinga, and Keli A. Bratiman. “Cell Phones and Driving: Review of Research”. In: *Traffic Injury Prevention* 7.2 (2006). PMID: 16854702, pp. 89–106. DOI: 10.1080/15389580600651103.

[34] Anthony D. McDonald, Thomas K. Ferris, and Tyler A. Wiener. “Classification of Driver Distraction: A Comprehensive Analysis of Feature Generation, Machine Learning, and Input Measures”. In: *Human Factors* 62.6 (2020), pp. 1019–1035. DOI: 10.1177/0018720819856454.

[35] D.T. McRuer and H.R. Jex. “A Review of Quasi-Linear Pilot Models”. In: *IEEE Transactions on Human Factors in Electronics* HFE-8.3 (1967), pp. 231–249. DOI: 10.1109/THFE.1967.234304.

[36] Barbara Metz, Nadja Schömig, and Hans-Peter Krüger. “Attention during visual secondary tasks in driving: Adaptation to the demands of the driving task”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 14.5 (2011), pp. 369–380. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2011.04.004. URL: https://www.sciencedirect.com/science/article/pii/S136984781100043X.

[37] Masahiro Miwata et al. “Performance Evaluation of an AI-Based Safety Driving Support System for Detecting Distracted Driving”. In: *Innovative Mobile and Internet Services in Ubiquitous Computing*. Springer International Publishing, 2022, pp. 10–17. ISBN: 978-3-031-08819-3.

[38] Kotaro Nakano and Basabi Chakraborty. “Real-Time Distraction Detection from Driving Data Based Personal Driving Model Using Deep Learning”. In: *International Journal of Intelligent Transportation Systems Research* 20.1 (2022), pp. 238–251. DOI: 10.1007/s13177-021-00288-9.

[39] Emma J. Nilsson et al. “Effects of cognitive load on response time in an unexpected lead vehicle braking scenario and the detection response task (DRT)”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 59 (2018), pp. 463–474. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2018.09.026. URL: https://www.sciencedirect.com/science/article/pii/S1369847817306198.

[40] Emma J. Nilsson et al. “On-to-off-path gaze shift cancellations lead to gaze concentration in cognitively loaded car drivers: A simulator study exploring gaze patterns in relation to a cognitive task and the traffic environment”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 75 (2020), pp. 1–15. ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2020.09.013. URL: https://www.sciencedirect.com/science/article/pii/S1369847820305325.

[41] International Transport Forum - OECD. *Road Safety Annual Report 2020*. 2020.

[42] Amit Paul et al. “Steering Entropy Changes as a Function of Microsleeps”. In: Oct. 2017, pp. 441–447. DOI: 10.17077/drivingassessment.1196.

[43] Tibor Petzoldt, Hanna Otto, and Josef Krems. “The Critical Tracking Task: A Potentially Useful Method to Assess Driver Distraction?” In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 56 (June 2014). DOI: 10.1177/0018720813501864.

[44] Thomas A Ranney, W Riley Garrott, and Michael J Goodman. *NHTSA driver distraction research: Past, present, and future*. Tech. rep. SAE Technical Paper, 2001.

[45] Michael A Regan, John D Lee, and Kristie Young. *Driver distraction: Theory, effects, and mitigation*. CRC press, 2008.

[46] Michael A Regan and Oscar Oviedo-Trespalacios. “Driver Distraction: Mechanisms, Evidence, Prevention, and Mitigation”. In: *The Vision Zero Handbook: Theory, Technology and Management for a Zero Casualty Policy*. Springer, 2022, pp. 1–62.

[47] Community Research and Development Information Service. *Human machine interface and traffic safety in Europe*. https://cordis.europa.eu/project/id/9665. 2001.

[48] Patrick Schneider and Fatos Xhafa. “Chapter 3 - Anomaly detection: Concepts and methods”. In: *Anomaly Detection and Complex Event Processing over IoT Data Streams*. Ed. by Patrick Schneider and Fatos Xhafa. Academic Press, 2022, pp. 49–66. ISBN: 978-0-12-823818-9. DOI: https://doi.org/10.1016/B978-0-12-823818-9.00013-4.

[49] Xiuyao Song et al. “Conditional Anomaly Detection”. In: *IEEE Transactions on Knowledge and Data Engineering* 19.5 (2007), pp. 631–645. DOI: 10.1109/TKDE.2007.1009.

[50] David L Strayer. “Is the technology in your car driving you to distraction?” In: *Policy insights from the behavioral and brain sciences* 2.1 (2015), pp. 157–165.

[51] Fredrick M Streff. “Driver distraction, aggression, and fatigue: a synthesis of the literature and guidelines for Michigan planning”. In: (2000).

[52] Tan et al. *Introduction to Data Mining*. May 2005.

[53] Fabio Tango and Marco Botta. “Real-Time Detection System of Driver Distraction Using Machine Learning”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), pp. 894–905. DOI: 10.1109/TITS.2013.2247760.

[54] Kari Torkkola, N. Massey, and C. Wood. “Driver inattention detection through intelligent analysis of readily available sensors”. In: Nov. 2004, pp. 326–331. ISBN: 0-7803-8500-4. DOI: 10.1109/ITSC.2004.1398919.

[55] John R Treat. “A study of precrash factors involved in traffic accidents.” In: *HSRI Research review* (1980).

[56] Anne Treisman. “Features and objects: The fourteenth Bartlett memorial lecture”. In: *The Quarterly Journal of Experimental Psychology Section A* 40.2 (1988), pp. 201–237.

[57] Panagiotis Tsiamyrtzis et al. “Imaging Facial Physiology for the Detection of Deceit”. In: *International Journal of Computer Vision* 71 (Feb. 2007), pp. 197–214. DOI: 10.1007/s11263-006-6106-y.

[58] JA Veltman and AWK Gaillard. “Physiological workload reactions to increasing levels of task difficulty”. In: *Ergonomics* 41.5 (1998), pp. 656–669.

[59] Gert-Jan H.A. Verkerk. “Classifying Human Manual Control Behaviour in Tracking Tasks with Various Display Types Using the Inception Time CNN”. In: *MSc. Thesis* (2021).

[60] Avinash Wesley, Dvijesh Shastri, and Ioannis Pavlidis. “A Novel Method to Monitor Driver’s Distractions”. In: *CHI ’10 Extended Abstracts on Human Factors in Computing Systems*. Association for Computing Machinery, 2010, pp. 4273–4278. ISBN: 9781605589305. DOI: 10.1145/1753846.1754138.

[61] Christopher Wickens. “Multiple resources and performance prediction”. In: *Theoretical Issues in Ergonomic Science* 3 (Jan. 2002), pp. 159–177. DOI: 10.1080/14639220210123806.

[62] Christopher D Wickens et al. *Engineering psychology and human performance*. HarperCollins Publishers Inc., 1992.

[63] Feng Xue and Weizhong Yan. *Multivariate Time Series Anomaly Detection with Few Positive Samples*. 2022. DOI: 10.48550/ARXIV.2207.00705.

[64] Lora Yekhshatyan and John D. Lee. “Changes in the Correlation Between Eye and Steering Movements Indicate Driver Distraction”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2013), pp. 136–145. DOI: 10.1109/TITS.2012.2208223.