UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Determining the projected frontal area of cyclists using motion capturing data

Gerrit Mark van Kruistum

MSc. Thesis
September 2020

Supervisors:
dr. M. Poel (EEMCS)
dr. G.W.J. Bruinsma (BMS)
dr. F.J. Wouda (Xsens)

Data Management & Biometrics Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Acknowledgements

This thesis marks not only the end of my graduation process or my Master's programme, but also the end of a 7-year adventure called student life. I like to believe I got out of it all I could, specifically by being part of a couple of associations, various boards and several committees. Through this way, I would like to thank the people who helped me to make my student life an unforgettable journey.

First, my supervisors, Mannes, Guido and Frank, thanks for guiding me in the right direction with your valuable feedback. Also thanks to my fellow interns and colleagues at Xsens, Elisabetta, Thijs, Eric, Xiaowen, for making the time in the office much more fun!

Second, my fellow *Huize de Kluis* occupants, thanks for the endless fun together, but foremost for making the house feel like our home!

Next, a long list of friends, Klaas, Rens, Jeroen, Jurjen, Joachim, Christiaan, Ramon, Ole, Joris, Nick, and everyone I am forgetting: Thanks for keeping me sharp and motivated to finish my studies, but even more for the uncountable mornings, afternoons, evenings and nights together!

Finally, mom, dad, Thirza, Rachelle and Susanne: Thanks for always having my back. I love you.

Summary

Even after three weeks and well over 80 hours of cycling, time differences in Grand Tours tend to be a matter of just a few minutes. A large portion of these minutes can be gained (or lost) in time trial stages. Aerodynamics play a large role, especially in these stages, and decreasing aerodynamic drag seems to be the key to winning valuable seconds. One of the main factors influencing aerodynamic drag in cycling is the projected frontal area. Decreasing this area, for example by altering a rider's position on the bike, leads to a smaller frontal area and thus to less aerodynamic drag.

In the current situation, this research is done in lab conditions, i.e. in a wind tunnel or a velodrome. This is not ideal: wind tunnel testing is expensive, riding positions are not tested during longer rides, while velodrome testing is tedious and does not deliver real-time results, all among other disadvantages. These problems can be overcome by a new testing method, that yields real-time results in a real-life setting.

For the development of such a method, 4 subjects were equipped with Xsens's MVN Awinda system. While riding on a bicycle trainer, the subjects' movements were recorded by sensors, and simultaneously they were filmed. Using unsupervised image segmentation, the video was converted to a frontal area value for each of the video's frames. Next, support vector regression and neural network models were built to predict the frontal area from motion data.

From several experiments, it follows that a support vector regression model, as developed in this thesis, is a viable proof of concept for predicting the frontal area of a cyclist from motion data. Experiments also show that such a model still works well when a reduced sensor set is used as input.

Concretely, this could lead to a change in the way of aerodynamic testing. Tests, for example, could take place over a longer distance. This gives insight into the role fatigue plays in aerodynamics, specifically when looking at frontal area. Another application for this research is virtual cycling. As of now, the position of a cyclist has no influence on the amount of resistance the smart trainer generates. With the implementation of this research, this would be made possible and increase the realisticness of virtual cycling.

All in all, it is expected that with further development, most importantly by increasing the size and quality of the dataset, this model can be a valuable addition to the current aerodynamic testing methods, able to bring real-time results in a real-life setting.

Contents

Chapter 1

Introduction

Even after three weeks and well over 80 hours of cycling, time differences in Grand Tours tend to be a matter of just a few minutes. A large portion of these minutes can be gained (or lost) in time trial stages. In this special discipline contestants ride against the clock, but maybe even more so against the wind: Research has shown that at high speeds (± 14 m/s), aerodynamic drag contributes up to 90% of the total resistance [1]. As such, decreasing aerodynamic drag seems the key to winning valuable seconds.

One of the main factors influencing aerodynamic drag in cycling is the projected frontal area. Decreasing this area, for example by altering a rider's position on the bike, leads to a smaller frontal area and thus to less aerodynamic drag. On the other hand, oftentimes the more aerodynamic a position is, the less comfortable and efficient it is, resulting in a loss of power output [2]. Therefore, among professional cyclists, there is a constant search for the position with the least projected frontal area, while also balancing the trade-off between aerodynamic drag and power output.

1.1 Problem statement

In the current situation, this research is done in lab conditions, i.e. in a wind tunnel or a velodrome. This is not ideal: wind tunnel testing is expensive, riding positions are not tested during longer rides, while velodrome testing is tedious and does not deliver real-time results, all among other disadvantages. These problems can be overcome by a new testing method, that yields real-time results in a real-life setting.

1.2 Research objective & research questions

From the introduction and the problem statement, it follows that a new method for the real-time determination of the frontal area of a cyclist in a real-life environment can be a solution to several problems present in nowadays aerodynamic testing. Therefore, this research focuses on the development of a machine learning model that takes motion data as input and can predict the frontal area of a cyclist. This way, the frontal area can be predicted in real-time and in a real-life environment. Rewording this goal into a research question gives the following:

> What is the performance of a motion data-based machine learning model to predict the frontal area of a cyclist?

The final objective of this thesis is a model that, given motion data as input, can predict the frontal area of a cyclist. To tune the hyperparameters of the model and to train and test the model, data is needed.

This data will be generated by equipping subjects with a motion capturing suit, while they adopt various positions on a bike. This data can be represented in two formats: quaternions or rotation matrices. Through the first subquestion, the best format for this cause will be determined:

> SQ1. What is the performance difference between quaternions and rotation matrices, when used as input for a model that predicts the frontal area of a cyclist?

The hypothesis is that a machine learning model that uses motion data is better suited for frontal area prediction than a non-machine learning model that uses, for example, a cyclist's height and weight as input. This hypothesis is tested through the answering of the second subquestion:

> SQ2. What is the performance difference between a machine learning model and a non-machine learning model, for predicting frontal area of a cyclist?

Both in terms of set-up time as of invasiveness for the subject, a reduced sensor set is preferred over a full sensor set. Therefore, experiments will be done to compare the performance of the original, full sensor set model, to the performance of a model with identical parameters, but with a reduced set of sensors as input data.

> SQ3. What is the performance difference between a full sensor set model and a reduced sensor set model?

1.3 Structure

The rest of this thesis is structured as follows: In the next chapter, background information is given and previous work related to this thesis's topics is discussed. In Chapter 3, the methods used in this thesis are discussed, after which the results are presented in Chapter 4. The results and its implications, as well as limitations of this study and recommendations, are discussed in Chapter 5. To conclude this thesis, conclusions are drawn in Chapter 6.

Chapter 2

Background & Related Work

In this chapter, background information, ranging from forces on a cyclist to machine learning principles, necessary to understand the rest of this thesis is given. This chapter also includes previous works, done by other researchers, on topics related to this thesis.

2.1 Xsens & motion capture

This thesis was executed in collaboration with Xsens, whose products are used in the motion capturing part of this research. In this section, Xsens as a company is briefly discussed. Next to that, general information regarding motion capture and more in-depth information regarding the Xsens systems specifically is provided.

2.1.1 Xsens

Xsens Technologies B.V. was founded as a University of Twente spin-off in 2000. Since then, Xsens has grown to an industry innovator with products ranging from inertial sensor modules, used in e.g. autonomous vehicles and robotics, to motion capturing solutions, used in e.g. film and sports.

MVN systems

Xsens has two motion capturing solutions: MVN Link and MVN Awinda. MVN Link consists of a full-body lycra suit that houses 17 inertial sensors. After a calibration pose, the accompanying software MVN Analyze processes the sensor data (sent through Wi-Fi) and provides orientations of 23 body segments, with a joint angle accuracy of 1 to 6° [3].

Xsens's other motion capturing solution, MVN Awinda, uses the same 17 sensors that the MVN Link system uses. However, instead of fitted on a lycra suit, the sensors are strapped to the body with elastic bands, making it a wireless system. Alike the MVN Link system, MVN Awinda has to calibrated before use, after which the data is processed by MVN Analyze.

MVN data

IMU-based motion capture systems, such as the MVN systems, use a biomechanical model to map the sensor data to a digital skeleton. This biomechanical model is made up of separate parts called segments. These segments largely resemble the separate parts of the human body, for example by mapping the shin and calf bone to a lower leg segment. Just like in the human body, the segments in the biomechanical model are connected by joints. As not all bones are mapped one-to-one with the segments, neither are all real joints mapped one-to-one to their digital counterparts. In Figure 2.2, the digital skeleton with its separate segments and joints can be found.

In default settings, the recorded motion data is saved in the proprietary .mvn file format. Next to .mvn files, MVN Analyze offers the possibility to

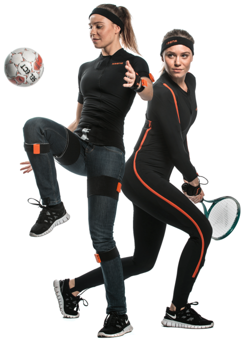


Figure 2.1: Two persons wearing the MVN Link system (tennis) and the MVN Awinda system (football). [4]

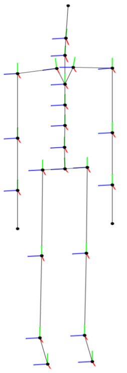


Figure 2.2: The digital skeleton created in the MVN software, consisting of segments (lines) connected by joints (dots). [5]

export data in other file formats, including .mvnx, which has an XML structure and .xlsx files. Although different in structure, these files can contain the same information. This information is organized in frames, together making up the full recording timeline. Frames are further split into elements, such as segment orientation, segment velocity and joint angles. All elements a frame can contain are described in Table 2.1 [5].

Not all elements provided by MVN Analyze are equally suitable to be included in a dataset created to train a frontal area predictor. Therefore, the usability of the various motion data elements was examined. Both segment orientations and joint angles could serve as suitable input for the machine learning algorithm. In recent works using motion data as input for a machine learning model, segment orientations are favoured as input, either described using quaternions [6]–[8] or rotation matrices [9].

Body surface area Besides the MVN elements to include, also a scaling factor to account for the size differences that exist between people, the body surface area (BSA), was investigated. Already a good 100 years ago, in 1916, Du Bois and Du Bois published a formula to determine a person's BSA (in cm^2) from their height (in cm) and weight (in kg): $BSA = 71.84 * h^{0.725} * w^{0.425}$ [10]. For many decades, this has been the golden formula in BSA estimation, despite the use of only 9 subjects in their calculations. Because of this sheer number of subjects, the Du Bois & Du Bois formula was re-evaluated by Shuter & Aslani, with a larger dataset of 42, after which they improved a refined formula: $BSA = 94.9 * h^{0.655} * w^{0.441}$ [11]. More recent advances in techniques have also advanced the search for the ultimate BSA equation. Villa et al. have performed CT scanning on 54 bodies after which their findings were compared to traditional equation estimates. They found that the CT scanning results had a very high correlation with traditional equations (e.g. by Du Bois & Du Bois, Shuter & Aslani) [12]. Similarly, Kuehnapfel et al. have performed a study using 3D laser scanning. As this modern method allows for fast and easy measurements, a large subject group of 1435 could be analyzed. They found that excellent correlations exist between the results of their 3D laser scanning and the estimates by traditional equations. Furthermore, they proposed a refined version of the BSA formula: $BSA = 151 * h^{0.5751} * w^{0.4259}$ [13].

2.1.2 Motion capture

While some movements can be analyzed easily, by just looking with the naked eye, there are many more complex movements that require deeper and more thorough analysis. This can be done through motion capture, a process in which movements are digitally recorded. While motion capture was first applied in biomechanical applications, such as gait analysis, the applications of motion capture have expanded and is nowadays also used by coaches and athletes to improve training programs, by video game and movie studios for the animation of characters and as an improvement to virtual/augmented reality, by capturing the movements in real-life and accurately copying these to the digital

Table 2.1: A description of all elements a .mvn file can contain. [5]

Element	Description
Orientation	1x4 quaternion vector describing the orientation of the segment with respect to the global frame.
Position	1x3 position vector of the origin of the segment in the global frame.
Velocity	1x3 velocity vector of the origin of the segment in the global frame.
Acceleration	1x3 acceleration vector of the origin of the segment in the global frame.
Angular velocity	1x3 angular velocity vector of the origin of the segment in the global frame.
Angular acceleration	1x3 angular acceleration vector of the origin of the segment in the global frame.
Sensor magnetic field	1x3 sensor magnetic field vector of the sensor.
Sensor orientation	1x4 sensor orientation quaternion vector of the sensor in the global frame.
Joint angle (ZXY)	1x3 Euler representation of the joint angle vector, calculated using the Euler sequence ZXY, using the ISB based coordinate system.
Joint angle (XZY)	1x3 Euler representation of the joint angle vector, calculated using the Euler sequence XZY, using the ISB based coordinate system.
Ergonomic joint angle (ZXY)	1x3 Euler representation of the ergonomic joint angle vector, calculated using the Euler sequence ZXY, using the ISB based coordinate system.
Ergonomic joint angle (XZY)	1x3 Euler representation of the ergonomic joint angle vector, calculated using the Euler sequence XZY, using the ISB based coordinate system.
Center of Mass	1x3 position vector of the body Center of Mass in the global frame.

environments. Motion capture is mainly done by two systems, either by using optical markers or by using inertial measurement units (IMUs).

Motion capture by optical systems

Optical motion capture systems rely on retro-reflective markers places onto rigid body parts. Cameras, placed around the recording area, emit infrared light which is then detected as it is reflected by the markers. Since the camera positions are known, the position of the markers can be inferred. Furthermore, because the markers are placed on known positions on body segments, the segment orientation can be inferred as well.

With a proper set-up of the optical system, its accuracy is very high, into the range of millimetres, for a long time making it the gold standard in motion capture [14]. However, because of this sensitivity, when the set-up is inaccurate and markers are slightly misplaced, this leads to significant errors in segment orientation [15]. This need for accuracy and the high number of cameras and markers that need to be set up, also leads to long set-up times, as well as a need for recording conditions nearing lab conditions [7]. Apart from set-up related difficulties, optical motion capture is also sensitive to occlusions and requires post-processing [16].

Motion capture by IMUs

The other main system for motion capture is based on inertial measurement units (IMUs). This system consists of small units, containing an accelerometer, a gyroscope and a magnetometer, that are placed on pre-determined body segments.

The accelerometer measures the acceleration of the unit, and thus the segment on which the unit is placed. The gyroscope measures the rate of angular rotation of the segment. Lastly, the magnetometer uses the Earth magnetic field to calculate the heading of the segment.

When information of these three sources is combined with a biomechanical model, it becomes possible to acquire segment orientations and segment positions. However, before this is possible, the system should be calibrated as to learn the sensor placements on the segments.

Motion capture by IMUs solves the optical system's problem of tedious set-up. IMU-based systems do not require lab-like conditions, although accurate sensor placement is still a prerequisite for accurate measurements. Next to that, IMU-based systems do not suffer from occlusion problems. The main disadvantage IMU-based systems have, is inaccuracy caused by sensor drift. As sensor positions and orientations are largely based on calculations, small measurement errors and sensor noise can easily add up to cause mistakes. Furthermore, IMU-based systems used to be influenced by magnetic disturbance, causing accuracy loss in heading. However, recent improvements in the algorithms have largely solved this issue [4].

2.2 Aerodynamics in cycling

As a certain legend once said: "Playing football is very simple, but playing simple football is the hardest thing there is." The same goes for cycling: whereas the act of cycling is easy, describing (and improving) the motion of a cyclist and its bike has been on the mind of researchers since the rise of the modern bicycle in the early 1900s. This motion consists of a complex mix of the cyclist's power input, aerodynamic drag, rolling resistance and mechanical resistance, besides optional changes in potential and kinetic energy [17]. Whereas some of the components making up the forces are difficult to alter, the frontal area of a cyclist can easily be decreased to improve a cyclist's aerodynamics, resulting in better cycling performance, and therefore is a hot topic within the cycling science.

In this section, the forces that work on a cyclist will be briefly explained. Following this, more in-depth explanations of measurement methods both for the aerodynamic drag, as well as for the frontal area are given.

2.2.1 Forces on a cyclist

Research concerning the power output of a road cyclist has been around for many decades and has steadily improved in accuracy, coming from simple equations [18] to a validated model including even the frictional losses in the wheel bearings [17]. Although such an extensive model could be useful, it is not needed for the goals of this thesis, as the losses in the wheel bearings or other frictional losses have no impact on the aerodynamic drag. Therefore, a simpler model, based on the works of Di Prampero et al. [19] is used. This simplified model will assume riding straight at a constant speed at a level surface. Furthermore, smaller resistive forces, such as the impact of the rotating wheels and frictional losses in the wheel bearings and the drivetrain have been simplified to an efficiency factor.

As such, the total power output P_{Tot} needed to overcome the resistive forces is equal to the sum of the power to overcome the aerodynamic drag P_D and the power to overcome the rolling resistance, divided by the efficiency factor η:

$$P_{Tot} = \frac{P_D + P_{RR}}{\eta} \tag{2.1}$$

Aerodynamic drag

The resistive force when an object, in this case a cyclist with its bike, moves through air is dependent on the air density ρ, the drag coefficient C_D, the frontal area of the cyclist A and its bike and the air velocity v, relative to the cyclist [19]:

$$F_D = 0.5 * \rho * C_D * A * v^2 \tag{2.2}$$

Crouch et al. explain the drag coefficient C_D as follows: “The drag coefficient describes the aerodynamic efficiency of a body independent of the size of the shape. (...) It depends on a number of factors including body shape, orientation, surface roughness, freestream flow conditions, and the Reynolds number.” [20]

As $P = F * v$, and the cyclist is assumed to ride in a straight line in windless conditions, the power needed to overcome this drag force is given by [20]:

$$P_D = 0.5 * \rho * C_D * A * v^3 \tag{2.3}$$

and is cubic in v.

Rolling resistance

Besides the drag force, the force due to rolling resistance is the other main resistive force during cycling. In contrast to the drag, this force is independent of the velocity of the rider. Instead, it is related to the weight of the cyclist and its bike, m, the gravitational acceleration, g, as well as to tyre and surface characteristics, e.g. the tyre pressure, tyre profile and the smoothness of the surface, combined into the rolling resistance coefficient C_{RR} [17]:

$$F_{RR} = C_{RR} * m * g \tag{2.4}$$

It follows that the power needed to overcome this force due to rolling resistance is given by:

$$P_{RR} = C_{RR} * m * g * v \tag{2.5}$$

and is linear in v.

Comparing the resistive forces

Of the two main resistive forces, the aerodynamic drag is, at racing speeds (± 14 m/s), by far the largest of the two and represents up to 90% of the overall resistance [1]. The increase of resistive forces due to speed can be seen in Figure 2.3. For the calculations, typical coefficient values as mentioned by Martin et al., of $C_{RR} = 0.0032$, $C_dA = 0.264$ m^2 and an efficiency $\eta = 98\%$ are used [17]. Besides, a total (cyclist + bike) mass of 80 kg, gravitational acceleration of 9.81 m/s^2 and an air density of 1.225 kg/m^3 are used.

Because of the high impact aerodynamic drag has on net power output, reducing drag is a key factor in improving the performance of professional cyclists and measuring it has been the subject of many studies. In the next section, these measuring methods will be discussed.

2.2.2 Measuring aerodynamic drag

To be able to lower the aerodynamic drag, one has to be able to measure the drag. Throughout decades of research, several methods to fulfil this purpose

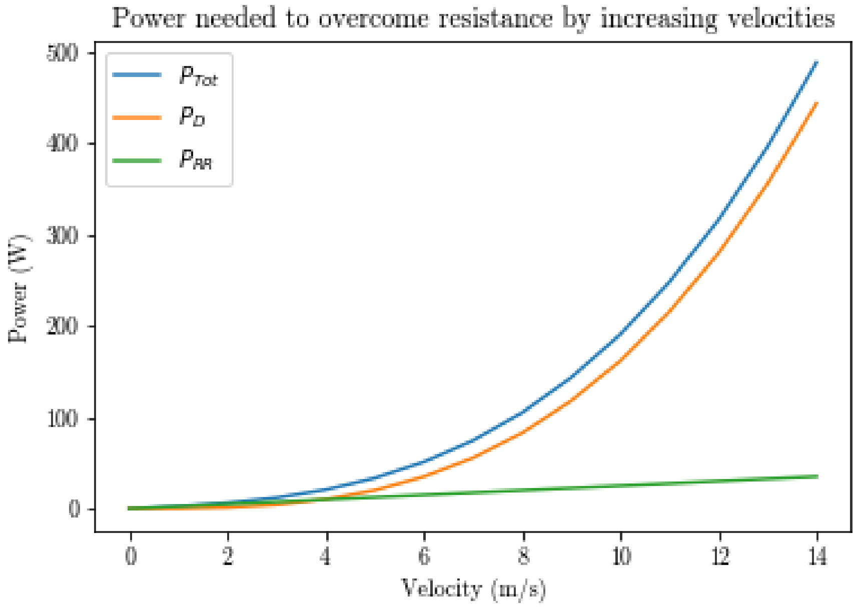


Figure 2.3: The power needed to overcome the total resistive forces (blue), aerodynamic drag (orange) and the rolling resistance (green) at different velocities.

have been developed. In this section, these methods, ranging from the simple coasting-down method to the advanced, state-of-the-art Ring of Fire are discussed.

Many methods for measuring aerodynamic drag make use of the so-called 'drag area' or 'effective frontal area'. This combined factor C_DA is the product of the drag coefficient C_D and the frontal area A. By using this combined factor, the uncertainty associated with measuring the frontal area is eliminated, while still providing a way to compare results between tests. After all, the drag area C_DA is the only remaining factor constituting the total drag force after the air density ρ and the velocity v, two easy to measure factors, are determined. [20]

Towing method

Starting with the oldest method, the towing or dynamometer method has been around for some 200 years when it was used in plough design [18]. Later, when the modern bicycles made their entrance, towing experiments were also carried out on cyclists, to measure the total resistive force acting on a bicycle during riding. These experiments are carried out by towing a bike with a car by means of a long cable, measuring the total force needed with a dynamometer. [19] When these experiments are then repeated at different speeds, linear regression analysis can be performed to determine the effective frontal area.

Although this method is relatively simple and can be performed in realistic conditions, the space, equipment and repetitions needed for accurate conclusions make this method less suitable for routine measurements [21]. Next to that, the air turbulence from the towing vehicle and changes in weather conditions can affect results, making it less accurate than e.g. indoor tests [1].

Coasting-down method

In this method, cyclists are asked to accelerate up until a certain speed and then stop pedalling. On the measuring site, there are several measuring spots, at which the time it took to reach that spot while coasting is measured. From this, the velocity at that moment is calculated. This method is based on Newton's second law of motion ($F = m * a$). In these experiments, the total weight is known and the acceleration/deceleration is measured. From this data, the resistive force (both the drag and the rolling resistance) is calculated. [21]

As with the towing method, the main advantage of this method is the opportunity to test in relatively realistic conditions. However, again similar to the towing method, this is also a disadvantage as climatic conditions can induce errors. [1] Also, several tries are needed to determine the effective frontal area, making it less suitable for routine tests [21].

Linear regression analysis

For this method, a power meter is used to measure the power needed to cycle at a certain speed. The cyclist is asked to maintain this speed for several laps, to ensure more accurate measurements. This procedure is then repeated on various speeds. From these measurements, the effective frontal area is calculated as follows.

From Equations 2.1, 2.2 and 2.4 it follows that the total resistive force (R_T) is given by:

$$R_{Tot} = 0.5 * \rho * C_D * A * v^2 + C_{RR} * m * g \tag{2.6}$$

Substituting $0.5 * \rho * C_D * A$ with a and $C_{RR} * m * g$ with b gives a linear relationship between the total resistive forces and the square of the velocity:

$$R_{Tot} = a * v^2 + b \tag{2.7}$$

It then follows that the effective frontal area C_DA is given by:

$$C_DA = a/(0.5 * \rho) \tag{2.8}$$

Using linear regression analysis and Equation 2.8, it is possible to determine C_DA from slope a of Equation 2.7.

When the input parameters are measured accurately, this method has high reliability, and the results are comparable to measurements in a wind tunnel [22], [23]. These reliable input parameters can be achieved by performing measurements in a velodrome, where climatological conditions do not affect the measurements. An additional benefit of velodrome measurements is the real-life condition in which they are performed [21]. A disadvantage of this method is the fact that results are only available in hindsight. This makes comparing positions a lengthy process, compared to when receiving real-time feedback on the consequences of a position change.

Wind tunnel

When measurements are done in a wind tunnel, a cyclist is positioned in front of a powerful fan. The bike is fixed to a force balance, which measures the forces working on the cyclist and its bike. From these forces, the effective drag area can be calculated.

When it comes to accuracy and reliability, wind tunnel measurements are the gold standard for measuring drag. The measurements are very accurate and are often taken as reference values for comparing other measuring methods. With the wind tunnel method, it is also very well possible to measure the aerodynamic differences of various riding positions and equipment. Furthermore, when also measuring the cyclist's frontal area, it is possible to determine the drag coefficient precisely. However, there are also several downsides to using the wind tunnel method. Firstly, the wind tunnel can cost up to 10,000 euros per day [21]. Secondly, in wind tunnel experiments is not standard practice to measure the drag while the cyclist is pedalling. Because of this, the effect of rotating legs and wheels is often ignored [23], [24], rolling resistance is not taken into account [25] and the rider's wind tunnel test position might be hard to sustain during actual locomotion [1]. Furthermore, because the bike is fixed to the floor, slight lateral movements are not present as they would in actual cycling [1].

Computational fluid dynamics

Computational fluid dynamics (CFD) is a relatively new technique, with publications using CFD on cycling aerodynamics starting around 2010 [25], [26]. For this method, a digital model of a cyclist is created, using a high-resolution 3D laser scanner. This model is then used in simulations, where drag force calculations are done for tiny parts of the model, which are then combined into a result for the full model. [25]

The main advantage of CFD is the increased insight into the flow field around the body of the cyclist. This helps in understanding the impact different adjustments to riding position have on the airflow and thus on the drag coefficient. [21] This insight also applies to body parts, making it possible to investigate the influence of small adjustments to riding position or equipment [27]. CFD is not (yet) as accurate as wind tunnel tests, although differences between them have dropped from 7-11% in 2010 [25] to 2-7% in 2019 [28]. This decrease can probably be explained by higher resolution grids, made possible by more computational power. As with the wind tunnel method, CFD does not support cyclist-bicycle locomotion and therefore ignores the effect of rotating legs and wheels [27].

Ring of Fire

This new technique to measure aerodynamic drag is based on large-scale stereoscopic particle image velocimetry. A cyclist rides through a tunnel, filled with helium-filled soap bubbles. With the means of lasers and high-speed cameras, the flow of these bubbles, and thus the air, is tracked and studied. From this, aerodynamic drag is calculated.

The main advantage of the Ring of Fire is its ability to be used with a moving cyclist instead of the (mostly) static measurements in wind tunnels. Next to that, instead of the wind blowing around the rider, as is the case in wind tunnel experiments, the cyclist rides into the wind. These two properties of the Ring of Fire installation help the experiments approach actual riding conditions. Furthermore, as the installation is mobile, it can be used both inside and outside, as well as in previously difficult locations, such as the corners of velodrome tracks. Although the Ring of Fire looks promising, it is not (yet) as accurate as the wind tunnel. [29]–[31]

2.2.3 Frontal area

From Section 2.2.1 it is clear that reducing the aerodynamic drag is vital in improving cycling performance. From Equation 2.2, it follows that, with a constant drag coefficient, the aerodynamic drag is proportional to the frontal area. Hence, a decrease in the frontal area will result in a decrease of aerodynamic drag. [25] However, the drag coefficient is also dependent on the rider's position and will almost always change when the position is altered. Nonetheless, the frontal area still has its use in providing a quick guess of the aerodynamic drag when elaborate tests (e.g. wind tunnel) are not available. [32] Next to that, accurate measurements of the frontal area are also of importance in determining the drag coefficient, for example, when studying the effect of position changes or equipment on the drag coefficient or when implementing mathematical prediction models [33].

2.2.4 Measuring frontal area

For measuring the frontal area of a cyclist, many different methods have been developed over the years. In this section, these methods will be discussed.

Anthropometric models

In early studies, researchers assumed a cyclist's frontal area is proportional to its total body surface area (BSA) [19]. This body surface area, in turn, was then estimated from the cyclist's body height and weight by applying an equation by the Du Bois brothers [10].

However, later research showed that frontal area is not proportional to BSA, as the BSA/mass ratio tends to be smaller in larger cyclists [34]. Despite this finding, more attempts have been made since to correlate frontal area and anthropometric parameters, mostly also taking into account other variables, such as trunk angle [35], bike position [36] or helmet length [37].

Nonetheless, because of the limited application and accuracy of anthropometric models, the interest is shifting towards direct methods of measuring frontal area [21].

Analogue methods

One of the main analogue methods for calculating frontal area is the weighing method. For this method, photographs of the cyclist and its bike are made and printed (or, in earlier times, developed). In the same photograph, also a reference frame of known dimensions should be visible. Both the cyclist with the bike and the reference frame are then cut out very precisely and weighed on an accurate balance. As both the weight and the area of the reference frame are known, the area of the cyclist can be calculated from the weight of the cyclist's cut-out. Although this method is very simple and has already been used for multiple decades, the method of weighing photographs is very accurate and has been used as a reference method when new methods were validated [32], [33], [35]. Yet, the need for cutting precisely, makes it a time-consuming method, with times up to 25 minutes per picture being reported [32]. Furthermore, this method calls for a reference plane, which makes it less suited for actual applications.

Next to weighing photographs, also a planimeter can be used to measure the frontal area. Similar to weighing photographs, this is an accurate, but time-consuming method, albeit taking about half the time of the weighing method. Another advantage of planimetry over weighing photographs is that a reference plane is not necessary. [32]

Digital methods

Digital planimetry is very similar to analogue planimetry. The difference is the outline being traced on a digitising tablet, instead of by a planimeter. Through this, a digital area is determined of which the area can be calculated [32]. Digital planimetry is similar to the so-called Computer-Aided Design method by Debraux et al. [33] Their main advantage is, as with analogue planimetry, that there is no need for a reference plane, making them useful for actual situations.

The digitizing method, described by, among others, Heil and Debraux et al. is comparable to the digital planimetry and the computer-aided design method [33], [35]. With the digitizing method, the desired frontal area is traced and blackened. From this, using the reference plane, the frontal area is calculated.

All digital methods have similar accuracy, also when compared to analogue methods, but are faster and more convenient than their analogue counterparts [21].

Computer vision-based approach

This is a newly developed method by Drory et al., performing near real-time automatic frontal area estimation [38]. First, the algorithm has to segment the frontal area. For this, it uses an active contours (AC) approach. To make sure the AC algorithm is more accurate, an object detection algorithm is used beforehand. This object detection algorithm is based on the active shapes models technique in which a sort of mean shape of all objects (the cyclist with the bikes) is created. Using this, the AC algorithm has an idea of what to search for and delivers better results.

One advantage of this approach is its speed. With an unoptimized implementation and suboptimal hardware, the algorithm takes about 1 second per photograph. Next to that, this method is highly automated, which saves much manual work. However, because of the lack of human supervision, the results of this approach are less accurate than those of the manual methods.

2.3 Machine learning

Large parts of this thesis are based on machine learning, specifically, by using it to segment images and to solve regression problems. Although machine learning is quite the buzzword and everyone seems to use it, the underlying principles are not always well understood by everyone. Therefore, in this section, background information and related work on machine learning is given. This will start off with a small introduction to machine learning, after which the principles and components of neural networks are covered. Lastly, two machine learning applications, namely image (co-)segmentation and regression analysis are discussed.

2.3.1 Machine learning principles

Machine learning is the name of a collection of techniques used to infer patterns and generate knowledge from large datasets. With the increasing availability of computing power and data, machine learning is used more and more in an increasing number of domains.

Machine learning techniques can be broadly categorized into supervised and unsupervised learning approaches. The difference between these two approaches is in the existence of data labels: supervised learning uses labelled data, whereas unsupervised approaches work without labelled data.

In this thesis, both approaches are used. First, unsupervised learning is used in the image segmentation algorithm (see Section 3.2.2). Later, supervised regression is used to correlate the frontal area and the riding position on the bike (see Section 3.5).

As machine learning can be applied to a wide variety of problems, there exists a wide range of machine learning techniques. In this thesis, only artificial neural networks are used, which is why only this technique is elaborated on in the next section.

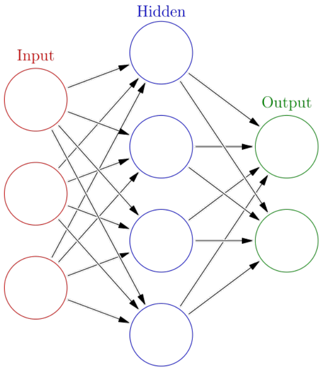


Figure 2.4: A simple neural network, with 3 input neurons, 1 hidden layer with 4 neurons and 2 output neurons. [39]

Neural networks

Artificial neural networks are a machine learning technique inspired by the workings of the human brain. Neural networks consist of multiple computing units, neurons, analogous to the cells in the human brain. The neurons are connected through connections called edges, see Figure 2.4. These edges have weights associated with them, while all neurons have a bias term, which can also be modelled as a weight. The 'strength‘ of these weights is continuously adjusted by optimizing a cost function, in order to achieve better results. This optimization and adjusting is called learning. Over the edges, signals (real numbers) are sent between neurons. The strength of the signal depends on the value of the input signal and the weight of the edge, the ultimate signal strength being the product of the two. In neural networks, the neurons are organized in layers. Typically, the signals sent between neurons travel from the input layer to the output layer, passing one or more so-called hidden layers in between. An illustration of a typical simple neural network can be found in Figure 2.4.

When neural networks are used for regression analysis, typically, the input layer of the network consists of multiple neurons, with each neuron representing an input variable. The output variable is represented by a neuron in the output layer of the network. In between, in the hidden layer(s), the network is left free. It usually does not have constraints in its regression type (logistic, polynomial, etc.), making it a very flexible option. Eventually, through learning, the network comes up with a fitting function.

Support vector machines

Support vector machines (SVMs) are a well-known approach to classification problems. In the most simple case, this is a linear binary problem. The SVM algorithm then builds a classifier by finding the best line for separating the two classes (see Figure 2.5). The goodness-of-fit is determined by the margin the

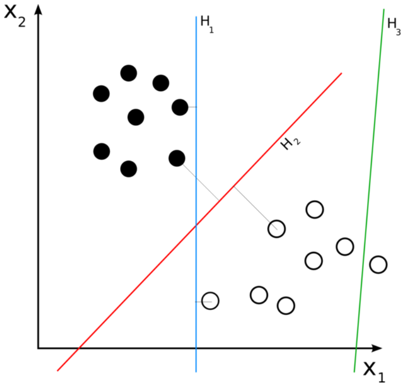


Figure 2.5: Graphic showing how a support vector machine would choose a separating hyperplane for two classes of points in 2D. H_1 does not separate the classes. H_2 does, but only with a small margin. H_3 separates them with the maximum margin. [41]

line has from the nearest points, with more distance being better. [40]

Fortunately, SVMs can also be used in cases with more than two classes to separate. This is done by adding more dimensions, transforming the decision boundary from a line into a hyperspace. Additionally, nonlinear problems can be solved by SVMs. This is achieved using a different kernel than a linear kernel, such as the radial-basis function (RBF) kernel or the polynomial kernel. By using a nonlinear kernel, the nonlinear data is mapped onto a space that makes the data linear, after which a decision boundary is determined. [40], [42]

Lesser known than the use of SVMs for classification, is the use of SVMs for regression problems, in which case the term support vector regression (SVR) is used. In a regression use case, the goal is, of course, not to separate the points. Instead, the goal is to find the hyperplane that is best able to describe the training samples. [42]

In SVR with an RBF kernel, a common choice, the hyperplane can be tuned by adjusting three parameters: C, γ and ϵ. C describes the cost of an error. With a high value of C, an error is penalized more than with a low value of C, resulting in a plane that more closely fits the training examples, but at the risk of overfitting. On the other hand, a low value of C results in a smooth decision boundary, but risks generalizing too much. The second parameter, γ, describes the weight a single training sample has, with a lower value giving more individual influence to the samples. The third parameter, ϵ, describes an acceptable error range from the hyperplane within which predictions are considered correct. With an ϵ value approaching 0, more and more support vectors will be needed and the risk of overfitting grows. As there is an unlimited amount of possible parameter tuples, often the best combination is found through experimentation. [43]

Between SVRs and neural network regression, there exist some differences.

Neural networks determine their best solution based on all training samples. Contrarily, SVRs base their solution only on boundary cases. These boundary cases are also what gives SVM & SVR their name: they are the support vectors. Another difference between SVRs and neural networks is their learning. While neural networks learn continuously through backpropagation and updating of the network's weights, SVRs determine one best solution and have no need for continuous learning.

2.3.2 Image segmentation

Image segmentation is the area in the field of image analysis that deals with extracting objects of interest from the rest of an image, for example when separating the fore- and background in an image. Being one of the fundamental areas of image analysis, image segmentation techniques have been developed since multiple decades. It is good to point out that contour detection and segmentation are related problems, but not identical. As Arbeláez and Fowlkes put it: "In general, contour detectors offer no guarantee that they will produce closed contours and hence do not necessarily provide a partition of the image into regions. But one can always recover closed contours from regions in the form of their boundaries." [44]

In the coming sections, various approaches for image segmentation will be discussed. For structure during the discussion, sections are related to the taxonomy presented in Figure 2.6. This taxonomy is created by combining three existing taxonomies [45]–[47]. The main split in the taxonomy is that of hard and soft-computing. Hard-computing methods can be seen as the more traditional and rigid methods. Soft-computing methods, in contrast, are more flexible and better able to deal with imprecise data. A separate group within the image segmentation field is that of image co-segmentation, which is discussed in the last section.

Image segmentation using hard-computing methods

As the name suggests, region-based methods are based on finding regions in pictures, by putting together pixels with similar values, e.g. creating regions containing pixels with the same colour.

One of the oldest methods used for image segmentation is thresholding. The most basic variant of this method works by dividing the image in a fore- and background by assigning pixels with a value below/equal to or above a threshold value to different regions. More advanced variants do not only look at the complete, global, histogram, but also at local properties. A well-known clustering method is k-means clustering. With this method, the image is segmented into a predefined k number of regions, by using an algorithm to minimise the sum of squared distances to assign pixels to regions.

Another category is that of graph partitioning methods. For these methods, the image is represented as a graph with the pixels being the nodes and with weighted edges between neighbouring nodes. The weights represent the

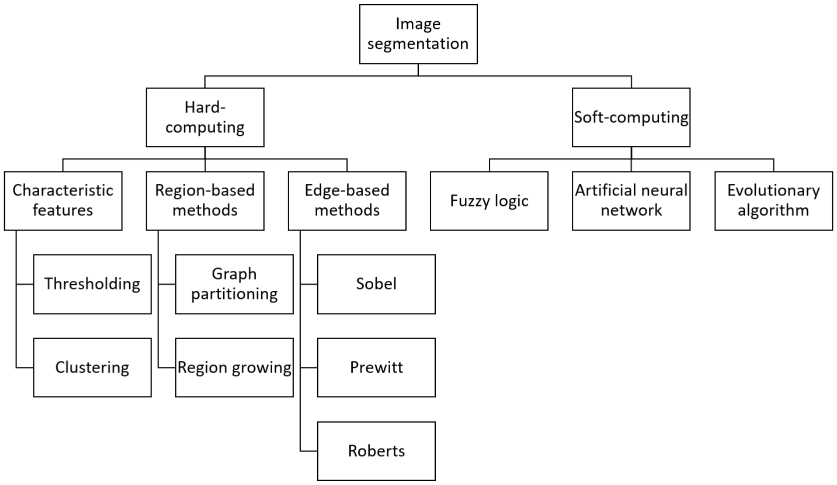


Figure 2.6: Taxonomy of image segmentation strategies, combined from [45]–[47]

similarity between the pixels. The eventual segmentation is dependent on the particular algorithm. For example, MinCut 'cuts' through the edges with the lowest values [48].

Another popular category of methods is region growing. In these methods, a starting pixel is compared to neighbouring pixels. If the compared pixel is similar, based on intensity, for example, it is added to the cluster. When it is not similar, a new cluster is created. This is repeated until all pixels belong to a cluster.

Edge-based methods are based on finding the edges between regions. This is done by, for example, looking for sharp changes in colour. Regions are then identified as the areas encapsulated by the found boundaries.

For detecting edges, several methods exist. A well-known method is applying an edge detection filter (e.g. Sobel, Prewitt or Roberts Cross) to the image. This is done by convolving the 2 kernels (1 for the horizontal direction, 1 for the vertical direction) over the image. By calculating the magnitude and the direction, this results in an image only containing the edges of the original image.

Image segmentation using soft-computing methods

Since recent years, soft-computing methods are preferred by researchers for image segmentation [49]. As neural networks, convolutional neural networks (CNNs) in particular, have shown promising results lately, only this soft-computing approach is further discussed.

Ronneberger et al. extended a fully convolutional network for supervised biomedical image segmentation, which achieved an average Intersection over Union (IoU) of 77.5% [50]. This model consists of two parts, a contracting

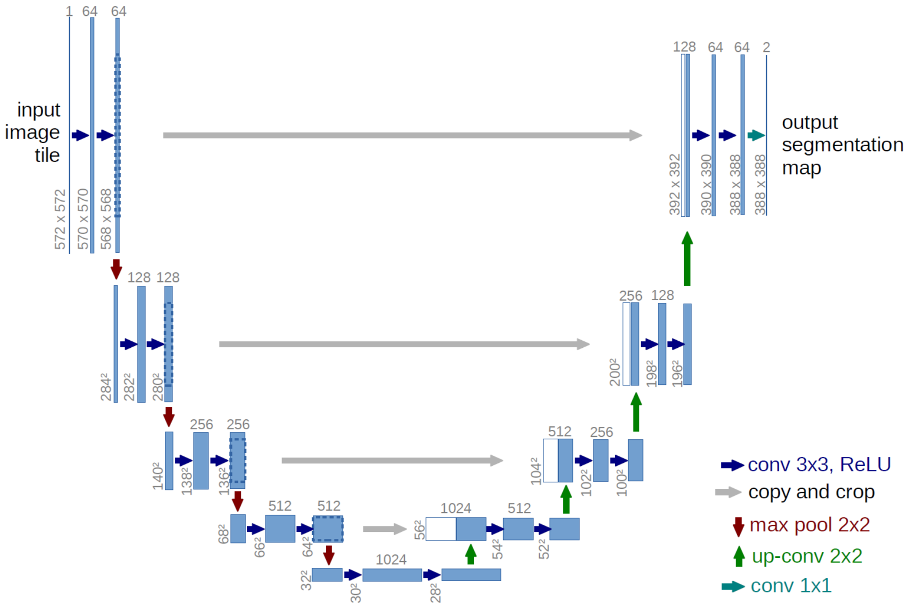


Figure 2.7: An example of the U-net model as designed by Ronneberger et al. It consists of two parts, the contracting path on the left side of the image and the expansive path on the right side. Every block is a layer and the different operations are portrayed by the different arrows. [50]

path and an expansive path, where the former extracts the features from the images, whereas the latter combines the feature and the spatial information. An example architecture for this U-shaped network can be found in Figure 2.7.

The U-net model by Ronneberg et al. gave promising results, but was a supervised approach. 2 years later, the U-net model was extended to an unsupervised version which ties two U-net models together to create a W-net, see Figure 2.8 [51]. The separate U-net models work very similarly to the original U-net model, with a contracting and an expanding path, the former used for feature extraction, the latter for feature localisation. Between the two nets, the segments resulting from the encoder net are smoothed by using conditional random fields (for sharper boundaries) and hierarchical segmentation (to merge small segments). In turn, the decoding net starts here and tries to reconstruct the original image. One of the loss functions used by W-Net is the error between the original input image and the reconstructed image, outputted by the second U-Net. Another loss function, optimized at the same time, is a variant of Normalized-Cut, made suitable for backpropagation. By using these two loss functions, the algorithm can simultaneously minimize the disassociation between segments (to create an accurate reconstruction of the original image), while maximizing the association within segments (to prevent over-segmentation).

Another approach for unsupervised image segmentation was developed by

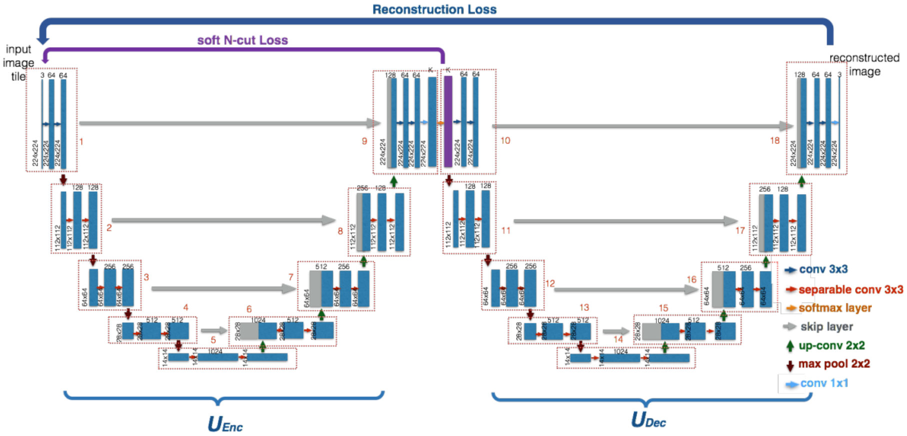


Figure 2.8: An example of the W-net model as designed by Xia et al. It consists of two U-net models, with an encoding net (left) and a decoding net (right). The different operations are given by the different arrows. [51]

Kanezaki and is inspired by deep clustering [52]. This algorithm is based on the continuous balancing of 3 criteria: (1) Pixels of similar features are desired to be assigned the same label. (2) Spatially contiguous pixels are desired to be assigned the same label. (3) The number of unique cluster labels is desired to be large. In this process, first, superpixels (a group of similar pixels) are created. These are then optimized according to the three criteria, by alternating label prediction and learning of network parameters. As such, this algorithm is capable of learning on the image itself and does not need a big training dataset.

An algorithm inspired by Kanezaki, in the sense that it also learns on the input image itself, was developed by Saha [53]. In contrast to the algorithm by Kanezaki, this approach does not use superpixels as a starting point. Instead, it leverages transfer learning and uses a part of the VGG16 network as a starting point, before a number of trainable layers.

Image co-segmentation

A special type of image segmentation is image co-segmentation. In these sorts of problems, the data set contains images with a shared foreground object. This common object is first identified by the algorithm, before segmenting this object. Since the (formal) introduction of co-segmentation in 2006 [54], various methods have been proposed. A classification of approaches, as recently presented by Xu et al, can be found in Figure 2.9 [55]. In their article, many co-segmentation methods and their applications are discussed and compared. As the interest of this thesis is in deep learning approaches, only this category will be discussed in the rest of this section.

Li et al. were the first to develop a fully convolutional network for image co-

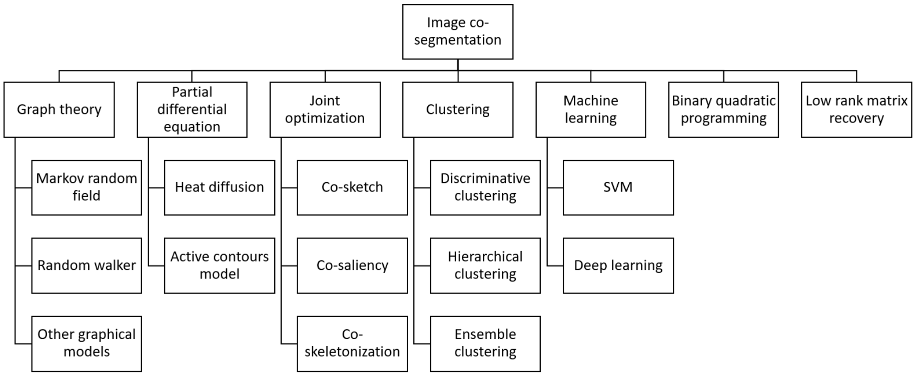


Figure 2.9: Taxonomy of image co-segmentation strategies [55]

segmentation. Their Deep Object Co-Segmentation (DOCS) algorithm is based on a Siamese encoder-decoder architecture, much like a double U-net. Instead of depending on hand-made features, like previous approaches, the encoder pair extracts high-level features of the foreground objects from both images. Then, a mutual correlation layer detects the common objects between the images, under the assumption that common objects should contain similar features. Finally, from the information given by the encoder and the correlation layer, the decoder generates the segmentation masks for each image. An example of the Siamese encoder-decoder architecture designed by Li et al. can be found in Figure 2.10. [56]

As an improvement to the model by Li et al., Banerjee et al. extended this model with a Siamese metric learning network. This Siamese metric learning network takes the two feature outputs from the decoder paths and transforms it into two feature vectors. These vectors represent the place of the objects in a latent space, in which similar objects are represented near each other and unrelated objects far from each other. With this extension, they have been able to obtain better results than Li et al. [57]

2.3.3 Regression analysis

In supervised machine learning, the two major problem categories are classification, with an output in the form as a class label and regression, with an output in the form of a continuous value. In the case of predicting the frontal area of

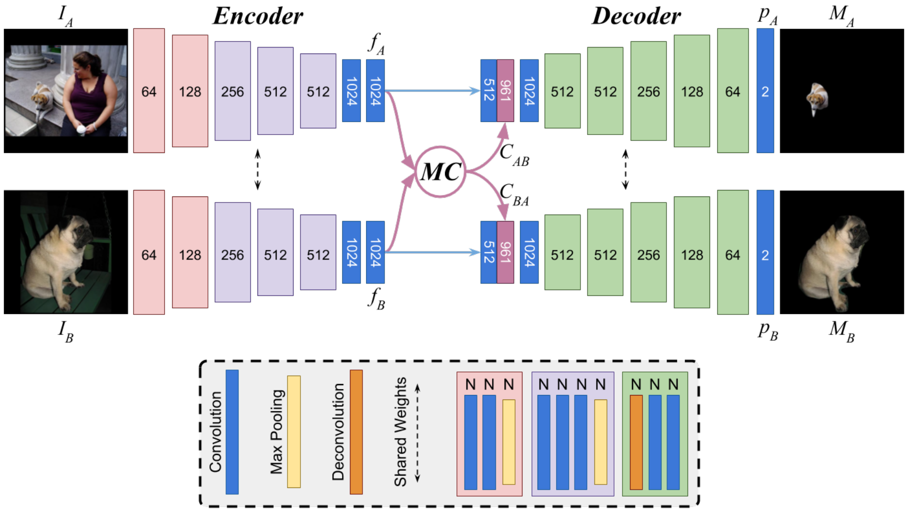


Figure 2.10: The network as designed by Li et al., containing the Siamese encoder (left) to extract feature maps f_A and f_B from input images I_A and I_B. The mutual correlation (MC) layer combines these feature maps to form correspondence maps C_{AB} and C_{BA}. Finally, Siamese decoder (right) uses the feature and correspondence maps to create the common object masks M_A and M_B. [56]

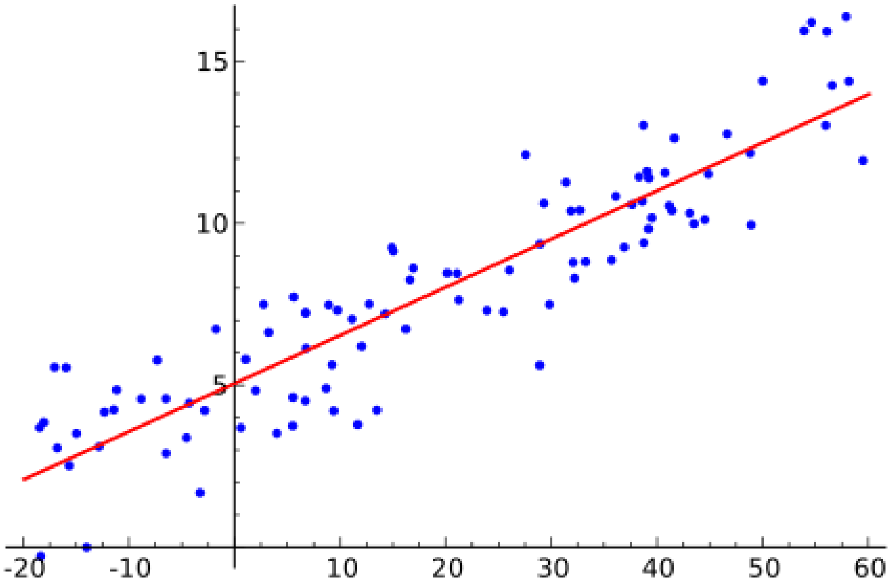


Figure 2.11: Random data points, including the best-fitting line [58]

a cyclist, it makes sense to use regression. In this section, different types of regression, regression analysis with aforementioned machine learning techniques, as well as common regression pitfalls are discussed.

Regression types

Regression analysis is the process of estimating the relationship between one or more input variables and one output variable. A simple way of looking at this, is that the goal of regression analysis is to find the best-fitting line through a set of data points, by altering parameters, as can be seen in Figure 2.11. The values of these parameters are usually determined by the method of least squares, but different methods exist for different cases.

The most simple form of regression is simple linear regression, with one input (X) and one output (Y) variable. The regression function for this model is given by:

$$Y = \beta_0 + \beta_1 X \tag{2.9}$$

Expanding on simple linear regression, there is multiple linear regression, with n input ($X_1...X_n$) variables, and still one output variable (Y). The regression function for this model is given by:

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n \tag{2.10}$$

Linear regression, obviously, only works in cases where the input variable(s) and output variable are linearly related. When this linear relationship is not present, nonlinear regression analysis can be used to find the best-fitting line through a set of points. In this case, the regression function includes nonlinear elements, such as exponential, logarithmic or trigonometric functions.

Common regression pitfalls

Although in essence, regression is a simple technique, there are a few different things that can go wrong and can lead to strange outcomes or false conclusions. In this section, several of these pitfalls are discussed.

One of the most common pitfalls, not only in regression, is overfitting. In regression, this happens when the regression function has optimised not only to the overall trend, but also to the individual observations. This, in turn, leads to a model that performs very well on the training set, but fails to predict unseen values. The best way to avoid overfitting is an easy one: keep the model at a bare minimum. In other words, do not at all times include all features in the model, but pick them wisely. For this, there are several methods, such as forward selection, backward elimination and stepwise selection. When neural networks are used for regression analysis, 'keeping the model simple' also applies to the number of hidden neurons in the network, with fewer hidden layers of smaller size being more resistant to overfitting. Another remedy against overfitting is regularisation. With regularisation techniques, such as ridge or lasso regression, penalties are added to the model's coefficients. This way, it is discouraged to have more complex models, thereby avoiding overfitting.

2.3.4 Metrics

When machine learning models are developed, it is critical to have clear and objective metrics, for example, to evaluate the effect of parameter changes, but also to compare different algorithms. In the coming paragraphs, several common metrics used to evaluate (co-)segmentation algorithms and regression models will be discussed.

Metrics for image (co-)segmentation

Before the metrics itself are discussed, first, several terms used in the explanation of the metrics will be discussed.

Foreground segmentation, where a foreground object (in this case the cyclist plus their bike) is extracted from a background, is a type of binary classification, with one background class and one foreground class, with all pixels belonging to only one class. After classification, a pixel can have one of four results: true positive (TP; pixel is correctly classified as foreground), true negative (TN; pixel is correctly classified as background), false positive (FP; pixel is classified as foreground, but belongs to background) or false negative (FN; pixel is classified as background, but belongs to foreground).

Accuracy Accuracy is one of the most intuitive evaluation metrics, giving the percentage of correctly labelled pixels over all pixels. Accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

[59]

Precision The precision metric gives the percentage of pixels correctly labelled as foreground over the total number of pixels labelled as foreground. As such, it can be seen as a quality or exactness indicator. Precision is given by:

$$Precision = \frac{TP}{TP + FP}$$

[59]

Recall Often mentioned together with precision is recall, or sensitivity. Recall describes the completeness by giving a percentage of pixels correctly labelled as foreground over the total number of pixels that should have been labelled as foreground. Recall is given by:

$$Recall = \frac{TP}{TP + FN}$$

[59]

F1 score/Dice coefficient As a way to combine the precision and recall scores into one metric, the F1 score can be used. Calculated as the harmonic mean of precision and recall, this score is an indicator of similarity. When used with binary data, as is the case with foreground segmentation, the F1 score is identical to the Dice coefficient and is given by:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

[59]

Intersection over Union/Jaccard index Another metric for similarity, and also positively correlated to the F1 score, is the Intersection over Union (IoU), or Jaccard index. This index divides the intersection (the correctly classified foreground pixels) by the union (all the pixels classified as foreground, both by the ground truth as well as by the segment) and is given by:

$$IoU = \frac{TP}{TP + FP + FN}$$

[59]

Metrics for regression analysis

To evaluate regression models, different metrics may be used. Unsurprisingly, they are all based on the difference the predicted value has from the real value. In the following equations, the predicted value is denoted by y_i, while the real value is denoted by x_i. Note that in this thesis only nonlinear regression is used, which is why the incompatible R^2-score is left out of this section [60].

Mean absolute error The mean absolute error (MAE) is probably the most intuitive metric. First, the difference is taken as absolute value, so negative differences do not cancel out positive differences. Then, the average of these absolute differences is taken. As an equation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i|$$

[61]

Mean squared error Another popular metric is the mean squared error (MSE). This is calculated similarly to MAE, except that the differences are squared. As a result, larger errors are penalized more than in MAE. This makes MSE a particular appropriate metric when an error of 2 is more than twice as worse than an error of 1.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$$

[61]

Root mean squared error The root mean squared error (RMSE) is, as the name suggests, simply the root of the MSE. Similar to MSE, this has the characteristic of penalizing larger errors. However, as a benefit over MSE, the unit of RMSE is the same as the unit of the data, making it easier interpretable.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2}$$

[61]

Chapter 3

Methodology

In this chapter, the approach taken towards the final regression model is described. To train the model, data was needed. This data was collected (Section 3.1) and processed (Sections 3.2 & 3.3). From this large data collection, several smaller data sets were created 3.4. Finally, these data sets were used in different experiments (Section 3.5).

This approach can be divided into different stages. First, both video and motion data was collected. These were then processed separately. An overview of all the steps taken, can be found in Figure 3.1.

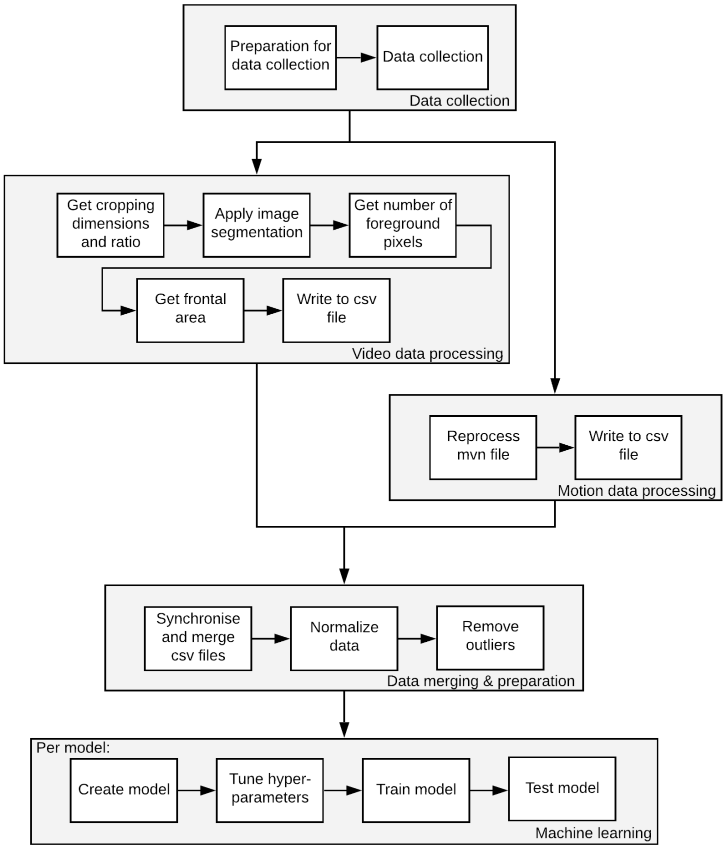


Figure 3.1: The steps taken towards the final regression model.

3.1 Data collection

3.1.1 Subjects

Due to the measures against the spread of the coronavirus (SARS-CoV-2) (i.e. social distancing & working from home), only subjects living within the same household as the author could be selected as a subject. As a result, the subject population consisted of 4 healthy, young (age: 22.75 ± 1.5 years) male subjects. All subjects have a slim to normal posture (height: 183 ± 4 cm; weight: 77.3 ± 8.5 kg; BMI: 23.0 ± 2.1 kg/m^2) and all have experience with road cycling and the various positions one can take on a bike.

3.1.2 Setup

The setup consisted of a road bike (owned by every individual subject), fixed in a bike trainer (Tacx Blue Motion; Garmin Ltd., Schaffhausen, Switzerland). This was placed in front of a camera (OnePlus 6; One Plus Technology (Shenzhen) Co., Ltd., Shenzhen, China), such that the camera was in a straight line with the bike. Next to the bike, a reference stick of known length was placed and behind the bike, a cloth of uniform colour was hung as background. The camera was equipped with the MVN Remote app (MVN Remote 2.0.1.; Xsens Technologies B.V., Enschede, The Netherlands) and connected to MVN Studio (MVN Studio 2019.2.1; Xsens Technologies B.V., Enschede, The Netherlands). A schematic overview of this setup can be found in Figure 3.2.

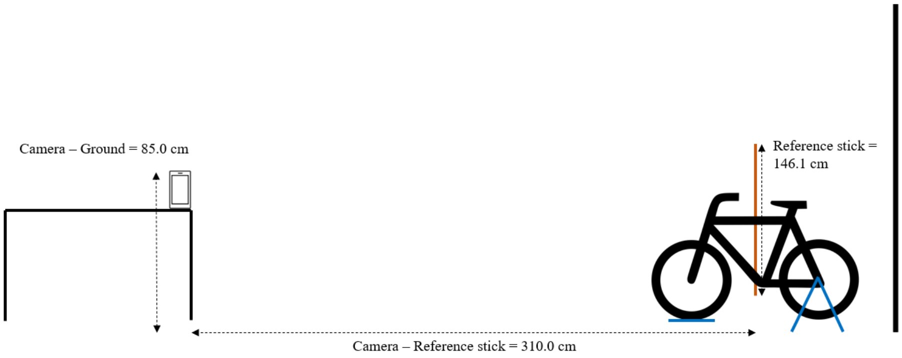

Camera – Ground = 85.0 cm
Reference stick = 146.1 cm
Camera – Reference stick = 310.0 cm

Figure 3.2: An overview of the setup used to generate data.

3.1.3 Procedure

At the start of each session, the subject was equipped with the MVN Awinda system (Xsens Technologies B.V., Enschede, The Netherlands). This system consists of 17 sensors, held in place by elastic bands (see Section 2.1 for more

Table 3.1: A description of the 9 bike positions used in the data collection. All positions were adopted twice by every subject.

	Position name	Position description
1	Hoods-high	Hands on the hoods; shoulders kept at normal height
2	Hoods-low	Hands on the hoods; shoulders kept low, near handlebars
3	Tops	Hands on the tops; shoulders kept at normal height
4	Sitting	Hands off the handlebars; sitting straight up
5	Sprint-sitting-high	Hands on the drops; sitting; shoulders at normal height
6	Sprint-sitting-low	Hands on the drops; sitting; shoulders near handlebars
7	Sprint-standing-high	Hands on the drops; standing; shoulders at normal height
8	Sprint-standing-low	Hands on the drops; standing; shoulders near handlebars
9	Time-trial	Underarms on handlebars; shoulders near handlebars

information on the system). The output was set to a 60 Hz frequency. Furthermore, the camera was connected to the MVN Studio software, using the MVN Remote app. With this app, it is possible to simultaneously start recording the motion data and the video data. As video output, there was chosen for 1080p at 15 frames per second, which is the maximum output when this link is used. Before recording, the system was calibrated. For this, the subject has to stand upright, with their hands alongside their body, walk straight forward, turn around and walk back to their starting position. The full calibration procedure takes about 30 seconds.

After calibration, the subject was instructed about the 9 different bike positions they had to adopt. A description of the positions can be found in Table 3.1. An example image for every position can be found in Appendix A. For every recording run, the subject was asked to adopt every position for approximately 10 seconds, verbally instructed when to switch position. Every subject was asked to perform 2 runs.

3.2 Video data processing

3.2.1 Pre-processing

After the data was collected, some pre-processing was done to prepare the data for the image segmentation. Previous tests, done as preparation for this thesis, showed that cropped input images provided better results than their uncropped counterparts. Therefore, cropping dimensions were determined for each video. This was done by the extraction of a single frame from each video, after which the smallest rectangle that would still include the subject and the bike in every frame was determined. The dimensions of this rectangle were the cropping dimensions. Then, given the dimensions, every video was split into frames with this size, resulting in a set of images ready for segmentation. Second, the length in pixels of the reference stick was determined. This information was required for the eventual frontal area calculation.

3.2.2 Image segmentation

The PyTorch implementation of the Deep Object Co-Segmentation (DOCS) algorithm [56] was chosen as the image segmentation algorithm for this thesis. As mentioned in Section 2.3.2, this algorithm is among the best-performing algorithms for unsupervised image co-segmentation. Next to its good performance, the complete implementation can be found on GitHub, making it very easy to implement [62].

Architecture

As stated before in the background chapter, the DOCS algorithm consists of three main parts, an encoder, a mutual correlation layer and a decoder. Firstly, the encoder extracts the high-level features of the objects in the images. These features are then fed to the mutual correlation layer, which detects the common objects in both images. Lastly, the decoder generates a mask for each image. This architecture can also be found in Figure 2.10.

One of the main benefits of this model, was the fact that it could be used right away, as the authors provided a pre-trained model, trained on a large co-segmentation dataset with image pairs from the PASCAL dataset.

Results

The authors of the DOCS algorithm reported very promising results, with the PyTorch implementation reaching precision scores of ±93% and Jaccard scores of ±79%. However, when the algorithm was applied to the video frames collected in the previous stage, the results were moderate. Whereas many images were segmented fairly accurate, there was also a substantial number of images that were segmented too aggressively, leaving cyclists without legs.

To quantify the severity, 32 images were randomly drawn from the image dataset. The ground truth of these images was manually annotated and then compared to the segments. Some examples of this can be found in Figure 3.3. Over these 32 images, several metrics were calculated, which turned out to be seriously lower than the values reported by the authors. The exact values can be found in Table 3.2.

Table 3.2: Performance of the image segmentation algorithm on the dataset, divided by subject. These values are based on 4 randomly drawn sample images per subject.

Subject	Precision	Recall	F1	Jaccard index
1	0.62	0.72	0.66	0.50
2	0.62	0.80	0.70	0.54
3	0.58	0.75	0.65	0.48
4	0.54	0.66	0.59	0.42
Average	0.59	0.73	0.65	0.49

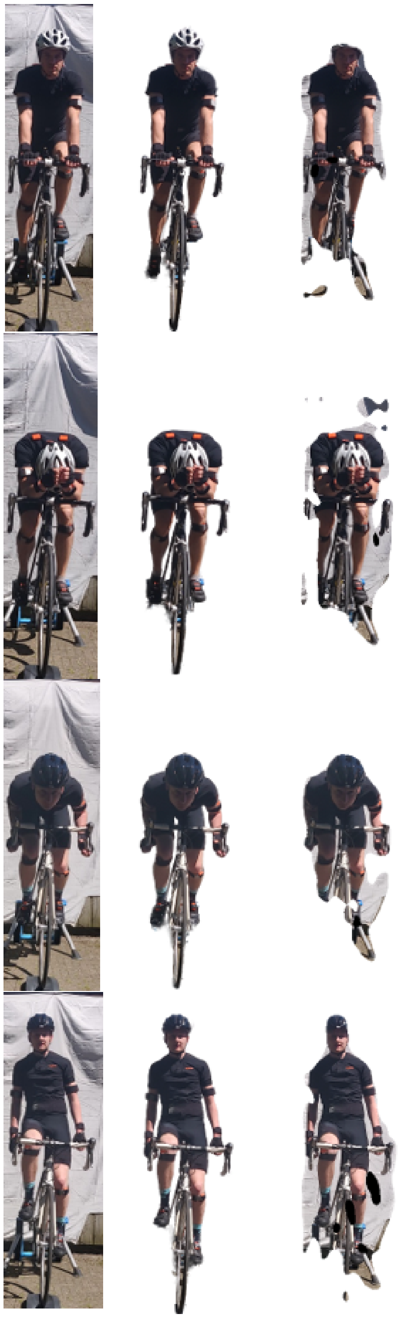


Figure 3.3: Example images, showing the original image (first column), the ground truth (second column) and the segment (third column).

3.3 Motion data processing

The default output of MVN Analyze, the recording software, contains many elements, as described in Section 2.1.1. There was chosen to export the minimal information: time-coded segment positions and segment orientations. Additionally, the file was downsampled from 60 Hz to 15 Hz and the first 90 seconds of the motion data were trimmed, to synchronize the video frames with the motion frames. As only the segment orientations mattered, these were extracted from the .mvnx and written to a .csv file. Besides the segment orientations, also experiment information (i.e. subject height, subject weight and bike position [1-9] per frame) was added to the .csv.

3.4 Data preparation

After data collection, the resulting dataset contained 10612 samples. As the cues for switching positions were given verbally, the positions are not exactly split equally between the samples. The distribution of the dataset over the different categories can be found in Table 3.3.

The full dataset was split on subject, with 3 subjects serving as training set and 1 subject serving as test set, resulting in 4 different train-test-combinations.

3.4.1 Normalization & outlier removal

All features in the dataset were within the [-1, 1] range, except for the subject height and subject weight. Therefore, these were normalized into the [0, 1] range in which 0 is 150cm/50kg and 1 is 210cm/100kg.

The image segmentation sometimes produced moderate results, by cutting off too much of the images. In these cases, the predicted frontal area is too low. To partly solve this problem, there was chosen to remove these outliers. For this, the samples were divided both on subject, as well as on position. Then the z-score over the predicted frontal area of these samples was calculated, with the samples being removed with an absolute z-score above 3. In Table 3.3, the distribution after outlier removal can be found.

3.5 Machine learning

3.5.1 Model architecture

In the experiments, two machine learning techniques were used: a neural network and support vector regression. In the following sections, the architecture, fixed between the experiments, is discussed.

Multi-layer perceptron

To prevent overfitting, there was chosen to keep the neural network simple, with 2 hidden layers between the input and the output layer. These are all fully-

Table 3.3: Number of samples before and after outlier removal, grouped by subject and position.

Subject	1		2		3		4		Total	
Position	Before	After	Before	After	Before	After	Before	After	Before	After
1	309	309	334	327	303	296	293	285	1239	1217
2	354	353	168	167	286	285	290	288	1098	1093
3	315	314	356	347	293	291	298	298	1262	1250
4	297	293	281	279	321	320	319	319	1218	1211
5	283	283	240	240	291	290	277	277	1091	1090
6	316	316	259	259	278	278	293	292	1146	1145
7	286	286	234	229	293	292	315	315	1128	1122
8	331	329	304	303	320	320	305	304	1260	1256
9	222	221	349	349	302	301	297	296	1170	1167
Total	2713	2704	2525	2500	2687	2673	2687	2674	10612	10551

connected layers, together forming the simplest form of a multi-layer perceptron (MLP). As implementation framework for the MLP, PyTorch was used [63]. The size of the input layer was dependent on the representation of the segment orientations, with 1 neuron per element (4 in quaternions, 9 in rotation matrices, per segment). Besides the segment orientations, two neurons were added to the input layer, to account for the height and the weight of the subject. The size of the output layer was always set to 1, representing the frontal area prediction. Between the layers, leaky ReLu activation functions were added, as they do not suffer from the dying ReLu problem, 'ordinary' ReLu functions suffer from. For the optimization of the network, the Adam optimizer was chosen, with the root-mean-square error (RMSE) as optimization/evaluation metric. Two parameters were chosen to vary with during hyperparameter tuning experiments: the size of the hidden layer and the learning rate.

Support vector regression

For the support vector regression (SVR), the scikit-learn implementation was used [64]. RBF was used as kernel, and an ϵ-value of 0.001 was chosen. The input, like in the MLP, was dependent on the orientation representation used. The output, again, was the frontal area prediction. As evaluation metric, also RMSE was used. The other two parameters, C and γ were chosen to optimize through hyperparameter tuning experiments.

3.5.2 Hyperparameter tuning

To optimize the model performance, experiments to tune the hyperparameters were done. These experiments were done separately for both SVR and MLP models, as well as separately for both quaternions and rotation matrices as input forms, for a total of 4 experiments.

The experiments consisted of a cross-validated grid search, in which 49 combinations of parameters per experiment were evaluated. For MLP, the values ranged from 50 to 200 for the size of the hidden layer, and from 1×10^{-7} to 1×10^{-3} for the learning rate, and every combination was trained for 50 epochs. For SVR, both parameters had values ranging from 1×10^{-4} to 1×10^{2}. The cross-validation was performed by splitting the original training set (3 subjects) into a new training set (2 subjects) and a validation set (1 subject). As there are 4 different training sets, the cross-validation was performed 4 times, after which the results were averaged over the subjects. The resulting heatmaps can be found in Figures 3.4 and 3.5, for quaternion and rotation matrix input, respectively.

The parameters with which the separate models performed best, were then used as the parameters for further experiments. For the SVR models, the C value was set to $1 \times 10^{-1}/1 \times 10^{-2}$ and the gamma value was set to $1 \times 10^{-4}/1 \times 10^{-3}$, for the model with quaternion input and rotation matrix input, respectively. For the MLP models, the hidden layer was set to have a size of 50 and the learning rate was set to 1×10^{-3}, for both the quaternion and the rotation matrix input models.

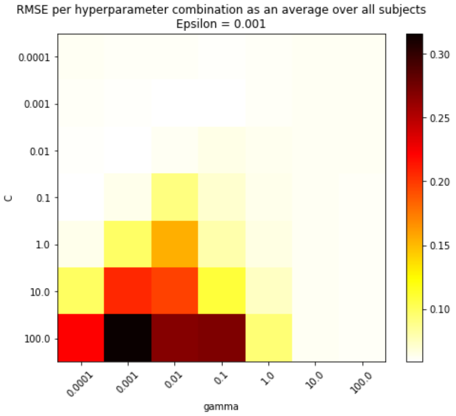


Figure 3.4: The heatmap, with RMSE scores averaged over all subjects, resulting from the cross-validated grid search to find the best performing hyperparameters for an SVR model with quaternion input.

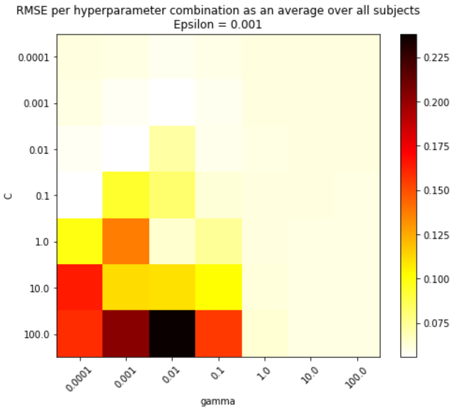


Figure 3.5: The heatmap, with RMSE scores averaged over all subjects, resulting from the cross-validated grid search to find the best performing hyperparameters for an SVR model with rotation matrix input.

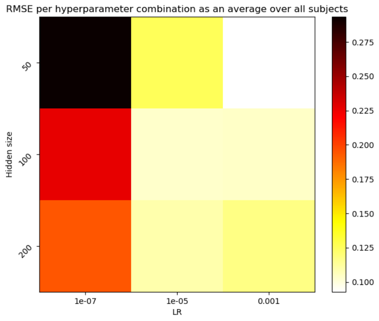


Figure 3.6: The heatmap, with RMSE scores averaged over all subjects, resulting from the cross-validated grid search to find the best performing hyperparameters for an MLP model with quaternion input.

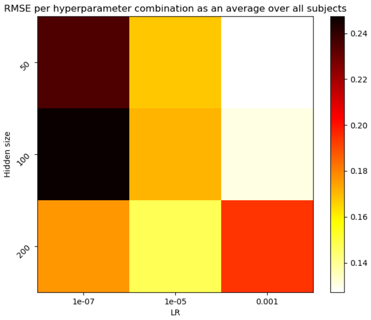


Figure 3.7: The heatmap, with RMSE scores averaged over all subjects, resulting from the cross-validated grid search to find the best performing hyperparameters for an MLP model with rotation matrix input.

3.5.3 Experiment 1

The goal of the first experiment, to answer subquestion 1, was to compare the impact the representation of the input data had on the performance of the model. As said before, the default representation of MVN segment orientations is in quaternions. This representation was compared to the rotation matrices representation. For this, the quaternions were transformed to rotation matrices, using SciPy's Rotation class [65]. As input data, all segment orientations, together with the (normalised) subject height and weight were used. The target variable was the frontal area in squared meters. After training (200 epochs, for the MLP), the model was evaluated on the test set using the root-mean-square error (RMSE) as metric. The results of this first experiment run showed a classic overfitting problem, with an increasing test loss after several epochs of training. Figures of this can be found in Appendix B. As a simple fix, early stopping was incorporated into the model: If the test loss of an epoch was higher than the test loss of the previous epoch, a counter (commonly called 'patience') would start. If the test loss would still not decrease within 10 epochs, the model would stop training and the model parameters of the model with the lowest test loss so far would be used. If the test loss would decrease within the 10 epochs, the counter would reset to 0 and the training process would continue as normal, before eventually again encountering an increase in test loss.

3.5.4 Experiment 2

A second experiment was done to answer subquestion 2, with the goal to compare the performance of the best-performing machine learning model from the previous subquestion with a non-machine learning model, i.e. the equation developed by Heil [36]. In his article, Heil proposed four equations, distinguishing four different positions, to approximate the total frontal area (cyclist + bike, identical to the frontal area as defined in this thesis), using only a cyclist's weight as further input.

The four positions used by Heil, correspond to four positions used in this research. The different names used in both researches, as well as Heil's proposed equation can be found in Table 3.4. These equations were applied to all the frames in which the position assumed by the subject matched one of the four positions as described by Heil, leading to a dataset with 4702 samples. Similar to the previous experiment, the machine learning model was trained on 3 subjects and evaluated for RMSE on the remaining subject.

3.5.5 Experiment 3

The third and last experiment was done to answer subquestion 3, to compare the performance of the best-performing model from the first subquestion with a reduced number of input features.

For the feature selection, the mutual information method, also known as information gain, was used. Using this method, the correlation between every

Table 3.4: Description of the matching positions from this and Heil's research, together with the accompanying equations as proposed by Heil. A_P = total frontal area, m_b = body mass of the cyclist. [36]

Description (this research)	Description (Heil)	Equation (Heil)
Tops	Stem position	$A_P(m^2) = 0.03176 * m_b^{0.650}$
Hoods-high	Brake hoods position	$A_P(m^2) = 0.04515 * m_b^{0.584}$
Sprint-sitting-high	Drops position	$A_P(m^2) = 0.04880 * m_b^{0.553}$
Time-trial	Traditional aero-position (TAP)	$A_P(m^2) = 0.03608 * m_b^{0.589}$

input variable and the target variable was analyzed. The full list of mutual information scores can be found in Appendix D. From this analysis, it turned out that a subject's height and weight have almost no correlation to the frontal area. Next to that, the lower body segments rank low on correlation. As one of the goals of this experiment was to find an accurate alternative with a minimal amount of sensors, there was chosen to also omit both shoulder and hand sensors from the used input features. As a result, a new experiment was done using only the segments corresponding to 5 upper body sensors, namely pelvis, sternum, head and both forearms. Again, the reduced sensor set model was trained on 3 subjects and evaluated for RMSE on the remaining subject.

Chapter 4

Results

In this chapter, the results of the experiments are presented. The results of the first experiment, comparing the performance of quaternions and rotation matrices as model input can be found in Section 4.1. In Section 4.2, the results of the second experiment, comparing the performance of a machine learning model and a non-machine learning model are discussed. The results of the final experiment, comparing the performance of a full sensor set model and a reduced sensor set model, are covered in Section 4.3.

4.1 Experiment 1

The first experiment was executed in order to give an answer to the first subquestion: "What is the performance difference between quaternions and rotation matrices, when used as input for a model that predicts the frontal area of a cyclist?" For this purpose, four different models were developed: A support vector regression (SVR) model with quaternion input, an SVR model with rotation matrix input, a multilayer perceptron with quaternion input and an MLP with rotation matrix input. The performance of these models was evaluated on the root-mean-square error (RMSE).

As can be seen in Table 4.2, the median difference of the SVR model with quaternion input is marginally closer to 0 (0.00835 vs. 0.00837). However, the spread of differences is narrower in the SVR rotation matrix model than in the SVR quaternion model, as can be seen in Figure 4.1. This results in a lower RMSE for the rotation matrix model, namely 0.05586 vs. 0.05182, see also Table 4.1. When looking at the results grouped by subject, the differences between quaternion and rotation matrix input are small. It is striking, however, that for both inputs the area difference of Subject 2 is larger than that of the other subjects. More detailed figures, showing the predicted area over time, can be found in Appendix C.1.2. In these figures, it can be seen that the predictions for both inputs are very constant, with the rotation matrix input being a little better in discriminating between positions, whereas the quaternion input is almost too constant to discriminate between positions. Another noteworthy finding that can be seen in the more detailed figures, is the jumpy behaviour of the 'real area'. Although at first sight this seems strange, a logical explanation would be the stance of the subject's feet. After all, when the cranks (the part connecting the bike's pedal and bottom bracket) are kept horizontal, the frontal area is smaller than if the cranks are kept vertical.

With the results grouped by position (see Figure 4.2), it can be seen that the RMSE of the quaternion input is lower for the first two positions (both hoods positions), but much higher for the rest of the positions, especially for positions 4 (sitting), 8 (standing sprint with shoulders low) and 9 (time-trial). A visual representation of the percentual data can also be found in Appendix C.1.1. All in all, the results show that for the SVR model, rotation matrix input delivers the best performance.

The results of the MLP model are more volatile. Overall, the results (see Tables 4.3 & 4.3 show that quaternions perform better. However, there are some striking features (see Figure 4.3). The spread of the difference is quite large in the case of Subject 2, with quaternion input. Furthermore, the model vastly underestimates the frontal area of Subject 3, especially when using rotation matrix input. These findings are also clear when looking at the predictions over time, which can be found in Appendix C.1.2. A positive aspect of the MLP model, is its ability to predict the jumpy course of the frontal area. Besides, it is quite well able to discriminate between positions. Similarly to the SVR model, the quaternion input is more constant, whereas the predictions of the rotation matrix input are more extreme. As mentioned before, it can clearly be seen that the results for Subjects 2 & 3 are well off, for both inputs. When looking at the differences per position (see Figure 4.4), it is particularly noteworthy that most of the positions with rotation matrix input are underestimated. From the table, it is clear that this is mainly caused by the underestimations of Subject 3. All in all, for this dataset, the quaternion input works best for the MLP model. This is due to the more constant predictions, causing the errors made to be smaller.

At a first glance, the differences between the SVR and MLP model are large, with the SVR model performing much better. However, after Subjects 2 & 3 were removed from the MLP results, the performance of the four models is much more similar. Albeit there being some positions and subjects for which the MLP model performs better than its SVR counterpart, the final model should be capable of generalizing over different subjects; an aspect in which the MLP model fails. Next to that, when looking at the RMSE scores, the SVR model still outperforms the MLP model. Therefore, it can be concluded that an SVR model with rotation matrix input is the best-performing model of the four tested models.

Table 4.1: RMSE scores of both SVR models, grouped by subject and position. Q = Quaternion input, RM = Rotation matrix input

Pos.	Subject 1		Subject 2		Subject 3		Subject 4		Average	
	Q	RM	Q	RM	Q	RM	Q	RM	Q	RM
1	0.0399	0.0461	0.0726	0.0701	0.0608	0.0653	0.0569	0.0578	0.0576	0.0598
2	0.0304	0.0354	0.0632	0.0646	0.0478	0.0501	0.0516	0.0450	0.0483	0.0488
3	0.0352	0.0363	0.0729	0.0652	0.0367	0.0286	0.0506	0.0454	0.0488	0.0439
4	0.0543	0.0384	0.0861	0.0620	0.0499	0.0590	0.0450	0.0370	0.0588	0.0491
5	0.0432	0.0411	0.0914	0.0935	0.0365	0.0326	0.0454	0.0409	0.0541	0.0520
6	0.0365	0.0257	0.0846	0.0941	0.0405	0.0402	0.0507	0.0394	0.0530	0.0499
7	0.0376	0.0378	0.0650	0.0633	0.0631	0.0497	0.0667	0.0635	0.0581	0.0536
8	0.0374	0.0290	0.0820	0.0768	0.0505	0.0521	0.0663	0.0535	0.0590	0.0529
9	0.0508	0.0332	0.0816	0.0811	0.0767	0.0712	0.0508	0.0406	0.0650	0.0565
Avg.	0.0406	0.0359	0.0777	0.0745	0.0514	0.0499	0.0538	0.0470	0.0559	0.0518

Table 4.2: Median values of the difference in m^2 of both SVR models, grouped by subject and position. Q = Quaternion input. RM = Rotation matrix input

Pos.	Subject 1		Subject 2		Subject 3		Subject 4		Average	
	Q	RM	Q	RM	Q	RM	Q	RM	Q	RM
1	0.0190	0.0206	0.0369	0.0328	-0.0024	-0.0168	0.0013	-0.0133	0.0137	0.0058
2	-0.0157	0.0153	0.0293	0.0337	-0.0299	-0.0361	-0.0322	-0.0239	-0.0121	-0.0028
3	0.0199	0.0138	0.0477	0.0409	0.0230	-0.0031	0.0129	-0.0096	0.0259	0.0105
4	0.0439	0.0219	0.0700	0.0400	-0.0081	-0.0286	0.0270	0.0034	0.0332	0.0092
5	0.0142	0.0156	0.0730	0.0725	0.0171	0.0014	0.0220	0.0154	0.0316	0.0262
6	-0.0279	-0.0014	0.0483	0.0673	-0.0054	-0.0137	-0.0375	-0.0230	-0.0056	0.0073
7	0.0062	0.0130	0.0592	0.0579	0.0580	0.0414	0.0424	0.0411	0.0415	0.0384
8	-0.0305	-0.0037	-0.0121	0.0144	-0.0266	-0.0324	-0.0450	-0.0250	-0.0285	-0.0117
9	-0.0389	-0.0112	0.0126	0.0303	-0.0423	-0.0373	-0.0288	-0.0123	-0.0244	-0.0076
Avg.	-0.0011	0.0093	0.0405	0.0433	-0.0019	-0.0139	-0.0042	-0.0052	0.0084	0.0084

Table 4.3: RMSE scores of both MLP models, grouped by subject and position. Q = Quaternion input, RM = Rotation matrix input

Pos.	Subject 1		Subject 2		Subject 3		Subject 4		Average	
	Q	RM	Q	RM	Q	RM	Q	RM	Q	RM
1	0.0458	0.0520	0.1154	0.1246	0.0616	0.0847	0.0588	0.0576	0.0704	0.0797
2	0.0380	0.0492	0.0744	0.0631	0.0450	0.0636	0.0447	0.0430	0.0505	0.0547
3	0.0457	0.0400	0.0966	0.0868	0.0460	0.1015	0.0490	0.0487	0.0593	0.0693
4	0.0412	0.0371	0.0989	0.0488	0.0674	0.1277	0.0414	0.0334	0.0622	0.0617
5	0.0436	0.0373	0.0961	0.0641	0.0386	0.0920	0.0422	0.0392	0.0552	0.0581
6	0.0339	0.0325	0.0858	0.1208	0.0496	0.0840	0.0393	0.0337	0.0522	0.0678
7	0.0443	0.0393	0.1019	0.0550	0.0535	0.1408	0.0642	0.0617	0.0660	0.0742
8	0.0430	0.0381	0.0905	0.1193	0.0510	0.0790	0.0508	0.0487	0.0588	0.0713
9	0.0481	0.0350	0.0949	0.0808	0.0553	0.1107	0.0436	0.0386	0.0605	0.0663
Avg.	0.0426	0.0401	0.0950	0.0848	0.0520	0.0982	0.0482	0.0450	0.0594	0.0670

Table 4.4: Median values of the difference in m^2 of both MLP models, grouped by subject and position. Q = Quaternion input. RM = Rotation matrix input

Pos.	Subject 1		Subject 2		Subject 3		Subject 4		Average	
	Q	RM	Q	RM	Q	RM	Q	RM	Q	RM
1	-0.0066	0.0190	-0.0138	-0.0905	0.0096	0.0684	-0.0129	-0.0141	-0.0059	-0.0043
2	-0.0011	0.0309	0.0123	-0.0028	0.0113	0.0500	-0.0262	-0.0244	-0.0009	0.0134
3	-0.0037	0.0132	0.0556	-0.0548	0.0339	0.0976	-0.0120	-0.0099	0.0184	0.0115
4	0.0197	0.0200	0.0800	-0.0119	-0.0271	0.1183	-0.0020	-0.0016	0.0176	0.0312
5	-0.0133	0.0007	0.0687	0.0130	0.0152	0.0895	0.0117	0.0138	0.0206	0.0292
6	-0.0096	0.0180	0.0077	0.0892	0.0283	0.0760	-0.0187	-0.0062	0.0019	0.0442
7	-0.0145	-0.0029	0.0909	0.0368	0.0385	0.1383	0.0381	0.0398	0.0383	0.0530
8	-0.0022	0.0115	0.0068	0.0952	-0.0003	0.0644	-0.0013	-0.0160	0.0007	0.0388
9	-0.0178	-0.0072	-0.0293	0.0088	0.0028	0.1024	0.0009	0.0018	-0.0109	0.0264
Avg.	-0.0055	0.0114	0.0310	0.0092	0.0125	0.0894	-0.0025	-0.0019	0.0089	0.0271

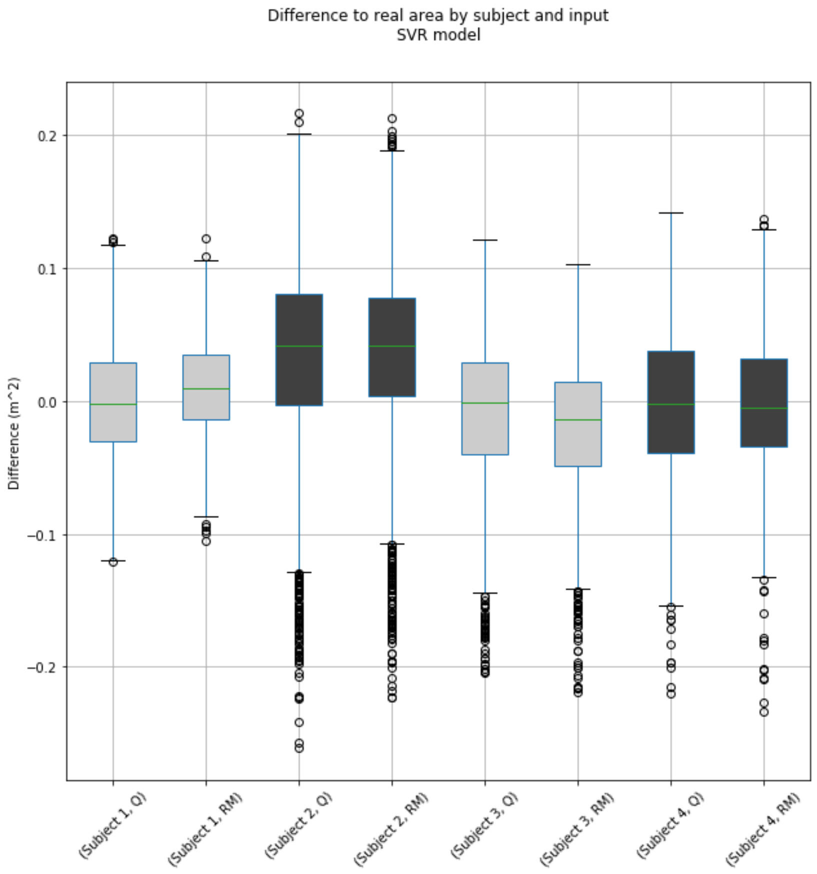


Figure 4.1: The difference between the area predicted by the SVR model and the real area, grouped by subject and input. Q = quaternion input, RM = rotation matrix input

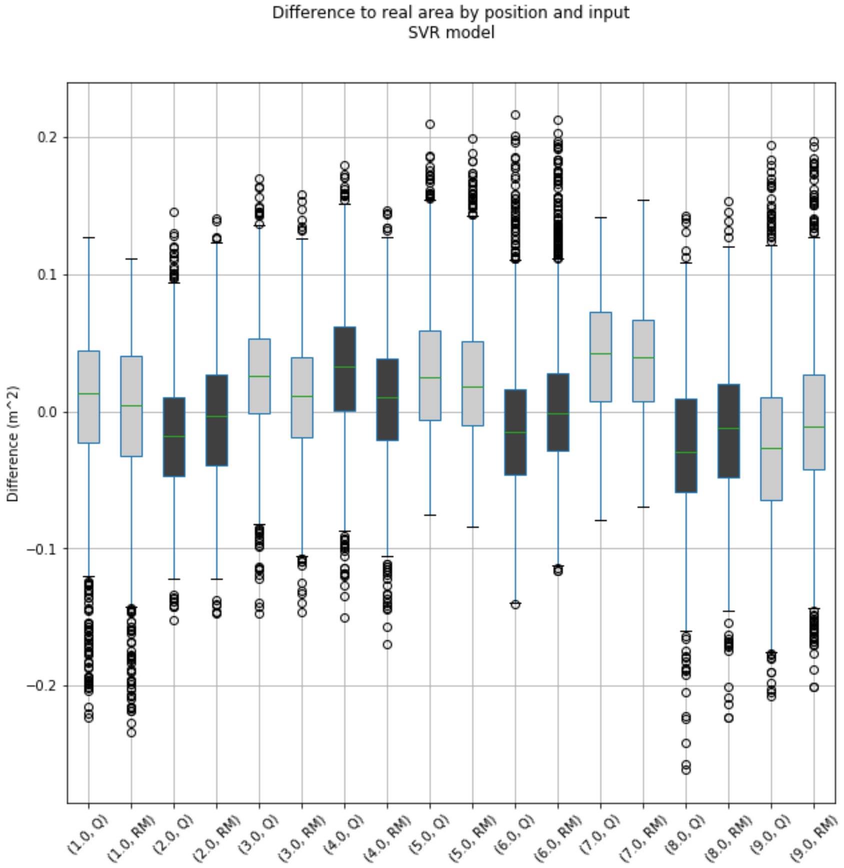


Figure 4.2: The difference between the area predicted by the SVR model and the real area, grouped by position and input. Q = quaternion input, RM = rotation matrix input

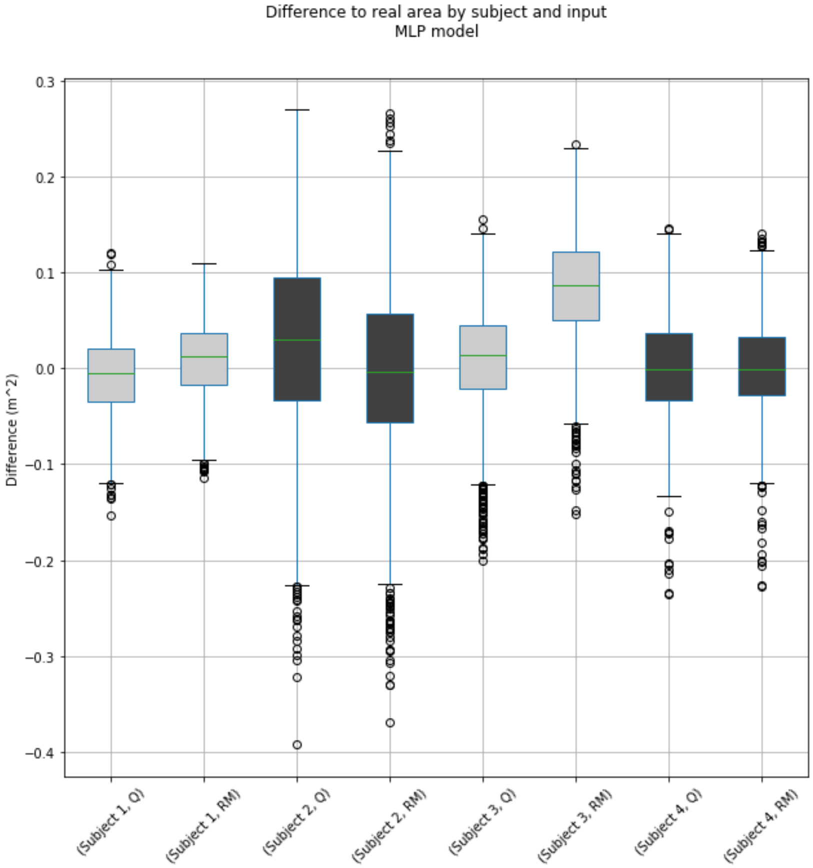


Figure 4.3: The difference between the area predicted by the MLP model and the real area, grouped by subject and input. Q = quaternion input, RM = rotation matrix input

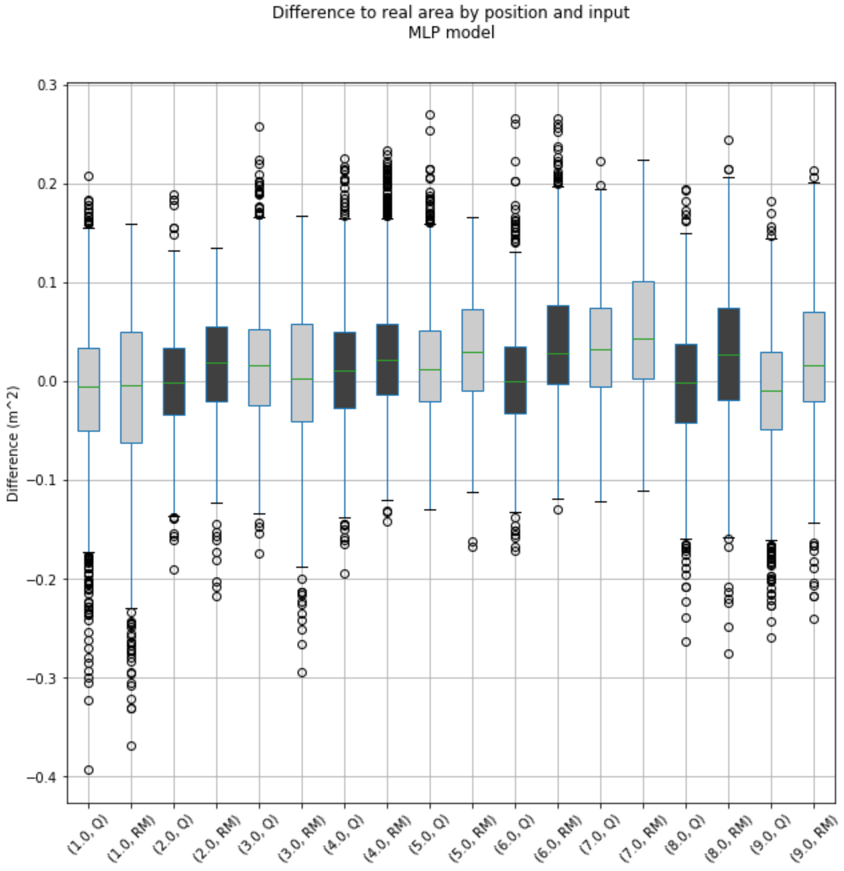


Figure 4.4: The difference between the area predicted by the MLP model and the real area, grouped by position and input. Q = quaternion input, RM = rotation matrix input

4.2 Experiment 2

After the best-performing input, with the best-performing model (i.e. SVR model with rotation matrix input) was identified, this model was compared to a non-machine learning model, in order to answer the second subquestion: "What is the performance difference between a machine learning model and a non-machine learning model, for predicting frontal area of a cyclist?" As non-machine learning model, a model that approximates frontal area, based on the bike position and the weight of the cyclist was applied to the data [36].

From the results (see Tables 4.5 & 4.6), it is clear that the non-machine learning approach consequently overestimates the area, with a median difference to the real area of 0.1854 m^2 or 55%. The poor performance of the non-machine learning approach is also reflected by its RMSE, which is almost 3.8 times higher than the RMSE of the SVR model. As can be seen in Figures 4.5 and 4.6 the subject and the bike position do not have much impact on the performance.

From the results, it is clear that there is a large performance difference between the SVR model and the Heil model, with the former performing much better, independent of subject or position.

Table 4.5: RMSE scores of the Heil model, grouped by subject and position.

Position	Subject 1	Subject 2	Subject 3	Subject 4	Average
1	0.1629	0.1552	0.1451	0.1448	0.1520
3	0.1366	0.1207	0.0885	0.1064	0.1130
5	0.1434	0.1259	0.1016	0.1092	0.1200
6	0.1201	0.1095	0.0889	0.1061	0.1062
Average	0.1408	0.1278	0.1060	0.1166	0.1228

Table 4.6: Median values of the difference in m^2 of the Heil model, grouped by subject and position.

Position	Subject 1	Subject 2	Subject 3	Subject 4	Average
1	0.1463	0.1236	0.0957	0.1034	0.1172
3	0.1239	0.1018	0.0702	0.0777	0.0934
5	0.1291	0.1125	0.0825	0.0953	0.1049
6	0.1067	0.0830	0.0561	0.0750	0.0802
Average	0.1265	0.1052	0.0761	0.0879	0.0989

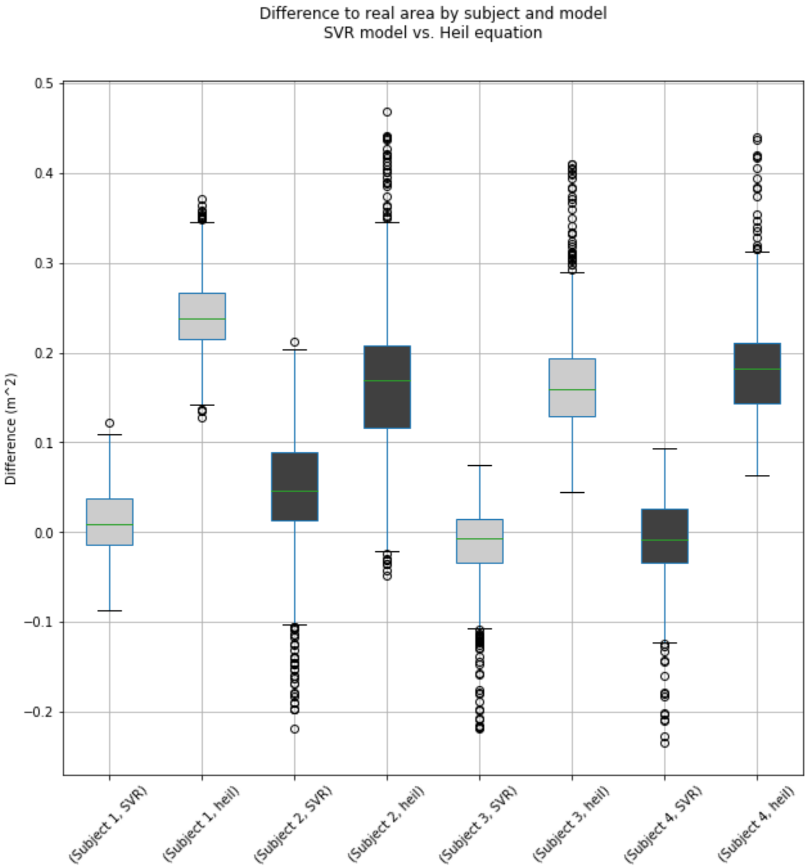


Figure 4.5: The difference between the area predicted by the SVR model/Heil model and the real area, grouped by subject and model.

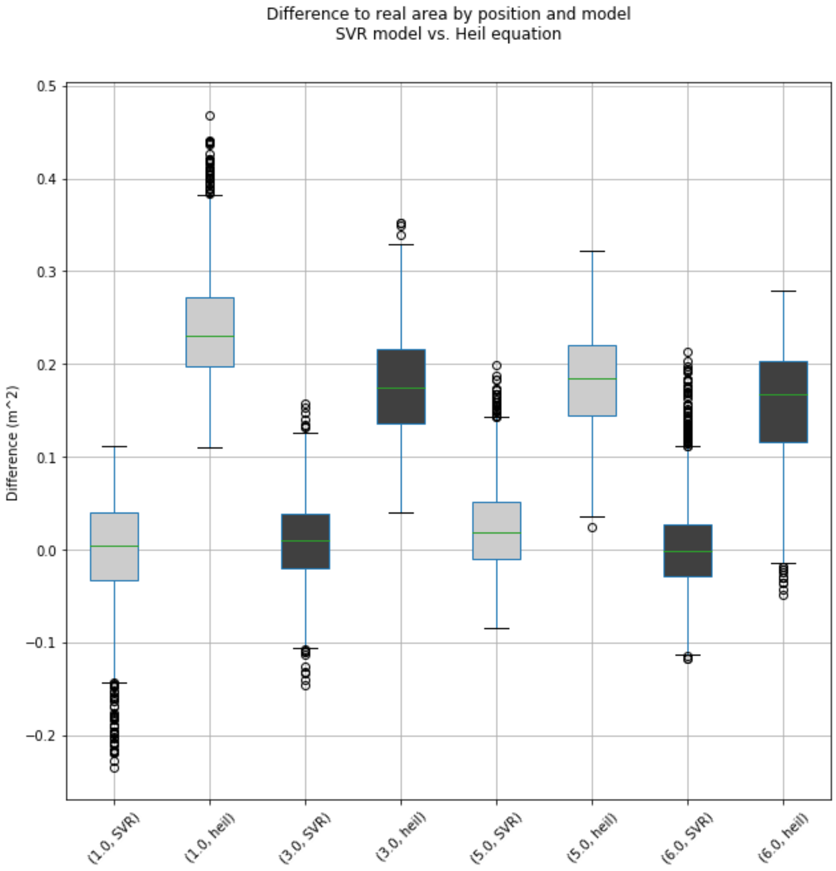


Figure 4.6: The difference between the area predicted by the SVR model/Heil model and the real area, grouped by position and model.

4.3 Experiment 3

The third and last experiment compares the performance between a model based on all sensors to a model based on a reduced sensor set, as to answer the third subquestion: “What is the performance difference between a full sensor set model and a reduced sensor set model?” In both cases, the best-performing model was used. However, in the reduced sensor set case, only segments that are deduced from five upper-body sensors are used as input.

As can be seen in Tables 4.7 & 4.8 and Figures 4.7 and 4.8, the full sensor set model performs only slightly better than the reduced sensor set model. These small differences are also visible when looking at the results grouped by subject, in which the reduced sensor set model outperforms the full sensor set model on RMSE in two of the four cases, albeit by a minimal margin. Also, when looking at the results grouped by position, results are very close.

All in all, it can be said that the performance difference between a full sensor set model and a reduced sensor set model is very minimal, although the full sensor set model outperforms the reduced sensor set model in almost all cases.

Table 4.7: RMSE scores of the reduced sensor set model, grouped by subject and position.

Position	Subject 1	Subject 2	Subject 3	Subject 4	Average
1	0,0441	0,0721	0,0636	0,0582	0,0595
2	0,0327	0,0678	0,0481	0,0464	0,0488
3	0,0350	0,0691	0,0291	0,0466	0,0450
4	0,0392	0,0670	0,0581	0,0375	0,0505
5	0,0428	0,0983	0,0333	0,0414	0,0539
6	0,0261	0,0979	0,0402	0,0418	0,0515
7	0,0386	0,0680	0,0503	0,0618	0,0547
8	0,0286	0,0773	0,0520	0,0567	0,0536
9	0,0331	0,0836	0,0711	0,0430	0,0577
Average	0,0356	0,0779	0,0495	0,0482	0,0528

Table 4.8: Median values of the difference in m^2 of the reduced sensor set model, grouped by subject and position.

Position	Subject 1	Subject 2	Subject 3	Subject 4	Average
1	0,0229	0,0384	-0,0124	-0,0121	0,0092
2	0,0138	0,0412	-0,0322	-0,0240	-0,0003
3	0,0158	0,0461	0,0023	-0,0092	0,0138
4	0,0227	0,0458	-0,0268	0,0037	0,0113
5	0,0180	0,0779	0,0051	0,0135	0,0286
6	-0,0032	0,0720	-0,0102	-0,0258	0,0082
7	0,0131	0,0632	0,0426	0,0377	0,0391
8	-0,0079	0,0173	-0,0317	-0,0297	-0,0130
9	-0,0133	0,0365	-0,0364	-0,0153	-0,0071
Average	0,0091	0,0487	-0,0111	-0,0068	0,0100

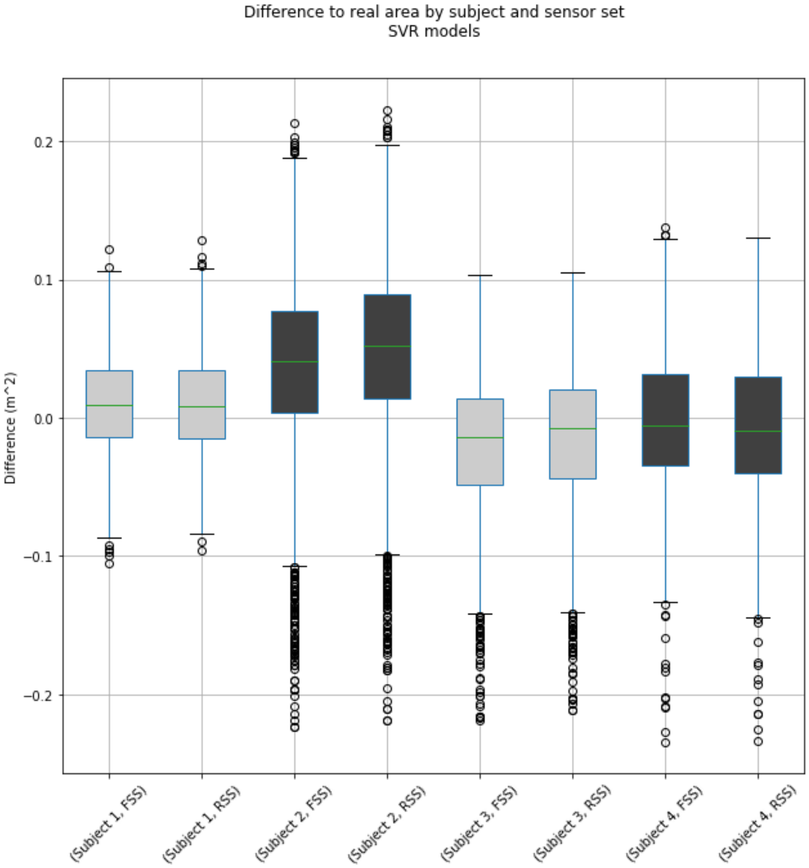


Figure 4.7: The difference between the area predicted by the SVR model and the real area, grouped by subject and input sensor set. FSS = full sensor set input, RSS = reduced sensor set input

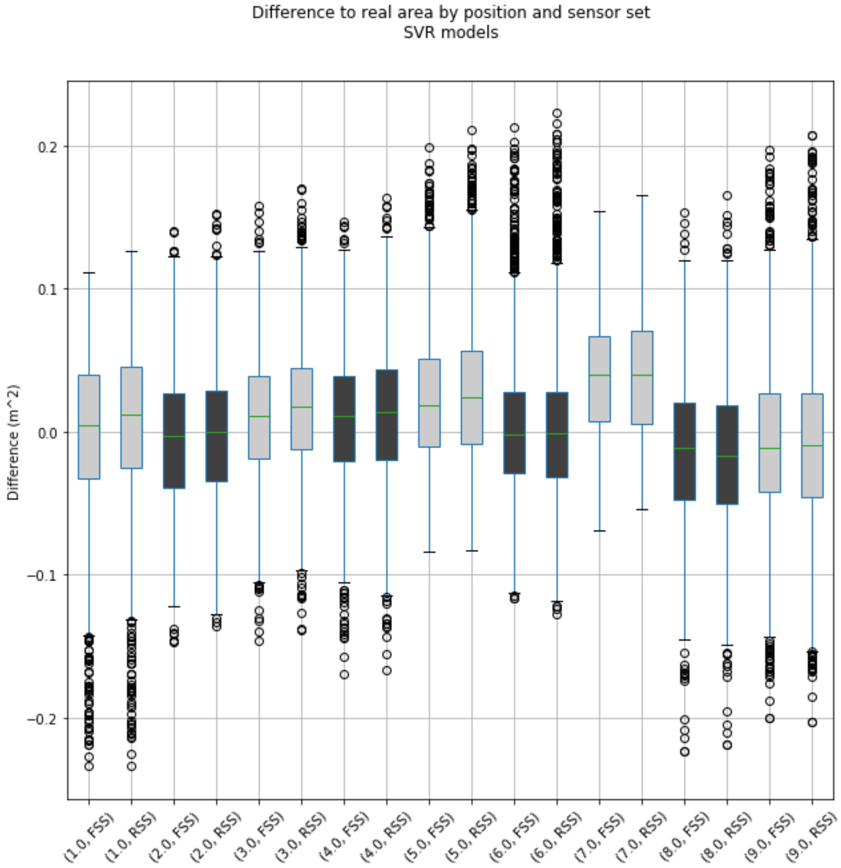


Figure 4.8: The difference between the area predicted by the SVR model and the real area, grouped by position and input sensor set. FSS = full sensor set input, RSS = reduced sensor set input

Chapter 5

Discussion

The results, presented in the previous chapter, will be further elaborated on in this chapter. Per experiment, expectations and explanations are discussed, also while looking back at the subquestions posed in the introduction (Sections 5.1, 5.2 & 5.3). Next to that, implications, limitations and further work are discussed in Sections 5.4, 5.5 & 5.6, respectively.

5.1 Experiment 1

When looking at the results of the SVR model, it can be concluded that using rotation matrix input outperforms using quaternion input. This could be caused by the fact that when using quaternions, a positive rotation over a positive axis, is the same as a negative rotation over the negative of an axis, which was not corrected for in the dataset. As a result, quaternions have identical orientations represented by two non-identical vectors, while rotation matrices can only represent identical orientations with identical matrices.

Then, when looking at the results of the MLP model, a striking result from the first experiment, was the volatility of the model. Both the RMSE and the median showed that the MLP model was not generalizing well over the different subjects. Most likely, this is caused by the number of training samples, which was probably too low for the MLP model to function properly. It can be expected a larger number of samples can improve the generalization ability of the MLP model. This will then probably be making it a better fit for this problem, as the MLP model scores better than the SVR model on predicting the jumpy course of the area.

5.2 Experiment 2

From the results, it is clear that the anthropometric by Heil [36], did not perform well and that the support vector regression (SVR) model performed much better. One possible explanation for the poor performance of the Heil model, is the accuracy of the equations. In the article, Heil already mentions R^2 values ranging from 0.397 to 0.681. These scores already indicate a suboptimal fit of the equations to the data. Another explanation could be the age of the Heil model. In this thesis, the frontal area of the cyclist together with the bike is used, a combination which Heil calls the 'total frontal area'. In the Heil model, the frontal area of a standard road bike has been incorporated in the calculations. However, since the publication of the article in 2002, the development of road bikes has not stood still. Therefore, it could be the case that the frontal area of a standard road bike in 2002 is quite a bit higher than that of a standard road bike produced more recently.

5.3 Experiment 3

The results showed that the SVR model that used a reduced sensor set, consisting of just 5 sensors (head, sternum, pelvis and both forearms), performed only marginally worse than the model that used a full sensor set. A noteworthy fact when looking at the feature selection, is the negligible correlation subject height and subject weight have with the frontal area. This is striking, as these are used as predictors for body surface area [10]–[13]. A possible explanation could be, that the correlation of height and weight is negligible when compared to the correlation of segment orientations. Another possible explanation, is the relative homogeneity of the subjects used in this research, with 3 out of 4 subjects having the same height and similar weight. The fact that lower-body segments have low correlation to the frontal area can be well explained. After all, this half of the body does not or merely change in frontal area when the position on the bike is altered. This automatically means that the upper-body segments have a relatively high correlation to the frontal area. This is quite a benefit of the model, because, when it comes to intrusiveness, the fewer sensors are required, the better. A limiting sidenote in this, however, is the fact that MVN Analyze does not (yet) support calibration with only 5 sensors. As of now, the closest is an upper body calibration, that also requires both shoulders and both upper arms to be equipped with sensors.

5.4 Implications

From the results, it can be deduced that it is possible to predict the frontal area of a cyclist, using motion data, specifically segment orientations. Furthermore, the results have shown that the performance of a reduced sensor set, only using sensors on the upper body, is similar to the performance of a full sensor set. This is desirable in a real-life application, as fewer sensors mean shorter set-up times and less intrusiveness.

As stated in the problem statement in the beginning of this thesis, there are several sub-optimal aspects to current methods of frontal area testing, e.g. costliness, absence of motion, or inability to deliver real-time results.

The final model (i.e. the SVR model with rotation matrix input and a reduced set of sensors) has the potential to solve these problems, as it is specifically designed for its use in real-life conditions, especially seeing that a model using a reduced, less intrusive, sensor set shows similar performance to a full sensor set model.

Furthermore, it should be very well possible to implement this model into an application that provides real-time results. Such an application could be developed to be run on a smartphone. The cyclist performing an aerodynamic test would then ride with sensors placed on the upper body. The output of the sensors would be processed by the smartphone, by running it through the machine learning model. The resulting frontal area can then immediately be displayed on the smartphone, possibly attached to the handlebars, or sent to a bike computer, also attached to the handlebars. Additional interesting features could be a running 3-second average of the frontal area, similar to modern power meters and the display of a small upward/downward pointing arrow, next to the frontal area prediction, indicating an increase/decrease in frontal area compared to the previous prediction.

The methods presented in this model are fairly simple. Besides the MVN system, there is no need for advanced materials. Also the measurements can be performed quickly and do not require subjects with specific features. Because of this, it is very well possible to extend the dataset with more subjects, probably leading to increased model accuracy. Next to that, because of the simplicity of the methods, it is easy to also apply this method to other sports in which aerodynamics play an important role, such as speed skating or alpine skiing.

Concretely, this could lead to a change in the way of aerodynamic testing. Tests, for example, could take place over a longer distance. This gives insight into the role fatigue plays in aerodynamics, specifically when looking at frontal area. Furthermore, this model could open doors to low-level aerodynamics testing. After all, a handful of sensors is enough for results. This way, frontal area tests could also become more accessible for amateur road cyclists.

Another application for this research is virtual cycling. In this indoor version of road cycling, cyclists use smart indoor trainers to race against other cyclists in a virtual environment. Whereas previously this was mostly popular as an alternative to road cycling during the winter, during the corona crisis this has gained popularity, also amongst professional cyclists. As of now, the position of a cyclist has no influence on the amount of resistance the smart trainer generates. With the implementation of this research, this would be made possible and increase the realisticness of virtual cycling.

5.5 Limitations

This research was largely executed during the 2020 corona crisis. Therefore, several aspects of this research were done differently than planned, leading to limitations.

The most prominent and important limitation is the small sample size ($n = 4$). During the data collection phase, The Netherlands was in lockdown, making it impossible to do a large-scale data collection in a responsible manner. Related to this, is the homogeneity of the subjects, as discussed before. The expectation is that the performance of the model will greatly increase with more, and more varying, data.

Another limitation of this research, also related to the corona crisis, is the test setup. Due to the restrictions that were in place, it was not possible to do measurements in lab-conditions. Instead, a backyard test setup, using common materials was built. The expectation is that this has, at the least, not contributed to the accuracy of the dataset.

A last, but important limitation is the performance of the image segmentation algorithm. Although the algorithm showed excellent performance on random test images taken from the internet, the performance on real video frames, containing a cyclist on a bike, was at most okay. The cause of this can only be guessed, but could have to do with the video data to be segmented being too similar. This way, the algorithm might not have recognized some parts of the cyclist as foreground, instead of (non-moving) background. Consequently, the quality of the dataset was not great, containing many inaccurate frontal area values. As described in Section 3.4, it was tried to solve this limitation by applying outlier removal to the dataset. Later on in the research, it appeared that the criteria for outliers (absolute z-score of 3) might have been too soft, since many outliers can be found when looking at the various results boxplots. Due to time constraints, a stricter outlier removal has not been applied, but this might help in increasing the quality of the dataset. After all, the expectation is that a better quality dataset leads to much better model performance.

5.6 Future work

There are several recommendations for further work on this thesis, mostly based on the limitations as stated in the previous section.

The most important recommendations concern the performance of the image segmentation and the size of the sample set. As said in the previous section, it is expected that a larger dataset of better quality, will greatly improve the performance of the suggested SVR model.

When more data is obtained, another interesting idea for future work is to do more research into the applicability of neural networks on this problem. It is generally assumed that especially neural networks profit from an increase in data, possibly making it a better performing model than SVR. Next to that, in this thesis only a very simple neural network structure was used, whereas there are many possible configurations. It would be interesting to see if implementing one of these more advanced configurations increases the model performance.

As a final recommendation, after further development of the model, it would be needed to validate the performance and accuracy of this model against the golden standard of this moment, the wind tunnel. Next to that, it should be tested for its usability over longer distances.

Chapter 6

Conclusions

As stated in the introduction of this thesis, the goal of this study was the development of a motion data-based model, able to predict the frontal area of a cyclist in real-time and in a real-life environment. To achieve this goal, several models with different configurations were developed and experimented with. As a results of these experiments, it followed that an SVR model was a good fit for solving the regression problem of predicting frontal area from segment orientations. Next to that, it was shown that using a reduced sensor set as input for the model, yielded almost identical performance when compared to using a full sensor set as input.

Due to various limitations, the performance of the final model is not yet good enough for real-life use and application in the professional cycling world. However, the results show enough indication that the model developed in this thesis can be called a successful proof-of-concept. Accordingly, it is expected that with further development, most importantly by increasing the size and quality of the dataset, this model can be a valuable addition to the current aerodynamic testing methods, able to bring real-time results in a real-life setting.

Appendix A

Examples of positions

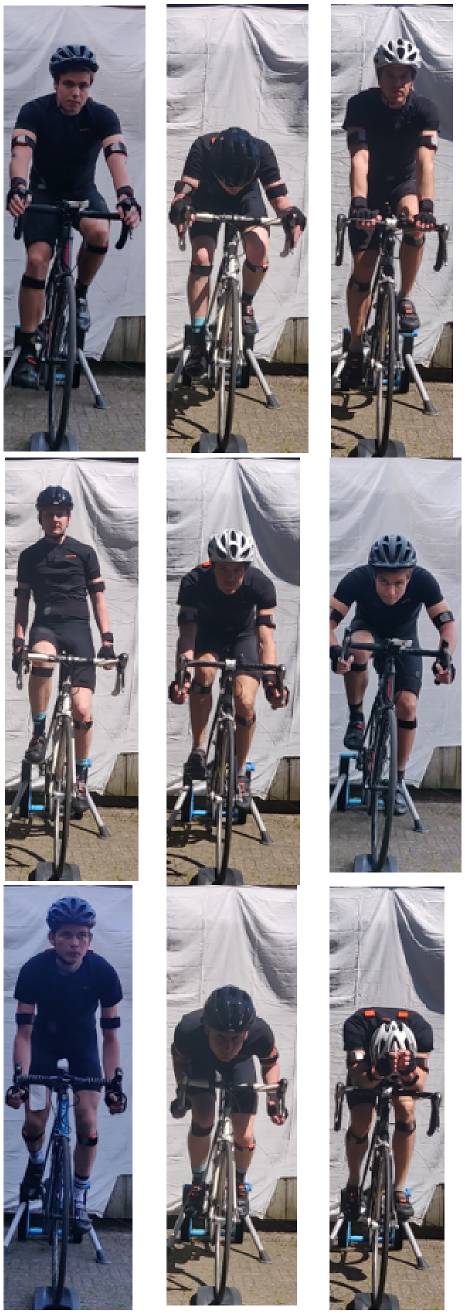


Figure A.1: Samples from each of the 9 measured positions. From top to bottom, left to right: (1) Hoods-high; (2) hoods-low; (3) tops; (4) sitting; (5) sprint-sitting-high; (6) sprint-sitting-low; (7) sprint-standing-high; (8) sprint-standing-low; (9) time-trial.

Appendix B

Training/test losses

B.1 Without early stopping

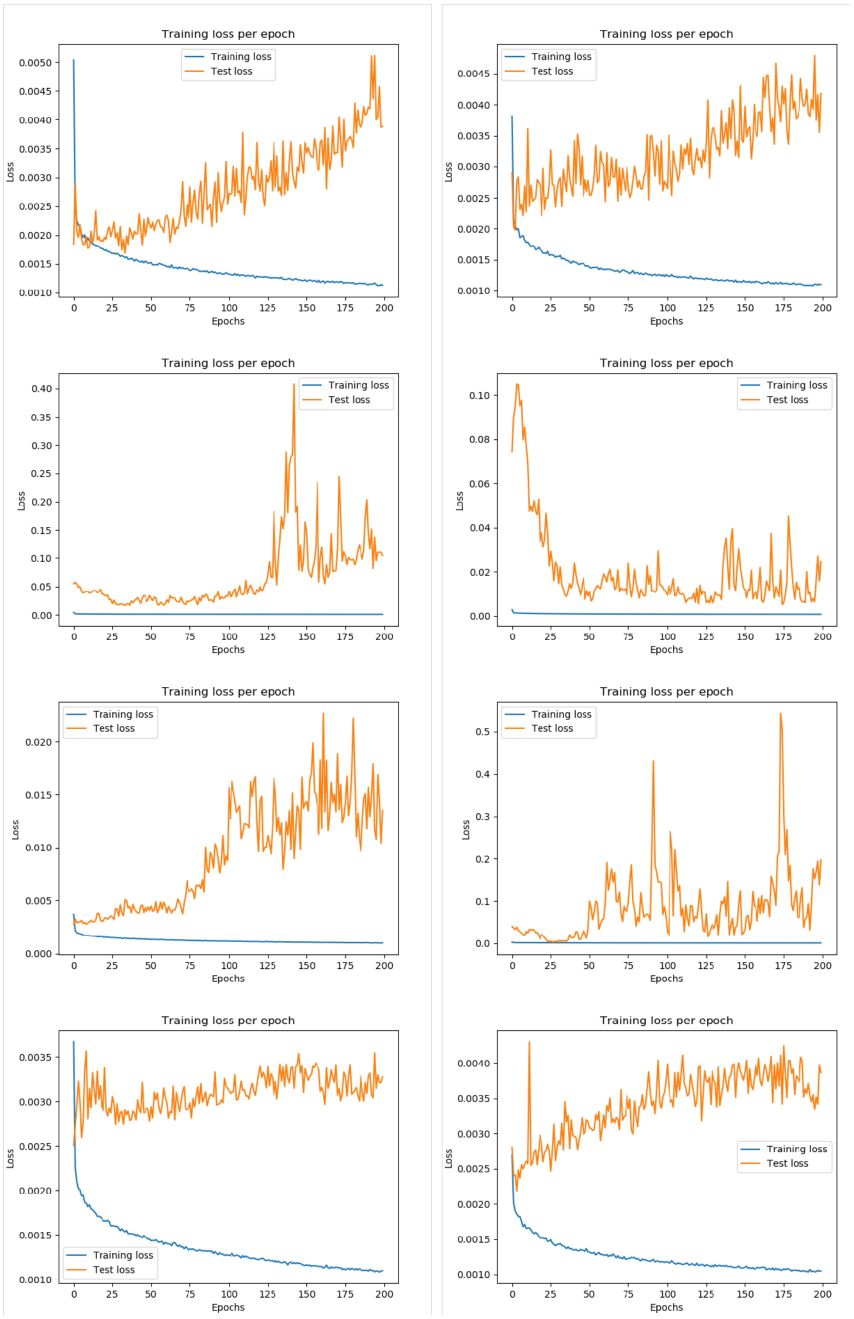


Figure B.1: Training and test losses of the MLP model, without early stopping. One subject per row. Left-side images are with quaternion input, right-side images are with rotation matrix input.

B.2 With early stopping

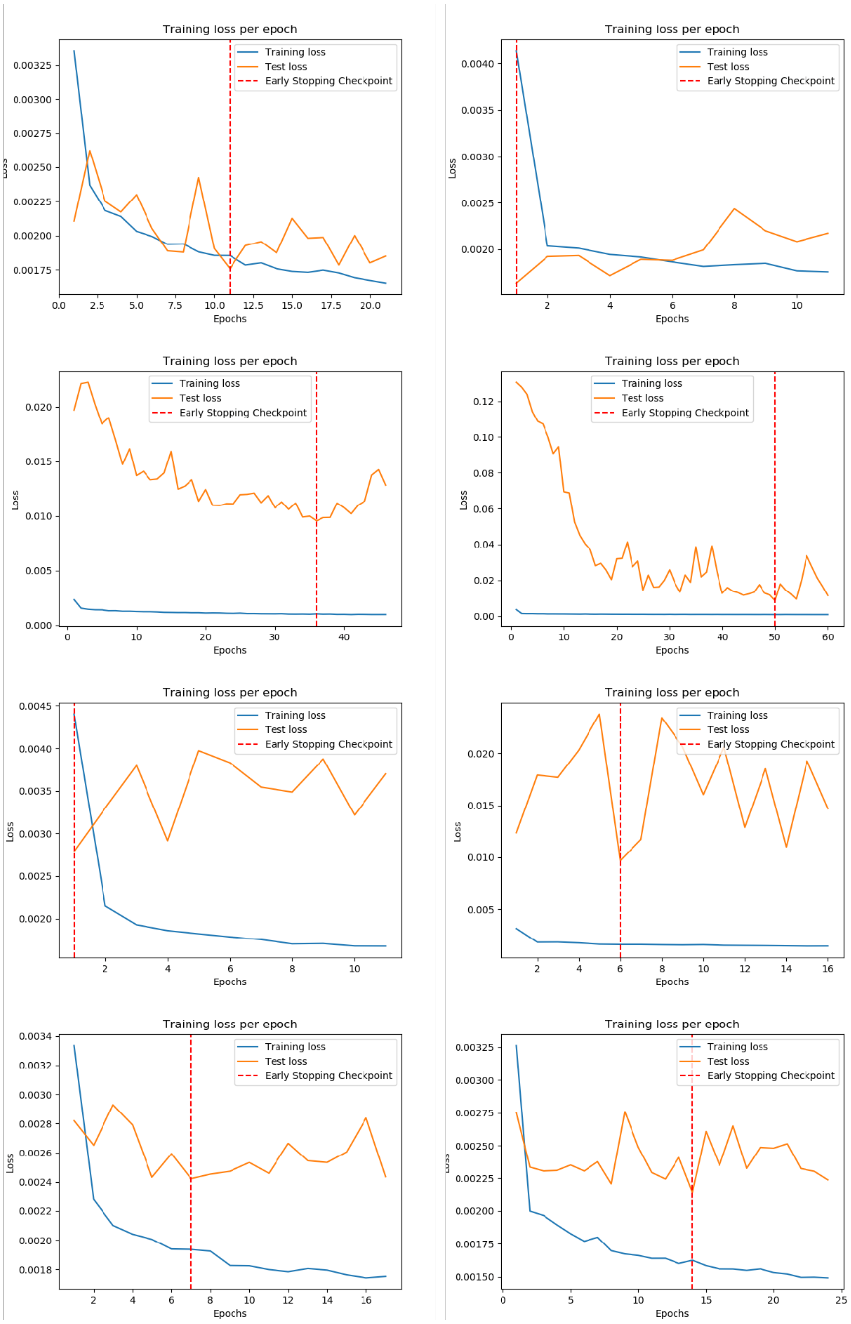


Figure B.2: Training and test losses of the MLP model, with early stopping. One subject per row. Left-side images are with quaternion input, right-side images are with rotation matrix input.

Appendix C

Results - More Figures

C.1 Experiment 1

C.1.1 Boxplots

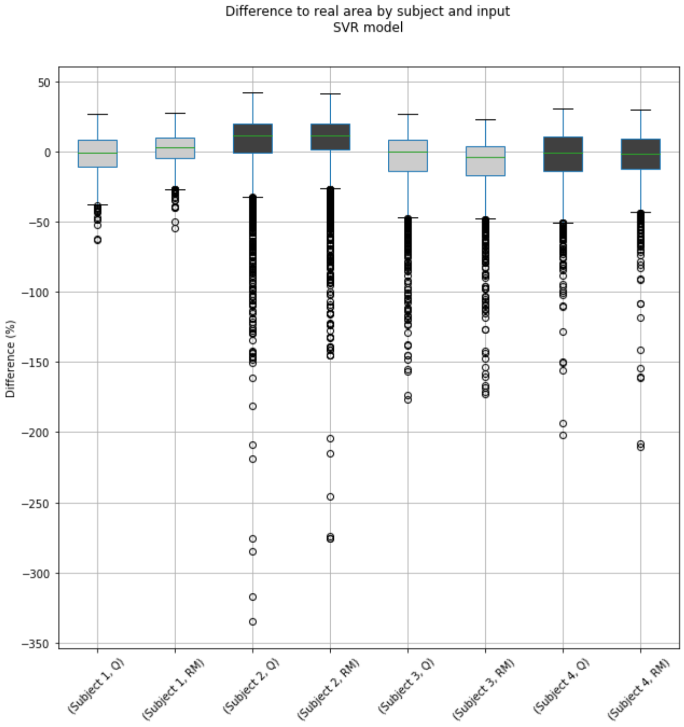


Figure C.1: The difference between the area predicted by the SVR model and the real area, grouped by subject and input. Q = quaternion input, RM = rotation matrix input

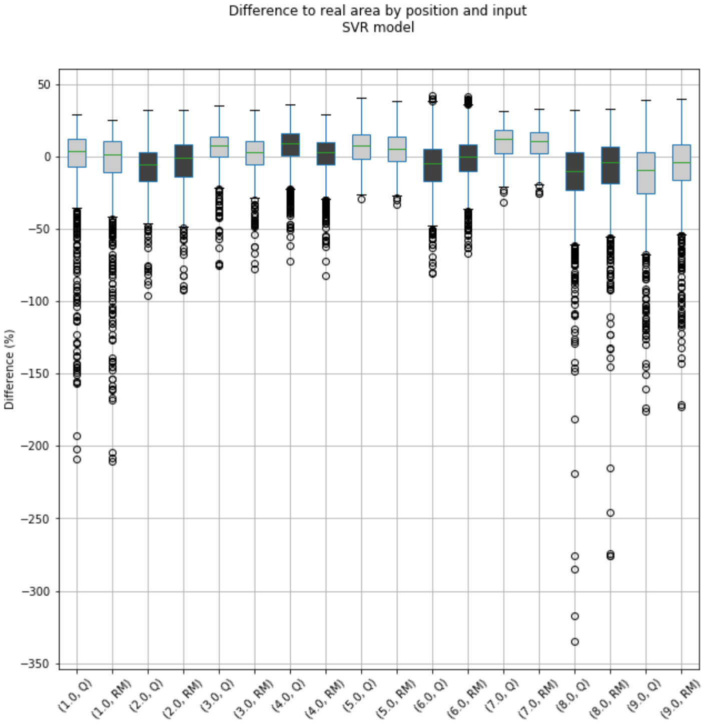


Figure C.2: The difference between the area predicted by the SVR model and the real area, grouped by position and input. Q = quaternion input, RM = rotation matrix input

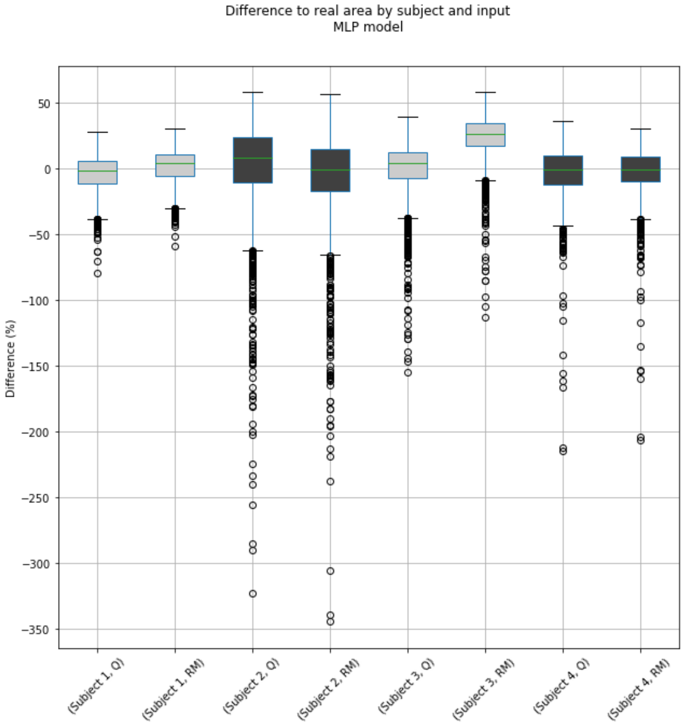


Figure C.3: The difference between the area predicted by the MLP model and the real area, grouped by subject and input. Q = quaternion input, RM = rotation matrix input

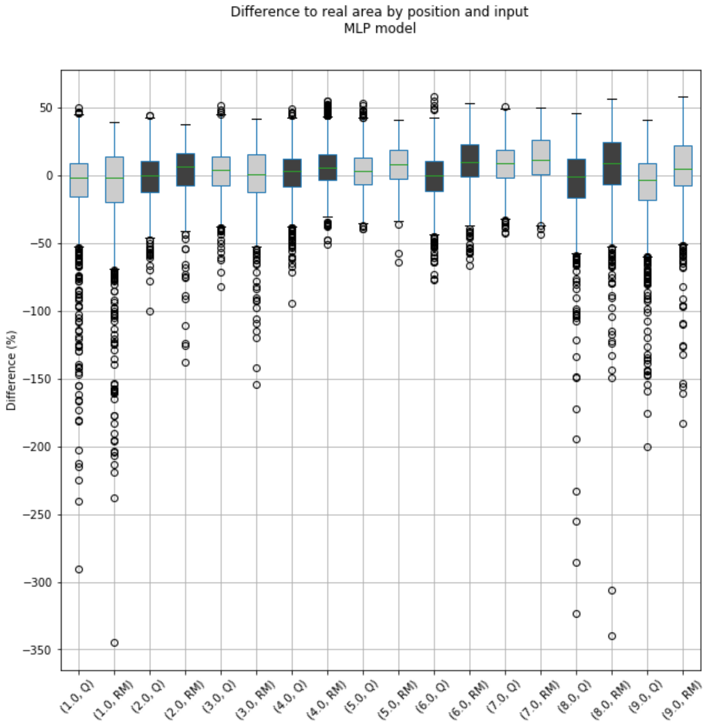


Figure C.4: The difference between the area predicted by the MLP model and the real area, grouped by position and input. Q = quaternion input, RM = rotation matrix input

C.1.2 Predictions over time

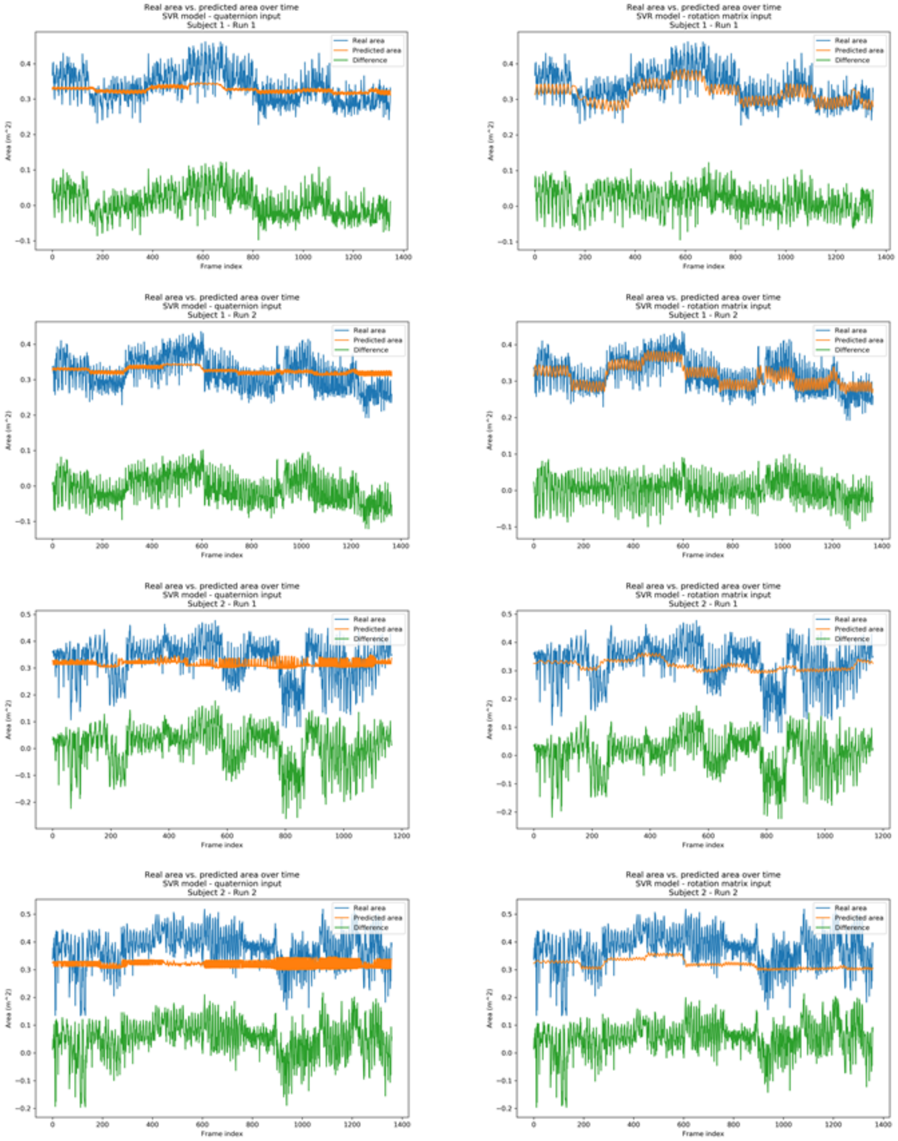


Figure C.5: The real area (blue), the predicted area (orange) and the difference between the two (green) for the SVR model, with quaternion input left and rotation matrix input right.

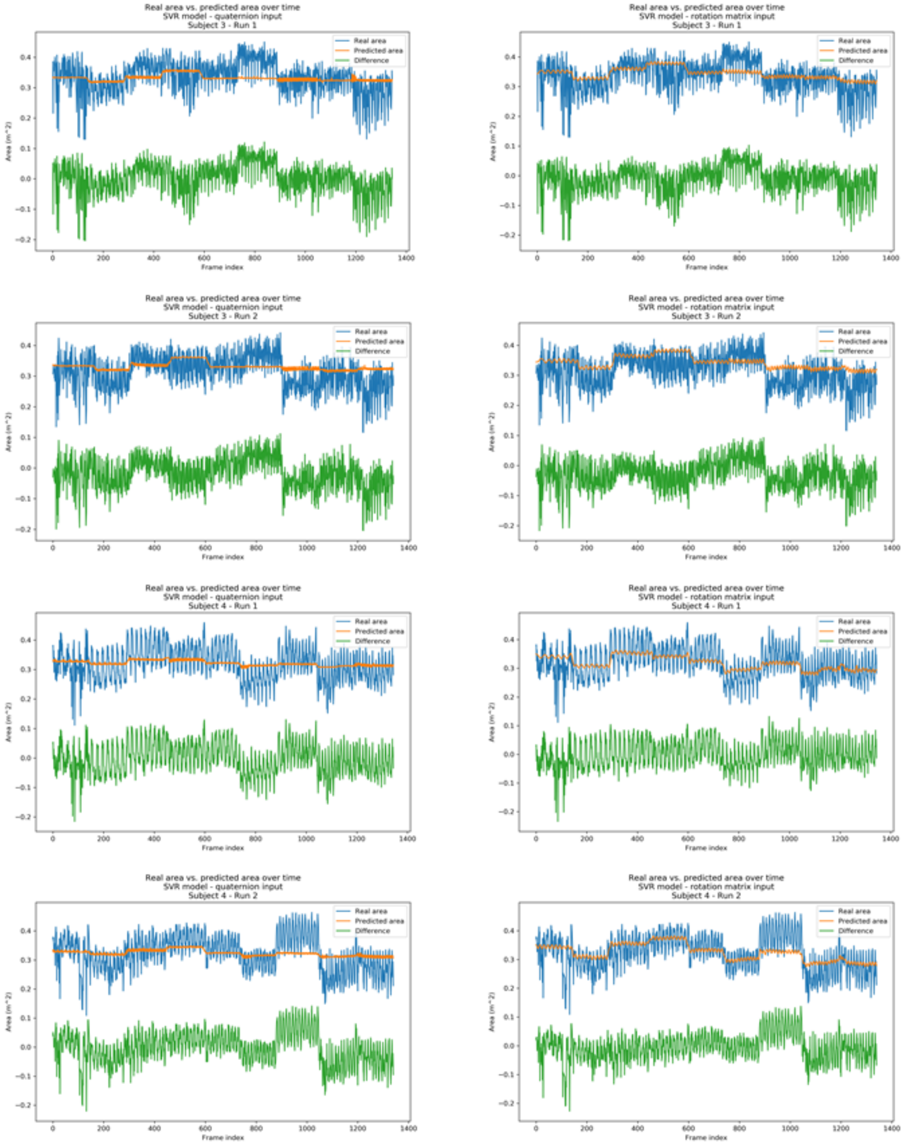


Figure C.6: The real area (blue), the predicted area (orange) and the difference between the two (green) for the SVR model, with quaternion input left and rotation matrix input right.

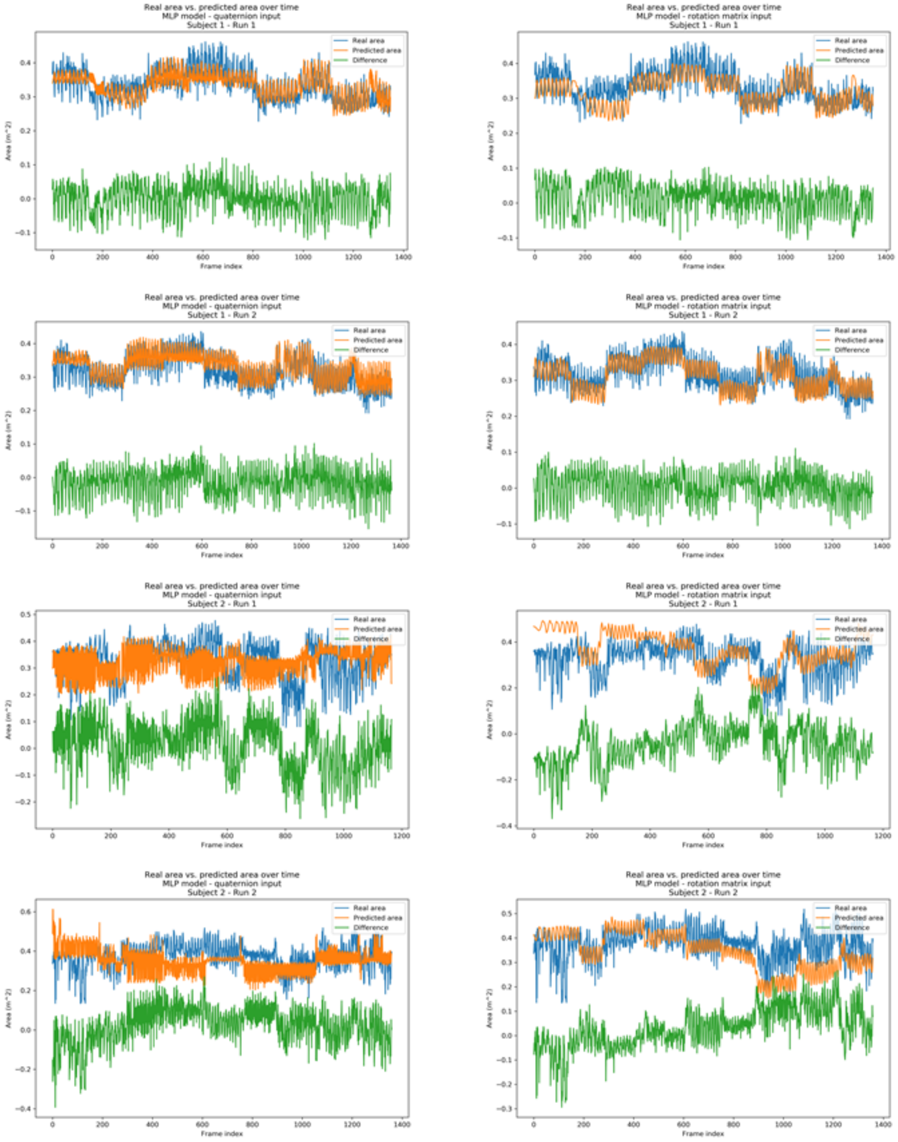
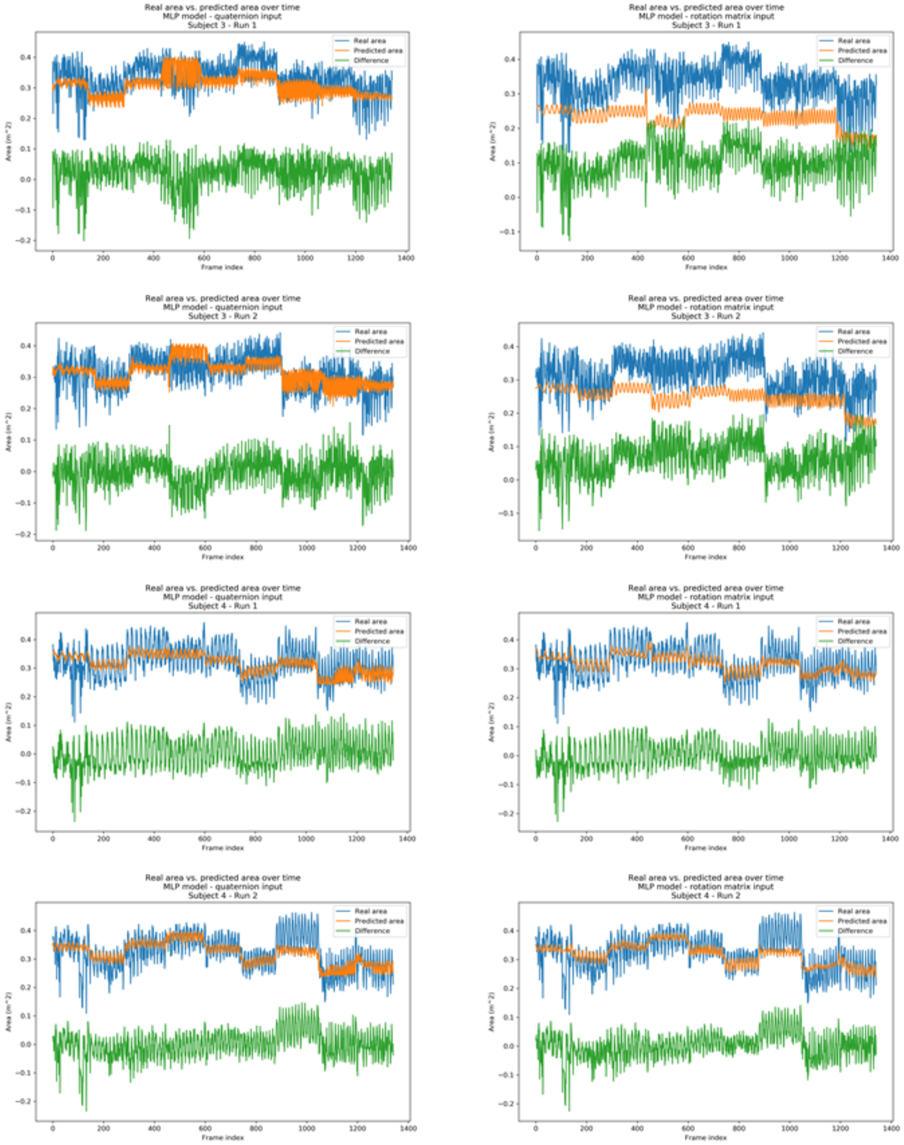


Figure C.7: The real area (blue), the predicted area (orange) and the difference between the two (green) for the MLP model, with quaternion input left and rotation matrix input right.



Real area vs. predicted area over time
MLP model - quaternion input
Subject 3 - Run 1

Real area vs. predicted area over time
MLP model - rotation matrix input
Subject 3 - Run 1

Real area vs. predicted area over time
MLP model - quaternion input
Subject 3 - Run 2

Real area vs. predicted area over time
MLP model - rotation matrix input
Subject 3 - Run 2

Real area vs. predicted area over time
MLP model - quaternion input
Subject 4 - Run 1

Real area vs. predicted area over time
MLP model - rotation matrix input
Subject 4 - Run 1

Real area vs. predicted area over time
MLP model - quaternion input
Subject 4 - Run 2

Real area vs. predicted area over time
MLP model - rotation matrix input
Subject 4 - Run 2

Figure C.8: The real area (blue), the predicted area (orange) and the difference between the two (green) for the MLP model, with quaternion input left and rotation matrix input right.

C.2 Experiment 2

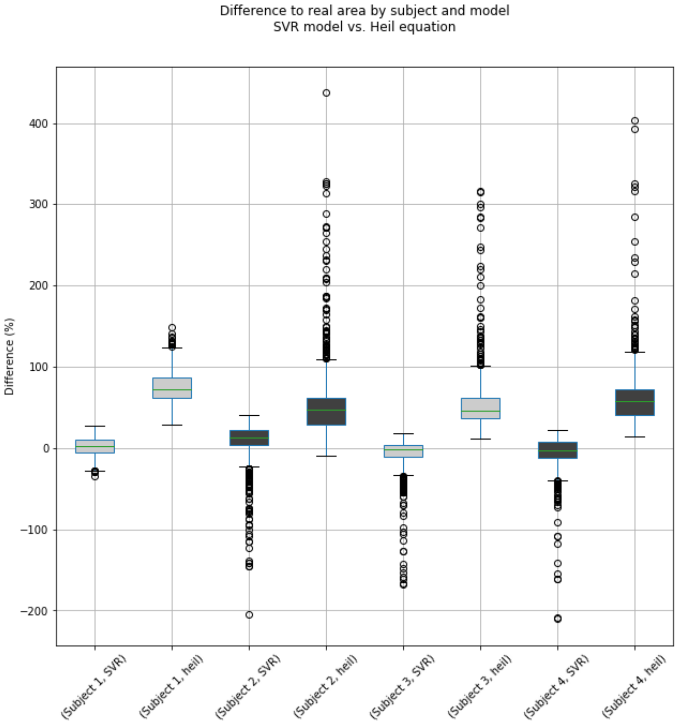


Figure C.9: The difference between the area predicted by the SVR model/Heil model and the real area, grouped by subject and model.

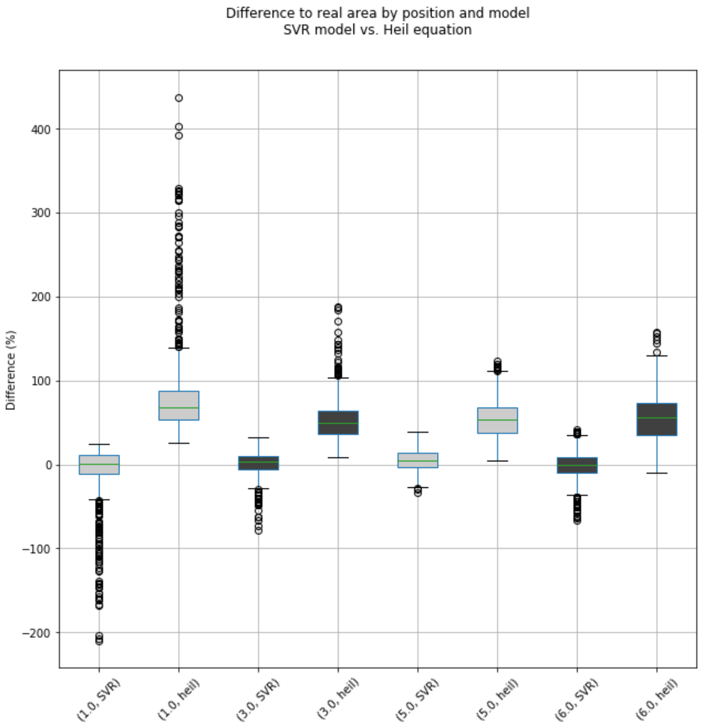


Figure C.10: The difference between the area predicted by the SVR model/Heil model and the real area, grouped by position and model.

C.3 Experiment 3

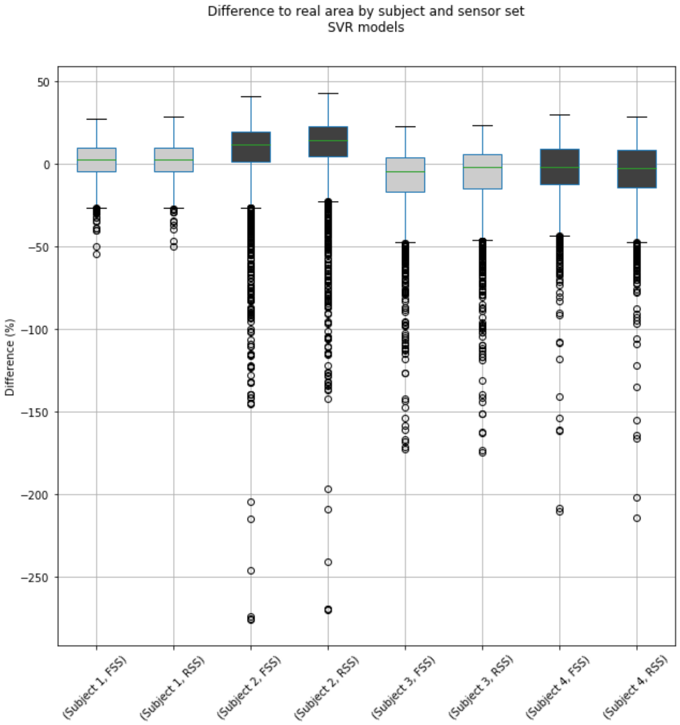


Figure C.11: The difference between the area predicted by the SVR model and the real area, grouped by subject and input sensor set. FSS = full sensor set input, RSS = reduced sensor set input

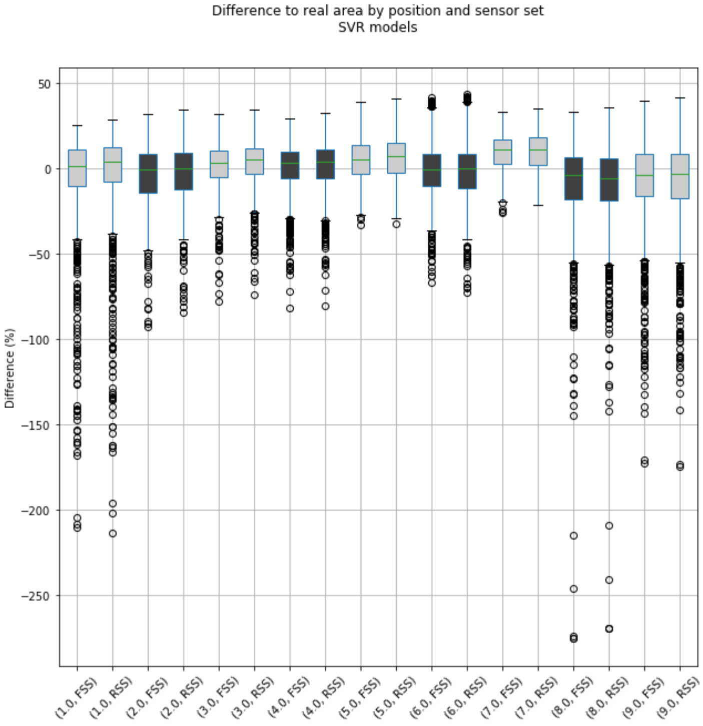


Figure C.12: The difference between the area predicted by the SVR model and the real area, grouped by position and input sensor set. FSS = full sensor set input, RSS = reduced sensor set input

Appendix D

Exp. 3 - Mutual information scores

Table D.1: Mutual information scores of the dataset, grouped by subject and sensor. Some segments use multiple sensors for their orientation (e.g. neck segment, using head and sternum (T8) sensor information).

Sensor	Subject 1	Subject 2	Subject 3	Subject 4	Average
height	0.0185	0.0070	0.0019	0.0245	0.0094
RightLowerLeg	0.0937	0.0542	0.0937	0.1068	0.0784
weight	0.0634	0.0342	0.0899	0.0909	0.0805
RightFoot	0.1091	0.0614	0.1024	0.1185	0.0921
RightToe	0.1091	0.0614	0.1024	0.1185	0.0921
RightUpperLeg	0.1145	0.0741	0.1230	0.1222	0.0993
LeftLowerLeg	0.1144	0.0780	0.1082	0.1200	0.1025
LeftUpperLeg	0.1180	0.0832	0.1076	0.1325	0.1032
LeftToe	0.1136	0.0766	0.1175	0.1294	0.1112
LeftFoot	0.1136	0.0766	0.1175	0.1294	0.1112
RightShoulder	0.1095	0.1082	0.1485	0.1379	0.1146
LeftUpperArm	0.1264	0.1235	0.1559	0.1509	0.1264
LeftShoulder	0.1382	0.1213	0.1502	0.1477	0.1271
Head, T8	0.1439	0.1310	0.1586	0.1543	0.1375
T8	0.1474	0.1358	0.1618	0.1546	0.1386
RightUpperArm	0.1392	0.1328	0.1674	0.1720	0.1393
Pelvis	0.1417	0.1310	0.1637	0.1650	0.1416
Head	0.1491	0.1287	0.1766	0.1825	0.1527
Pelvis, Head, T8	0.1542	0.1503	0.1735	0.1717	0.1538
RightForeArm	0.1712	0.1742	0.1931	0.2006	0.1690
LeftForeArm	0.1743	0.1643	0.2092	0.2003	0.1717
LeftHand	0.1782	0.1785	0.2187	0.2134	0.1814
RightHand	0.1784	0.1811	0.2142	0.2229	0.1846

References

[1] R. B. Candau, F. Grappe, M. Ménard, B. Barbier, G. Y. Millet, M. D. Hoffmann, A. R. Belli, and J. D. Rouillon, "Simplified deceleration method for assessment of resistive forces in cycling," *Medicine & Science in Sports & Exercise*, vol. 31, no. 10, p. 1441, Oct. 1999, ISSN: 0195-9131. DOI: 10.1097/00005768-199910000-00013. [Online]. Available: https://insights.ovid.com/crossref?an=00005768-199910000-00013.

[2] D. Fintelman, M. Sterling, H. Hemida, and F.-X. Li, "Optimal cycling time trial position models: Aerodynamics versus power output and metabolic energy," *Journal of Biomechanics*, vol. 47, no. 8, pp. 1894–1898, Jun. 2014, ISSN: 00219290. DOI: 10.1016/j.jbiomech.2014.02.029. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021929014001407.

[3] J.-T. Zhang, A. C. Novak, B. Brouwer, and Q. Li, "Concurrent validation of Xsens MVN measurement of lower limb joint angular kinematics," *Physiological Measurement*, vol. 34, no. 8, N63–N69, Aug. 2013, ISSN: 0967-3334. DOI: 10.1088/0967-3334/34/8/N63. [Online]. Available: http://stacks.iop.org/0967-3334/34/i=8/a=N63?key=crossref.b8cddbd89301a6c8dae28bde2e307266.

[4] M. Schepers, M. Giuberti, and G. Bellusci, "Xsens MVN : Consistent Tracking of Human Motion Using Inertial Sensing," *Xsens Technologies*, no. March, pp. 1–8, 2018. DOI: 10.13140/RG.2.2.22099.07205.

[5] Xsens Technologies B.V., *Xsens MVN User Manual*, Revision Y, July. Enschede, 2019. [Online]. Available: https://www.xsens.com/hubfs/Downloads/Manuals/MVN_User_Manual.pdf.

[6] F. Wouda, M. Giuberti, G. Bellusci, and P. Veltink, "Estimation of Full-Body Poses Using Only Five Inertial Sensors: An Eager or Lazy Learning Approach?" *Sensors*, vol. 16, no. 12, p. 2138, Dec. 2016, ISSN: 1424-8220. DOI: 10.3390/s16122138. [Online]. Available: http://www.mdpi.com/1424-8220/16/12/2138.

[7] F. J. Wouda, M. Giuberti, G. Bellusci, E. Maartens, J. Reenalda, B.-J. F. van Beijnum, and P. H. Veltink, "Estimation of Vertical Ground Reaction Forces and Sagittal Knee Kinematics During Running Using Three Inertial Sensors," *Frontiers in Physiology*, vol. 9, no. MAR, pp. 1–14, Mar. 2018,

ISSN: 1664-042X. DOI: 10.3389/fphys.2018.00218. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fphys.2018.00218/full.

[8] F. J. Wouda, M. Giuberti, N. Rudigkeit, B.-J. F. van Beijnum, M. Poel, and P. H. Veltink, "Time Coherent Full-Body Poses Estimated Using Only Five Inertial Sensors: Deep versus Shallow Learning," *Sensors*, vol. 19, no. 17, p. 3716, Aug. 2019, ISSN: 1424-8220. DOI: 10.3390/s19173716. [Online]. Available: https://www.mdpi.com/1424-8220/19/17/3716.

[9] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll, "Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–15, Oct. 2018, ISSN: 0730-0301. DOI: 10.1145/3272127.3275108. [Online]. Available: https://dl.acm.org/doi/10.1145/3272127.3275108%20http://arxiv.org/abs/1810.04703.

[10] D. Du Bois and E. Du Bois, "A formula to estimate the approximate surface area if height and weight be known," *Archives of Internal Medicine*, vol. XVII, no. 6_2, pp. 863–871, Jun. 1916, ISSN: 0730-188X. DOI: 10.1001/archinte.1916.00080130010002. [Online]. Available: http://archinte.jamanetwork.com/article.aspx?doi=10.1001/archinte.1916.00080130010002.

[11] B. Shuter and A. Aslani, "Body surface area: Du Bois and Du Bois revisited," *European Journal of Applied Physiology*, vol. 82, no. 3, pp. 250–254, 2000, ISSN: 14396319. DOI: 10.1007/s004210050679.

[12] C. Villa, C. Primeau, U. Hesse, H. P. Hougen, N. Lynnerup, and B. Hesse, "Body surface area determined by whole-body CT scanning: need for new formulae?" *Clinical Physiology and Functional Imaging*, vol. 37, no. 2, pp. 183–193, Mar. 2017, ISSN: 14750961. DOI: 10.1111/cpf.12284. [Online]. Available: http://doi.wiley.com/10.1111/cpf.12284.

[13] A. Kuehnapfel, P. Ahnert, M. Loeffler, and M. Scholz, "Body surface assessment with 3D laser-based anthropometry: reliability, validation, and improvement of empirical surface formulae," *European Journal of Applied Physiology*, vol. 117, no. 2, pp. 371–380, Feb. 2017, ISSN: 1439-6319. DOI: 10.1007/s00421-016-3525-5. [Online]. Available: http://link.springer.com/10.1007/s00421-016-3525-5.

[14] P. Eichelberger, M. Ferraro, U. Minder, T. Denton, A. Blasimann, F. Krause, and H. Baur, "Analysis of accuracy in optical motion capture – A protocol for laboratory setup evaluation," *Journal of Biomechanics*, vol. 49, no. 10, pp. 2085–2088, 2016, ISSN: 18732380. DOI: 10.1016/j.jbiomech.2016.05.007. [Online]. Available: http://dx.doi.org/10.1016/j.jbiomech.2016.05.007.

[15] J. L. McGinley, R. Baker, R. Wolfe, and M. E. Morris, "The reliability of three-dimensional kinematic gait measurements: A systematic review," *Gait and Posture*, vol. 29, no. 3, pp. 360–369, 2009, ISSN: 09666362. DOI: 10.1016/j.gaitpost.2008.09.003.

[16] F. J. Wouda, "Towards Motion Capture with Minimal Sensing," PhD thesis, University of Twente, Enschede, The Netherlands, Feb. 2020, ISBN: 9789036549233. DOI: 10.3990/1.9789036549233. [Online]. Available: http://purl.org/utwente/doi/10.3990/1.9789036549233.

[17] J. C. Martin, D. L. Milliken, J. E. Cobb, K. L. McFadden, and A. R. Coggan, "Validation of a Mathematical Model for Road Cycling Power," *Journal of Applied Biomechanics*, vol. 14, no. 3, pp. 276–291, Aug. 1998, ISSN: 1065-8483. DOI: 10.1123/jab.14.3.276. [Online]. Available: https://journals.humankinetics.com/view/journals/jab/14/3/article-p276.xml.

[18] F. R. Whitt, "A Note on the Estimation of the Energy Expenditure of Sporting Cyclists," *Ergonomics*, vol. 14, no. 3, pp. 419–424, May 1971, ISSN: 0014-0139. DOI: 10.1080/00140137108931261. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/00140137108931261.

[19] P. E. di Prampero, G. Cortili, P. Mognoni, and F. Saibene, "Equation of motion of a cyclist," *Journal of Applied Physiology Respiratory Environmental and Exercise Physiology*, vol. 47, no. 1, pp. 201–206, Jul. 1979, ISSN: 01617567. DOI: 10.1152/jappl.1979.47.1.201. [Online]. Available: https://www.physiology.org/doi/10.1152/jappl.1979.47.1.201.

[20] T. N. Crouch, D. Burton, Z. A. LaBry, and K. B. Blair, "Riding against the wind: a review of competition cycling aerodynamics," *Sports Engineering*, vol. 20, no. 2, pp. 81–110, 2017, ISSN: 1369-7072. DOI: 10.1007/s12283-017-0234-1. [Online]. Available: http://link.springer.com/10.1007/s12283-017-0234-1.

[21] P. Debraux, F. Grappe, A. V. Manolova, and W. Bertucci, "Aerodynamic drag in cycling: Methods of assessment," *Sports Biomechanics*, vol. 10, no. 3, pp. 197–218, Sep. 2011, ISSN: 14763141. DOI: 10.1080/14763141.2011.592209. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/14763141.2011.592209.

[22] F. Grappe, R. Candau, A. Belli, and J. D. Rouillon, "Aerodynamic drag in field cycling with special reference to the Obree's position," *Ergonomics*, vol. 40, no. 12, pp. 1299–1311, Dec. 1997, ISSN: 0014-0139. DOI: 10.1080/001401397187388. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/001401397187388.

[23] J. García-López, A. Ogueta-Alday, J. Larrazabal, and J. A. Rodríguez-Marroyo, "The Use of Velodrome Tests to Evaluate Aerodynamic Drag in Professional Cyclists," *International Journal of Sports Medicine*, vol. 35, no. 5, pp. 451–455, Sep. 2013, ISSN: 0172-4622. DOI: 10.1055/s-0033-1355352. [Online]. Available: http://www.thieme-connect.de/DOI/DOI?10.1055/s-0033-1355352.

[24] J. García-López, J. A. Rodríguez-Marroyo, C.-E. Juneau, J. Peleteiro, A. C. Martínez, and J. G. Villa, "Reference values and improvement of aerodynamic drag in professional cyclists," *Journal of Sports Sciences*, vol. 26, no. 3, pp. 277–286, Feb. 2008, ISSN: 0264-0414. DOI: 10.1080/02640410701501697. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/02640410701501697.

[25] T. Defraeye, B. Blocken, E. Koninckx, P. Hespel, and J. Carmeliet, "Aerodynamic study of different cyclist positions: CFD analysis and full-scale wind-tunnel tests," *Journal of Biomechanics*, vol. 43, no. 7, pp. 1262–1268, May 2010, ISSN: 00219290. DOI: 10.1016/j.jbiomech.2010.01.025. [Online]. Available: http://dx.doi.org/10.1016/j.jbiomech.2010.01.025%20https://linkinghub.elsevier.com/retrieve/pii/S002192901000059X.

[26] ——, "Computational fluid dynamics analysis of cyclist aerodynamics: Performance of different turbulence-modelling and boundary-layer modelling approaches," *Journal of Biomechanics*, vol. 43, no. 12, pp. 2281–2287, 2010, ISSN: 00219290. DOI: 10.1016/j.jbiomech.2010.04.038. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021929010002629.

[27] ——, "Computational fluid dynamics analysis of drag and convective heat transfer of individual body segments for different cyclist positions," *Journal of Biomechanics*, vol. 44, no. 9, pp. 1695–1701, Jun. 2011, ISSN: 00219290. DOI: 10.1016/j.jbiomech.2011.03.035. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021929011002880.

[28] B. Blocken, T. van Druenen, Y. Toparlar, and T. Andrianne, "CFD analysis of an exceptional cyclist sprint position," *Sports Engineering*, vol. 22, no. 1, p. 10, Mar. 2019, ISSN: 1369-7072. DOI: 10.1007/s12283-019-0304-7. [Online]. Available: http://link.springer.com/10.1007/s12283-019-0304-7.

[29] W. Terra, A. Sciacchitano, and F. Scarano, "Aerodynamic drag of a transiting sphere by large-scale tomographic-PIV," *Experiments in Fluids*, vol. 58, no. 7, pp. 1–14, Jul. 2017, ISSN: 07234864. DOI: 10.1007/s00348-017-2331-0. [Online]. Available: http://link.springer.com/10.1007/s00348-017-2331-0.

[30] W. Terra, A. Sciacchitano, and Y. Shah, "Aerodynamic drag determination of a full-scale cyclist mannequin from large-scale PTV measurements," *Experiments in Fluids*, vol. 60, no. 2, p. 29, Feb. 2019, ISSN: 0723-4864. DOI: 10.1007/s00348-019-2677-6. [Online]. Available: http://dx.doi.org/10.1007/s00348-019-2677-6%20http://link.springer.com/10.1007/s00348-019-2677-6.

[31] A. Spoelstra, L. de Martino Norante, W. Terra, A. Sciacchitano, and F. Scarano, "On-site cycling drag analysis with the Ring of Fire," *Experiments in Fluids*, vol. 60, no. 6, p. 90, Jun. 2019, ISSN: 0723-4864. DOI: 10.1007/s00348-019-2737-y. [Online]. Available: https://doi.org/

10.1007/s00348-019-2737-y%20http://link.springer.com/10.1007/s00348-019-2737-y.

[32] T. S. Olds and S. Olive, “Methodological considerations in the determination of projected frontal area in cyclists,” *Journal of Sports Sciences*, vol. 17, no. 4, pp. 335–345, Jan. 1999, ISSN: 0264-0414. DOI: 10.1080/026404199366046. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/026404199366046.

[33] P. Debraux, W. Bertucci, A. Manolova, S. Rogier, and A. Lodini, “New Method to Estimate the Cycling Frontal Area,” *International Journal of Sports Medicine*, vol. 30, no. 04, pp. 266–272, Apr. 2009, ISSN: 0172-4622. DOI: 10.1055/s-0028-1105940. [Online]. Available: http://www.thieme-connect.de/DOI/DOI?10.1055/s-0028-1105940.

[34] D. P. Swain, J. R. Coast, P. S. Clifford, M. C. Milliken, and J. Stray-Gundersen, “Influence of body size on oxygen consumption during bicycling,” *Journal of Applied Physiology*, vol. 62, no. 2, pp. 668–672, Feb. 1987, ISSN: 8750-7587. DOI: 10.1152/jappl.1987.62.2.668. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/3558226%20https://www.physiology.org/doi/10.1152/jappl.1987.62.2.668.

[35] D. P. Heil, “Body mass scaling of projected frontal area in competitive cyclists,” *European Journal of Applied Physiology*, vol. 85, no. 3-4, pp. 358–366, Aug. 2001, ISSN: 1439-6319. DOI: 10.1007/s004210100424. [Online]. Available: http://link.springer.com/10.1007/s004210100424.

[36] ——, “Body mass scaling of frontal area in competitive cyclists not using aero-handlebars,” *European Journal of Applied Physiology*, vol. 87, no. 6, pp. 520–528, Oct. 2002, ISSN: 1439-6319. DOI: 10.1007/s00421-002-0662-9. [Online]. Available: http://link.springer.com/10.1007/s00421-002-0662-9.

[37] C. Barelle, V. Chabroux, and D. Favier, “Modeling of the time trial cyclist projected frontal area incorporating anthropometric, postural and helmet characteristics,” *Sports Engineering*, vol. 12, no. 4, pp. 199–206, Aug. 2010, ISSN: 1369-7072. DOI: 10.1007/s12283-010-0047-y. [Online]. Available: http://link.springer.com/10.1007/s12283-010-0047-y.

[38] A. Drory, H. Li, and R. Hartley, “Estimating the projected frontal surface area of cyclists from images using a variational framework and statistical shape and appearance models,” *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, vol. 231, no. 3, pp. 169–183, Sep. 2017, ISSN: 1754-3371. DOI: 10.1177/1754337117705489. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1754337117705489.

[39] Glosser.ca, *Colored neural network.svg*, 2013. [Online]. Available: https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg.

[40] R. Berwick, *An Idiot's guide to Support vector machines (SVMs)*, Cambridge, MA, 2003. [Online]. Available: http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf.

[41] Cyc, *Svm separating hyperplanes.png*, 2008. [Online]. Available: https://commons.wikimedia.org/wiki/File:Svm_separating_hyperplanes.png.

[42] S. Sayad, *Support Vector Machine - Regression (SVR)*. [Online]. Available: https://www.saedsayad.com/support_vector_machine_reg.htm.

[43] scikit-learn developers, *RBF SVM parameters*, 2020. [Online]. Available: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.

[44] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.161. [Online]. Available: http://ieeexplore.ieee.org/document/5557884/.

[45] N. M. Zaitoun and M. J. Aqel, "Survey on Image Segmentation Techniques," *Procedia Computer Science*, vol. 65, no. Iccmit, pp. 797–806, 2015, ISSN: 18770509. DOI: 10.1016/j.procs.2015.09.027. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2015.09.027%20https://linkinghub.elsevier.com/retrieve/pii/S1877050915028574.

[46] S. S. Chouhan, A. Kaul, and U. P. Singh, "Image Segmentation Using Computational Intelligence Techniques: Review," *Archives of Computational Methods in Engineering*, vol. 26, no. 3, pp. 533–596, Jul. 2019, ISSN: 1134-3060. DOI: 10.1007/s11831-018-9257-4. [Online]. Available: https://doi.org/10.1007/s11831-018-9257-4%20http://link.springer.com/10.1007/s11831-018-9257-4.

[47] K. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, Jan. 1981, ISSN: 00313203. DOI: 10.1016/0031-3203(81)90028-5. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0031320381900285.

[48] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2004.60. [Online]. Available: http://ieeexplore.ieee.org/document/1316848/.

[49] P. Mesejo, Ó. Ibáñez, Ó. Cordón, and S. Cagnoni, "A survey on image segmentation using metaheuristic-based deformable models: state of the art and critical analysis," *Applied Soft Computing*, vol. 44, pp. 1–29, Jul. 2016, ISSN: 15684946. DOI: 10.1016/j.asoc.2016.03.004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494616301053.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015, ISSN: 16113349. DOI: `10.1007/978-3-319-24574-4{\_}28`.

[51] X. Xia and B. Kulis, "W-Net: A Deep Model for Fully Unsupervised Image Segmentation," Nov. 2017. [Online]. Available: `http://arxiv.org/abs/1711.08506`.

[52] A. Kanezaki, "Unsupervised Image Segmentation by Backpropagation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Apr. 2018, pp. 1543–1547, ISBN: 978-1-5386-4658-8. DOI: `10.1109/ICASSP.2018.8462533`. [Online]. Available: `https://ieeexplore.ieee.org/document/8462533/`.

[53] S. Saha, S. Sudhakaran, B. Banerjee, and S. Pendurkar, "Semantic Guided Deep Unsupervised Image Segmentation," in *Image Analysis and Processing - ICIAP 2019*, E. Ricci, S. Rota Bulò, C. Snoek, O. Lanz, S. Messelodi, and N. Sebe, Eds., vol. 11752, Cham, 2019, pp. 499–510, ISBN: 9783030306441. DOI: `10.1007/978-3-030-30645-8{\_}46`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-030-30645-8_55%20http://link.springer.com/10.1007/978-3-030-30645-8_46`.

[54] C. Rother, V. Kolmogorov, T. Minka, and A. Blake, "Cosegmentation of image pairs by histogram matching - Incorporating a global constraint into MRFs," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 994–1000, 2006, ISSN: 10636919. DOI: `10.1109/CVPR.2006.91`.

[55] H. Xu, G. Lin, and M. Wang, "A Review of Recent Advances in Image Co-Segmentation Techniques," *IEEE Access*, vol. 7, pp. 182 089–182 112, 2019, ISSN: 2169-3536. DOI: `10.1109/ACCESS.2019.2960162`. [Online]. Available: `https://ieeexplore.ieee.org/document/8933459/`.

[56] W. Li, O. Hosseini Jafari, and C. Rother, "Deep Object Co-Segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11363 LNCS, no. 2, pp. 638–653, Apr. 2018, ISSN: 16113349. DOI: `10.1007/978-3-030-20893-6{\_}40`. [Online]. Available: `http://arxiv.org/abs/1804.06423`.

[57] S. Banerjee, A. Hati, S. Chaudhuri, and R. Velmurugan, "CoSegNet: Image Co-segmentation using a Conditional Siamese Convolutional Network," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, California: International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 673–679, ISBN: 978-0-9992411-4-1. DOI: `10.24963/ijcai.2019/95`. [Online]. Available: `https://www.ijcai.org/proceedings/2019/95`.

[58] Sewaqu, *Linear regression.svg*, 2010. [Online]. Available: `https://commons.wikimedia.org/wiki/File:Linear_regression.svg`.

[59] X. Liu and X. Duan, “Automatic Image Co-Segmentation: A Survey,” pp. 1–12, Nov. 2019. [Online]. Available: `http://arxiv.org/abs/1911.07685`.

[60] J. Frost, *R-squared Is Not Valid for Nonlinear Regression.* [Online]. Available: `https://statisticsbyjim.com/regression/r-squared-invalid-nonlinear-regression`.

[61] M. Michailidis, *Regression Metrics’ Guide*, 0. [Online]. Available: `https://www.h2o.ai/blog/regression-metrics-guide/`.

[62] W. Li, O. Hosseini Jafari, and C. Rother, *Deep Object Co-Segmentation (DOCS) - PyTorch*, 2018. [Online]. Available: `https://github.com/ohosseini/DOCS-pytorch`.

[63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[64] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[65] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, and P. van Mulbregt, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, ISSN: 1548-7091. DOI: `10.1038/s41592-019-0686-2`. [Online]. Available: `http://www.nature.com/articles/s41592-019-0686-2`.