Eye Movement Classification Using Deep Learning

Group 1002
Shagen Djanian

Master thesis

Fall Semester 2018 - Spring Semester 2019

Aalborg University
Vision, Graphics and Interactive Systems

Resumé

Denne rapport omhandler hvad et godt datasæt til øjenbevægelses klassificering er og hvordan deep learning håndterer at klassificere øjenbevægelser. En grunding gennemgang er blevet gjort af øjenbevægelses datasættet GazeCom og Lund 2013. Ved at kigge på histogrammer af hastigheden og retning af fikseringer, sakkader, smooth pursuit og post-sakkade oscillering (PSO) har det været muligt at undersøge kvaliteten af annoteringerne i GazeCom. Det viste sig at især den manuelle annotering af sakkader ikke levede op til de fysiologiske egenskaber ved sakkader og dette var et tegn på at datasættet er annoteret forkert. De samme problemer var ikke tilstede i Lund 2013. Derfor blev Lund 2013 valgt til at træne et neuralt netværk til at klassificere fikseringer, sakkader, smooth pursuit af PSO'er. Forskellige modeller blev evalueret for at finde ud af hvilke der er bedst til klassificering. Diverse features blev også brugt for at se hvilke der har den største effekt på klassificeringen. Der blev foretaget en evaluering på både datapunktniveau og på helhedsniveau. Helhedsniveau evalueringen blev implementeret af forfatteren da ingen implementering var tilgængelig. Evalueringen viste at det bedste netværk var et multi resolution neural netværk, men det var ikke markant bedre end de andre netværker der blev testet. Den største udfordring var at skelne mellem fikseringer og smooth pursuit og dette problem blev ikke løst.

AALBORG UNIVERSITY
STUDENT REPORT

Vision, Graphics and Interactive Systems
Aalborg University
http://www.aau.dk

Title:
Eye Movement Classification Using Deep Learning

Theme:
Eye tracking, machine learning, deep learning

Project Period:
Fall Semester 2018 - Spring Semester 2019

Project Group:
Group 1002

Participants:
Shagen Djanian

Supervisor:
Zheng-Hua Tan

Number of Pages: 76

Date of Completion:
June 6, 2019

Abstract:

This projects outlines how to choose a good eye movement dataset and evaluate a deep learning approach to eye movement classification. A thorough investigation of the annotation of the GazeCom dataset was performed. By looking at different feature distributions and event durations of eye movements it was possible to show that the annotations did not comply with the physiological properties of said movements. A similar investigation was performed on the Lund 2013 dataset and this showed that the features were in agreement with physiological properties. A 1D Convolutional Neural Network Bidirectional Long Short-Term Memory (1D-CNN-BLSTM) neural network was trained on the Lund 2013 dataset to classify fixations, saccades, smooth pursuit and post-saccadic oscillation. Different model parameters and eye movement features were tested. The best performing model was a multi resolution 1D-CNN-BLSTM but it was not outperforming the other neural networks by much. The biggest challenge was differentiating between fixations and smooth pursuit and this was not solved.

Contents

Chapter 1

Introduction

Eye tracking is a field that has been in rapid growth over the last 20 years. It has seen success in various academic field like usability analysts, cognitive psychologist, reading research and sports science. It has been used study schizophrenia, reading strategies, and human computer interaction by gaze Holmqvist et al. [2011]. Newer uses of eye trackers are foveated rendering [Patney et al., 2016], biometrics [Friedman et al., 2017] and integration with Virtual Reality [Tobii, 2019]. It has been also been used as assistance technology for people with various disabilities. In other words it's a useful tool which has been helpful in a slew of different area. Companies like Tobii make affordable, easy to use eye trackers that can integrate with modern gaming becoming more of an everyday item. Some of the major producers of eye trackers include Tobii, SensoMotoric Instruments (SMI) and EyeLink. Eye trackers differ depending on their usage. Some operate in low frequencies like 30 Hz while other operate in 1000 Hz. The recording techniques are different and the algorithms that classify the eye movements also often differ from eye tracker to eye tracker and from company to company. They are often proprietary and classify different eye movement. Some only classify saccades and fixations while others might classify micro saccades or smooth pursuit. A survey of 112 eye tracking researchers conducted by Hessels et al. [2018] show that 62% of the researchers use manufacturers classification software while 25% use algorithms described in literature and 58% use self-written software. It was possible to answer with multiple answers. This can make it difficult to compare research as this also changes the definition of the eye movements depending on the algorithm. The same survey also showed that conceptually there is a difference in how researches defined fixations and saccades. In recent years the classification algorithms have started to move threshold based algorithms [Salvucci and Goldberg, 2000] to machine learning based algorithms [Hoppe and Bulling, 2016; Startsev et al., 2018; Zemblys et al., 2018]. This projects focus will be eye movement classification as it is the heart of eye trackers and has implications on the eye trackers usage.

Chapter 2

Eye movement classification

This chapter is a review of the state of the art algorithm for eye movement classification. It contains a summary of the different datasets that are used and their availability, the evaluation metrics for the classification algorithms and the algorithms themselves.

2.1 Eye movements

This project will only concern itself with the four major eye movements; fixation, saccades, smooth pursuit and Post-Saccadic Oscillation (PSO). According to Hessels et al. [2018] definitions of these events differ from researcher to researcher and there does not seem to be a consensus amongst the eye movement community. This section consists of description of the movements and their characteristics. To be as clear as possible a definition of some events is also given.

2.1.1 Fixation

From Leigh and Zee [2004] the definition of a fixation is *"Holds the image of a stationary object on the fovea by minimizing ocular drifts."*. In other words it is when the eye is looking at target that is not moving. The fovea is the part of the eye which has a high concentration of cones which allow for high acuity vision. The fovea takes up a very small portion of visual field, approximately the size your thumbnail if you extend your arm straight from yourself. The rest of the visual field is low acuity vision which means that the eye has to move it self to get an object into the part of the visual field where the fovea is. While the eye is fixating there are small tremors and drifts that occur so the eye never truly still. There is no maximum duration of a fixation but they typically span between 200 - 400 ms [Holmqvist et al., 2011].

2.1.2 Saccade

A saccade *"brings images of objects of interest onto the fovea"* [Leigh and Zee, 2004]. They are very rapid and short movements. Their durations are typically between

between 30-80 ms spanning amplitudes of 4deg - 20deg and having a velocity of 30 deg/s to 500 deg/s [Holmqvist et al., 2011]. They are ballistic in nature and predetermined, meaning that that once the movement has started it doesn't change until it reaches the predetermined end. An example if a saccade can be seen in Figure 2.1. There are also movements called micro saccades which are very small saccades having amplitudes of around 0.16 deg - 0.66 deg [Holmqvist et al., 2011]. Micro saccades are beyond the scope of this project.

2.1.3 Post-Saccadic Oscilation

PSO are small movements that can occur at the end of a saccade. They have previously been called dynamic overshoot/undershoot and glissades but PSO is the term that is currently being used. They are characterized by having a small wobble leading to the fixation after a saccade. The cause of the PSO is not entirely clear as some believe it is caused be the recording equipment Deubel and Bridgeman [1995], others believe it has a neural cause Bahill et al. [1975a] and Holmqvist et al. [2011] believes the eye itself wobbles after a saccade. This disagreement also reflects in the problem that PSOs are a eye movement raters struggle to agree on. Never the less they are events that occur when recording with an eye tracker and they can influence the characteristics of saccades and fixations Nyström et al. [2013]. They are typically very short event lasting 10-40 ms with an amplitude of 0.5deg - 2 deg and having velocities between 20 deg/s to 140 deg/s [Holmqvist et al., 2011]. They also don't occur after every saccade. An example can be seen in Figure 2.1.

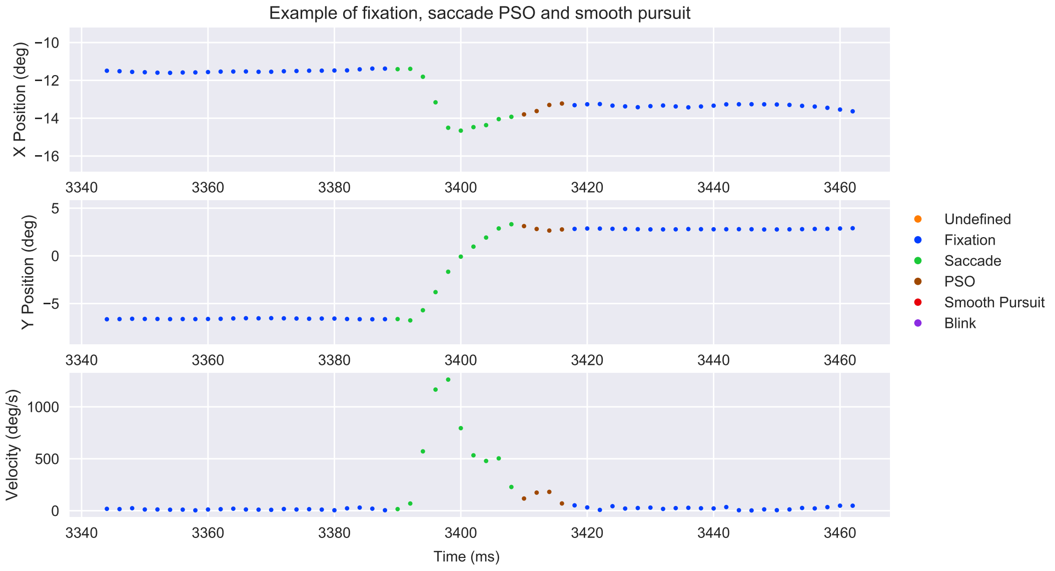


Figure 2.1: An example of a fixation, saccade and PSO in a real recording.

2.1.4 Smooth pursuit

Smooth pursuit is when the eye is following a moving target. It is typically in the range of 10deg $/s$ to 30 deg $/s$ [Holmqvist et al., 2011] but it can reach up to 100 deg $/s$ Meyer et al. [1985] but those movements typically consists of both smooth pursuit and saccades termed catch-up saccades.

2.2 Datasets

The datasets can be split into two main categories. Low frequency eye trackers, 25-30 Hz, and high frequency eye trackers, 300-1000 Hz.

2.2.1 Driving data

There are three datasets that deal with recording eye movements while driving. They all use the Diklabis mobile infrared eye tracker at 25 Hz.

Tafaj et al. [2012] recorded a dataset that was used in a broader study. It contained three groups of subjects; 1. subjects suffering from homonymous visual field defects. 2. subjects suffering from glaucoma 3. control subjects. They do not state the amount of participants used or the amount of data recorded. The subjects were tasked with driving a car while their eye movements were recorded. The dataset does not contain labels and is not publicly available.

Tafaj et al. [2013] recorded 27 subjects driving in an extensive virtual driving simulation. The subjects drove a route of 37.5 km that contained 10 hazardous situations. The situations labelled by manually annotating a bounding box around the object that causes hazardous situation. This is because the study concerns itself with whether the subjects would perceive and react to the presented hazards.

Braunagel et al. [2016] recorded 85 subjects driving and doing secondary tasks in a virtual driving simulation. The subjects drove for 35 minutes and this resulted in 35.5 hours of recording. 1.5 hours of this was manually labelled by two coders resulting in 6623 fixation samples and 1384 saccades samples.

2.2.2 Clinical data

Santini et al. [2015] recorded 6 subjects doing what they described as visual stimuli[1]. A Diklabis mobile infrared eye tracker at 30 Hz was used. The simuli is a dot presented on a uniform background and which will move to produce fixation, saccades and smooth pursuits of varying length, velocity and amplitude. They divided smooth pursuit into circular movement and straight movement. Four datasets were collected from each subject; I) Fixations, saccades, and all possible straight pursuits. II) Fixations and saccades. No pursuits. III) Fixations, saccades, and all circular pursuits. IV) Fixations, saccades, straight and circular pursuits. As the algorithm

[1] Publicly available at http://ti.uni-tuebingen.de/Eye-Movements-Identification.1845.0.html?&L=1

for eye movement Santini et al. [2015] proposed does not use gaze position but pupil position, only the pupil was recorded. The eye tracker was not calibrated before each subject. The datasets were manually coded one coder. This produced 18682 fixation samples, 1296 saccade samples and 4143 smooth pursuit samples. Noise was also classified but only stated as ≈ 1.76% of the entire dataset and not in number of samples. In total approximately 13.4 minutes of recording was recorded.

2.2.3 Hoppe and Bulling

Hoppe and Bulling [2016] recorded 16 subjects with a Tobii TX300 remote eye tracker at 300 Hz. The subjects were presented with dot stimulus, 10 static images, 7 videos and 4 reading task. A subset made up of random amounts of dot stimulus task, one random image viewing task, one random video viewing task and two reading tasks per subject were manually annotated. Number of coders was not stated. This resulted in a total of 400000 labelled samples spanning over 1626 fixation events, 2647 saccades events and 1089 pursuit movements events. The duration of events or sample distribution was not stated. In total approximately 22.2 minutes of labelled recording.

2.2.4 GazeCom

Dorr et al. [2010] recorded 76 subjects with a SR Research EyeLink II eye tracker at 250 Hz. The subjects were spread out over three experiments. First experiment had 54 subjects that were tasked with viewing 18 movies of real-world scenes in and around Lübeck. Second experiment 11 subjects were brought in two days in a row for repeated measures. They watched four Hollywood movie trailers six selected movies from the 18 real-world movies. Third experiment also contained 11 subjects. They were shown nine stop motion movies made from the real-world footage movies in first experiment. Afterwards they were shown static images from the remaining nine movies from the first experiment. Agtzidis et al. [2017] labelled the whole dataset by first automatically labelling the whole dataset and then having manual coders go through it and correct it[2]. Only the data from experiment one was labelled resulting in about 4.3 million samples spread across 38629 fixations events, 39217 saccades events, and 4631 smooth pursuits events. This dataset seems to be the largest publicly available manually labelled dataset and is about ≈4.8 hours of recording. It could be viewed as a benchmark dataset since multiple state of the art eye movement classification algorithms have been tested on the data and their performance is also publicly available.

2.2.5 Lund 2013

Larsson et al. [2013] recorded 31 subjects with a Hi-Speed 1250 eye-tracker from SensoMotoric Instruments at 500 Hz. The subjects were presented with static images,

[2] Publicly available at http://michaeldorr.de/smoothpursuit/

reading, video clips, moving dot stimuli and vertical scrolling text. Only a subset of images, moving dot stimuli and video clip was manually labelled by two raters Marcus Nyström (MN) and Richard Andersson (RA)[3] . The data was annotated into fixation, saccades, PSOs, smooth pursuit, blinks and undefined. The full size of the labelled recordings were 12.75 minutes but Zemblys et al. [2018] used only the image task and reported that it consisted of 151639 samples (303.28 seconds) with 77.78% fixations, 8.93% saccades, 4.96% PSO, 5.03% blinks and 0.17% undefined.

2.2.6 gazeGenNet

Zemblys et al. [2018] used a subset of the Lund 2013 dataset that contained the image viewing task to generate synthetic data. The subset was 43.78 second and contained 85.57% fixations, 10.45% saccades and 3.98% PSOs. These were fed to a sequence-to-sequence Long Short-Term Memory (LSTM) with a Mixture Density Network as an output layer. This made it possible to train a network that would generate 10 second long synthetic recordings. The recordings were then heuristicly augmented to make the signal close to real signal. This entails enforcing different rules e.g. forcing maximum saccade duration, removing too short fixations or saccades, removing saccades of certain amplitudes. After the heuristics the result is ≈5.4 hours of synthetic recording. The dataset is not publicly available but the code for gazeGenNet is[4].

2.2.7 Reading study

Friedman et al. [2018] used a subset from a larger eye-tracking study. 20 subjects were recorded with EyeLink 1000 at 1000 Hz. Only the first 26 seconds of each subject were labelled. Four different algorithms automatically labelled the data into fixation, saccade, PSO, noise or artifact, and unclassified which is publicly available[5], but no manual labelling was done.

A total of of 9 datasets have been reviewed with 5 being publicly available. Only one dataset contained labelling of fixations, saccades, PSOs and smooth pursuit while the rest contained a subset of these movements. The stimuli used in the datasets is varying from real world situations to clinically controlled environments which also makes them difficult to compare to each other. The summary of all the dataset can be seen in Table 2.1.

[3] Publicly available at https://www.humlab.lu.se/en/person/MarcusNystrom/

[4] Publicly available at https://github.com/r-zemblys/gazeGenNet

[5] Publicly available at https://digital.library.txstate.edu/handle/10877/6975

Author	Eye Tracker	Sampling rate (Hz)	No. Subjects	Fixation	Saccades	Smooth Pursuit	PSO	Blink	Noise	Unlabelled	Labelled	Publicly available
Tafaj et al. [2012]	Diklabis	25										
Tafaj et al. [2013]	Diklabis	25	27									
Braunagel et al. [2016]	Diklabis	25	87	x	x						1.5 h	
Santini et al. [2015]	Diklabis	30	6	x	x	x			x		13.4 m	x
Hoppe and Bulling [2016]	Tobii TX300	300	16	x	x	x					22.2 m	
Agtzidis et al. [2017]	EyeLink II	250	54	x	x	x					4.77 h	x
Larsson et al. [2013]	Hi-Speed 1250	500	31	x	x	x	x	x	x		12.75 m	x
Zemblys et al. [2018]†	Hi-Speed 1250	500		x	x		x				5.43 h	x
Friedman et al. [2018]†	EyeLink 1000	1000	20	x	x		x		x	x	8.66 m	x

Table 2.1: Summary of the datasets that have been described in section 2.2. It shows what type of eye tracker was used for recording, the sampling frequency, the manually coded classes, the amount of labelled data and the availability of the dataset. † indicates that the labelling was not done by manual coders.

2.3 Evaluation metrics

There does not seem to be a commonly accepted method among the eye tracking community as to how to evaluate an algorithms performance. This sections contains some of the different evaluation metrics that have been used to evaluate classification algorithms.

2.3.1 Behavioural scores

Komogortsev et al. [2009] introduced metrics to measure accuracy when the presented stimuli is known. Since the stimuli is known an expected accuracy based on physiological traits can be calculated and this can be compared to the classification of an algorithm. Fixation Quantitative Score (FQnS) compares the number of shown fixation stimuli to the number of fixations detected, Equation 2.1. $fixation\_detection\_counter$ is the amount of samples correctly detected as fixation while $stimuli\_fixation\_points$ is the total number of fixation samples. This metric will never reach 100% due to the the brain taking approximately 200 ms to calculate and initiate a saccade.

$$\text{FQnS} = 100 * \frac{fixation\_detection\_counter}{stimuli\_fixation\_points} \tag{2.1}$$

Fixation Qualitative Score (FQlS) compares the proximity of the known stimuli to the positional accuracy of the detected fixation. It is not for measuring classification accuracy but instead a measurement that the subject is actually looking at the presented stimuli. Equation 2.2 takes the distance between the position measured in degrees of the sample labelled as fixation and the position of the presented sample of the stimuli for each sample labelled as fixation and averages over number of samples.

$$\mathrm{FQlS} = \frac{1}{N} * \sum_{\mathrm{i=1}}^{N} fixation\_distance_i \tag{2.2}$$

FQlS is ideally 0° but due to inaccuracies in eye trackers it will never be that low and should preferably remain below 0.5° If the FQlS is too high this is either an indication that the subject was not looking at the stimuli or that something is wrong with the eye tracker. Lastly the Saccade Quantitative Score (SQnS) compares the amount of detected saccades to the number of presented saccade. A saccade is defined here as a jump from one fixation stimuli to another fixation stimuli. The distance between the two fixations in the stimuli is added to *total_stimuli_saccade_amplitude* while the distance between detected fixations is added to *total_detected_saccade_amplitude*, Equation 2.3

$$\mathrm{FQnS} = 100 * \frac{total\_detected\_saccade\_amplitude}{total\_stimuli\_saccade\_amplitude} \tag{2.3}$$

This value can be above 100% if behaviour such as PSO or overshooting is present.

2.3.2 Hierarchical error rules

Friedman et al. [2018] developed a method to manually inspect the labelling from an algorithm and the raw signal to determine the accuracy of the algorithm. A hierarchical rule set of 32 types of error was created. The rater is presented with the signal and algorithm classification and will decide if the event is detected correctly or classified as something else and if the timing is too early or too late. This rule set was developed for 1000 Hz data and is difficult to generalize to other sampling frequencies as some of the rules are defined by the number of samples. It is also very time consuming and difficult to replicate. Because of the hierarchical nature of the rules some errors will never be present because they will already be classified by a different type i.e. saccades timing errors are labelled before PSO therefore the errors will increase in saccades.

2.3.3 Global metrics

Accuracy defined as Equation 2.4 shows a simplistic picture of the algorithms performance which can be lacking.

$$\mathrm{Accuracy} = \frac{\mathrm{True\ Postitive + True\ Negative}}{\mathrm{Positive + Negative}} \tag{2.4}$$

Since there is naturally a much higher occurrences of fixation in the signal, classification of fixations will increase the overall accuracy but will not be very informative of how well the algorithm would perform in the other classes. This is sometimes known as the accuracy paradox. To consider how the classification does with fairer representation multiple metrics have been found during the literature review. An overview of a metrics occurrence can be seen in Table 2.2

Authors	**Accuracy**	**F1- Score**	**Specificity**	**Recall**	**Precision**	**Cohen's Kappa**
Tafaj et al. [2012]						
Tafaj et al. [2013]	X		X	X		
Santini et al. [2015]	X		X	X		X
Braunagel et al. [2016]		X		X	X	
Hoppe and Bulling [2016]	X	X		X	X	
Startsev et al. [2018]		X				X
Zemblys et al. [2018]		X				X

Table 2.2: An overview of how many times a metric has been reported in the reviewed literature.

Precision

Precision was used by [Braunagel et al., 2016; Hoppe and Bulling, 2016] and is defined as Equation 2.5.

$$\text{Precision} = \frac{\text{True Postitive}}{\text{True Positive} + \text{False Positive}} \tag{2.5}$$

Where True Positive (TP) are samples that have been classified as belonging to the class that it actually belonged to. False Positive (FP) are samples that have been classified to belong to the class when they actually belonged to a different class. Precision, also called Post Predictive Value (PPV), represents the ratio of TP samples from the total amount of positively classified samples. In other words; of all the samples that were classified as positive how many were actually correct.

Specificity

Specificity was by [Chen and Epps, 2013; Chen and Chien, 2015] and is defined as Equation 2.6.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}} \tag{2.6}$$

Where True Negative (TN) are samples that have been classified as not belonging to the class correctly. False Negative (FN) Are samples that have been wrongly classified as not belonging to the class, when they did belong to it. Specificity, also known as selectivity or True Negative Rate (TNR), represents the ratio of TN samples from the total amount of negative classified samples. In other words; of all the samples classified as negative how many where actually negative.

Recall

Recall was used by [Tafaj et al., 2013; Chen and Chien, 2015; Braunagel et al., 2016; Hoppe and Bulling, 2016] and is defined as Equation 2.7.

$$\text{Recall} = \frac{\text{True Postitive}}{\text{True Positive} + \text{False Negative}} \tag{2.7}$$

Recall, also known as sensitivity, hit rate, and True Positive Rate (TPR), represents the ratio of classified TP samples from actual total amount of samples belonging to that class. In other words; from the total amount of samples belonging to a class how many were correctly classified.

F1 score

The F1 score was the most commonly used metric [Braunagel et al., 2016; Hoppe and Bulling, 2016; Startsev et al., 2018; Zemblys et al., 2018]. It is defined as Equation 2.8.

$$\text{F1} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{2.8}$$

It is a harmonic mean so when using the F1 score both recall and precision have to be high for the resulting metric to be high. This means that it does not suffer from the accuracy paradox. This makes global comparison easier.

Cohen's Kappa

Cohen's Kappa is a measurement between how similarly two raters rate the same signal. It is used to compare how much manual raters agree when rating an eye movement signal. Another use is to compare how much an algorithm agrees with a manual rater. Cohen's Kappa was used by [Chen and Chien, 2015; Startsev et al., 2018; Zemblys et al., 2018]İt is defined as Equation 2.9

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{2.9}$$

where p_o is the relative observed agreement among the raters. It can be thought of as an accuracy because it represents the fraction of samples that the voters agree on relative to total number of samples. p_e the hypothetical probability that the raters would agree by chance. It is given by Equation 2.10

$$p_e = \frac{1}{N^2} \sum_k n_{\text{k1}} n_{\text{k2}} \tag{2.10}$$

where N is number of samples, k is number of classes and n_{ki} is number of times the rater i predicted category k. The metric lies between a value of 0 to 1 with 0 being no agreement and 1 being total agreement. The Kappa is commonly interpreted as such:

- 0.1 – 0.20 = slight agreement.
- 0.21 – 0.40 = fair agreement.
- 0.41 – 0.60 = moderate agreement.
- 0.61 – 0.80 = substantial agreement.
- 0.81 – 0.99 = near perfect agreement

2.3.4 Sample and Event level difference

A common evaluation of the signal in the literature has been to look at both the sample level and event level. The difference between them is that a sample is one single point of a recording while an event is a number of consecutive points that form the event that eye perform, e.g. fixation. While it is straight forward to segment the signal into event it is not straight forward to match an predicted events to events from the ground truth. Hoppe and Bulling [2016] have used the naive approach of simply using the majority as an indicator of which event to match. So an event from the ground truth is matched with the event from prediction that has the majority of samples that occur during ground truth. They then reported the confusion matrix on the event level.

Hooge et al. [2017] used a different method of event matching. An event is matched with the earliest event that overlaps it. It is then labelled as a hit if they are both the same class. If an event from the predicted stream is not matched with an event from the ground truth because the ground truth is matched with an earlier event, it is labelled as a false alarm (FA). If an event from ground truth end up being unmatched it is labelled as a miss. The different scenario can be seen depicted in Figure 2.2 from Hooge et al. [2017]. These are then used to calculate an event level F1 score defined as Equation 2.11

$$\text{F1} = \frac{2 * \#\text{Hits}}{2 * \#\text{Hits} + \#\text{Misses} + \#\text{False Alarms}} \tag{2.11}$$

where #Hits can be thought of as TP, #Misses as FN and #False Alarms as FP. They also introduce two other metrics derived from this method of event matching called Relative Timing Offset (RTO) and Relative Timing Difference (RTD). RTO is the mean of the offsets between events while RTD is the deviation of the same distribution. These are calculated in two passes, one for the beginning offset of events and one for the ending offset of events. To match the ends of event the matching algorithm is simply flipped and the matching is done from reverse. The paper does not specify on what pass the F1 score is calculated, which is important since this matching would produce asymmetric F1 score.

Zemblys et al. [2018] tried to combine Hoppe and Bulling [2016] and Hooge et al. [2017]. This is done by using the same classifications of TP, FN and FP but instead of matching with the earliest event they match the ground truth with the event that has the largest overlap.

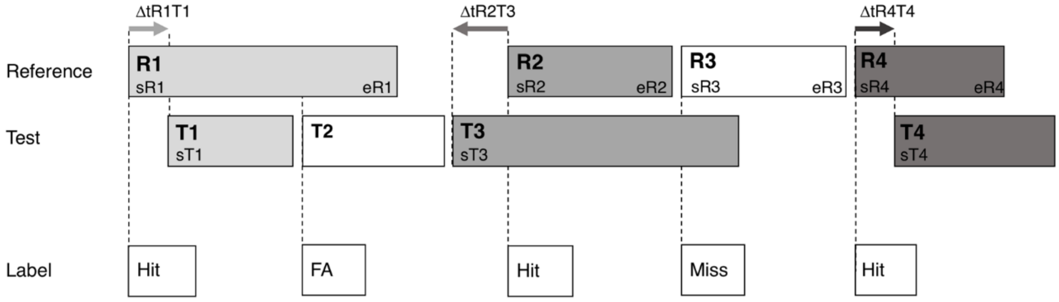


Figure 2.2: Event matching used in Hooge et al. [2017]. Reference is the ground truth and test is an algorithms output. The figures is taken from Hooge et al. [2017].

Startsev et al. [2018] tried to improve Hooge et al. [2017] by using a metric from computer vision called Intersection over Union (IoU). It is a measurement of how much two areas overlap defined as Equation 2.12 which ranges from 0 to 1:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2.12}$$

When matching with the first event that occurs it is used as a threshold. Only events with an IoU of 0.5 are registered as hits so that two events cannot be matched with the same ground truth. Instead of calculating the RTO and RTD they use the average IoU as a an overall metric for a class. For this FP, FN and hits with with and IoU below 0.5 are set to an IoU of 0.

2.4 Algorithms

The common approaches to eye movement classification algorithms have traditionally been threshold based or dispersion based. Newer approaches have tried to utilise statistical model and machine learning to improve classification. This section contains the explanation of some modern eye move classification algorithm.

2.4.1 Classical Algorithms

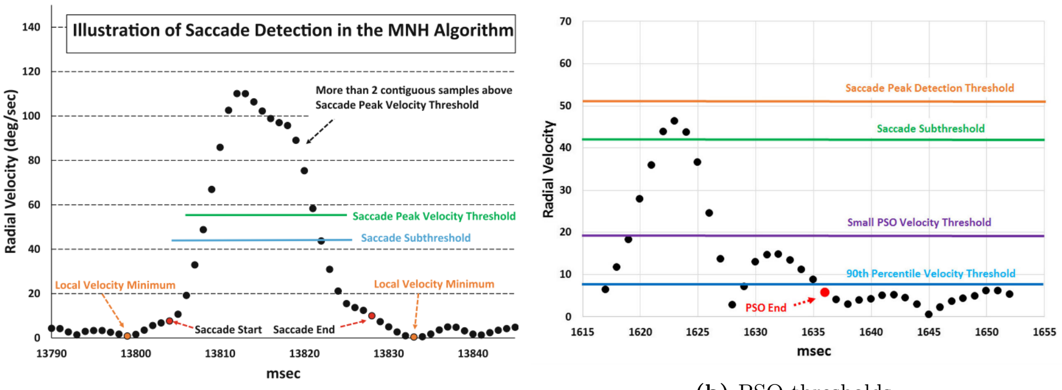
Velocity-Threshold Identification (I-VT) is a classical thresholding algorithm. The instantaneous velocity is used and an empirically chosen threshold is applied. Everything below the threshold is classified as fixation and everything above as saccade. Dispersion-Threshold Identification (I-DT) is a dispersion based algorithm that uses the gaze position. A moving window is applied to the signal and when the dispersion of the window crosses an empirically set threshold threshold everything in the window is set to fixation. Samples then not marked as fixation are marked as saccade. Hidden Markov Model Identification (I-HMM) is probabilistic approach where a two state Hidden Markov Model (HMM) uses the instantaneous velocity to learns the distributions and transitional probabilities of fixations and saccades [Salvucci and Goldberg, 2000].

The classical approaches mainly classify saccades and fixations, as smooth pursuit is a difficult movement to classify. Velocity and Velocity Threshold Identification (I-VVT) is an expanded I-VT that tries to classify smooth pursuit. Here two velocity based threshold are used; T_{vs} and T_{vp}. Everything above T_{vs} is classified as saccades, unclassified samples below T_{vp} and the rest are classified as smooth pursuit. Velocity and Movement Pattern Identification (I-VMP) uses the same approach as I-VVT to classify the saccades. For the unclassified points an empirically selected temporal window is applied. Inside the window the angle between each adjacent point and the horizontal axis is computed. These angels are mapped to the unit circle and if the mean of these angles is above an empirically chosen threshold they are marked as smooth pursuit. Velocity and Dispersion-Threshold Identification (I-VDT) has the same approach as I-VVT for computing the saccades. On the remaining points a dispersion threshold is applied as in I-DT and the points below the threshold are marked as fixation and above as saccades [Komogortsev and Karpov, 2013].

A more recent algorithm using similar approaches to the classical is the Modified Nyström and Holmqvist (MNH) by Friedman et al. [2018] which classifies fixations, saccades and PSOs. It is velocity based, but the velocity is smoothed by a Savitysky-Golay filter and transformed to a radial velocity using Equation 2.13 where Vel_x is the horizontal velocity and Vel_y is the vertical velocity.

$$\text{radial velocity} = \sqrt{Vel_x^2 + Vel_y^2} \tag{2.13}$$

Empirically set velocity and acceleration thresholds are used to remove noise and artifacts. A saccade peak velocity threshold of 55 deg/s are used to identify potential saccades and a saccade subthreshold of 45 deg/s to identifying when to search for a local minimum. Because the Sovitsky-Golay filter introduces a delay into the velocity the beginning and end of a saccade is set to be 5 samples before or after the local minimum. These thresholds can be seen depicted in Figure 2.3a. To classify PSOs the only adaptive threshold in the algorithm is used; 90th percentile of velocity noise distribution. The first 5 consecutive samples that appear below that threshold mark the end of the PSO. The PSOs are then categorised as small, moderate and large depending on if they cross the saccade peak velocity threshold, the subsaccade peak velocity threshold or the small PSO velocity threshold which set at 20 deg/s, respectively. The PSO thresholds are depicted in Figure 2.3b. Fixations are then whatever has not been classified as saccade, PSO or noise or artifact. The MNH has been hand crafted specifically for reading tasks using the EyeLink 1000 eye tracker at 1000 Hz.



(a) Saccade thresholds.

(b) PSO thresholds.

Figure 2.3: A depiction of how the thresholds used for saccade and PSO detection of the MNH taken from [Friedman et al., 2018].

2.4.2 Bayesian Mixed Model

Introduced by Tafaj et al. [2012] the Bayesian Mixed Model (BMM) is a machine learning based approach designed for classification of a live signal from a low frequency recording of 25 Hz. It was developed to be used for assistance during driving and was tested used driving data, section 2.2.1. The instantaneous velocity is used to learn the parameters of Gaussian distributions representing fixations and saccades, seen in Equation 2.14

$$p(x) = \sum_{\mathrm{i=1}}^{K} \phi * \mathcal{N}(x|\mu_i, \sigma_i) \tag{2.14}$$

with ϕ being the prior of the distribution and μ being the mean and σ is the standard deviation. This is done by letting the recording run for 200 samples which correspond to 8 seconds and using Expectation-Maximization (EM) to learn the parameters. Whenever a new sample is recorded the parameters are recalculated. Braunagel et al. [2016] claimed that the BMM adapted too slowly and could not handle changes in the underlying distributions which could be cause by i.e. the subject suddenly doing a secondary task. They proposed a new algorithm Moving Estimate Classification (MERCY) which uses weights and threshold to decide how to update the parameters. When a new sample is predicted new parameters are learned as with BMM. A second set of parameters are estimated by using Equation 2.15

$$\mu_{\mathrm{k_{n+1}}} = \frac{\omega\mu_{\mathrm{k_n}} + \upsilon_{\mathrm{n+1}}}{\omega + 1} \tag{2.15}$$

where $\mu_{\mathrm{k_{n+1}}}$ is the estimated mean,$\mu_{\mathrm{k_n}}$ is the previous mean, ω is the size of the weight and $\upsilon_{\mathrm{n+1}}$ is the velocity of the newest sample. The standard deviation and prior is estimated in a similar fashion. If the distance between the learned parameters and the estimated parameters are larger than a threshold l, which they set to zero, the estimated parameters are used for Gaussians. This causes the model to be able to handle shifts in the underlying distribution and the ω decides how fast or slow the algorithms adapts, which was chosen to be $\omega = 10$.

Tafaj et al. [2013] expanded BMM to classify smooth pursuit. Principal Component Analysis (PCA) together with an empirically set threshold t is determine if the last k fixation samples are classified as fixation or smooth pursuit, Equation 2.16.

$$\frac{\sigma_2^2 * ||u_2|}{\sigma_2^1 * ||u_1|} = \frac{\sigma_2^2}{\sigma_2^1 *} =< t \tag{2.16}$$

Here σ_1 and σ_2 are the largest and second largest eigenvalues and u_1 and u_2 are the corresponding eigenvectors. The logic being that smooth pursuit would be classified as fixations by BMM since it's velocity is usually slower than saccades. The variation between the samples should be small for fixations while smooth pursuit would have larger ions.

Santini et al. [2015] added classification of smooth pursuit to BMM with a different approach. Fixations and saccades are classified the same way as BMM while the probability of smooth pursuit is calculated by Equation 2.17

$$p(sp) = r_i = \frac{1}{N_w} \sum([W_i > 0]) \tag{2.17}$$

where r_i is the movement ratio over the window W_i which contains the last N velocities. $[W_i > 0]$ means that velocities above zero counted as 1 and velocities below as 0. This produces a value between 0 and 1 depending on how much movement there is during the window which reflects the probability of smooth pursuit $p(sp)$ as occurring. The window size was set to be 1.5 times maximum saccade duration which they claim is 80 ms.

2.4.3 Deep Learning

Deep learning has been used in various fields which great success but to the field of eye movement classification it is relatively new. To the authors knowledge currently three papers using deep learning for this task with the earliest being Hoppe and Bulling [2016]. They claim to use an end-to-end Convolutional Neural Network (CNN) architecture to classify a single eye movement sample at a time from a 30 sample window containing the horizontal and vertical gaze position. The network is not end-to-end though since the input of the network is the Fast Fourier Transform (FFT) of the 30 second sample window. The architecture consists of a convolution layer of size 10x1, maxpooling, a fully connected layer and output layer of three classes. It classifies fixations, saccades and smooth pursuit and was trained with the dataset described in section 2.2.3.

Zemblys et al. [2018] proposed a architecture they named gazeNet to do sequence-to-sequence classification. The network consists of two convolutional layers with kernel sizes of 2x11 followed by three bi-directional Gated Recurrent Unit (GRU) layers and a fully connected layer at the end. The network classifies fixation, saccade and PSOs. The network was trained with synthetic data generated by gazeGenNet, section 2.2.6. It was compared with several state-of-the-art algorithms for PSO detection including the MNH and seemed to outperform them on multiple datasets.

Startsev et al. [2018] a 1D-CNN with BLSTM sequence-to-sequence network that classifies fixations, saccades and smooth pursuit. It has three convolutional layers with kernel size of 3x1 followed by a fully connected time distributed layer, BLSTM layer with 16 units, fully connected time distributed layer and an output layer. The output layer has 5 classes as it also classifies noise and unknown. As input different features and their combination were tested; gaze position (x and y coordinates), speed, acceleration and direction. The best combination was speed and direction. Different input sizes were also tested with a input window corresponding to 1 second of recording being the best. It was trained on the GazeCom dataset, section 2.2.4. Various feature combinations were compared to several state-of-the-art fixation and saccade only detections algorithms as well as some smooth pursuit algorithms on the GazeCom dataset. Most feature combinations were either competitive or outperformed the competition with the best neural network results having an average F1 score of 0.830 and the best non deep learning algorithm being 0.769. On other datasets it also performed competitively but not outperforming as much as with GazeCom.

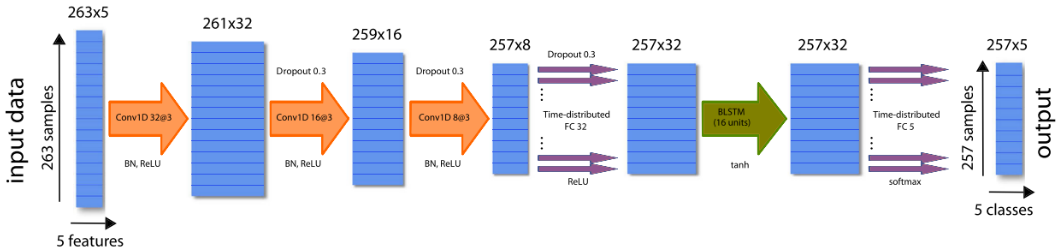


Figure 2.4: The 1D-CNN with BLSTM architecture, figure from Startsev et al. [2018].

2.5 Problem statement

Eye movement classification is not a field that has been figured out and there seems to be room for improvement, especially in regards to PSOs and smooth pursuit. There does also not seem to be a common agreement of how to event evaluate the algorithms which can make it difficult to compare their performances. Deep learning is a relatively new approach in this field and not much research has been done on the topic. One of the limitations of deep learning is that is requires large amounts of labelled data, which there is unfortunately not a lot of publicly available. It is very time consuming and resource exhaustive to manually score eye movement recoding and there seem to be disagreement in the community about the specifics of scoring. The largest available manually annotated gaze eye movement gaze dataset is GazeCom which contains around ~4.7 hours of recording with the second largest being the Lund 2013 dataset at ~12.75 minutes. GazeCom also seems to be the only dataset with publicly available benchmark scores for different algorithms on it. The few networks that do exist seem to perform competitively or even outperform the state-of-the-art non deep learning algorithms. In all three instance of deep learning there were no exploration as to how the networks actually labelled the movements, only global metrics were reported. The problem statements, based on the literature review in this chapter, for this project is:

> *What are the characteristics of a good eye movement dataset?*
> *How well do deep learning algorithms perform eye movement classification?*

Chapter 3

Dataset exploration and selection

The largest available dataset is GazeCom [Dorr et al., 2010]. Unfortunately the GazeCom dataset was found to be lacking in quality. This chapter shows the shortcomings of GazeCom and proposes to use the Lund 2013 dataset instead. GazeCom's classification methodology and examples are shown together with their influence on the features extracted from the dataset. The same is done for the Lund 2013 dataset.

3.1 GazeCom features

The set contains recordings of 18 videos with 47 subjects in each. The experiment for this dataset is described in section 2.2.4. The subjects were seated 45 cm from a looking at screen of size 40 cm by 30 cm with images at a 1280 by 720 resolution. The stimulus covered 48 by 27 degrees of visual angle, with 1 degree corresponding to about 26.7 pixels [Dorr et al., 2010]. Binocular calibration was performed but only monocular data was recorded which resulted in a mean validation error of 0.62 deg across subjects. The recordings are stored in the ARFF format which is easily accessible through Python. In the metadata of the recording the specifications of the experiment are stored. The dataset contains the raw gaze positions, ground truth labels, the outputs from different eye movement algorithms and extracted features. The ground truth contains the time in microseconds, x (horizontal) and y (vertical) position as gaze position on the screen, the confidence of the eye tracker and the scoring of each scorer and a final combined scoring. The dataset contains 5 classes unknown, fixations, saccades, smooth pursuit and noise labelled from 0 to 4 in the respective order.

The features extracted are the velocity, direction and acceleration of the signal. After they have been extracted the gaze position, velocity and direction are converted to pixels per degree by dividing each sample by an average pixel per degree value calculated by Equation 3.1.

$$ppd = \frac{\left(\frac{width(px)}{2*\text{arctan}\left(\frac{width(mm)}{2*distance(mm)}\right)} * \frac{180}{\pi}\right) + \left(\frac{height(px)}{2*\text{arctan}\left(\frac{height(mm)}{2*distance(mm)}\right)} * \frac{180}{\pi}\right)}{2} \tag{3.1}$$

Velocity is defined as Equation 3.2, direction as Equation 3.4 and acceleration as Equation 3.4:

$$v = \frac{\sqrt{\Delta x^2 + \Delta y^2}}{\Delta time} \tag{3.2}$$

$$dir = \arctan \frac{\Delta y}{\Delta x} \tag{3.3}$$

$$acc = \sqrt{\left(\frac{\Delta v_x}{\Delta time}\right)^2 + \left(\frac{\Delta v_y}{\Delta time}\right)^2} \tag{3.4}$$

where Δx is the difference between the two horizontal gaze samples and Δy is the difference between two vertical gaze samples. $\Delta time$ is the difference between two samples used. v is in $\deg/s$. The direction is the angle in radians between the horizontal plane and the sample, so it ranges between $-\pi$ and π. Acceleration is $\deg/s^2$ and Δv_x and Δv_y is the difference between two horizontal or vertical velocity samples respectively. Each feature has been computed using different window sizes which dictate the temporal distance between the two samples used in the feature extraction. The window sizes corresponded to 4, 8, 16, 32 and 64 ms.

3.1.1 Classification methodologgy

It has been a bit unclear as to what criteria exactly Agtzidis et al. [2017] used to determine the different classes, as the paper focused more on software used for the manual labelling or handscoring as they called it. The recordings were automatically labelled by three algorithms; SP-DBSCAN, I-VVT and I-DT where a majority vote was used to determine the label of a sample. This labelling was then presented and the rater could make adjustments to it. Two raters were used for this process and their labels are present in the ground truth dataset for GazeCom. Agreement between the two raters has also not been reported. A final label is then produced to be the ground truth but it is unclear exactly how this was decided. The paper claimed that a 20 second recording took between 3 and 5 minutes to label.

3.1.2 Class examples

Since no clear classification criteria were specified this section contains examples of how the recordings and their classes look. Multiple recordings have been manually inspected and the presented figures here are representative of the general trends of the labelling. Figure 3.1 shows a whole recording plotted. The range of horizontal and vertical values should not go below zero, since $(0, 0)$ marks the bottom left corner

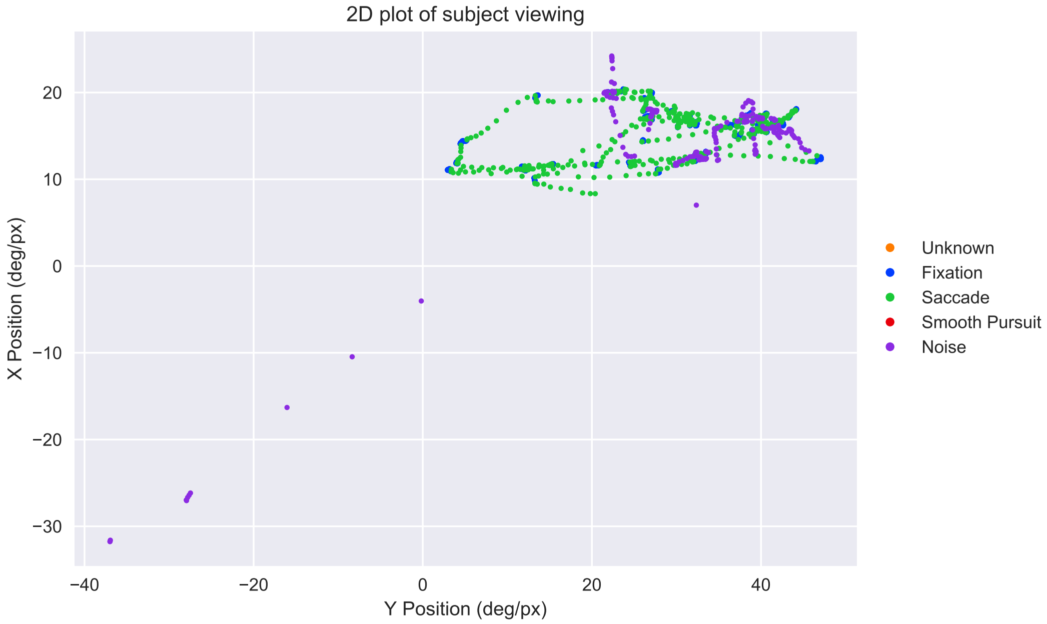


Figure 3.1: The 2D plot of subject JJK viewing the beach recording.

of the screen the subject was looking at. The negative values have been marked as noise. Looking at a zoomed view of the horizontal position, see Figure 3.2, the noise class contains multiple different things. The negative degrees seem like artefacts from the eye tracker "losing" track of the eye. The noise also contains what looks like fixations and saccades. It seems that samples before and after the eye tracker loses track are also classified as noise even though they could be classified as fixations or saccades

Figure 3.3 shows examples of some saccades. It is quite clear that the saccades are not rigorously marked as the some samples which should have been labelled as fixations are marked as saccades especially the ending of the saccade near 12200 ms. There are also what appear to be a PSO at the end of the first saccade. Agtzidis et al. [2017] have chosen not to mark PSOs even though it is clear in the velocity trace that they do occur in the recordings.

Figure 3.4 shows examples of a smooth pursuit in the recordings. Both smooth pursuits have been broken off in the middle by a saccade, which looks like a catch up saccade. While the second smooth pursuit one has a clear saccade in between them, the first one could just as easily have been normal jitter, especially since the end of that small saccade looks very stable so the actual length of the saccade looks to be about 3 samples. This inconsistent ending labelling occurs multiple times in the segment, especially the ending of the first saccade. Even when viewed on a larger time axis it is visible that the saccades are marked into what should have been fixations. A case of noise is also present where the eye tracker loses tracks and drops

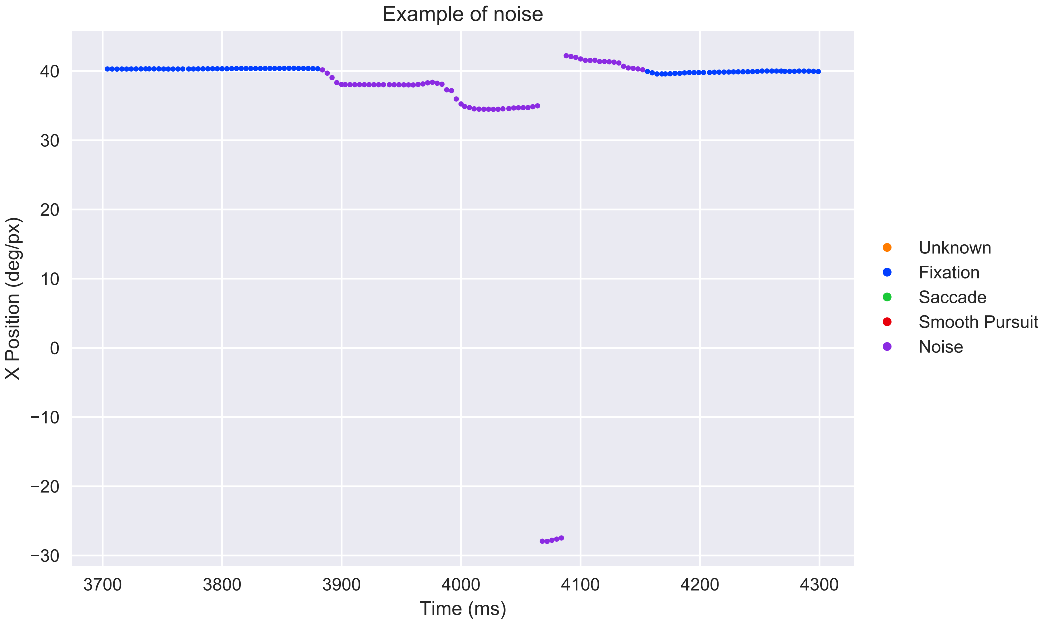


Figure 3.2: An example of a noise event.

the positional signal to zero. The jittery signal surrounding the zeros are also marked as noise. At around 4700 ms there appears the be a movement that has been marked as fixation but does not appear as either fixation, saccade or smooth pursuit.

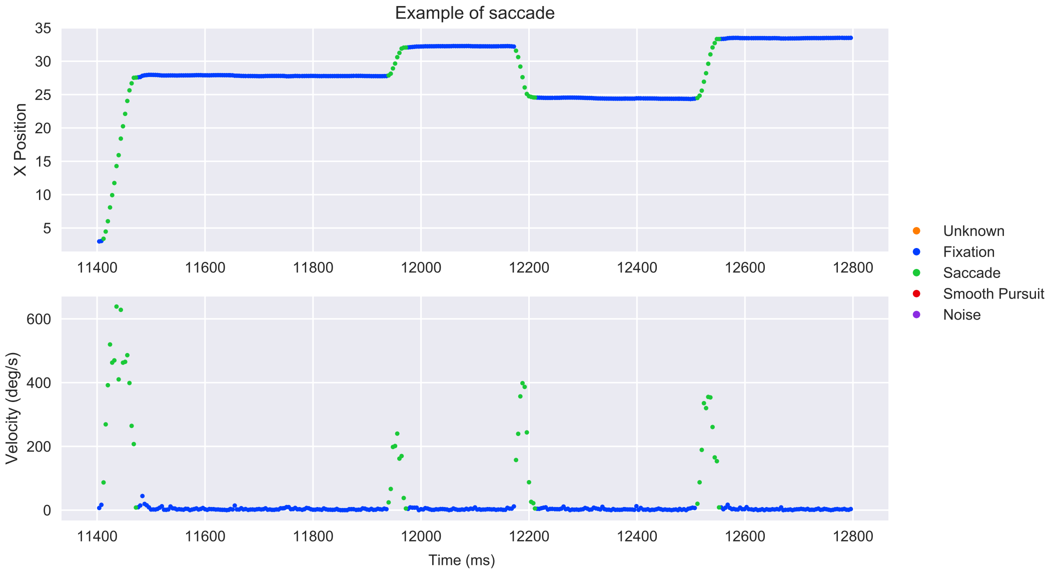


Figure 3.3: An example of saccades in the beach recording of subject JKK.

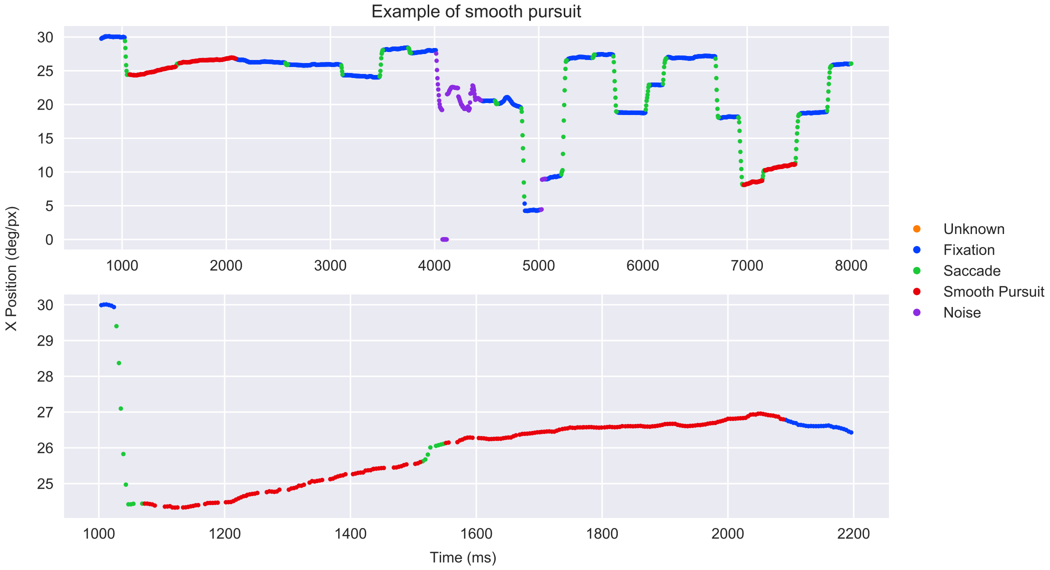


Figure 3.4: An example of smooth pursuit in the beach recording of subject AAF. The bottom figure shows a zoomed in plot of first smooth pursuit event.

3.1.3 Duration

This section investigates the properties and characteristics of the labelled classes. Figure 3.5 shows the distribution of events duration for each class. There were no instance of the class "unknown" in the entire dataset. The first thing that comes to attention is there are saccades that have been marked that are longer than usual. Typical saccades are in the range of 20-80 ms with an amplitude of 4-20 deg [Holmqvist et al., 2011] and with such a large number being above that, it is alarming.

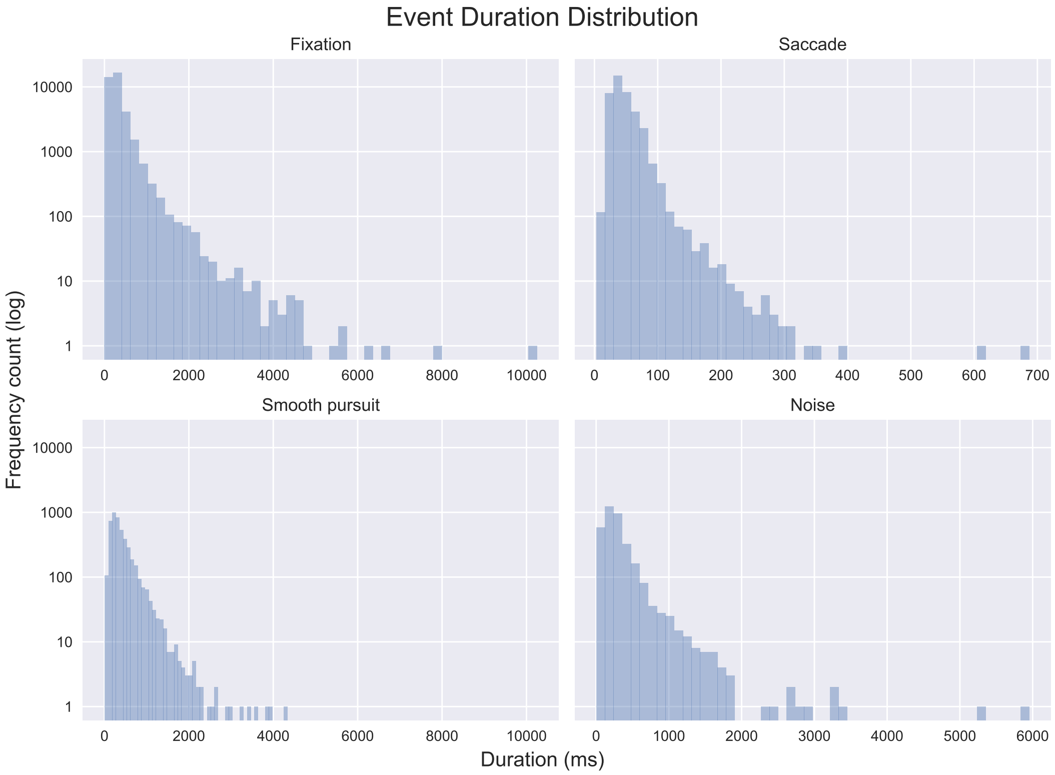


Figure 3.5: Distributions of event duration

The main sequence [Bahill et al., 1975b] is tool to view saccades. When a combination of the duration, magnitude or peak velocity for saccades are plotted on a logarithmic scale a largely linear relationship should be seen. Figure 3.6 shows the main sequence of the classified saccades. It becomes very apparent that there are several saccade under 10 ms and several with magnitudes below 1 deg. There are even some saccades with magnitudes below 0.1 deg. There also appear to be a large a number of saccades above 80 ms. There also appear to be several questionable outliers. These are all indications that some saccades have been misclassified which could heavily impact the underlying distribution of what they were misclassified as.

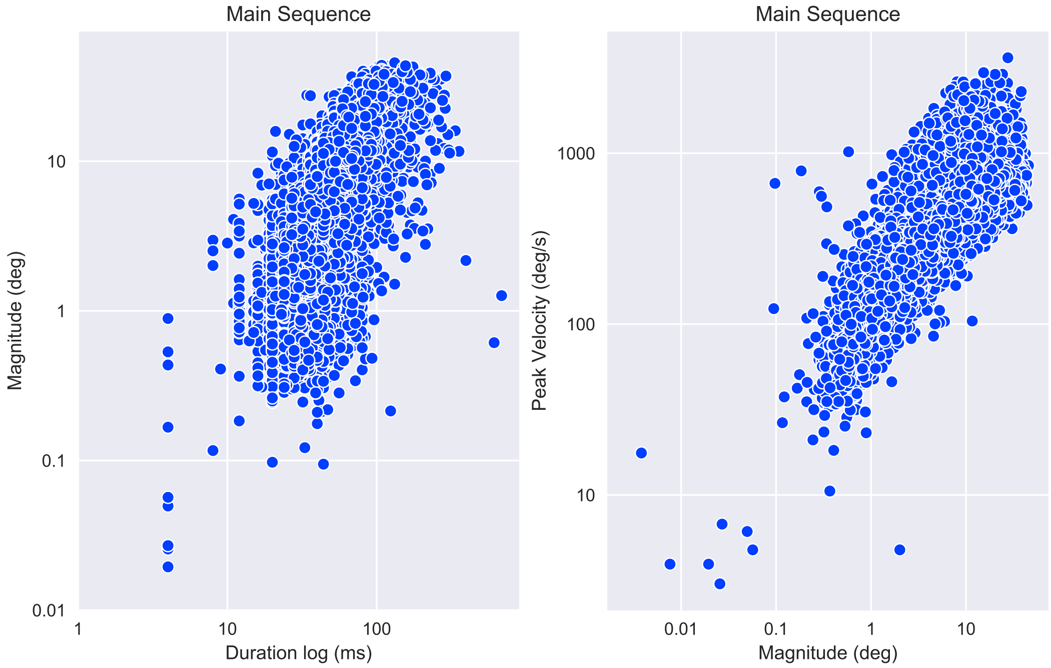


Figure 3.6: The main sequence for the classified saccades of GazeCom

3.1.4 Inspection of saccades

To understand what the saccades have been classified as the easiest approach is to look at examples of the recording in those regions. There were 16 instances of saccades below 10 ms. By looking at Figure 3.7 it can be seen that by the using only the velocity trace the event could be interpreted as a saccade while in reality it is just noise since the positional signal does not have the spatial movement of a saccade. The beginning and end of the "saccade" are also wrong as the "saccade" itself is only two samples. This is a case of bad classification.

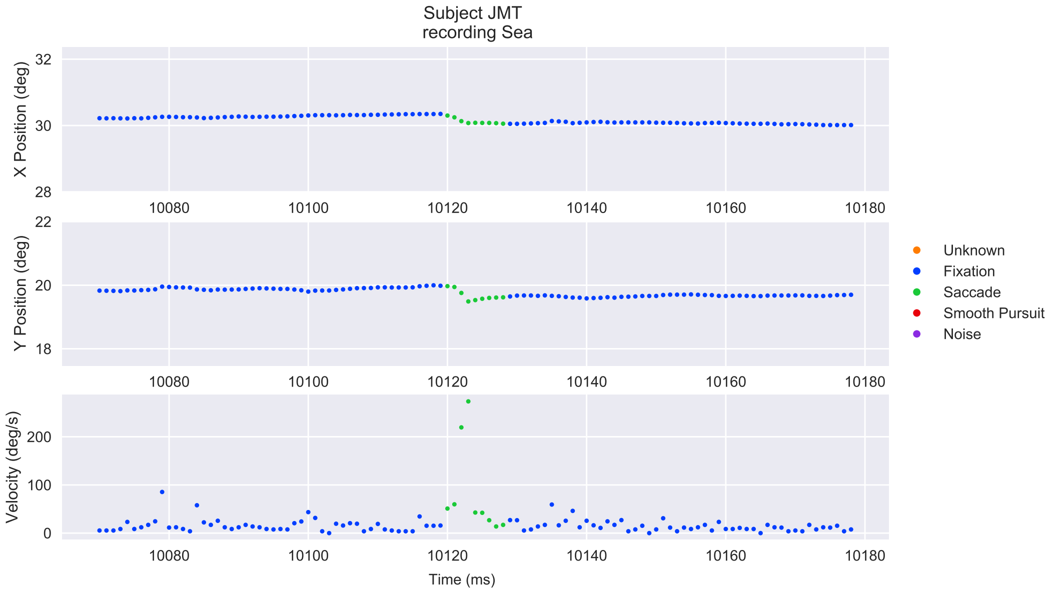


Figure 3.7: Example of a regular recording noise being classified as saccade.

Figure 3.8 shows that the duration of the event is only 1 sample, which corresponds to 4 ms duration. After some investigation this seems to be caused by the fact that there are two raters and the final label is based off of some unknown merging of those two. These composed the majority of sub 10 ms samples.

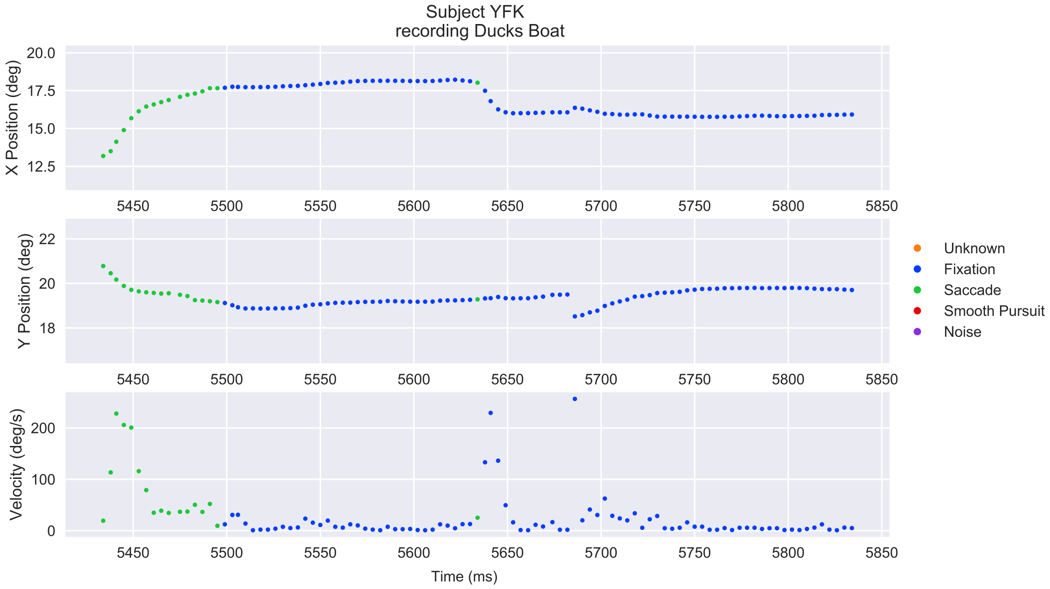


Figure 3.8: Example of 1 sample classifications which can be seen between 5600 ms to 5650 ms.

When looking at the events around 20 ms some are marked correctly and some are signal noise with very long start or ends as Figure 3.7. The closer to 10 ms the more errors. When looking at event around 80-120 ms classification events such as Figure 3.9 also to appear where it is clear that a PSO has been marked as part of a saccade.

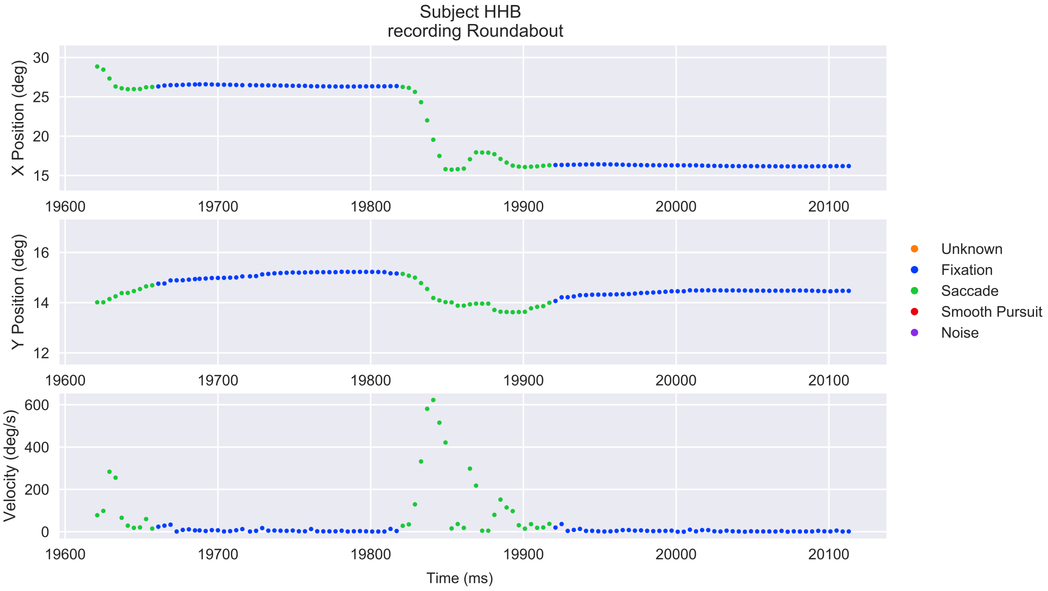


Figure 3.9: Example of saccade with a PSO that has been classified as saccade.

When look at around 180-200 ms the saccades begin to look more like smooth pursuit. Figure 3.10 is an example it can be seen in the vertical trace that those movements are not saccadic. It is hard to interpret as there is a very large saccade in the horizontal saccade but the vertical trace looks more like smooth pursuit or perhaps noise. Figure 3.11 looks like a huge saccade in the horizontal trace but it lasts almost 200 ms. When looking at the vertical trace it becomes clear that it looks much more like a partial blink instead. Figure 3.12 is actually two saccades with a fixation in between that has been labelled as one saccade instead. The saccades that do exist in this duration are actually saccades where the end has been severely misclassified and the saccade goes on for too long.

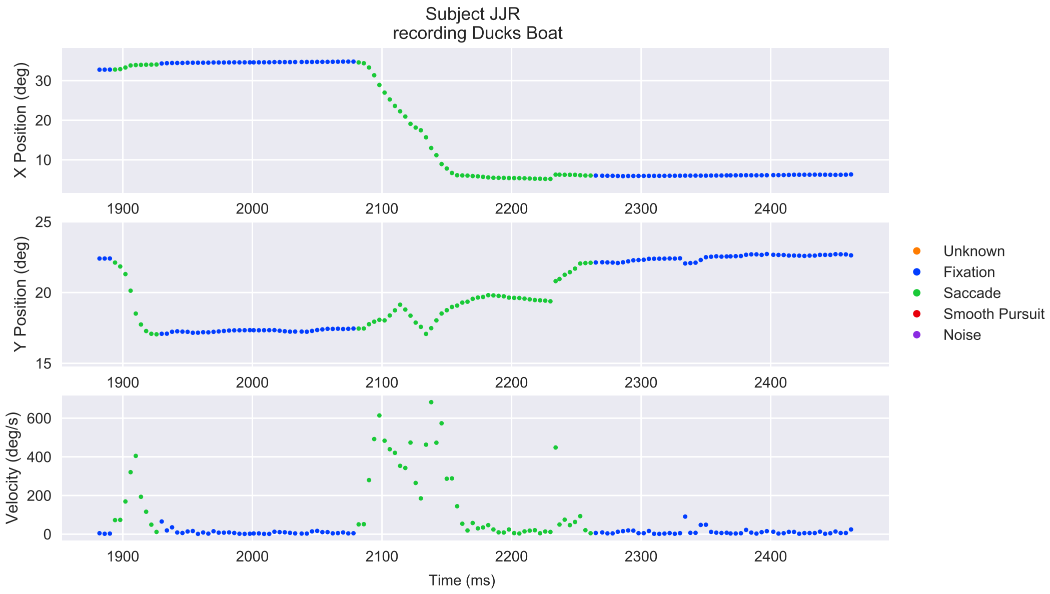


Figure 3.10: Example of saccade that is mislabelled at around 180-200 ms

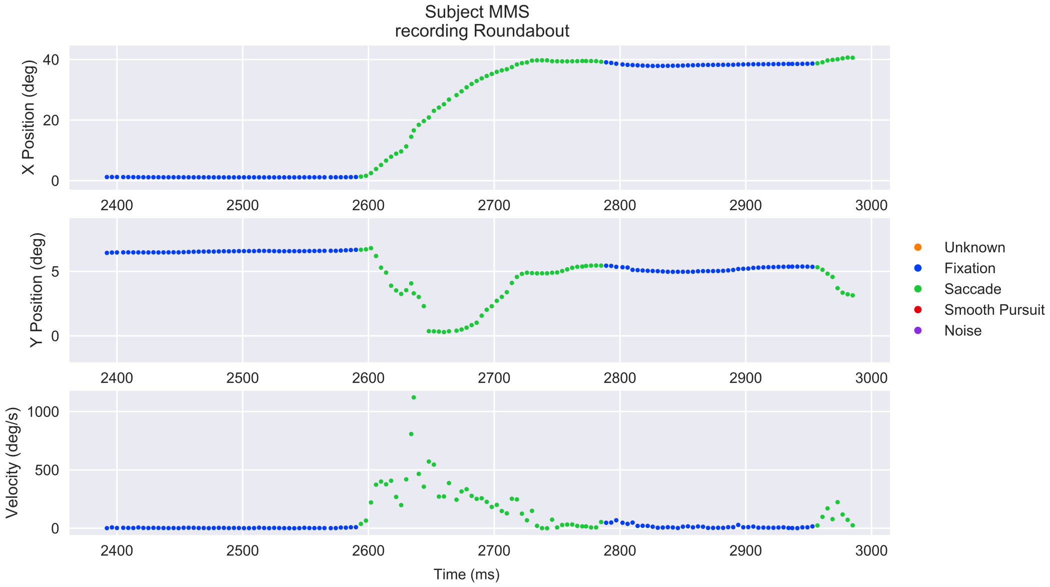


Figure 3.11: Example of a partial blink mislabelled as a saccade at around 180-200 ms .

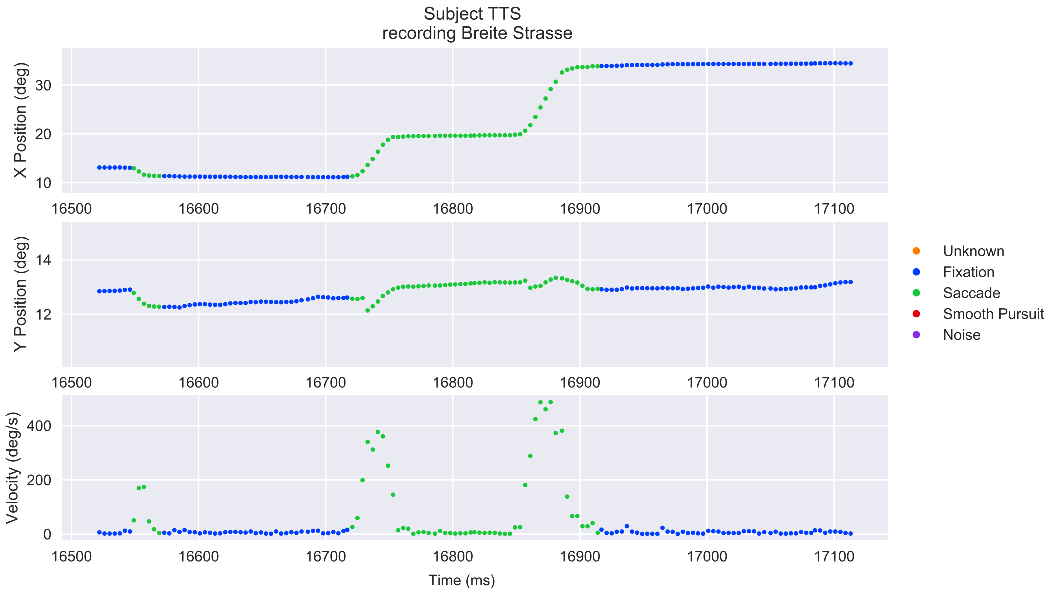


Figure 3.12: Example two saccades misclassified as one saccade

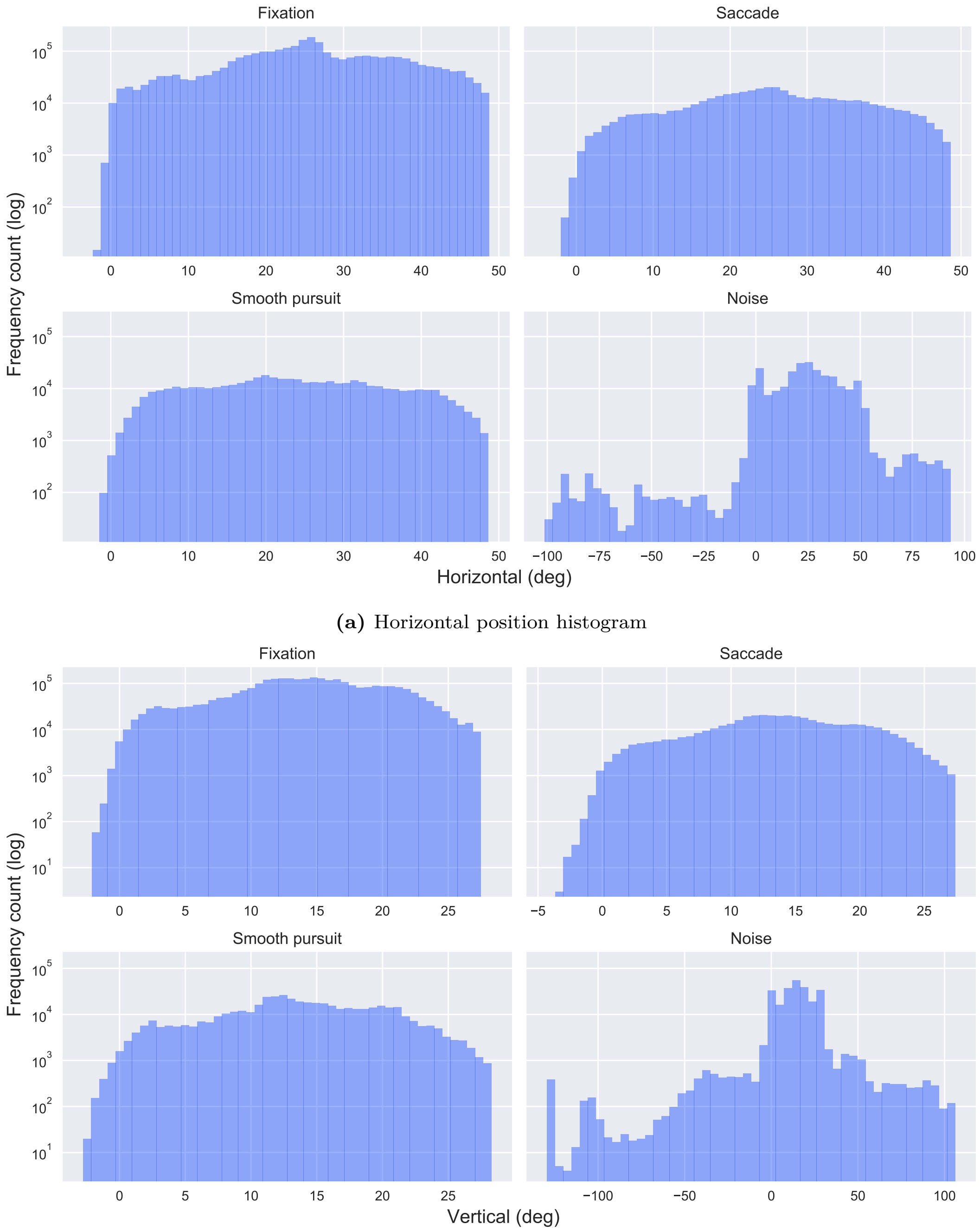
3.1.5 Feature distribution

Startsev et al. [2018] use the horizontal position, vertical position, velocity, acceleration and direction as features for their network. The distribution for velocity, acceleration and direction shown here are all based on the windows size that corresponds to 4 ms as it is the instantaneous velocity at a sampling frequency of 250 Hz. Figure 3.13 shows the positional distribution for the different classes. There were very large outliers that distorted the histograms so only the 99.9th percentile at a sample level has been plotted. There doesn't seem to be distinct differences between the classes except for the noise class. There are a lot more samples around and below 0 than the other classes. There are also samples that go to above 50 deg, while the actual movements stay between 0 deg and 50 deg horizontal and 0 deg and 25 deg vertical. There seems to be some samples a little bit below 0 in the other classes and this could be caused by the eye-tracker loosing track of the eye.

Figure 3.14 shows the distribution of the velocity and acceleration. The velocity of the fixations are spread between 0 and 100 deg/s which indicates that there is a lot of signal noise in the recordings. The smooth pursuit does not seem much different which is strange since smooth pursuit should have velocities larger than the fixations. The saccades have a distinctive range of distribution above 100 deg/s but they also contain a lot of samples that are below. This could indicate that many saccades are not clearly marked and contain a lot of fixation or smooth pursuit samples. The noise velocities are magnitudes above the rest which is likely caused large oscillations that have been shown in the previous examples. They also contain a large amount of samples in the realm of the other classes velocities which is likely caused by the samples surrounding the dropped samples also being classified as noise. The acceleration looks the same except for the saccades.

The distribution of directions can be see in Figure 3.15. The most distinctive class is the saccade which has clear peaks at 0, π, $\frac{\pi}{2}$, $-\pi$, and$-\frac{\pi}{2}$. These could correspond to largely horizontal saccades at 0, π and $-\pi$ while vertical saccades are centred around $\frac{\pi}{2}$ and$-\frac{\pi}{2}$. Fixation just looks like a scaled up version smooth pursuit while in theory smooth pursuit should be more distinctive. Noise looks like less distinctive version of the saccade distribution which could indicate that a lot of saccadic like movements have been labelled as noise.

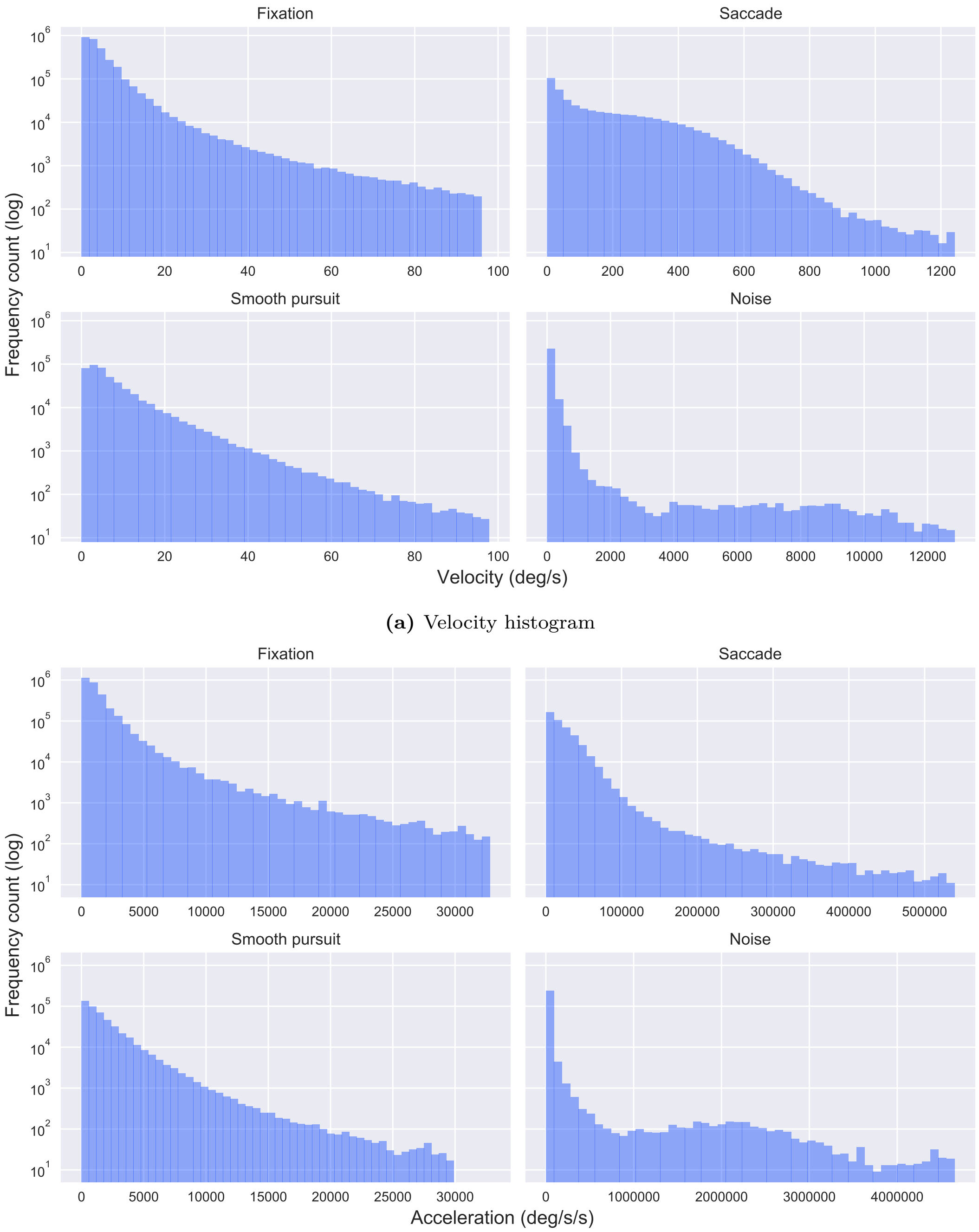
This all indicates that the dataset has not been classified properly. This can be especially seen in Figure 3.6 where a large portion of the marked saccades does not line up with the physical properties of saccades. Further inspection of individual cases confirms that many of the saccade classifications are questionable and some are plainly wrong. The results of using such a dataset to train a network would be very questionable. Either this dataset must be reclassified or some clean up is need. Since that is beyond of the scope of this project a different dataset will be used.



(a) Horizontal position histogram

(b) Vertical position histogram

Figure 3.13: Distribution of the positional gaze signal



(a) Velocity histogram

(b) Acceleration histogram

Figure 3.14: Distribution of the instantaneous velocity and acceleration of the gaze

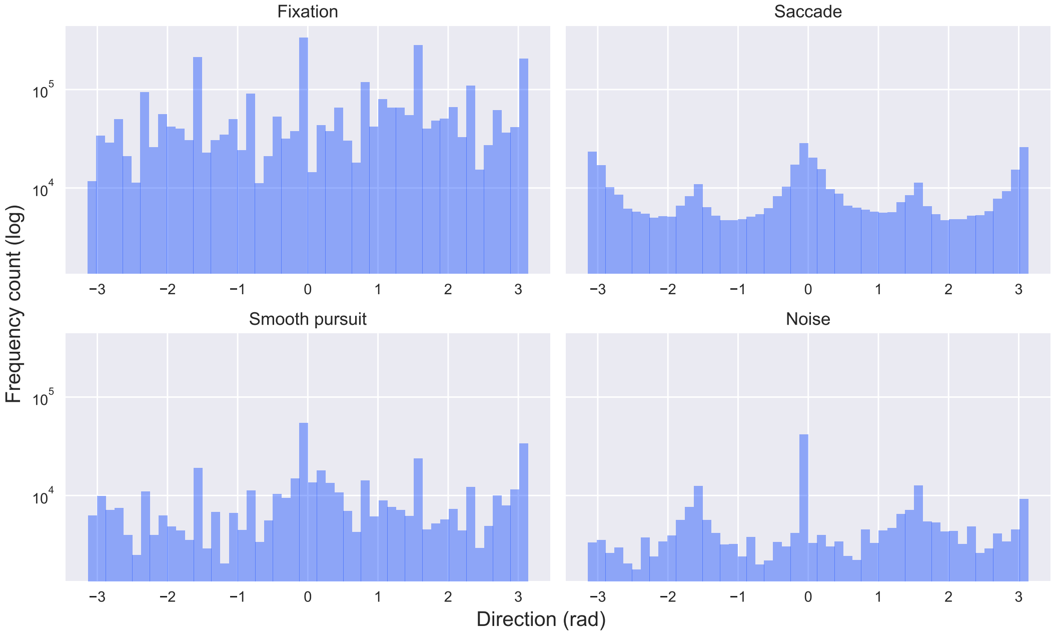


Figure 3.15: Direction histogram.

3.2 Lund 2013 dataset

The Lund 2013 dataset, described in section 2.2.5, was recorded on 38 cm by 30 cm monitor with a resolution of 1024 by 768 pixels. The subjects were seated 67 cm from the screen. The recordings were presented to the raters without the knowledge of which stimuli it was from. The horizontal and vertical coordinates over time were shown together with a velocity over time trace as well. Marcus Nyström (MN) and Richard Anderson (RA), who are experts in the eye tracking field with many years of experience, classified the recordings based on their own internal understanding of the event definitions without explicitly stating what definitions they each used or accounting for difference in opinion. The eye movements were classified into fixations, saccades, PSOs, smooth pursuit, blinks and unidentified. It contains recordings from three experiment; dot stimuli, image viewing and video viewing. In the dot and video viewing there is smooth pursuit stimuli presented while the images are static and therefore should not contain smooth pursuit. Zemblys et al. [2018] found a mistake in the dataset where a saccade should have been marked as a fixation and reclassified that event and this fix is also applied in the following presentation. Table 3.1 shows the amount of samples in the different experiments. The dots stimuli has far greater smooth pursuit than fixations. The video stimuli has more of an even spread of fixation and smooth pursuit. There are not many saccade and even less PSO samples compared to fixation and smooth pursuit which makes sense since they are both usually much shorter events. The Cohen's kappa reported are samples level comparison in the recordings which both RA and MN labelled. There is a good agreement with fixations and saccades, and slightly worse with smooth pursuit. PSOs are the lowest at 0.73 which is not unusual for PSOs. The duration of each experiment and how much each rater has labelled can also be seen. Rater RA has by far the most labels with a total of 764.58 seconds labelled. Going forward only recordings labelled by rater RA will be used.

		Fixations	Saccade	PSO	Smooth Pursuit	Blink	Undefined	Duration
Rater RA	Dots	12.85%	4.72%	1.43%	79.53%	0.53%	0.95%	40.0 s
	Images	76.46%	9.18%	4.76%	4.78%	4.68%	0.14%	175.58 s
	Videos	33.63%	4.41%	2.64%	57.88%	1.36%	0.08%	548.19 s
Total		42.38%	5.53%	3.06%	46.81%	2.08%	0.14%	765.58 s
Rater MN	Dots	8.99%	4.53%	2.01%	81.64%	1.42%	1.42%	23.73 s
	Images	79.6%	8.59%	5.24%	0.85%	5.51%	0.2%	127.70 s
	Videos	42.97%	5.17%	3.38%	46.38%	2.03%	0.06%	58.06 s
Total		61.45%	7.19%	4.36%	22.62%	4.09%	0.3%	209.50 s
Cohen's Kappa		0.82	0.9	0.73	0.79	0.54	0.91	

Table 3.1: Distribution of the manual labelling done by rater RA and and MN for the Lund dataset. Rater MN labels are a subset of rater RA. The distribution of samples for each class in each experiment is shown. Cohen's Kappa was calculated on sample level on the recordings across all experiments that both RA and MN had labelled.

3.2.1 Lund 2013 movement examples

Figure 3.16 shows an example of how rater RA scores fixations, saccades PSO and smooth pursuit. These are much clearer markings than in the GazeCom dataset, since saccade beginnings and ends corresponds to what is normally defined as a saccade. Figure 3.17 shows how a blink looks like in these recordings. Because a different eye-tracker is used in the Lund 2013 dataset compared to the GazeCom dataset blinks are handled differently. In Lund 2013 dataset the gaze position is set to NaN whenever the eye-tracker losses track of the eye. After viewing a couple of examples the tendency of blinks in this dataset is that the event before and after the blink is the same and the gaze position being relatively unchanged. Figure 3.18 show an example of an unidentified event which in this case is an event in the end of a recording. This is probably because RA judged that it was a different event that did not finish and therefore should be excluded. Other cases unidentified labels are NaNs that can occur during recording. Here the NaN and a few samples before and after are marked as unidentified. There are no NaNs present in labels other than blinks and unidentified.

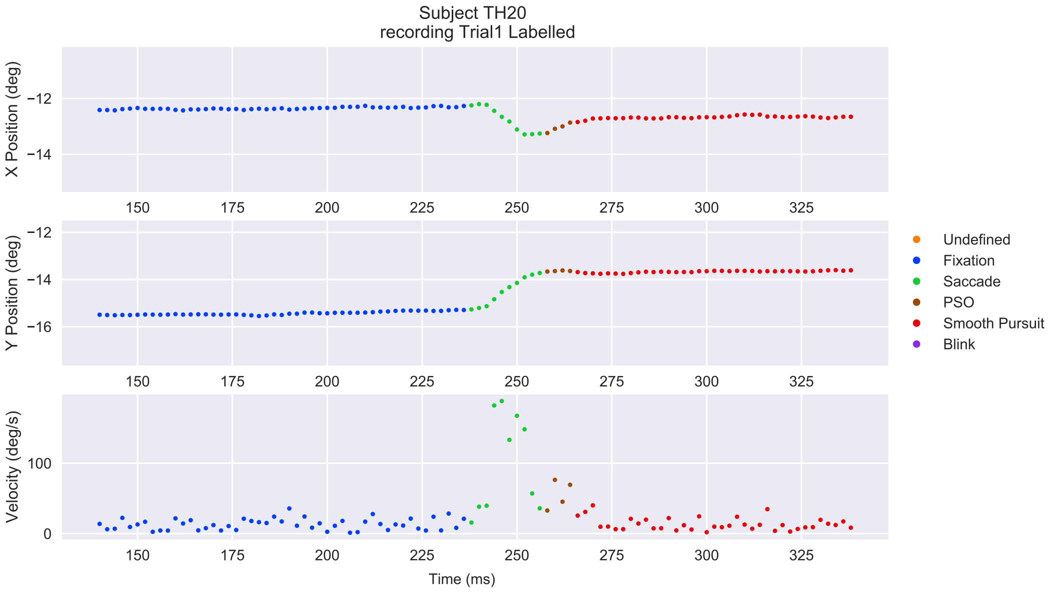


Figure 3.16: Example of how rater RA scores fixation, saccades, PSO and smooth pursuit. The velocity is instantaneous velocity.

3.2.2 Lund 2013 feature distribution

The gaze position is transformed into degrees. The point $(0, 0)$ is not the bottom left corner as in GazeCom but is the center of the screen instead. Since Larsson et al. [2013] do not calculate the features mentioned in section 3.1 so these have

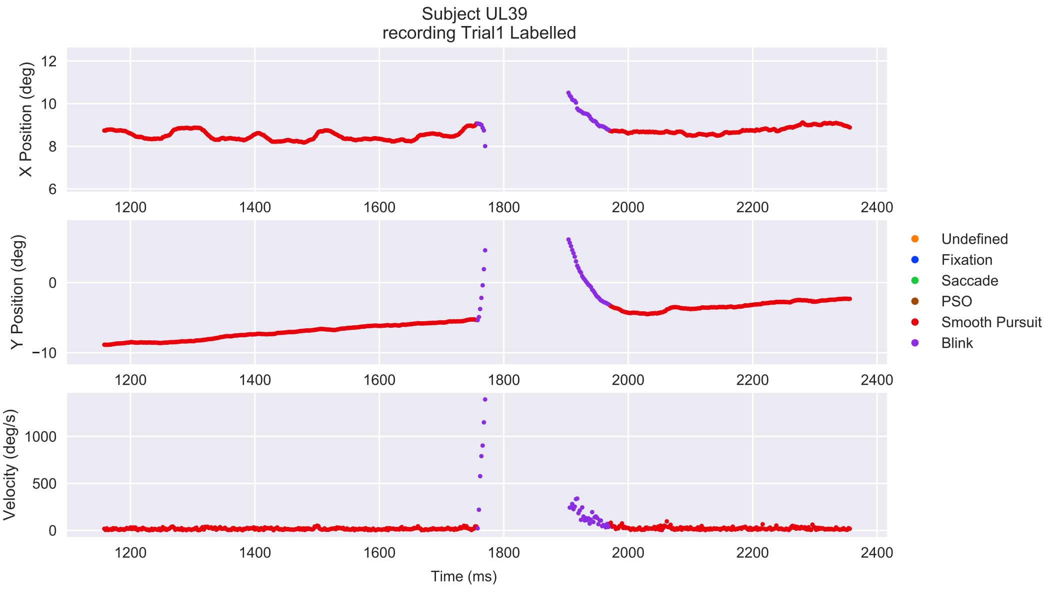


Figure 3.17: Example of how rater RA scores blinks. The missing samples are NaNs. The velocity is instantaneous velocity.

been calculated with the same approach as Startsev et al. [2018]. Since Lund 2013 recordings are 500 Hz and not 250 Hz the temporal windows for feature extraction can therefore be lower. The windows used are 2, 4, 8, 16 and 32 ms. Since the recording contains NaNs some of the features will also be NaNs. Figure 3.19 shows that there are not many blinks and unidentified events and in the recording and since blink detection is out of the scope for this project blinks and unidentified events are simply dropped and the events before and after are stitched together. This has the unfortunate effect that it create events that are longer than they should be e.g. long fixations. There was also one instantaneous acceleration sample that was a NaN and that has been removed as well. There does not seem to be any general approach to how to deal with NaNs or blinks in the reviewed literature.

Figure 3.19 also shows that there are no the durations for the events are also reasonable. There doesn't seem to be many saccades above 75 ms and the PSOs are shorter than the saccades. Figure 3.20 shows the main sequence after blinks and unidentified have been removed. The very shortest saccades seem questionable and after looking at examples of some of them it becomes apparent that they look saccades in the velocity trace but the gaze does not actually change positing from before and after that saccade. There are only 9 saccades above 90 ms and two of those are result of removing blinks in between two saccades which result in a saccade that is 206 ms and another that is 242 ms. The relationship also seems fairly linear without too many outliers.

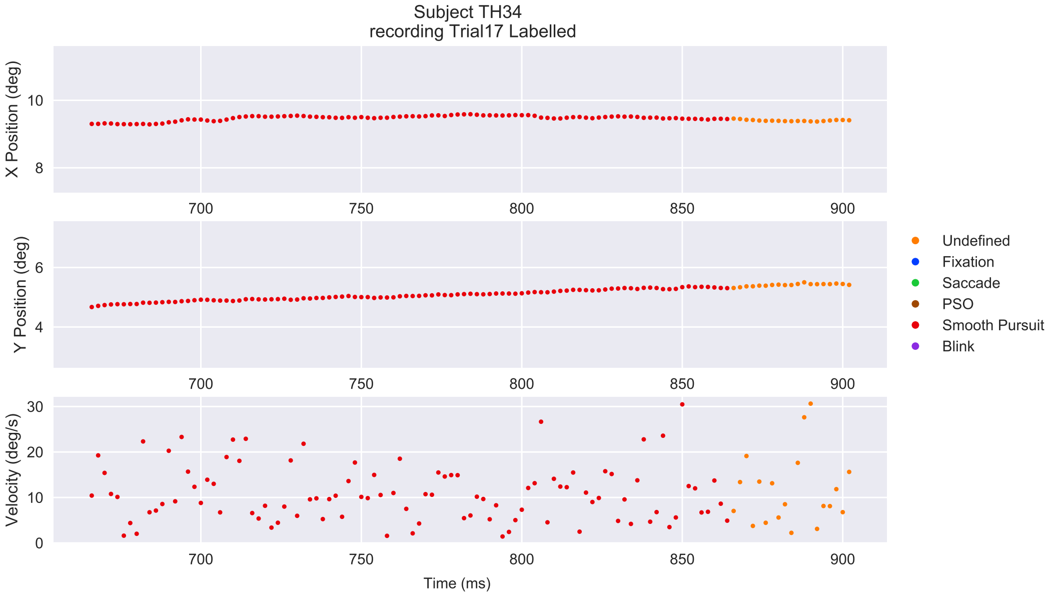


Figure 3.18: Example of how rater RA scores the end of a recording as unidentified. The velocity is instantaneous velocity.

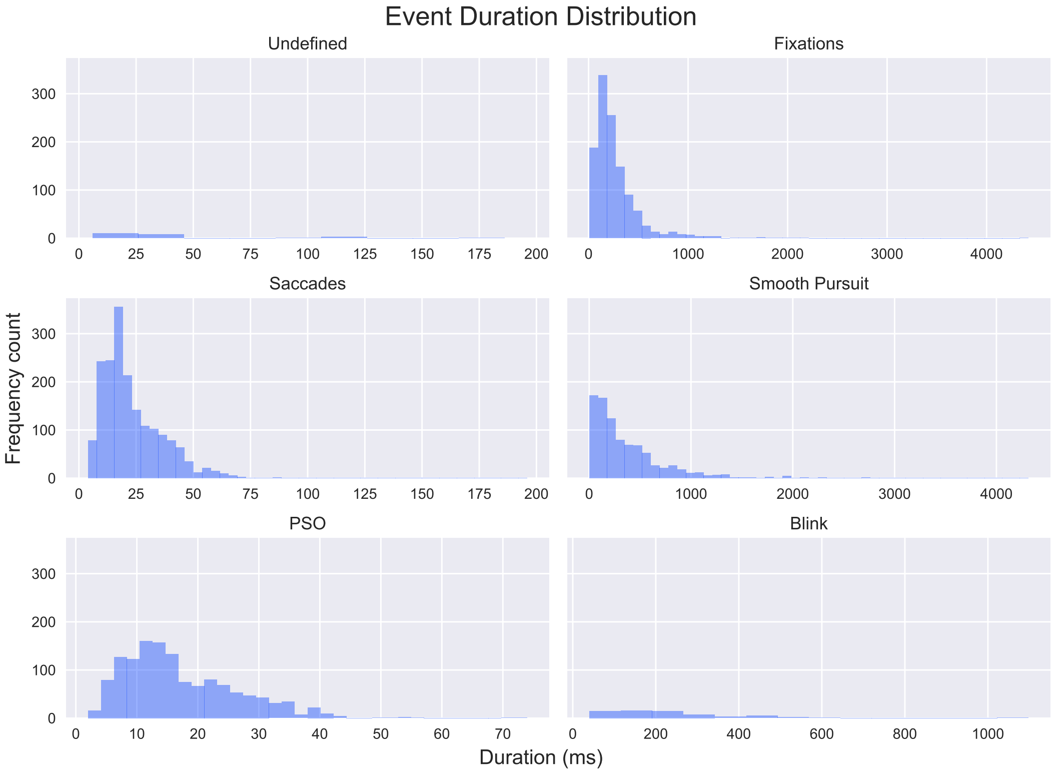


Figure 3.19: Event duration in milliseconds for the Lund dataset by rater RA

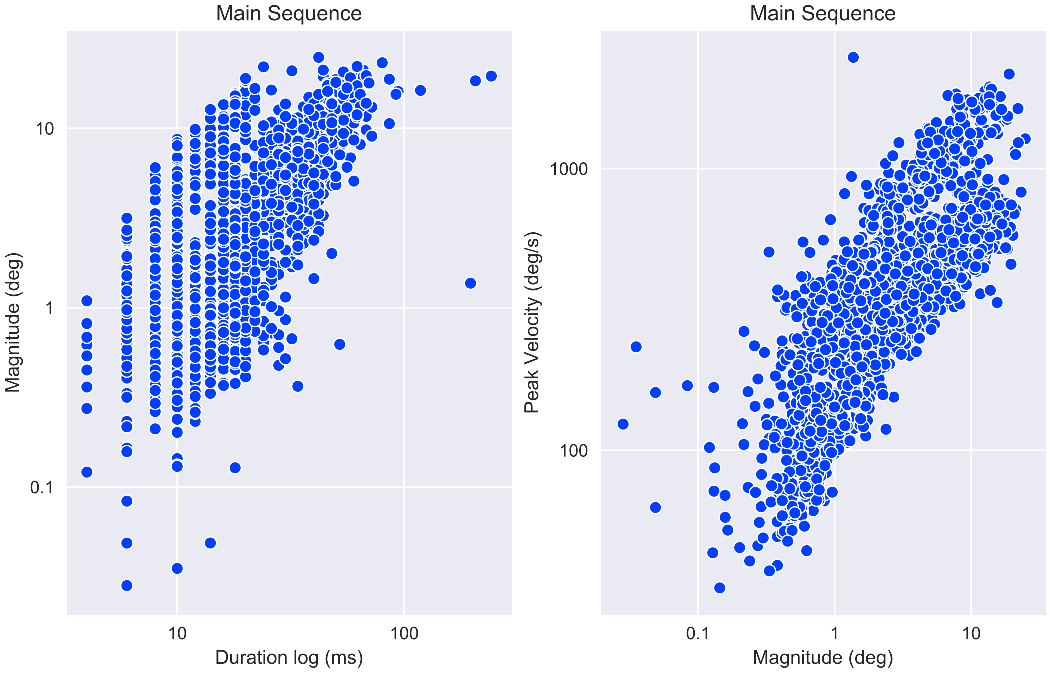
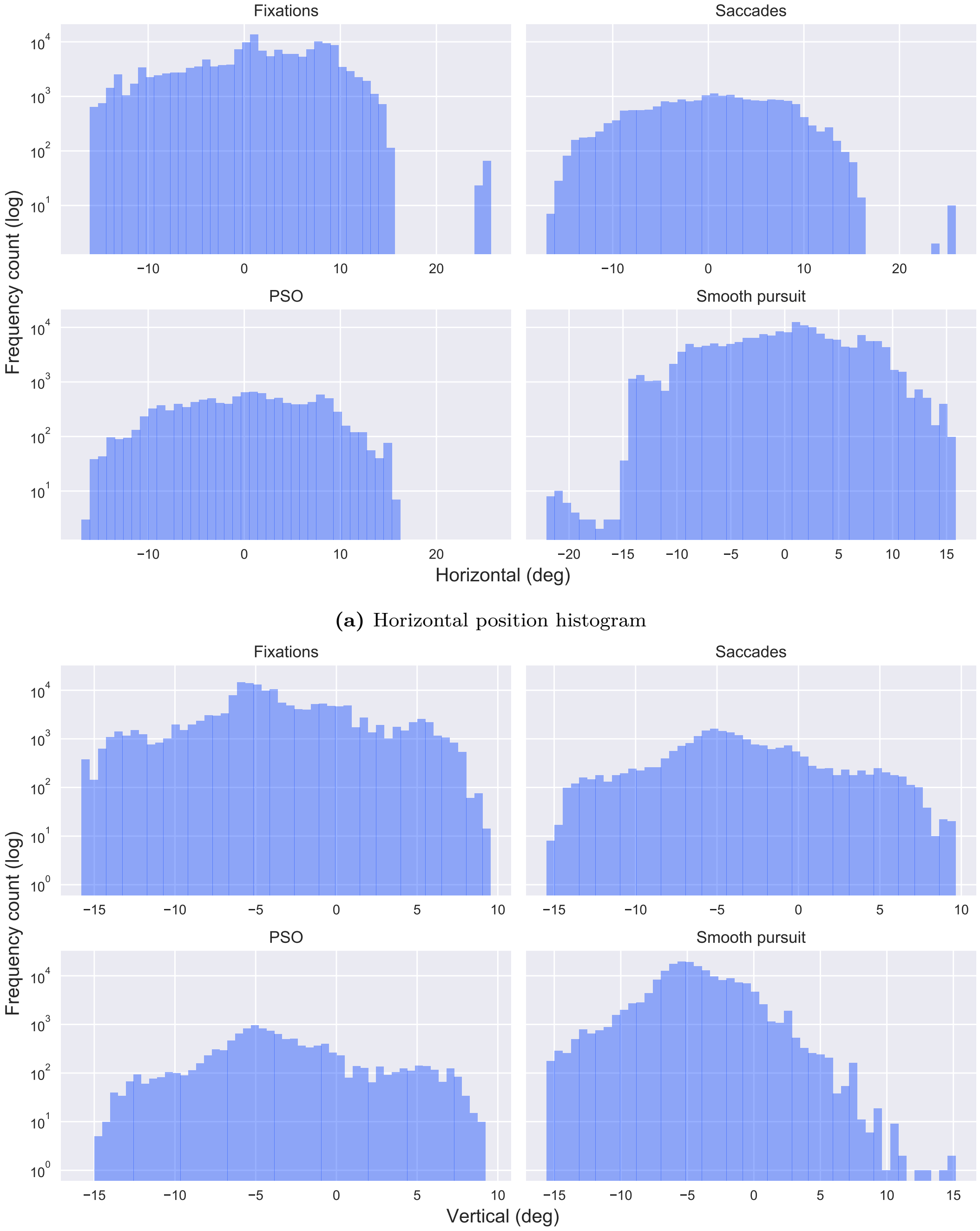


Figure 3.20: The main sequence for the Lund dataset by rater RA after blinks and unidentified have been removed.

The distribution of the features shown are all calculated with a 2 ms window. No outlier removal was performed on these distributions. Figure 3.21 shows the positional features. There does not seem the anything distinctive about them. Figure 3.22 shows the velocity and acceleration distribution. The saccade velocities are clearly higher than the rest which is as expected. Smooth pursuit and fixation have similar distributions with spikes in the lower velocities. PSO has a similar distribution as saccades except it is not in the same range. The accelerations seem like a scaled version of the velocities. There also appear to be a few outliers but nothing that seems major, except for 1 sample in the acceleration for smooth pursuit.

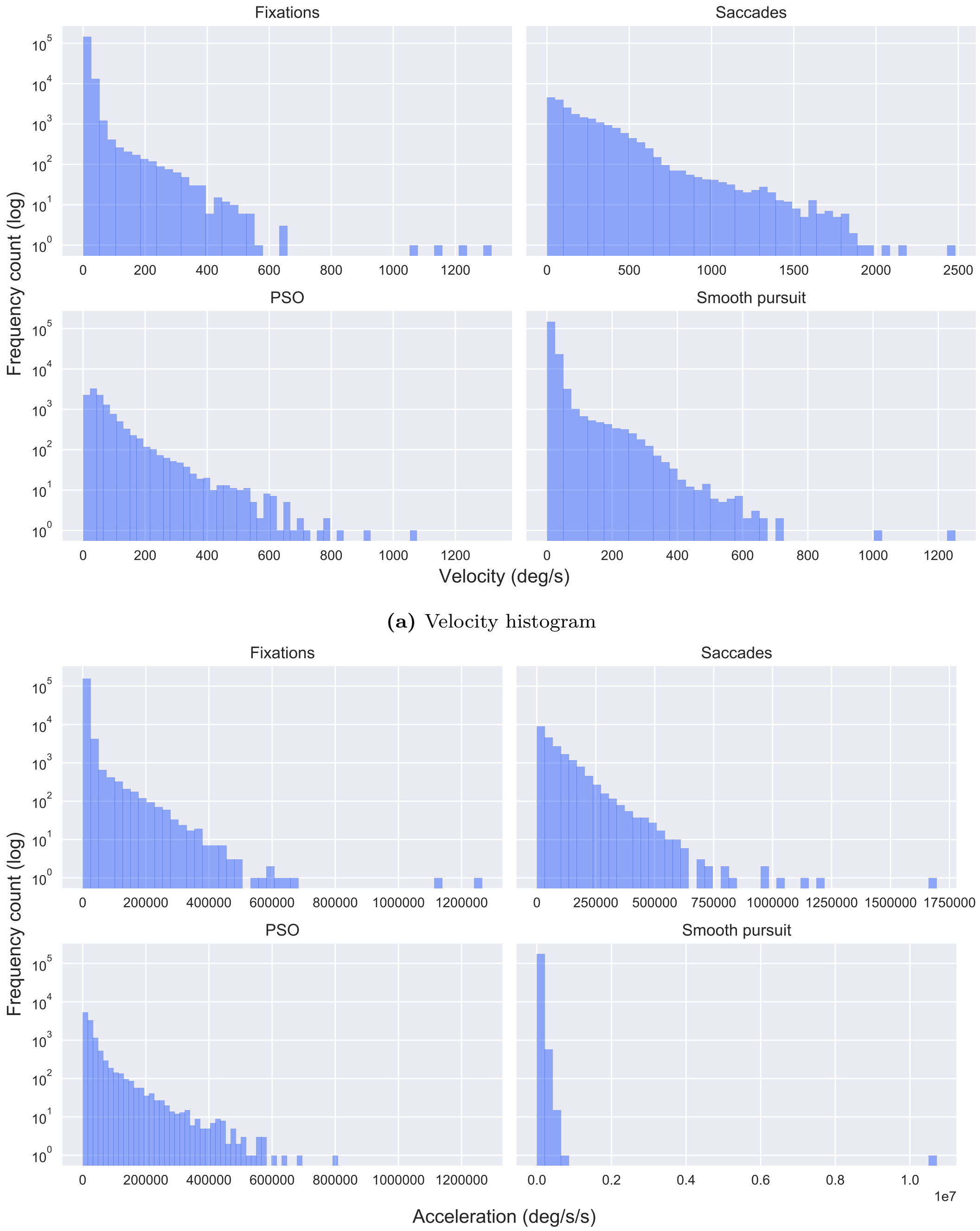
Figure 3.23 shows the directional feature. Compared to the same feature presented in GazeCom this seems much more distinctive which could indicate a better labelling. The saccade has the same distinctive peaks as mentioned previously. PSO looks like a less distinctive version with the same peaks. Smooth pursuit and fixations look similar but smooth pursuit seems to have sharper peaks.



(a) Horizontal position histogram

(b) Vertical position histogram

Figure 3.21: Distribution of the positional gaze signal in the Lund dataset



(a) Velocity histogram

(b) Acceleration histogram

Figure 3.22: Distribution of the instantaneous velocity and acceleration of the gaze in the Lund dataset.

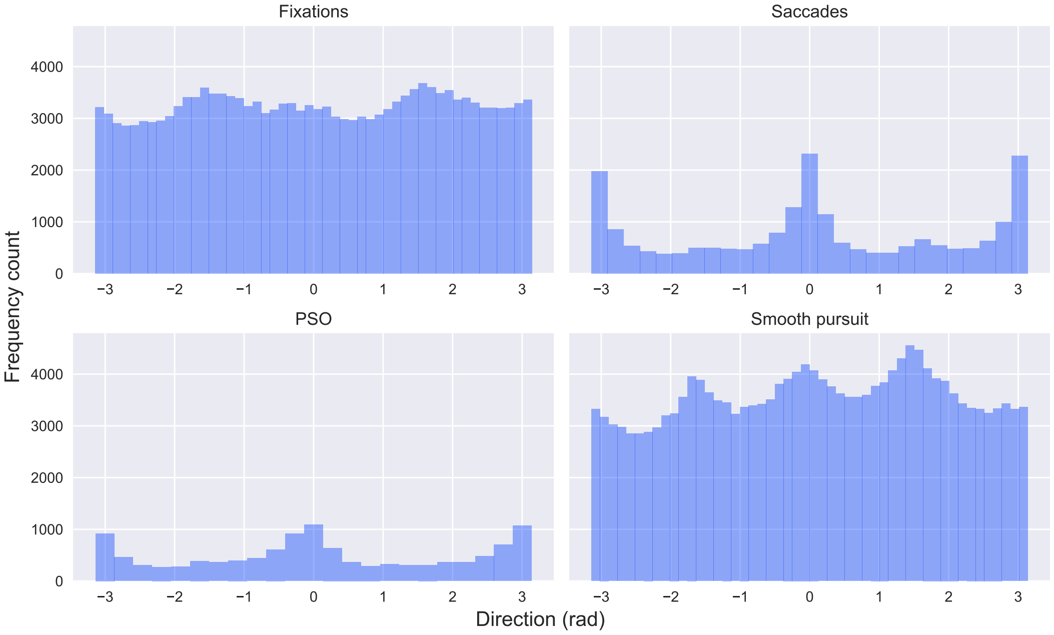


Figure 3.23: Distribution for the directions in the Lund dataset.

Chapter 4

Methodology

This chapter contains the theory behind 1D-CNN-BLSTM network, the implementation of the network and the evaluation metrics chosen.

4.1 Neural Networks

The 1D-CNN-BLSTM network from Startsev et al. [2018] is made up of two main parts; a CNN part and a Recurrent Neural Network (RNN) part.

4.1.1 Convolutional Neural Network

A CNN consists of a kernel with weights that convolves over the signal. The output of this is then put through an activation function which results in what is often called a feature map. Depending on the weights, different feature maps are extracted. A CNN then has a set number of kernels with different weights and will learn those weights. The features that the kernel learns are translation invariant meaning that the features are local and not dependent on their location in the signal. Typically multiple convolutional layers are used as the features extracted in the early layers are simple e.g. horizontal or vertical line while the feature maps of the later layers are more abstract e.g. edges [Goodfellow et al., 2016]. The typical parameters to set of a convolutional layer are the stride, i.e how many steps the kernel will move at each step, the kernel size itself, number of kernels and what activation function is used. In 1D-CNN-BLSTM the convolution is in 1D.

4.1.2 Long Short-Term Memory

RNNs can be thought of as a layer that loops. The input to the layer is processed by some function and put back into itself which allows it to have knowledge of all the time steps that came before it. This is especially useful with sequential data as the order of what came before is important in predicting what comes next. The LSTM network is a type of RNN that can learn long-term dependencies. It has a similar repeating structure except instead of just having a function each cell of the LSTM contains an input gate, a forget gate, cell state, output gate and output. The cell state is like a highway of information that passes through all the cells and passes information from one cell to the next. The gates then determine what information is passed along. First the forget gate decides what information to throw away. The output of the previous layer $h_{\mathrm{t-1}}$ and the current input x_t are put through a sigmoid function to get a number between 0 and 1 to determine how important the previous information is. This is then multiplied with the current cell state $C_{\mathrm{t-1}}$. The equation for the forget get can be seen in Equation 4.1 where subscript f denotes the forget gate, W are the weights, b is the bias and σ is the sigmoid function.

$$f_t = \sigma(W_f * [h_{\mathrm{t-1}}, x_t] + b_f) \tag{4.1}$$

The input gates is similar to the forget gate except it is instead multiplied with the *tanh* function of the previous gate. These are then used to update the cell state as in Equation 4.2, where i_t is the input gate and C_t is the new cell state. Each state also has it's own respective weights W and bias b

$$\begin{aligned} i_t &= \sigma(W_i * [h_{\mathrm{t-1}}, x_t] + b_i) \\ \widetilde{C} &= tanh(W_C * [h_{\mathrm{t-1}}, x_t] + b_C) \\ C_t &= f_t * C_{\mathrm{t-1}} + i_t * \widetilde{C} \end{aligned} \tag{4.2}$$

The output of the cell block h_t is then based upon the current state combined with the previous output, Equation 4.3. This output is then fed to next cell block [Goodfellow et al., 2016].

$$\begin{aligned} o_t &= \sigma(W_o * [h_{\mathrm{t-1}}, x_t] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{4.3}$$

The LSTM used in this network is bidirectional which is actually two independent LSTM layers that run in the opposite direction of each other time wise. In other words one layer has knowledge about past samples in respect to the current sample while the other layer has knowledge about the future samples. They are typically treated as one layers since their outputs are concatenated. The typical parameters of an LSTM layers is called units which represents the number of outputs at each time step.

The full 1D-CNN-BLSTM architecture used be seen in Table 4.1. This also shows that batch normalization and dropout layers are added. The time distributed layers are a wrapper that ensures that when flattening or creating a dense layer the appropriate slice is taken so that each time step is kept separate. The input, 520×5, is shown when using all five features. The input is 520 samples because it has been mirror padded. This ensures that the output has the correct size of 514 samples. The number of classes is 4. The input size depends on the kernel size as larger kernels will require more padding to maintain the 514 sample output.

Layer Type	Feature Map Size	Kernel	Activation	Other
Input				520 × 5
Conv1D	32	3		
BatchNormalization				
Activation			ReLU	
Conv1D	16	3		
BatchNormalization				
Activation			ReLU	
Dropout				0.3
Conv1D	8	3		
BatchNormalization				
Activation			ReLU	
Dropout				0.3
TimeDistributedFlatten				
TimeDistributedDense	32		SoftMax	
BLSTM	16			
Output	514 × Num. of Classes		SoftMax	

Table 4.1: Parameters set for the 1D-CNN-BLSTM

4.2 Implementation

The original network from Startsev et al. [2018] was made for data recorded at 250 Hz and since the Lund dataset was recorded at 500 Hz some changes were made. The recordings were split into windows of 514 samples with an overlap of 130 samples between recording, which is twice what was used at 250 Hz. The windows were padded by using mirror padding in each windows. Rater RA was chosen as the ground truth. Leave-One-Video-Out (LOVO) cross validation was performed where one recording was keep as test each time. There were a total of 10 different recordings; (dots) trial 17, (images) Europe, Vy, Rome, Konijntjes, (videos) Bergo Dalbana, Dolphin, Biljardklipp, TrafikEhuset and Triple Jump. The recordings being used as test set did not have any overlap when being split into windows. The features position, velocity, acceleration and direction were used and a combination of those features was also tried. The network was coded in Python using Keras with Tensorflow backend. The networks were trained on a NVIDIA GeForce GTX 970 graphics card. All models were trained for 500 epoch. Since there are many more fixation and smooth pursuit samples in the data a weighted cross entropy loss was used. The weights are $w = 1 - (\frac{\sum x_n}{\sum x_m})$ where x_n are samples of the class and x_m are all samples. The RMSprop optimizer with default settings was used. A batch size of 1500 was used on most models except some models which due to GPU memory could not run with such a large batch size. In these cases a batch size of 500 was used. With this setup it took approximately 1.5h of training each model which becomes 11.5h when doing LOVO.

Different variations of the 1D-CNN-BLSTM were tried in this project as the work by Startsev et al. [2018] did not show the effect of different model parameters. The kernel sizes of 3, 9 and 29 were tried. A Residual Network (Resnet) [He et al., 2016] was also tried. The intuition behind larger kernel sizes is that smooth pursuit is a long event where the change is gradual and often needs more time be salient. Different features and feature combination were also used to see if any of the features were particularly good.

4.3 Evaluation metrics chosen

To evaluate the performance the metrics mentioned in section 2.3 were considered. On a sample level the F1-score and Cohen's Kappa were chosen. On an event level the F1-score was chosen. To get an event level matching the approach from Startsev et al. [2018] was used. There was no implementation of their matching available online so based on the paper the algorithm was been reimplemented from scratch. The pseudo code can be seen in Algorithm :

An example of how the matchings look can be seen in figure Figure 4.1. It can be visually seen how an event is matched into hits, miss and false alarms. False alarm matching has been shown as fa and fa2 where fa is when IoU< 0.5 and fa 2 are events that happens during the algorithm but not during the ground truth. In the calculation of the F1 - score they are grouped together into a single false alarm number. This subjects F1 - scores can be seen in table Table 4.2. The biggest difference is that smooth pursuit has a sample level F1 of 0.61 but a 0.00 event level F1. This is because the long smooth pursuit event in ground truth is split into smaller events in the algorithm. Even though more than half the samples occurring in the large smooth pursuit event are classified as smooth pursuit by the algorithm, none of the events have an IoU > 0.5 causing them all to be misses.

	F1 - score	
	Sample	Event
Fixation	0.46	0.30
Saccade	0.88	1.00
SP	0.61	0.00
PSO	0.55	0.20

Table 4.2: Sample and event level F1 - scores for the recording in Figure 4.1

Algorithm Event level matching: The inputs are sample level labels for ground truth and a comparison e.g. another rater or output of a classification algorithm. The event types can be fixation, saccades, smooth pursuits and PSOs depending on what events were labelled. The hits, misses and false alarm can be used to compute an event level F1 score.

```
Get starts and ends of events in ground truth and comparison
for event type in number of event types do
    set current event type to 1 every other event to 0
    Keep track of events marked in ground truth and in comparison
    for event in ground truth do
        if no comparison event occurred then
            miss counter ++
            mark ground truth event as marked
        else
            for event occurring in comparison during ground truth event do
                calculate IoU
                if $IoU \geq 0.5$ then
                    hit counter ++
                    mark ground truth event as marked
                    mark comparison event as marked
                else
                    false alarm counter ++
                    mark comparison event as marked
        if event in ground truth not marked then
            miss counter ++
            mark ground truth event as marked
        for event in comparison do
            if comparison event is not marked then
                false alarm counter ++
                mark comparison event as marked
```

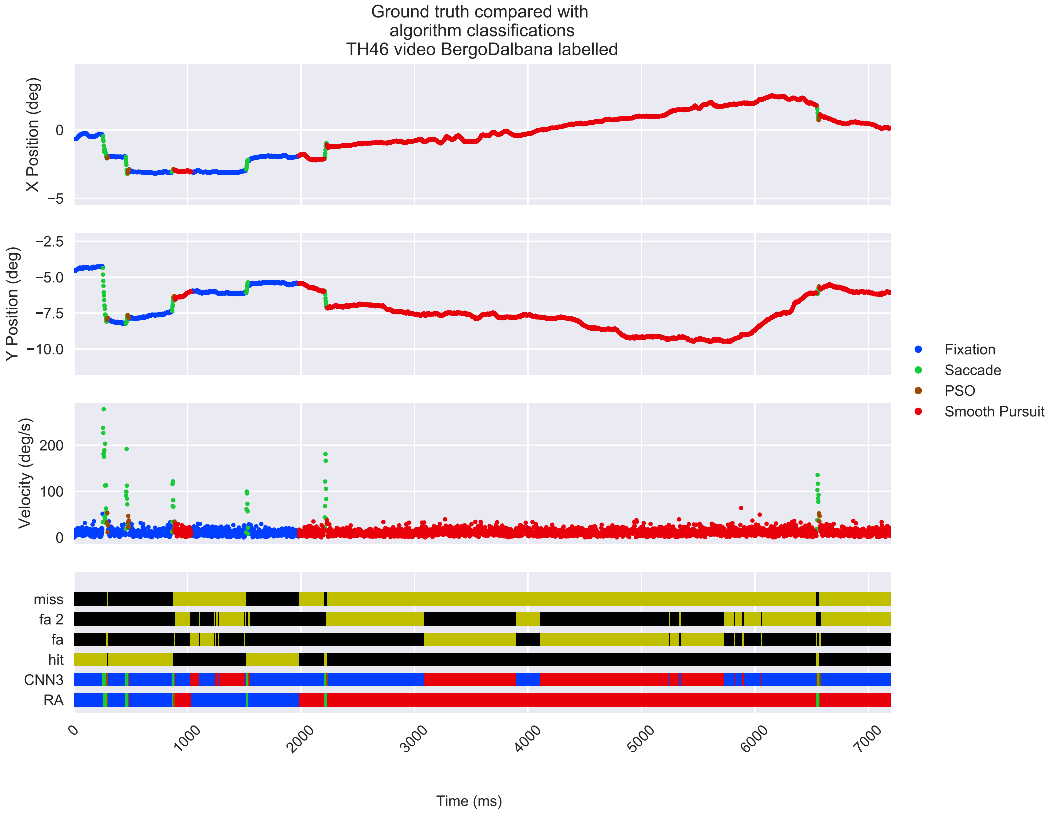


Figure 4.1: Example of the output of event level matching on one recording. RA is the ground truth and CNN3 is a model. Hits, misses and false alarms can be seen for all events.

Chapter 5

Experiment

This chapter contains the results of the initial experiment and the models classifications. An improvement of the models is attempted and its results are also shown.

5.1 Results of LOVO cross validation

Figure 5.1 shows the sample level F1 -scores from the cross validation. The errors bars for both saccade and PSO are smaller than fixation and smooth pursuit which could indicate that the networks are pretty consistent in those to eye movements across different recordings. When looking at the event level F1- scores it is quickly noticeable that smooth pursuit performs much worse at the event level than at the sample level. E.g. when looking at the model which used speed and directions the lowest performing model, CNN3, has a sample level F1 - score of 0.57 but a 0.17 event level F1- score. The event level Cohen's Kappa are very low for fixation and smooth pursuit in all cases indicating that these two classes are where the networks struggles the most. Even though the sample level F1-score for fixation is high, the the low Cohen's Kappa this could indicate that because there are a lot more fixation samples the F1 - score could be higher based by simple chance which is something the Kappa takes into account.

The best performing single feature across all metrics and models is the velocity with acceleration being behind it which make sense since they are very similar. There doesn't seem to be much variation in performance for velocity across the models. The positional feature on the other hand has more variation across models. It seems like a kernel size of 9 performs best but there is no difference between a kernel size of 3 and 29. With direction the largest kernel with size 29 seems to perform the worst. When the features are combined there doesn't seem be much improvement across models or classes and it looks similar to just using speed. In Startsev et al. [2018] the best performing model was using velocity and direction but they did not show error bars nor did they try positional features with velocity and direction.

The minimum and maximum metrics for the models that used velocity and combination of the other features can be seen in Table 5.1. The general trend is that saccades are the class it with the best performance in both sample level and event level evaluation followed by PSOs. Fixation and smooth pursuit both perform similarly in sample level F1 - score but smooth pursuit performs much worse when looking at the event level F1 - score and sample level Cohen's Kappa while fixation performs slightly worse in event level F1 - score but has a very low Kappa.

	Fixation	PSO	Saccade	SP
Sample level F-1	0.44 - 0.64	0.53 - 0.64	0.77 - 0.83	0.53 - 0.64
Event level F-1	0.38 - 0.55	0.43 - 0.59	0.81 - 0.88	0.10 - 0.20
Sample level Cohen's Kappa	0.12 - 0.31	0.52 - 0.63	0.75 - 0.82	0.07 - 0.19

Table 5.1: The minimum and maximum F-1 - score and Cohan's kappa for fixation, PSO saccade and smooth pursuit (SP) for the models that used velocity, and a combination of direction, position and acceleration.

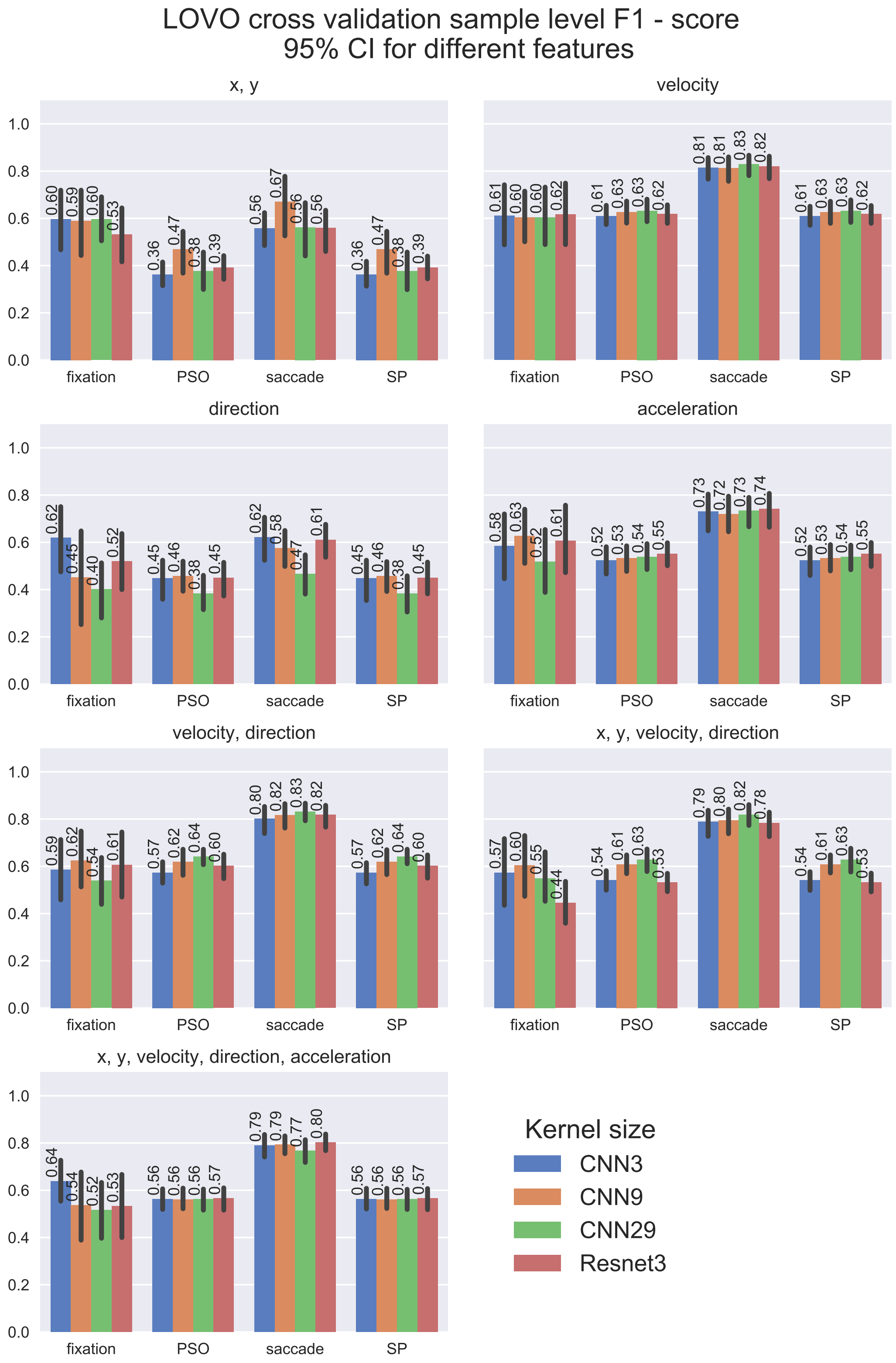


Figure 5.1: Sample level F1 score for LOVO cross validation with features and feature combination. Resnet3 has a kernel size of 3 while the CNN have the kernel size of 3, 9 and 29. The error bars are the 95% confidence interval for the 10 recordings used for LOVO.

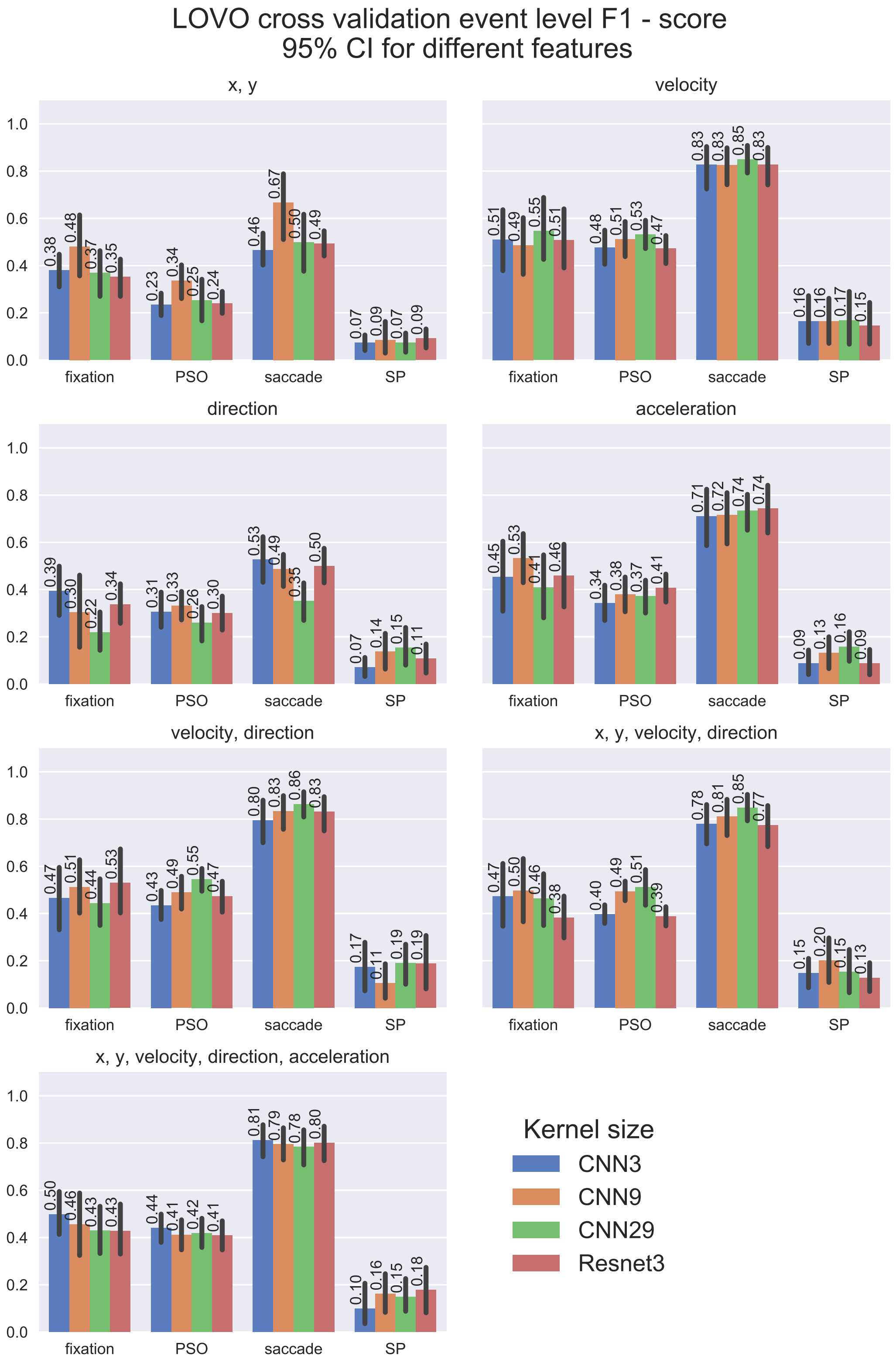


Figure 5.2: Event level F1 score for LOVO cross validation with features and feature combination. Resnet3 has a kernel size of 3 while the CNN have the kernel size of 3, 9 and 29. The error bars are the 95% confidence interval for the 10 recordings used for LOVO.

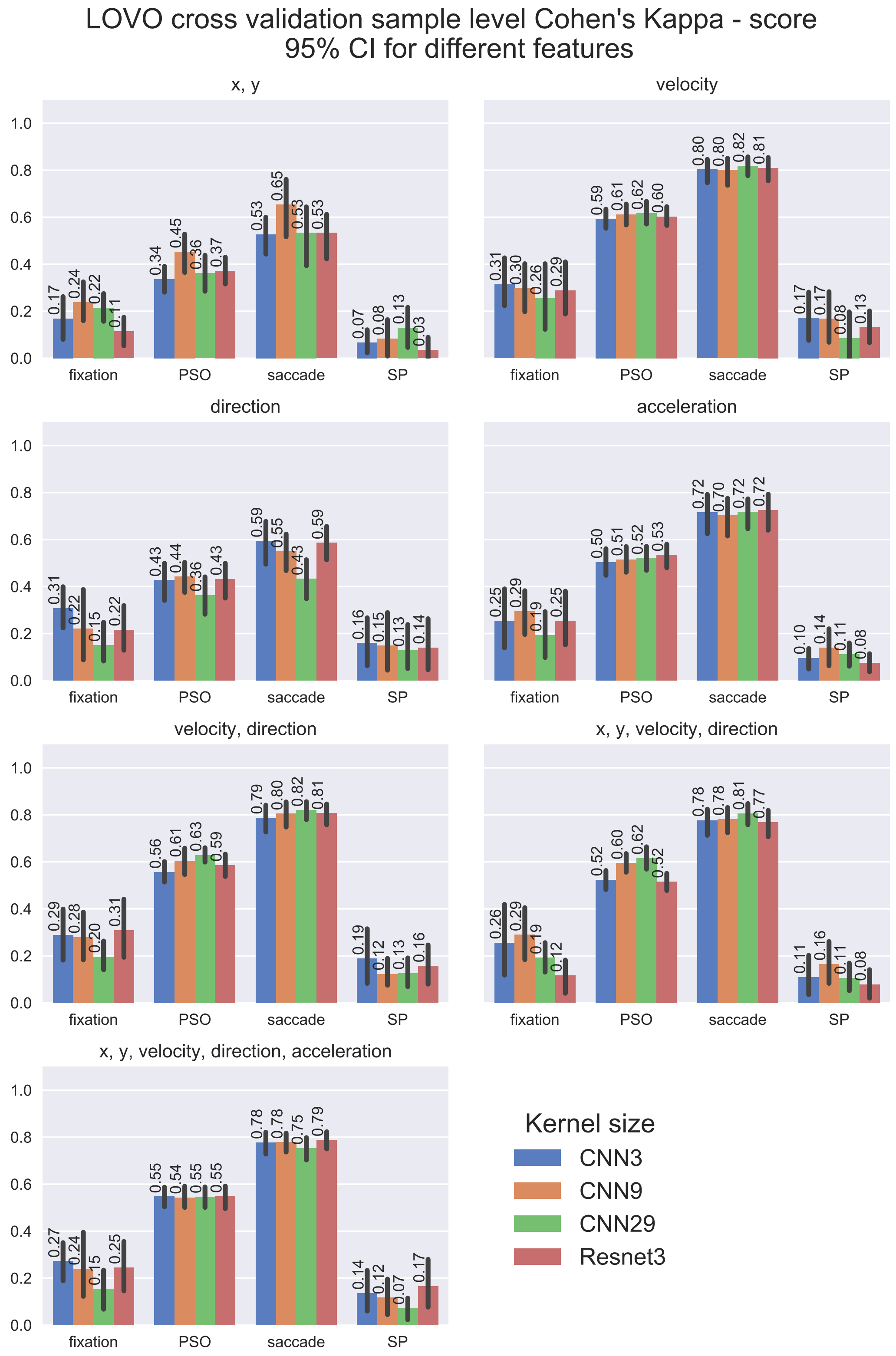


Figure 5.3: Sample level Cohen's kappa for LOVO cross validation with features and feature combination. Resnet3 has a kernel size of 3 while the CNN have the kernel size of 3, 9 and 29. The error bars are the 95% confidence interval for the 10 recordings used for LOVO.

5.2 1D-CNN-BLSTM network output

To get a better understanding of the actual output of the network manual inspection was done. The networks that used x, y, velocity, direction and acceleration were chosen to be inspected. Random recordings from their respective test set were looked at. Figure 5.4 show an example of how the classification looks. The positional trace, x and y can be seen together with velocity trace, while the bottom panel are the classifications from rater RA as the ground truth and CNN 3, 9 and 29 are the 1D-CNN-BLSTM with different kernel sizes and Resnet3 was the Resnet with kernel size 3. The window size of the network is 1028 ms.

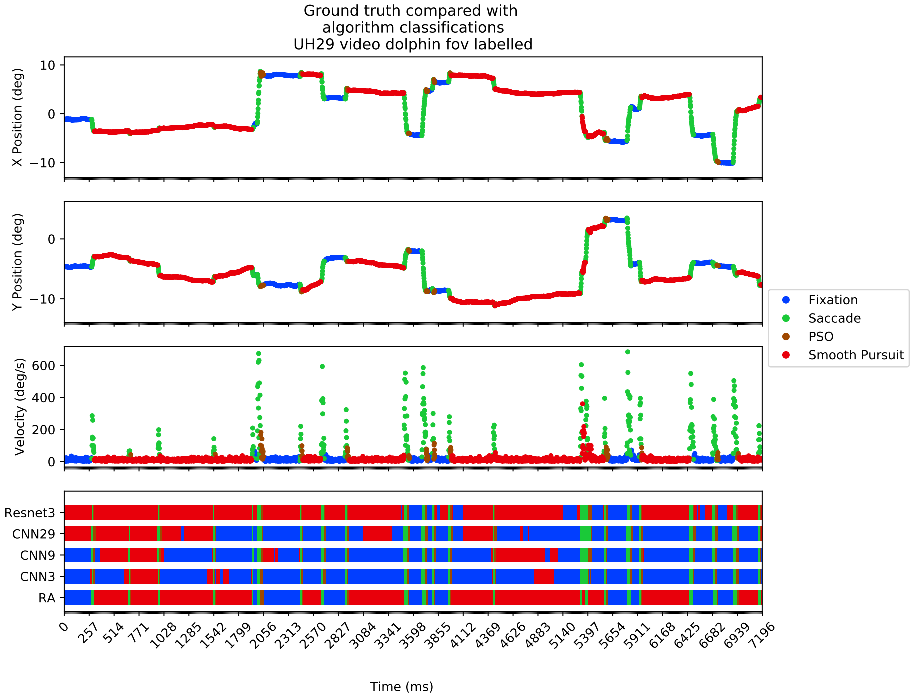


Figure 5.4: Rater RA is the ground truth and CNN 3, 9 and 29 are the 1D-CNN-BLSTM with different kernel sizes and Resnet3 was the Resnet with kernel size 3. Recording is dolphin with subject UH29.

In this subject it appears as though the Resnet3 performs better since it's classifying smooth pursuit better but this is not a general tendency. What can be seen is that some events appear very short. Figure 5.5 shows example of a close up of Figure 5.4. None of the networks are performing particularly well as CNN3 has impossibly small fixations and smooth pursuits and the saccade is very short, CNN9 also has an impossible smooth pursuit and misclassified the smooth pursuit as fixation, CNN29 misclassified smooth pursuit as fixation and Resnet3 didn't detect the PSO and had very short saccade as well. Figure 5.6 shows the event level fixation distribution and it is very obvious that all algorithms have some very short fixations and classified more fixations that were present, except for Resnet3. The short events seem to be a general problem amongst the recording which could be the cause of the event level F1 - scores for fixation and smooth pursuit being so low. The many short events break up the longer events, so even if a the majority of sample during e.g. a smooth pursuit event are classified as smooth pursuit, if a few samples in the middle are classified as fixation none of the smooth pursuit events are hits. This indicates that post processing to reclassify the impossible events is needed. A different problem is that the networks all have a hard time differentiating between fixations and smooth pursuit. The general boundaries for the fixations or smooth pursuit are usually good but the actual class itself is wrong which can be seen in Figure 5.4

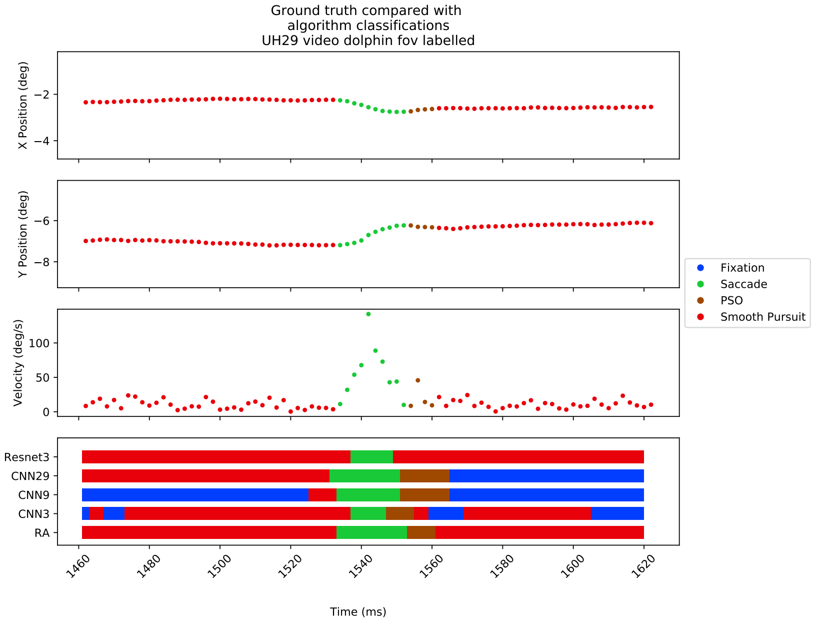


Figure 5.5: A closer looks at Figure 5.4

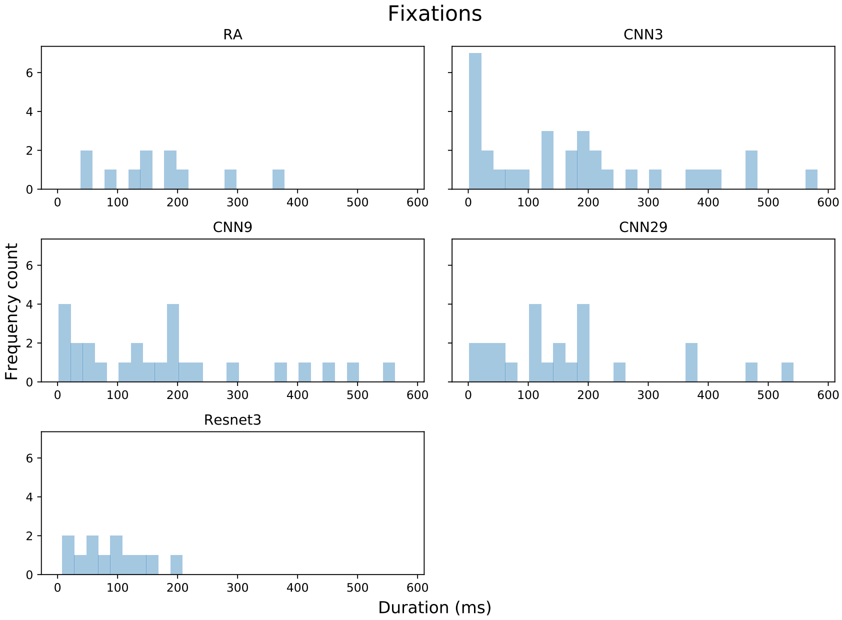


Figure 5.6: Event level fixation distribution from the ground truth and different networks for Figure 5.4

5.3 Results of improvements

One approach to try and solve the issue with the alternating too short events is to change the temporal aspect of the networks. The following was done; increase the units in the Bidirectional Long Short-Term Memory (BLSTM) layer from 16 to 64, add another BLSTM layer and lastly try a multi resolution network consisting of low resolution part that uses kernel size 3 and a high resolution part that has a kernel size of 9 or 29. The features extracted from each CNN stream are concatenated before flattening them followed by the BLSTM layers. The parameters of the model multi resolution model can be seen in Table 5.2.

5.3.1 Changing the LSTM layer

The models were trained with the same setting as section 5.1. The kernel size for the models was kept to 3 and only the LSTM layer was changed. LSTM 16 is the same network as CNN 3 used in section 5.1, the name is just changed to showcase the important settings of the network in this evaluation. LSTM 64 is the same network except with 64 units. LSTM with 2 layers has 64 units in both layers. Only the feature combination of position, velocity, acceleration and direction was used. The LOVO cross validation results can be seen in Figure 5.7 and Figure 5.7. On a sample level Cohen's Kappa doesn't reveal any changes as all scores are very comparable. The sample level F1 - scores shows a similar trend. The event level F1 - scores a

Type	Size	Kernel	Activation	Other	Type	Size	Kernel	Activation	Other
Input				520 × 5	Input				598 × 5
Conv1D	32	3			Conv1D	32	29		
BatchNorm					BatchNorm				
Activation			ReLU		Activation			ReLU	
Dropout				0.3	Dropout				0.3
Conv1D	16	3			Conv1D	16	29		
BatchNorm					BatchNorm				
Activation			ReLU		Activation			ReLU	
Dropout				0.3	Dropout				0.3
Conv1D	8	3			Conv1D	8	29		
BatchNorm					BatchNorm				
Activation			ReLU		Activation			ReLU	
Dropout				0.3	Dropout				0.3
Concatanate	16								
TDFlatten									
TDDense	32		SoftMax						
BLSTM	64								
BLSTM	64								
TDDense	514x4		SoftMax						

Table 5.2: Parameters set for the multi resolution 1D-CNN-BLSTM. The two different resolution streams are concatenated in the concatenate layer. TD are TimeDistributed layers and Classes are the number of classes.

very small improvement of 0.8 in smooth pursuit but at the expense of the other classes. With LSTM 64 it looks like it is as the expense of saccades and PSOs while the 2 layer LSTM 64 has similar performance as LSTM 16 in saccades and PSO but a decrease of 0.8 in fixation. Overall there does not seem to be significant change in the performance of any metric.

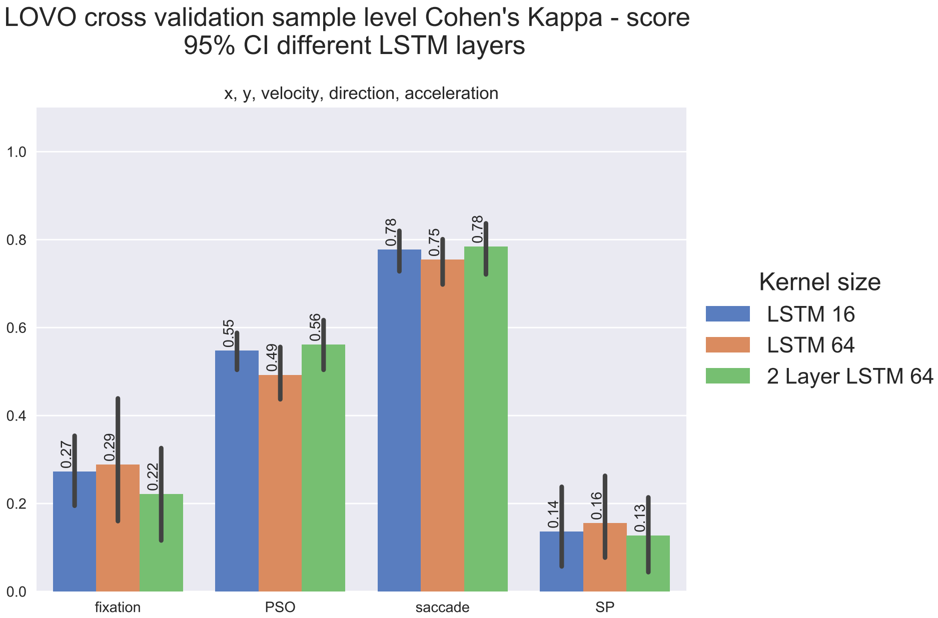
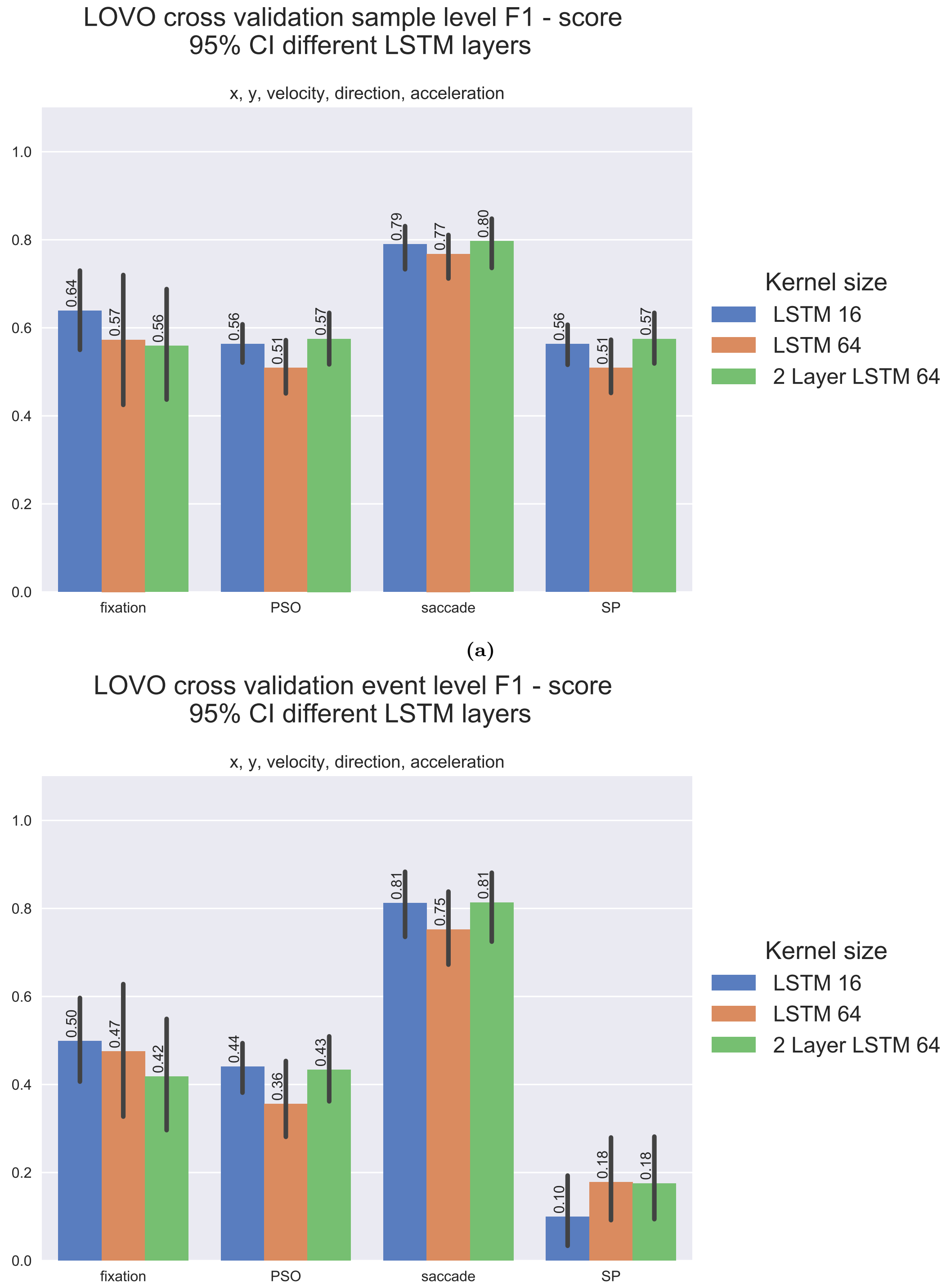


Figure 5.7: Sample level Cohen's Kappa for LOVO cross validation when using position, velocity, acceleration and direction. All models had a kernel size of 3, only the LSTM layer was changed. The error bars are the 95% confidence interval for the 10 recordings used for LOVO.



(a)

(b)

Figure 5.8: Sample level (Figure 5.8a) and event level (Figure 5.8b) F1 score for LOVO cross validation when using position, velocity, acceleration and direction. All models had a kernel size of 3, only the LSTM layer was changed. The error bars are the 95% confidence interval for the 10 recordings used for LOVO.

5.3.2 Multi resolution network

The multi resolution network had a significant increase in training time. 500 epochs took around 30.5 hours for one model so cross validation would have taken over two weeks. To look at the effect a model with kernel size (3, 9) and (3, 29) was trained for 100 epochs holding out the recording "video BergoBalbana" as a test set. A single model with kernel size (3, 9) was trained for 500 epochs. All models had a single LSTM layer of 64 units. Their results can be seen in Figure 5.9 and Figure 5.10. For the Kappa the model with kernel (3, 9) outperforms the (3, 29) model, especially for PSOs. The 500 epochs (3, 9) model performs better with saccades and PSOs then the same model with only 100 epochs but performs much worse with fixation and smooth pursuit. None seem to be superior and they are comparable to LSTM 16. When looking at sample level F1 - score the model the (3, 9) 500 epoch model outperforms both the other multi resolution models but does not outperform LSTM 16 as they are very comparable. A the event level a difference does appear as the smooth pursuit F1 - score is much higher for the multi resolution networks. The (3, 9) models perform comparably with fixations and saccades, and a drop of 0.04 and 0.1 in PSO. These could indicate that the multi resolution networks had an effect on the small alternating events described in section 5.2.

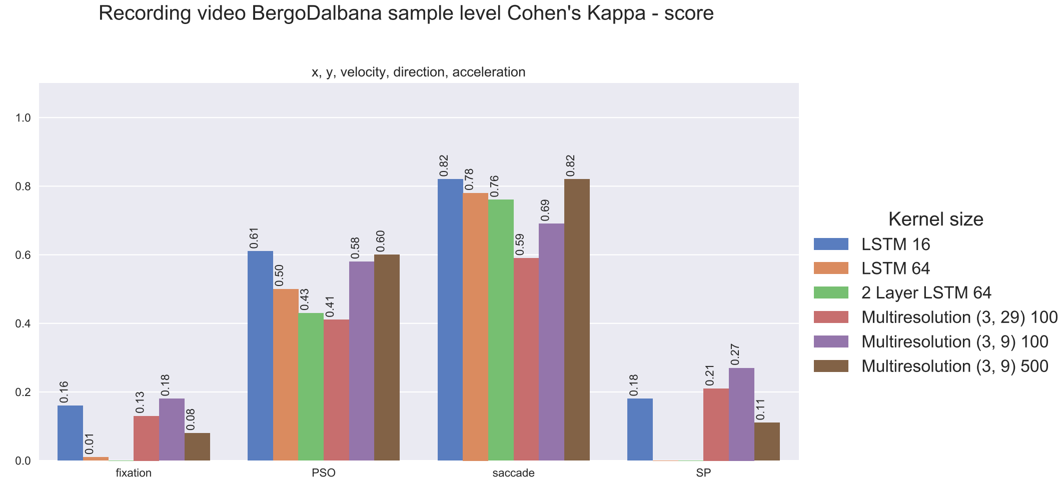
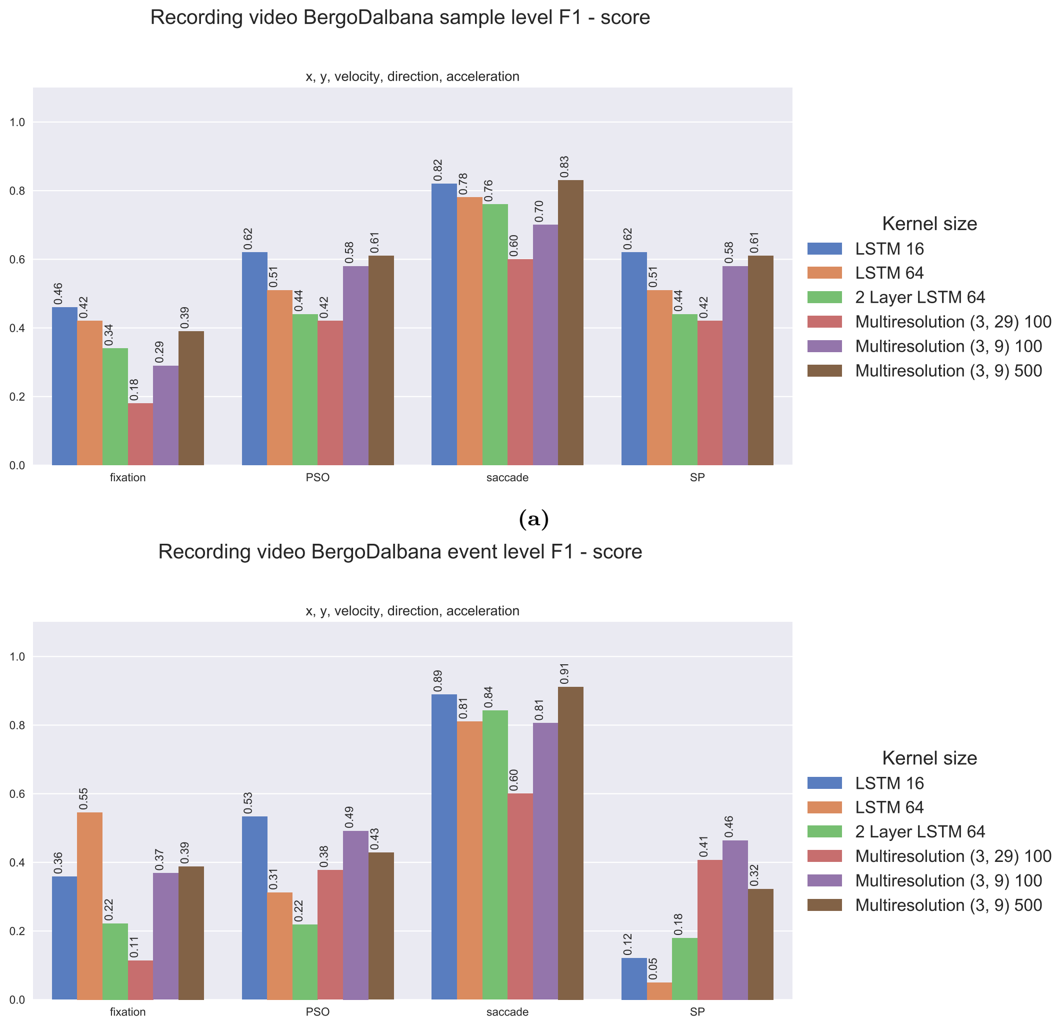


Figure 5.9: Sample level Cohen's Kappa for the recording "video BergoBalbana" when using position, velocity, acceleration and direction. The model Multi Resolution (3, 9) 100 was trained for 100 epochs while Multi Resolution (3, 9) 500 was trained for 500 epochs. The 2 layer LSTM 64 fixation and smooth pursuit score has a negative Cohen's Kappa at -0.04 both. The LSTM 64 smooth pursuit score is at -0.01. Since the figure is capped at 0 they are not shown.



(a)

(b)

Figure 5.10: Sample level (Figure 5.10a) and event level (Figure 5.10b) F1 score for the recording "video BergoBalbana" when using position, velocity, acceleration and direction. The model Multi Resolution (3, 9) 100 was trained for 100 epochs while Multi Resolution (3, 9) 500 was trained for 500 epochs.

5.4 Output of improvements

To see if the changes have had an effect on the output manual inspection is performed again. Figure 5.11 shows the models with different LSTM layers and the multi resolution models. RA is again the ground truth. What the multi resolution networks should output were less shorter events and more continuous fixations and smooth pursuit. The Multiresolution (3, 29) 100 epoch networks looks promising as it is only around 500 ms that an alternation is found and the fixation event is also not unreasonably small. On the other hand it did miss an obvious saccade that the other networks found. The (3, 9) kernel model in both 100 and 500 epochs seem to have the impossibly small events present but not in the same places as single kernel models. Figure 5.12 shows the network outputs of the same recording but a different subject. The Multiresolution (3, 29) model has completely missed most of the saccades which is also reflected in F1 - scores in Figure 5.10. There are again impossibly small events on the other multi resolution networks.

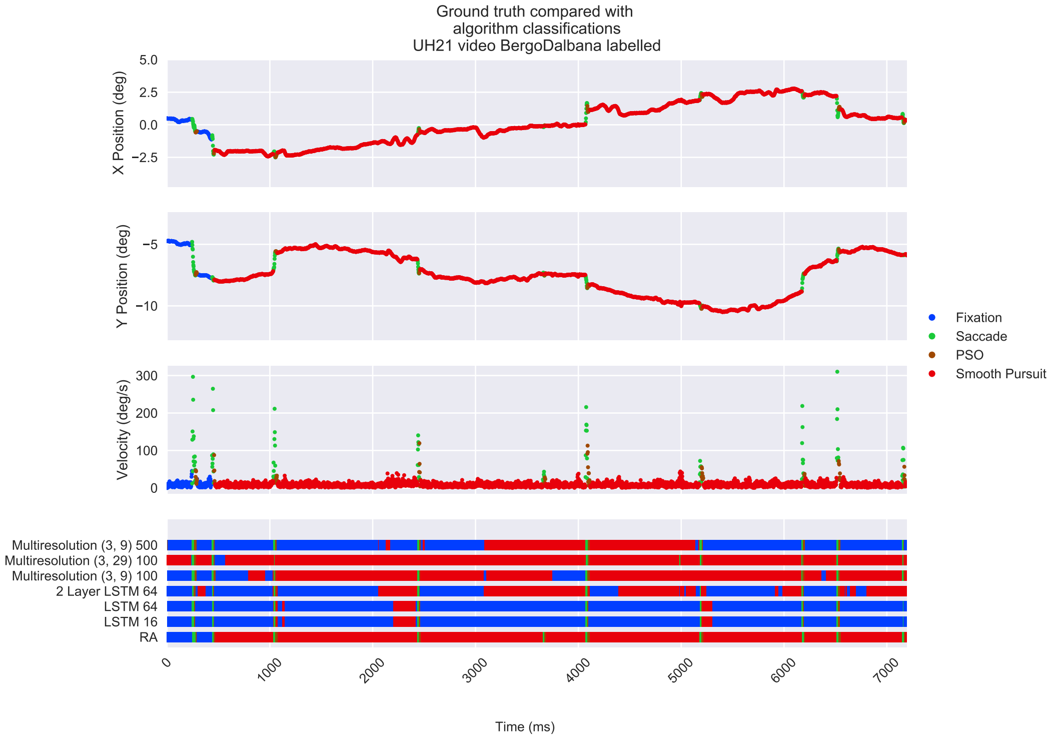


Figure 5.11

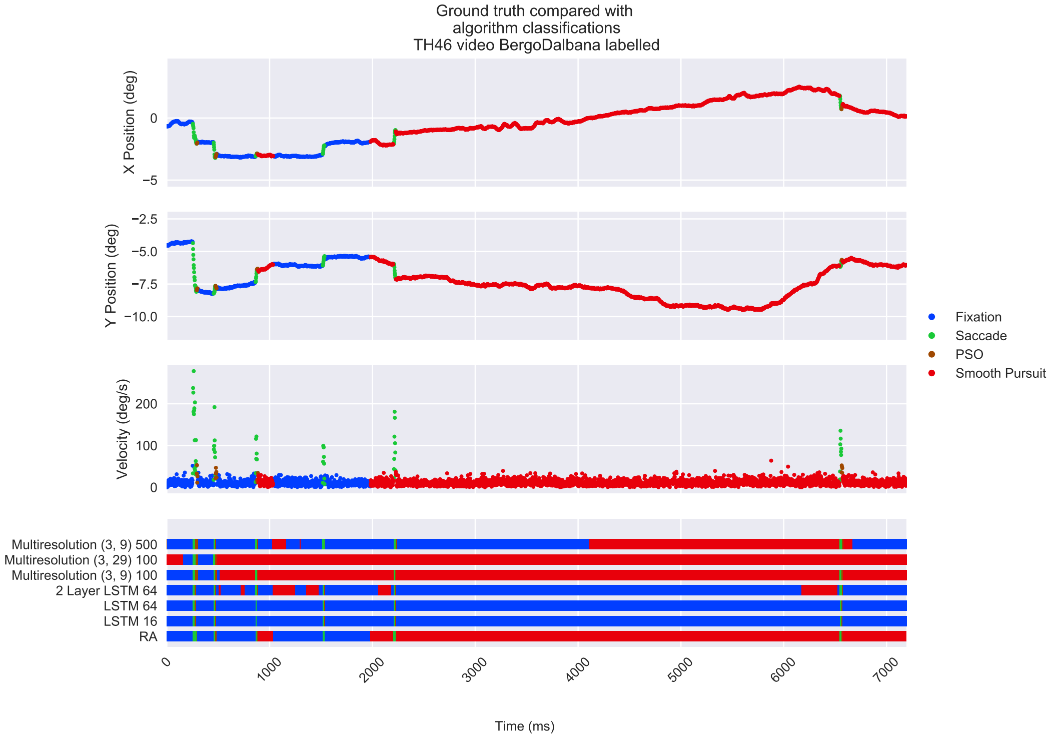

Ground truth compared with
algorithm classifications
TH46 video BergoDalbana labelled
X Position (deg)
0
−5
Y Position (deg)
−2.5
−5.0
−7.5
−10.0
Velocity (deg/s)
200
100
0
Fixation
Saccade
PSO
Smooth Pursuit
Multiresolution (3, 9) 500
Multiresolution (3, 29) 100
Multiresolution (3, 9) 100
2 Layer LSTM 64
LSTM 64
LSTM 16
RA
0
1000
2000
3000
4000
5000
6000
7000
Time (ms)

Figure 5.12

Chapter 6

Discussion

For a machine learning approach choosing the right dataset is one of the most important steps. The scarce availability of datasets and their varying number and definitions of eye movement makes it challenging to choose the right dataset. It is important to know the methodology and the definitions of eye movements that were used when the dataset was labelled. Looking at the velocity, direction and event durations distributions for each eye movement should be done to sanity check that the labels do indeed make sense. The main sequence [Bahill et al., 1975b] is also an excellent tool to investigate the labelled saccades and locate possible errors or outliers. In chapter 3 an example of bad labelling is showcased with the large dataset GazeCom [Dorr et al., 2010]. The dataset is described as manually annotated while it is in reality automatically annotated and manually looked through afterwards. The lack of proper definitions of the eye movements show up in impossibly short and impossibly long saccade durations. Multiple examples have been shown with questionable labels. This can have serious implications as all the machine learning research done with this dataset can become moot as the outputs of networks are not based on real eye movements. An alternative dataset, the Lund 2013 dataset [Larsson et al., 2013], is proposed. It is much smaller totalling in 12.75 minutes of manually labelled compared to GazeComs 4.8 hours of labelled recording. On the other hand it 500 Hz instead of GazeCom's 250 Hz and it was labelled by Marcus Nyström (MN) and Richard Anderson (RA) who are both distinguished eye tracking researchers with many years of eye movement experience. No definitions of movements were used to label the dataset other than their respective experiences but the difference in quite clear when looking at each datasets feature distributions. Lund 2013 is much sharper in the feature distribution shapes especially the directions. The event durations and velocities also match the physiological properties of eye movements described in section 2.1.

A different issue is how to evaluate an algorithms classifications of the dataset against the ground truth. Especially in cases where multiple raters are involved as raters often disagree between each other to a certain degree [Hooge et al., 2017]. [Dorr et al., 2010] attempted to solved this issue by having two human raters produce annotations and then merging those annotations on an unknown criteria. This did produce some problems as it created multiple events that were 1 sample long. With the Lund 2013 there are no final annotations, only each raters annotations. The rater with most annotations was chosen as the ground truth and the other raters labels were discarded. After a ground truth is chosen the question becomes what metrics to use. There is not a consensus amongst the eye movement community on what metrics to use and many papers develop their own metrics. The most common metrics is sample level F1 - score and sample level Cohen's Kappa. But as seen in section 5.1 it is possible to have a high sample level F1 - score but when looking at output of the algorithms it is clear that it is wrong. This is something that can be seen in an event level evaluation. There currently exists four different approaches to match events from ground truth to events from the algorithm. These matchings can produce true positives, false positives and false negatives which can be used to calculate and F1 - score [Hooge et al., 2017]. IoU can then be used to determine how close an overlap the matched events need to have to be classified as a true positive [Startsev et al., 2018]. This event matching algorithm has been described by Startsev et al. [2018] but no implementation of it has been available so it has been implemented from scratch and pseudo code for the implementation can be found in section 4.3. These metrics are global metrics and make it easier to compare performance between algorithms but they do not have any informations about the shortcomings of the algorithms. There exists two metrics, RTD and RTO [Hooge et al., 2017] that provide information about the timing differences between matched events but they still only produces one number. A more informative version of RTD and RTO could be to report the histograms of the timing differences. Since RTD and RTO have a different matching criteria than what was used in this project these were not used. Instead manual and more a qualitative inspection was done with examples of how the algorithms classifies eye movements. These provided a deeper but more abstract insight into where the algorithms make mistakes and what type of mistakes it makes.

The neural network architecture 1D-CNN-BLSTM by Startsev et al. [2018] was chosen as the basis network for this project. It was made up of three CNN layers with a 1D kernel of size 3 and a BLSTM layer of 16 units. The network took 1028 ms windows as input and performed sequence to sequence classification fixations, saccades, PSO and smooth pursuit. There was no clear test set so LOVO cross validation was used. The features instantaneous velocity, acceleration, direction and gaze position and a combination of said features were used to see what features performed well. Velocity was the feature that performed best on its own which is not surprising as many eye movement classification algorithms are velocity based. Acceleration performed similarly albeit had a slightly lower F1 and Kappa score which was to be expected as distribution of velocity and acceleration are very similar. The positional features seemed to perform the worst, while direction seemed to be slightly better than positional which is surprising as the distributions of direction for the different movements seemed to have distinct patterns. When combining features there were no real improvement gain which is contradictory to Startsev et al.

[2018] finding that combining velocity and direction should improve the performance. Multiple kernel sizes were also tried with rationale that smooth pursuit is a movement with slow gradual change which is easier to detect over a longer duration. Kernel size 3, 9 and 29 were used but there was no real difference when combining features. On the other hand a difference can be seen when only using direction as the longest kernel, 29, seemed to perform worse especially for saccades. This was not the case when using velocity as it had no effect on it which also explains why it did not have an effect on the combined features.

To try and understand how the networks actually labelled the movements manual inspection of various random recordings and subjects was done. The general trend was that most of the saccade errors seemed to stem from small saccade with amplitudes of less than 0.5-1°. These are tough to label even for a rater as there can be multiple things causing those movements like noise, micro saccades or fixation drift. PSOs are ill defined so even raters disagree amongst them selves what categorizes a PSO. This shows in the network output as many PSOs marked by the network were not in labelled PSOs by ground truth but upon insepction of the velocity trace it becomes move clear why it could be interpreted as a PSO. In general saccades and PSOs had the highest sample Kappas being in the area of 0.8 and 0.55 respectively. In comparison fixations and smooth pursuit were around 0.25 and 0.15. This seemed to be cause by two issues; 1) the networks had a hard time distinguishing fixations from smooth pursuit 2) the networks created impossibly short fixations and smooth pursuit and would create alternating sequences of short fixations and smooth pursuit when the ground truth was just one long event. The first issue is a well known problem as differentiating fixation from smooth pursuit is notoriously difficult as their characteristics are very similar. During a fixation the eye stands still but due to recording noise, tremors in the eye and fixation drift the eye is actually constantly moving at a very slow velocity. Smooth pursuit is actual movement whose speed is determined by the object that the eye is following. The slow movement makes is difficult to differentiate between them. The second issue stems from the network not knowing the physical properties of the movements e.g. a fixation below 200 ms are uncommon, section 2.1 or that the short 10 ms fixations which it outputs are impossible events. This problem could be handled in two ways; improve the network or do post processing to detect and merge the impossible events. An attempt to improve the model was done by changing the parameters of the temporal part of the network - the BLSTM layer. The amount of units was changed from 16 to 64 and a second BLSTM was added. These did not seem to have desired effect as the impossible events were still present but they were just found in other parts of the recording than before. A multi resolution network was also tried with the reasoning being that a larger kernel, size 9 or 29, might be better for smooth pursuit while the shorter kernel of size 3 would be better for the shorter events like saccades and PSO. Since it took 30.5 hours to train the multi resolution model for 500 epochs it was only trained once with the video BergoDalbana kept as test set instead of using the LOVO cross validation. The (3, 29) kernel did produce less of the impossibly short events but that was at cost of not detecting the saccades and overly classifying events as smooth pursuit. It also had less of the alternating sequences with multiple small fixations and smooth pursuit. The (3, 9) kernel with 500 epochs did seem to produce higher event level F1 - scores for smooth pursuit than the LSTM 16, 0.32 and 0.12 respectably, while having slightly higher scores for fixation and saccades, 0.39 and 0.91 compared to 0.36 and 0.89 respectively. This seemed to come at the cost of having lower PSO scores, 0.43 and 0.53. This indicates that there could be potential in a multi resolution model, especially if it did indeed create less of the impossible events but more research would be needed. Since the comparison was only made with one test set, cross validation could show if the higher performance was due to variance or if was and actual improvement. This would also make post processing easier as less events would have to be merged which can be difficult to determine when there is an alternating sequence present.

Chapter 7

Conclusion

This project showcased some of the difficulties in making a deep learning based eye movement classification algorithm. Since neural networks have to train on a dataset, the importance of having a good data set is imperative. Choosing a good dataset is difficult as manually annotating eye movements is an exhaustive and time consuming task so the availability of public databases is scarce. The biggest contribution from this project was showing that by looking at the distributions of the gaze position, velocity and direction for each eye movement type it is possible to determine how well the dataset was classified. The main sequence and event duration distribution can also be used to ensure that the labelled saccades make sense compared the physiological properties of saccades. It was shown that the classifications of the GazeCom dataset do not make physiological sense and even though it was claimed to be manually annotated it was in fact automatically labelled by three different eye movement classification algorithms and then superficially inspected by raters. This has important implications as previous work done based on this dataset becomes questionable as a model trained on a wrongly labelled dataset will produce meaningless classifications. Instead the much shorter dataset Lund 2013 was proposed as it was actually manually annotated by eye movement experts. It was shown that its feature distributions were much more distinct for the different eye movements and the saccades make physiological sense. To evaluate an algorithms performance sample level F1 - score and Cohen's Kappa were chosen as these were the most common metrics used in the field although many other metrics exist and there does not seem to be an agreement as to what to use. An event level evaluation was implemented by matching events in the ground truth with events in the algorithm stream to count the number of hits, false alarms and misses. These could be used to calculate an event level F1 - score. Intersection over Union (IoU) with a threshold of 0.5 was used to determine the criteria for when an event is a hit. A sequence to sequence 1D Convolutional Neural Network Bidirectional Long Short-Term Memory (1D-CNN-BLSTM) neural network by Startsev et al. [2018] was used the baseline model to classify fixations, saccades, smooth pursuit and Post-Saccadic Oscillations (PSOs). A kernel size of 3, 9 and 29 and a Residual Network (Resnet) version of 1D-CNN-BLSTM were tried with the features gaze position, velocity, acceleration and direction and a combination of said features. Leave-One-Video-Out (LOVO) cross validation was performed on the ten stimuli from the Lund 2013 dataset. The best performing single feature was velocity. When combining features there was no real improvement. Between the three kernel sizes there were no real difference in model performance for velocity and combinations of features that used velocity. There was a performance difference between models when using direction as kernel size kernel size 9 and 29 performed worse. A manual inspection of the network outputs showed that the networks had a hard time differentiating between fixation and smooth pursuit. This created impossibly short events that would sometimes result in alternating sequences of short fixations and smooth pursuit instead of a longer event of either type. This resulted in lower event level F1 - scores. To improve the temporal aspect of the network different Bidirectional Long Short-Term Memory (BLSTM) configurations were tried. The unit size of 16 and 64 for the BLSTM were tried together with adding an additional BLSTM layer. This did not seem have a noticeable improvement on the event level F1 - score. A multi resolution model with a low resolution channel of kernel size 3 and a high resolution channel of kernel size 9 and 29 was tried. The multi resolution seemed to have less of the impossibly short events and had higher scores but it was only compared with 1 test set instead of cross validation as the model took 30.5 hours to train. The sample level F1 - scores, sample level Cohen's Kappa and event level F1 - scores for the multi resolution model were respectively; fixations - 0.39, 0.08, 0.39, saccades - 0.83, 0.82, 0.91, smooth pursuit - 0.61, 0.11, 0.32 and PSO - 0.61, 0.60, 0.43. The multi resolution model is promising but a full cross validation will have to be done to properly compare its performance to the other models. The problem of distinguishing fixations from smooth pursuit was not solved as it is a difficult task that needs more research.

Chapter 8

Furher work

There are several things that need to be done. A post processing algorithm will have to be implemented by choosing criteria for unacceptable events and merging them with other events. A common criteria is event duration to ensure that saccades or fixations cannot be too short. The challenges lies in how to determine what the unacceptable event should be classified as in stead. In cases where both the adjacent events are of the same type it can be just labelled as them. The challenge arises when the adjacent events are two different events. A similar but slightly different challenge are the alternating sequences of fixations and smooth pursuit that were present in the manual inspection of the output. It is not straight forward to decides if they should all be either fixations or smooth pursuit.

Comparison of the neural networks performance with other algorithms will also be nessecary to determine how well it actually performs. This can be a challenge as many of the classic algorithms have thresholds that need to be manually set. One approach could be to perform a grid search over the entire data set for the different algorithm parameters but this could artificially increase their performance as they would have knowledge of the test set while the network does not. It becomes difficult with algorithms like the Modified Nyström and Holmqvist (MNH) that were designed specifically for a certain frequency and task. There are other challenges when implementing other researchers deep learning networks as they may not have the model weights available or have done some specific pre processing of the data they do not mention. There is also the issue that many algorithms do not classify both smooth pursuit and PSO.

To evaluate the neural networks generalisability other datasets will have to be used. This poses multiple challenges as the datasets are often recorded at different frequencies with different eye trackers. This creates inherent differences that may give neural networks trained on them an inherent edge. It also poses the issue that not many datasets are available and their definition of events also vary. It is reasonable to think that a network trained on GazeCom will produce much different saccades than one trained on Lund 2013 as the saccade boundaries in the ground truth are much different. This would create lower scores because the ground truth is operating on a different definition of where the movements start and end.

To improve the difficult task of differentiating fixations from smooth pursuit a different approach could be necessary. The first step would be to create a new dataset with clinical smooth pursuit stimuli of varying velocities and directions. This would allow investigation into what types of smooth pursuit the network struggles with and which it is able to detect. This could give some insight into how to improve the smooth pursuit detection. It would also be a good approach to review the literature for alternative features that others have used for smooth pursuit and see how it would impact the classification

Bibliography

Agtzidis, I., Startsev, M., and Dorr, M. In the pursuit of (ground) truth: A hand-labelling tool for eye movements recorded during dynamic scene viewing. In *Proceedings of the 2nd Workshop on Eye Tracking and Visualization, ETVIS 2016*, pp. 65–68. IEEE, oct 2017. ISBN 9781509047314. doi: 10.1109/ETVIS.2016.7851169. Available at: <http://ieeexplore.ieee.org/document/7851169/>.

Bahill, A. T., Clark, M. R., and Stark, L., 1975. Dynamic overshoot in saccadic eye movements is caused by neurological control signal reversals. *Experimental Neurology*, jul, 48(1), pp. 107–122. ISSN 10902430. doi: 10.1016/0014-4886(75)90226-5. Available at: <https://www-sciencedirect-com.zorac.aub.aau.dk/science/article/pii/0014488675902265>.

Bahill, A., Clark, M. R., and Stark, L., 1975. The main sequence, a tool for studying human eye movements. *Mathematical Biosciences*, jan, 24(3-4), pp. 191–204. ISSN 00255564. doi: 10.1016/0025-5564(75)90075-9. Available at: <https://www.sciencedirect.com/science/article/pii/0025556475900759http://linkinghub.elsevier.com/retrieve/pii/0025556475900759>.

Braunagel, C., Geisler, D., Stolzmann, W., Rosenstiel, W., and Kasneci, E., 2016. On the necessity of adaptive eye movement classification in conditionally automated driving scenarios. *Etra*, (2), pp. 19–26. doi: 10.1145/2857491.2857529. Available at: <http://dx.doi.org/10.1145/2857491.2857529>.

Chen, H. Y. and Chien, J. T., 2015. Deep semi-supervised learning for domain adaptation. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2015-Novem, pp. 1–6. ISSN 21610371. doi: 10.1109/MLSP.2015.7324325.

Chen, S. and Epps, J., 2013. Automatic classification of eye activity for cognitive load measurement with emotion interference. *Computer Methods and Programs in Biomedicine*, may, 110(2), pp. 111–124. ISSN 0169-2607. doi: 10.1016/J.CMPB.2012.10.021. Available at: <https://www.sciencedirect.com/science/article/pii/S0169260712002830{#}bib0125>.

Deubel, H. and Bridgeman, B., 1995. Fourth Purkinje image signals reveal eye-lens deviations and retinal image distortions during saccades. *Vision Research*, feb, 35 (4), pp. 529–538. ISSN 00426989. doi: 10.1016/0042-6989(94)00146-D. Available at: <https://www.sciencedirect.com/science/article/pii/004269899400146Dhttps://linkinghub.elsevier.com/retrieve/pii/004269899400146D>.

Dorr, M., Martinetz, T., Gegenfurtner, K. R., and Barth, E., 2010. Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision*, aug, 10 (10), pp. 28–28. ISSN 1534-7362. doi: 10.1167/10.10.28. Available at: <http://jov.arvojournals.org/Article.aspx?doi=10.1167/10.10.28>.

Friedman, L., Nixon, M. S., and Komogortsev, O. V., 2017. Method to assess the temporal persistence of potential biometric features: Application to oculomotor, gait, face and brain structure databases. *PLoS ONE*, jun, 12(6), p. e0178501. ISSN 19326203. doi: 10.1371/journal.pone.0178501. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/28575030http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5456116https://dx.plos.org/10.1371/journal.pone.0178501>.

Friedman, L., Rigas, I., Abdulin, E., and Komogortsev, O. V., 2018. A novel evaluation of two related and two independent algorithms for eye movement classification during reading. *Behavior Research Methods*, aug, 50(4), pp. 1374–1397. ISSN 15543528. doi: 10.3758/s13428-018-1050-7. Available at: <http://link.springer.com/10.3758/s13428-018-1050-7>.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning.* MIT Press, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pp. 770–778, dec 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.90. Available at: <http://arxiv.org/abs/1512.03385>.

Hessels, R. S., Niehorster, D. C., Nyström, M., Andersson, R., and Hooge, I. T. C., 2018. Is the eye-movement field confused about fixations and saccades? A survey among 124 researchers. *Royal Society open science*, aug, 5(8), p. 180502. ISSN 2054-5703. doi: 10.1098/rsos.180502. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/30225041http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6124022>.

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Halszka, J., and van de Weijer, J. *Eye tracking: A comprehensive guide to methods, and measures.* Oxford University Press, 2011. ISBN 9780199697083.

Hooge, I. T., Niehorster, D. C., Nyström, M., Andersson, R., and Hessels, R. S., 2017. Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior Research Methods*, oct, 50(5), pp. 1864–1881. ISSN 15543528. doi: 10.3758/s13428-017-0955-x. Available at: <http://link.springer.com/10.3758/s13428-017-0955-x>.

Hoppe, S. and Bulling, A., 2016. End-to-End Eye Movement Detection Using Convolutional Neural Networks. sep. Available at: <http://arxiv.org/abs/1609.02452>.

Komogortsev, O. V. and Karpov, A., 2013. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods*, mar, 45(1), pp. 203–215. ISSN 1554351X. doi: 10.3758/s13428-012-0234-9. Available at: <http://link.springer.com/10.3758/s13428-012-0234-9>.

Komogortsev, O. V., Jayarathna, S., Koh, D. H., and Gowda, S. M., 2009. Qualitative and Quantitative Scoring and Evaluation of the Eye Movement Classification Algorithms. *Proceedings of ACM Eye Tracking Research & Applications Symposium, Austin, TX*, p. 10. doi: 10.1145/1743666.1743682.

Larsson, L., Nystrom, M., and Stridh, M., 2013. Detection of Saccades and Postsaccadic Oscillations in the Presence of Smooth Pursuit. *IEEE Transactions on Biomedical Engineering*, sep, 60(9), pp. 2484–2493. ISSN 0018-9294. doi: 10.1109/TBME.2013.2258918. Available at: <http://ieeexplore.ieee.org/document/6504734/>.

Leigh, J. R. and Zee, D. S. *The Neurology of Eye Movements.* 4 edition, 2004. ISBN 978-0-19-530090-1.

Meyer, C. H., Lasker, A. G., and Robinson, D. A., 1985. The upper limit of human smooth pursuit velocity. *Vision Research*, jan, 25(4), pp. 561–563. ISSN 0042-6989. doi: 10.1016/0042-6989(85)90160-9. Available at: <https://www.sciencedirect.com/science/article/pii/0042698985901609>.

Nyström, M., Hooge, I., and Holmqvist, K., 2013. Post-saccadic oscillations in eye movement data recorded with pupil-based eye trackers reflect motion of the pupil inside the iris. *Vision Research*, nov, 92, pp. 59–66. ISSN 00426989. doi: 10.1016/j.visres.2013.09.009. Available at: <https://www.sciencedirect.com/science/article/pii/S0042698913002356?via{%}3Dihub{#}b0085>.

Patney, A., Kim, J., Salvi, M., Kaplanyan, A., Wyman, C., Benty, N., Lefohn, A., and Luebke, D. Perceptually-based foveated virtual reality. In *ACM SIGGRAPH 2016 Emerging Technologies on - SIGGRAPH '16*, pp. 1–2, New York, New York, USA, 2016. ACM Press. ISBN 9781450343725. doi: 10.1145/2929464.2929472. Available at: <http://dl.acm.org/citation.cfm?doid=2929464.2929472>.

Salvucci, D. D. and Goldberg, J. H., 2000. Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the symposium on Eye tracking research & applications - ETRA '00*, pp. 71–78. ISSN 10960384. doi: 10.1145/355017.355028. Available at: <http://portal.acm.org/citation.cfm?doid=355017.355028>.

Santini, T., Fuhl, W., Kübler, T., and Kasneci, E., 2015. Bayesian Identification of Fixations, Saccades, and Smooth Pursuits. *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*, nov, pp. 163–170. ISSN 1471-2105. doi: 10.1145/2857491.2857512. Available at: <http://dl.acm.org/citation.cfm?doid=2857491.2857512http://arxiv.org/abs/1511.07732>.

Startsev, M., Agtzidis, I., and Dorr, M., 2018. 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods*, nov, pp. 1–17. ISSN 1554-3528. doi: 10.3758/s13428-018-1144-2. Available at: <http://link.springer.com/10.3758/s13428-018-1144-2>.

Tafaj, E., Kasneci, G., Rosenstiel, W., and Bogdan, M. Bayesian online clustering of eye movement data. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12*, p. 285, New York, New York, USA, 2012. ACM Press. ISBN 9781450312219. doi: 10.1145/2168556.2168617. Available at: <http://dl.acm.org/citation.cfm?doid=2168556.2168617>.

Tafaj, E., Kübler, T. C., Kasneci, G., Rosenstiel, W., and Bogdan, M. Online Classification of Eye Tracking Data for Automated Analysis of Traffic Hazard Perception. In *IEEE Signal Processing Magazine*, volume 30, pp. 442–450. sep 2013. ISBN 978-3-642-40728-4. doi: 10.1007/978-3-642-40728-4_56. Available at: <http://www.csie.ntu.edu.tw/{~}cjlin/talks/rome.pdfhttp://ieeexplore.ieee.org/document/6582713/http://link.springer.com/10.1007/978-3-642-40728-4{_}56>.

Tobii, 2019. *Tobii and HTC Bring Eye Tracking to Next Generation VR Headset.* Available at: <https://www.tobii.com/siteassets/tobii-and-htc-bring-eye-tracking-to-next-generation-vr-headset-press-release-8-jan-2019/?v=1>.

Zemblys, R., Niehorster, D. C., and Holmqvist, K., 2018. gazeNet: End-to-end eye-movement event detection with deep neural networks. *Behavior Research Methods*, oct, pp. 1–25. ISSN 1554-3528. doi: 10.3758/s13428-018-1133-5. Available at: <http://link.springer.com/10.3758/s13428-018-1133-5>.