Detecting skier pose on treadmill by machine learning

Juha Hulkkonen

Master's Thesis

UNIVERSITY OF
EASTERN FINLAND

School of Computing

Computer Science

April 2022

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Joensuu
School of Computing
Computer Science

Student, Juha Hulkkonen: Detecting skier pose on treadmill by machine learning
Master’s Thesis, 69 p.
Supervisors of the Master’s Thesis: Prof. Pasi Fränti, D.Sc (Tech) Markus Koskela
April 2022

Abstract: Motion capture is a growing and significant field of research with many applications in sports. Exercise technology research helps top athletes and enthusiasts improve their performance, and on the other hand, it also helps scientists to understand human body activity during exercise performance. My thesis explores whether the traditional reflective marker based motion capture systems used in the skiing research, could be replaced by a video camera and machine vision algorithm in the pose estimation of an athlete skiing on a treadmill. The research environment is the University of Jyväskylä's Sport Technology Unit and its VICON motion capture system located in the skiing laboratory in Vuokatti. The work will shed light into the uses of motion capture, its implementation and the use of machine learning in motion capture. For the practical part of the work, a video recording event was held in Vuokatti to collect data from skiers of different levels. This data was used to form training data for the machine learning algorithms. Three different data sets were created for training, the first of which identifiable joint points were manually marked for each frame image. The joint points of the second data set were produced algorithmically by calibrating the 3D joint points of point cloud data produced by VICON into a two-dimensional view. A third, smaller than the previous, data set was created for the calibration algorithm. General-purpose models produced by AlphaPose pose estimation algorithms were fine tuned using self created data sets, and finally models trained using different data sets were compared with each other. The accuracy of existing pose estimation models was improved by fine tuning the models. However, the accuracy of the models produced was not good enough to replace VICON. More research is needed on the subject. The creation and calibration of training data rose to play a major role in the research. The summary will go through the lessons learned during the work, and what should be taken into account in future studies. My thesis has been conducted in collaboration with my employer CSC - IT Center for Sciences and the Department of Sport Technology of the University of Jyväskylä in the CEMIS Consortium HYTELI project funded by the Kainuun Liitto, the European Regional Development Fund, and municipality funding from Kajaani and Sotkamo.

Keywords: pose estimation, machine vision, machine learning, sports science, cross-country skiing

CR Categories (ACM Computing Classification System, 1998 version): I.2.10 Vision and Scene Understanding

ITÄ-SUOMEN YLIOPISTO, Luonnontieteiden ja metsätieteiden tiedekunta, Joensuu
Tietojenkäsittelytieteen laitos
Tietojenkäsittelytiede

Opiskelija, Juha Hulkkonen: Detecting skier pose on treadmill by machine learning
Pro gradu –tutkielma, 69 s.
Pro gradu –tutkielman ohjaajat: Prof. Pasi Fränti, TkT Markus Koskela
Huhtikuu 2022

Tiivistelmä: Liikkeentunnistus on kasvava ja merkittävä tutkimusala, jolla on monia sovelluksia urheilussa. Liikuntateknologinen tutkimus auttaa huippu-urheilijoita ja harrastajia parantamaan suorituskykyään ja toisaalta se auttaa myös tutkijoita ymmärtämään ihmisekehon toimintaa liikuntasuorituksen aikana. Tutkielmassani selvitetään, voisiko hiihtotutkimuksessa käytettyjä perinteisiä, heijastaviin markkereihin perustuvia liikkeenkaappausjärjestelmiä korvata videokameralla ja konenäköalgoritmilla rullasuksilla hiihtomatolla hiihtävän urheilijan asennontunnistuksessa. Tutkimuskohteena on Jyväskylän yliopiston Liikuntateknologian yksikön Vuokatissa sijaitseva hiihtolaboratorio ja sen VICON-liikkeenkaappausjärjestelmä. Työssä tutustutaan liikkeenkaappauksen käyttötarkoituksiin, sen toteuttamistapoihin ja koneoppimisen hyödyntämiseen liikkeenkaappauksessa. Työn käytännön osuutta varten Vuokatissa järjestettiin kuvaustapahtuma, jossa kerättiin dataa eritasoisilta hiihtäjiltä. Tästä datasta muodostettiin koulutusdataa työssä käytetyille koneoppimisalgoritmeille. Koulutusta varten luotiin kolme eri datajoukkoa, joista ensimmäiseen tunnistettavat nivelpisteet merkittiin käsin kuvaruutu kerrallaan. Toisen datajoukon nivelpisteet tuotettiin algoritmisesti kalibroimalla VICONin tuottaman pistepilvidatan 3D-nivelpisteet kaksiulotteiseen näkymään. Kolmas, edellisiä pienempi, datajoukko luotiin kalibrointialgoritmia varten. Työssä jatkokoulutettiin yleiskäyttöisiä AlphaPose-asennontunnistusalgoritmilla tuotettuja malleja itse luoduilla datajoukoilla ja lopuksi vertailtiin eri datajoukkojen avulla koulutettuja malleja keskenään. Jatkokouluttamalla olemassa olevia asennontunnistusmalleja asennontunnistuksen tarkkuutta saatiin parannettua. Tuotettujen mallien tarkkuus ei kuitenkaan ollut riittävän hyvä, jotta niillä voisi korvata VICONin. Aiheesta tarvitaan lisää tutkimusta. Koulutusdatan luominen ja kalibrointi nousivat merkittävään rooliin tutkimuksessa. Yhteenvedossa käydään läpi työssä opittuja, tulevissa tutkimuksissa huomioon otettavia asioita. Tutkielmani on tehty yhteistyössä työnantajani CSC - Tieteen tietotekniikan keskus Oy:n ja Jyväskylän Yliopiston Liikuntateknologian yksikön kanssa Kainuun Liiton, Euroopan aluekehitysrahaston sekä Kajaanin ja Sotkamon kuntien rahoittamassa CEMIS-konsortion HYTELI-hankkeessa.

Avainsanat: asennontunnistus, konenäkö, koneoppiminen, urheilutieteet, hiihto

ACM-luokat (ACM Computing Classification System, 1998 version): I.2.10 Vision and Scene Understanding

Foreword

This thesis is made for the University of Eastern Finland School of Computing during the academic years 2020-2021 and 2021-2022.

Covid-19 pandemic delayed this work like many others in 2020 and 2021. Video recording session was moved, meetings were cancelled and moved to online platforms. Asking for help and support changed to more difficult since you couldn't just go, show and ask help with your problems.

Thank you Professor Pasi Fränti for your support and guidance and also for understanding the pressure and stress which may occur when combining remote work and remote studies during this challenging pandemic time. Thanks to my manager Aleksi Kallio and colleagues Katja Mankinen and Markus Koskela for great support and suggestions for improvement! This thesis would never have been completed without you.

I recently watched a video lecture about machine learning by Professor Andrew Ng (2021). He was talking about his view of the shift from model-centric to data-centric machine learning. In his speech Prof. Ng referred to the old saying that 80% of data science projects are about cleaning and preparing data. After this thesis work I can sincerely sign that statement. My work wasn't about developing machine learning algorithms, but to utilise and fine tune existing ones. Still the amount of work and encountered issues when cleaning and preparing data for fine tuning the existing models were very much more than I expected and how much I initially scheduled time for it when planning this project.

In his speech Prof. Ng also stated that according to observations of his research group, improving training dataset quality can often yield the same benefit as doubling the amount of data. Even though that observation was not the result of any specific, scientific research, I dare to assume that learning data cleaning and manipulating skills will be an extremely beneficial asset in the world of ever growing data. These skills really improved during this thesis work.

Kajaani 12.4.2022 Juha Hulkkonen

List of Abbreviations

2D	Two dimensional
3D	Three dimensional
AVI	Audio Video Interleave
CNN	Convolutional neural network
COM	Center of Mass
CSC	CSC - IT Center for Science Ltd.
CSV	Comma-Separated Values, file format
DLT	Discrete Linear Transform
Fc	Propulsion force with Göpfert et al. (2017) model
Fpropulsive	Propulsion force
Fr	Resultant force
Fro	Rotational force
Ft	Translational force
GPU	Graphic Processing Unit
H.264	Video codec
HAKE	Human Activity Knowledge Engine
HRnet	High-Resolution Network
JSON	JavaScript Object Notation
JYU	University of Jyväskylä in Finland
LED	Light Emitting Diode
mAP	mean average precision
MOV	Abbreviation for QuickTime File Format by Apple
MP4	Extension format for QuickTime File Format
MPII	Human Pose dataset, collected from human curated YouTube videos
MS COCO	Microsoft Common Objects in Context dataset

NMS	Parametric Pose Non-Maximum-Suppression
PC	Personal Computer
PFA	Point of Force Application
PGPG	Posed-Guided Proposals Generator
ResNet	Residual Network, form of convolutional neural network
RGB	Red Green Blue
RMPE	Regional Multi-person Pose Estimation
SPPE	Single-Person Pose Estimator
SSTN	Symmetrical Spatial Transformer Network
VFR	Variable Frame Rate
VICON	Motion capture system by Vicon Motion Systems Ltd.
YAML	Yet Another Markup Language or YAML Ain’t Markup Language
mAP	mean average precision
MPII dataset	Human Pose dataset, collected from human curated YouTube videos. Maintained by Max Planck Institute[1]
MS COCO dataset	Microsoft Common Objects in Context dataset commonly used for training and validating machine vision algorithms
VICON	VICON motion capture system used in Vuokatti skiing laboratory. Created by Vicon Motion Systems Ltd [2]

[1] http://human-pose.mpi-inf.mpg.de/

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

Table of content

1 Introduction

This thesis was created in a cooperation project between CEMIS consortium members CSC - IT center for science (CSC) and the Sports Technology department of University of Jyväskylä (JYU) in Vuokatti, Kainuu, Finland. As a specialist working in CSC, the author's role in the project was to study possibilities to utilise machine vision in skiing coaching and research. Jyväskylä University is doing research in nordic snow sports in their laboratory in Vuokatti and cross country skiing is one of their subjects. The laboratory has a motion capture system created by Vicon Motion Systems Ltd. That system is later referred to as *VICON,* which uses 8 infra-red cameras to track the small reflective markers attached to subjects' joints and to collect the location information in three dimensional space. That data is later referred to as *VICON data*. Researchers use the data to detect skiers body position when the subject is skiing on a treadmill. The body position information is later used in coaching applications and scientific studies e.g. by calculating the forces skier applies to the treadmill. These forces are calculated from the body position and speed information and the skier's technique is fine tuned with the help of the coach to optimise the skiing performance. The body position as a set of joint location coordinates is later referred to as *pose*.

VICON is a precise instrument but is expensive and difficult to operate. Attaching the reflective markers is a time consuming task and a specialist is needed to attach them to correct spots. A specialist is also needed to operate the system during measurement.

In a former collaborative project between CSC and University of Jyväskylä it was discovered that machine vision algorithms can be used to estimate skier pose from video recorded with an ordinary video camera. One such algorithm is *AlphaPose*[3]

[3] https://www.mvig.org/research/alphapose.html (4.1.2021)

(Fang et al. 2017). As such, the precision was however not sufficient compared to VICON and further research is needed in this topic. This thesis addressed needs and studies whether the existing models can be fine tuned to increase performance.

The main research question of this work was to find out if there is a way to replace the VICON motion capture system with an ordinary video camera and a machine vision algorithm in context of skier pose estimation when skiing on a treadmill in a laboratory environment. In the current setup, detecting the skier pose is a difficult process. Even though VICON is a precise tool, it is difficult and time-consuming to set up and operate, which hinders the collection of skier pose data during daily operation of the skiing laboratory. We studied the possibility to streamline this process by finding alternative ways to accomplish skier pose estimation with a normal video camera and machine learning algorithms. Since we had in an earlier project benchmarked the existing models and found out that the resulting pose estimation accuracy was not high enough to challenge VICON, our approach was to create a custom dataset to fine tune those existing models to gain better results. An easier and faster measurement process was originally requested by the personnel using the skiing laboratory for research and coaching.

Other issues to be addressed in this thesis are:

- How to fine-tune an existing open source pose estimation algorithm to better work in pose estimation when skiing on a treadmill
- How to create a training dataset by utilising accurate VICON data
- How to compare pose estimation algorithm output to VICON data
- How to decide whether the developed model is good enough to fulfil customers needs in the skier pose estimation task compared to VICON

The work started with a data collection event where we had four volunteer skiers with roller skis according to a custom test protocol on top of a treadmill. We used VICON to collect so-called ground truth data and a video camera to collect data for machine vision training. We had to use reflective markers when recording the training data because of VICON data collection, even the markers can possibly alter algorithm performance. The markers were essential to collect VICON data from the same skiing runs as the video camera recording.

This thesis consists of 9 main chapters between Introduction and Conclusion.

Chapter 2 sheds light on the background of this research by explaining the partners involved, introducing the skiing laboratory, what is the location this work is focused on and how the pose estimation fits into the field of cross-country skiing research.

Chapter 3 dives into the topic of pose estimation, problems with traditional methods and then introduces how machine vision can be utilised for pose estimation tasks.

Chapter 4 is about relevant machine vision algorithms and the kind of datasets used when applying machine vision to pose estimation. The AlphaPose algorithm used in this work is introduced and a short introduction to other similar algorithms and applications is given.

The description of the practical part of this thesis starts in Chapter 5. First we go through the steps for gathering the data. In Chapter 6 we discuss the essential steps performed during the data pre-processing phase. This chapter includes the important Section 6.5 where we discuss the challenges encountered during pre-processing and how those affected the thesis.

Chapter 7 is about the model training part of the work and Chapter 8 describes the performance evaluation and the results of the experiments. In Chapter 9 we discuss the findings, how well the research questions were answered during the work and give suggestions for future research in this topic.

2 Background of the sports research in Vuokatti

This chapter gives a short introduction of sports science, and how it is studied at the Sports Technology department of Jyväskylä University located in Vuokatti. Section 2.2 describes the skiing laboratory which is the environment where the data collection for this work was conducted and for which the machine vision system developed in this work is targeted. Section 2.3. is about a deeper explanation of the reason behind this work. If the studied machine vision method turns out to be precise enough to fulfil customers needs, it can be utilised in many use cases in the laboratory to analyse skier pose to improve performance instead of the VICON motion capture system.

2.1 Sport science

Sports science is a field of study aiming to maximise performance and endurance of an athlete. At the same time it studies ways to reduce the risk of injury. It applies the principles of science to sporting activities, like nordic ski sports. Sports science is a multidisciplinary field containing studies such as exercise physiology, biomechanics, motor control and motor development, exercise and sport psychology and combinations of those. The field of study is approached with a close collaboration with the athletes in a way that both are benefiting from the symbiosis. The researcher has a subject to study and the athlete can gain improvement in his or her performance from the results of the research[4]. One example of this is the training of cross-country skiing in the Vuokatti skiing laboratory where the coach can guide the athlete through a training routine while researchers are studying the biomechanics of the skier. The effectiveness of the skiing can be measured and two training runs can be compared. The researchers can get valuable data from real world use cases while athletes can get information about the performance and effect of different techniques.

[4] https://ssep.com.au/what-is-sport-science/ (12.3.2022)

2.2 Skiing laboratory in Vuokatti

Jyväskylä University has a Sports Technology department in Vuokatti. The department is specialised in multidisciplinary and applied sports biology research. There are also masters and doctoral schools in Vuokatti. The Nordic Ski Sports laboratory in Vuokatti is focused for applied and technology research in the areas of cross country skiing, ski jump and biathlon. The researchers are able to study nordic snow sports in an advanced environment and in close cooperation with the coaches and athletes. There is also the Vuokatti-Ruka Sports Academy that trains young competitive athletes. In a laboratory environment the coach and the athlete can produce real world data for researchers to study with the latest computational methods. There is a skiing treadmill and a skiing tunnel where one can ski on a real snow track around the year.

The skiing laboratory contains an advanced skiing treadmill that is used for athlete training and scientific studies. The treadmill can be tilted to simulate steep hills and its speed can be adjusted quickly and precisely. The athlete can monitor his or her speed and other environmental data from a screen in front of the treadmill. The treadmill can be programmed to follow the profile of some known race track while recorded video or an animation of that particular track is shown on screen. Figure 1 is showing a subject skiing on the treadmill with 8° inclination during data collection for this thesis work. The faces of operating personnel in the background are blurred in this and forthcoming figures.

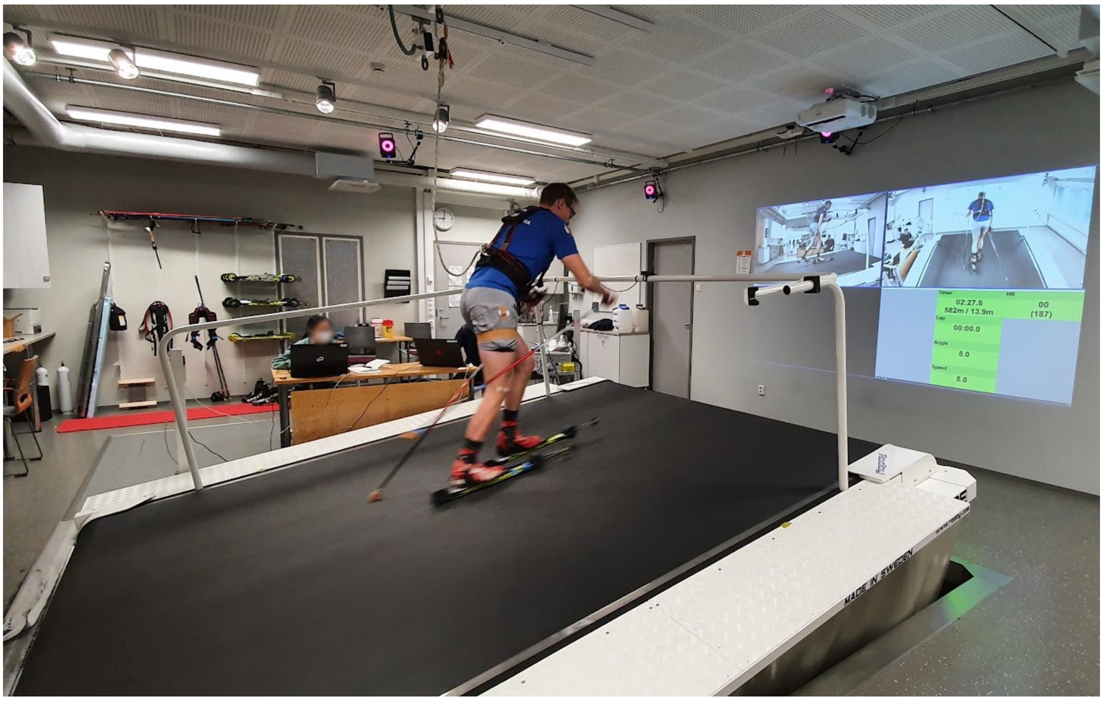


Figure 1: Athlete skiing on Vuokatti skiing laboratory treadmill during data collection event. Treadmill is set to 8° inclination.

In addition to Vuokatti, nordic snow sports are studied in Paris Lodron University Salzburg (PLUS) in Austria[5]. Norway is also a major player in nordic ski sports, and research in this field is done in several Norwegian universities, such as the Norwegian University of Science and Technology (NTNU) in Trondheim.[6]

2.3 Using pose estimation to improve skiing performance

As a CSC employee I was tasked with participating in the *HYTELI*[7] project as a machine learning consultant. HYTELI was a cooperation project between the CEMIS consortium members: Jyväskylä University, Oulu University, Kajaani University of Applied Sciences, Technology Research Center VTT and CSC - It Center for Science. The project was funded by Kainuun Liitto, European Regional Development Fund, and municipality funding from Kajaani and Sotkamo. The target

[5] https://www.plus.ac.at/research/plus/?lang=en (12.3.2022)
[6] https://www.ntnu.edu/inb (12.3.2022)
[7] https://www.jyu.fi/sport/fi/liikuntateknologia/hankkeet/hyteli (17.12.2021)

of the project was to develop advanced innovation platforms and environments to increase regional competence in technology. The aim of our work package was to increase knowledge of participants on machine learning and create a pilot for applying machine learning to skiing research.

The purpose of this work was to create a generic machine learning solution that could be used for any competitive skier in the skiing laboratory. The AlphaPose[8] machine vision algorithm was used for detecting skier body position from video camera recordings. AlphaPose was chosen because it was found to be the most promising in a former collaborative project between CSC and University of Jyväskylä where various algorithms for this task were compared. For this reason the algorithm was familiar to both research partners and it was observed that further research to improve its precision in skier pose estimation was needed.

Ohtonen (2019) proposed that the effectiveness of cross-country skiers can be analysed with a novel propulsion component analysis method (Göpfert 2017) on treadmill skiing with motion capture equipment. Göpfert et al. used a VICON motion capture system at the Vuokatti Ski tunnel to detect skier joint locations in three-dimensional (3D) space. The joint locations were used to calculate skiers Center of Mass (COM). The propulsion components that are used to calculate propulsion forces are illustrated in figure 2. Based on COM and force sensors installed in ski bindings and ski poles, the propulsion forces that affect the acceleration of the skier can be calculated. The force sensors were installed in custom-made ski bindings (Figure 3) made in the Neuromuscular Research Center, University of Jyväskylä, Finland. The sensors measure the front and rear foot forces while skiing and thus provide directional force data from the appropriate directions.

[8] https://www.mvig.org/research/alphapose.html (4.1.2021)

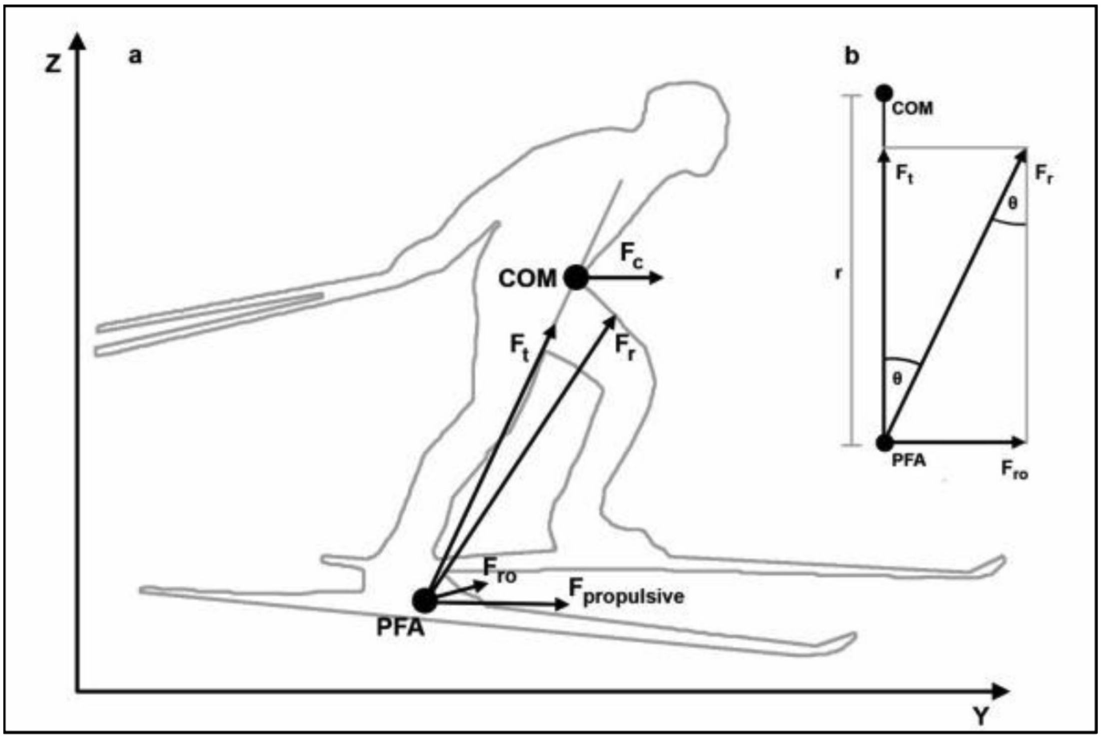


Figure 2: Propulsion components used to calculate propulsion forces. COM means center of mass, PFA means point of force application. Fc, propulsion force with Göpfert et al. (2017) model; Fro, rotational force; Fr, resultant force; Ft, translational force; Fpropulsive, propulsion force calculated with earlier methods. (Ohtonen et al. 2020)

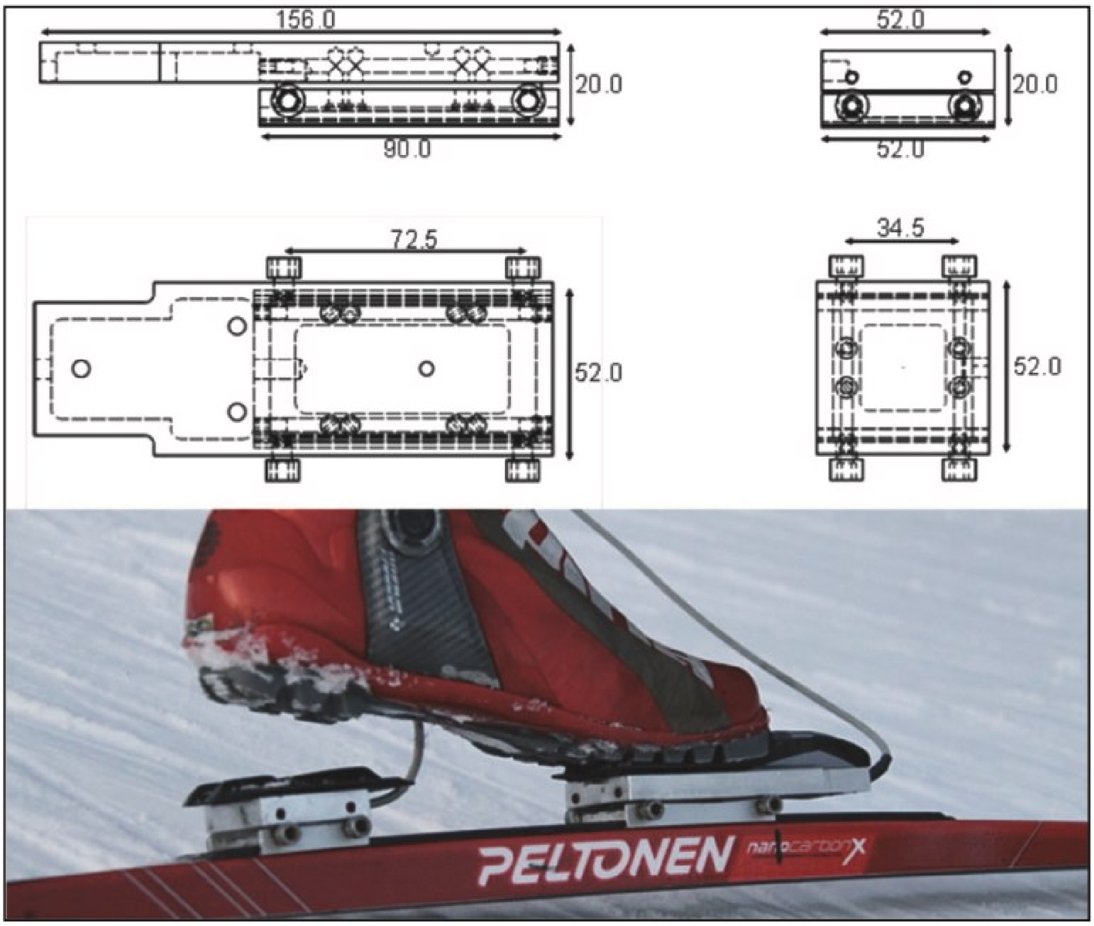


Figure 3: Schematics and photo of a custom force binding developed during Ohtonen's research work. Dimensions in schematics are in millimetres. (Ohtonen. 2019)

Pole sensors (Hottinger–Baldwin Messtechnik GmbH, Darmstadt, Germany) were installed on the pole grip (Figure 4). Pole sensors provide data about upper body forces.

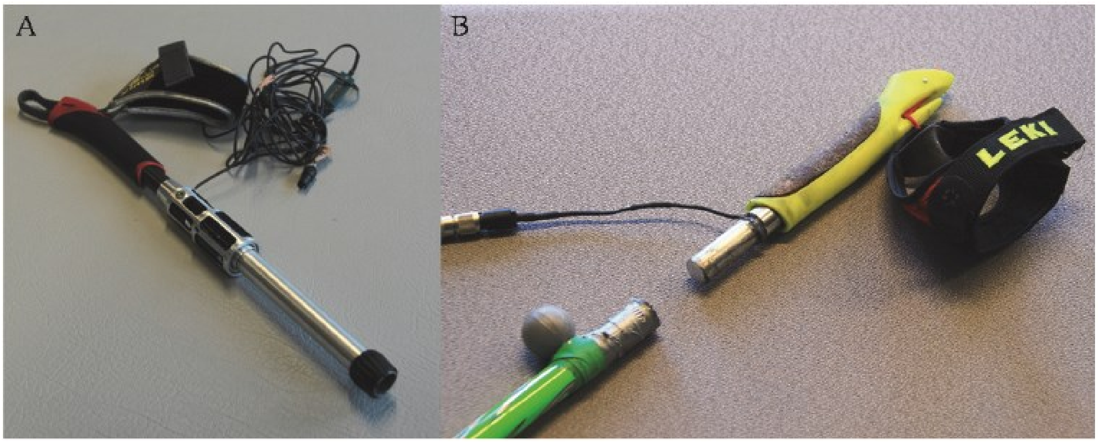


Figure 4: Pole force sensor (A) used in Ohtonen's dissertation study Experiment I and (B) in Experiment III. (Ohtonen. 2019)

The measurements in Ohtonen's work were conducted in the Vuokatti Ski tunnel. A conclusion of the work was that propulsion component analysis can offer valuable technique and performance optimization tools for athlete diagnostics to the coach. These tools have been combined with the Coachtech instant feedback system created by JYU. Coachtech is a versatile system aimed for coaches and athletes to analyse and compare training trials in various sports. In addition to the video of two cameras, Coachtech includes wireless measurement nodes, access points, ethernet components, a PC equipped with the application and a Web user interface. The wireless nodes collect data like, for example, the treadmill speed and angle, and force sensors from ski poles and ski bindings. The application combines and synchronises the signals from the various sources and provides the sport related feedback based on the inputs. In cross-country skiing, the parameters can be, for example, cycle length, the impulse of force and the side differences of impulses. The training recordings can be uploaded to a web server for athletes and coaches to access with credentials later. (Ohtonen. 2016)

Ohtonen et al. (2020) used 12 infra-red cameras (T-Series T 40S, 100 Hz, Vicon, Oxford, UK) in their research. The same cameras (8 of those) were also used in this research to collect ground truth data to be used for training and validation. One part of this work was to study if it is possible to create a training and validation dataset from the infra-red cameras of the VICON motion capture system. This dataset is later referred to as VICON data. If a sufficiently accurate can be created in this way, it would speed up the other parts of this work significantly by removing the need for manual work in labelling the data and thus allowing more time to be used to fine tune the models and to compare results to the ground truth. The comparison part is essential for deciding the quality of the machine vision output and for studying the main research question: could the VICON be replaced with an ordinary video camera and machine vision algorithm.

3 Pose estimation

In this chapter we briefly discuss what pose estimation is and why it is needed. First, we will review some traditional pose estimation methods, what are the problems with traditional approaches and then we will look how pose estimation can be approached with machine vision.

3.1 What is pose estimation

Pose estimation is a problem where the goal is to detect human body position and orientation precisely from the real world and simulate these in a virtual environment. Pose in this thesis means skier body position on top of a treadmill. The problem has been traditionally approached with marker-based motion capture systems like VICON. Motion capture systems are used e.g. for diagnosing clinical problems, biomechanical studies and animating characters in the movie industry. The most common way to capture body motion is to use reflective markers attached to skin and tracked with optical cameras. There are however problems related to the skin markers.

In addition to the reflective marker-based motion capture systems like VICON, there are other ways to achieve the same goal. The markers can be passive, like the reflective ones, or active like *LED* or acoustic markers. Active markers need wires and electricity so those are even more difficult to use in sports science than the passive ones. There are also non-optical systems that collect the position from inertia or magnetic-based markers. Traditional optical tracking systems used to track airplanes and satellites consist of the camera, computer and the mechanical tracking platform.[9]

Virtual reality is a promising and growing field of study because of its possibilities in both education and entertainment. Human pose estimation can be used to enhance the

[9] https://en.wikipedia.org/wiki/Motion_capture (12.3.2022)

interaction of humans and the virtual reality environment. Pose estimation can also be utilised in video surveillance to track, identify and recognize the actions of people in a monitored area. Medical assistance can be provided with pose estimation by detecting the movement of physical therapy patients. In self-driving cars the detection of persons with very high accuracy is an essential task. (Chen et al. 2020)

The latest video camera and machine vision based pose estimation methods have increased the number of possible applications of pose estimation. Since well-being has become a popular topic, pose estimation is used to create virtual personal trainers for yoga and other exercises. In robotics, pose estimation is used in simulated environments to train reinforcement learning algorithms. Motion capture and augmented reality are areas where pose estimation can be utilised in the entertainment sector. The most interesting application of pose estimation in the context of this thesis is athlete pose detection.[10] Machine vision based pose estimation is discussed more in Section 3.3.

3.2 Problems with marker-based pose estimation

The reflective markers should be placed on the skin precisely on top of the underlying bone. During movement, the skin can move in relation to the bone. The markers are also difficult to attach and attaching them is time consuming. The markers can also be an impediment to the movement of the subject. (Corazza et al. 2006)

In the studied application, clothes are an additional problem. Since some markers are attached to clothes instead of skin, there is yet another moving layer between the bone and the marker. Figure 5 shows an example of an athlete with markers attached. Some markers are attached to skin, but others are in cloth. In this case clothes are also not very tight fitting. While athletes are usually wearing tight fit training clothes, it is still possible that the marker is moving in relation to bone.

[10] https://www.v7labs.com/blog/human-pose-estimation-guide (12.3.2022)

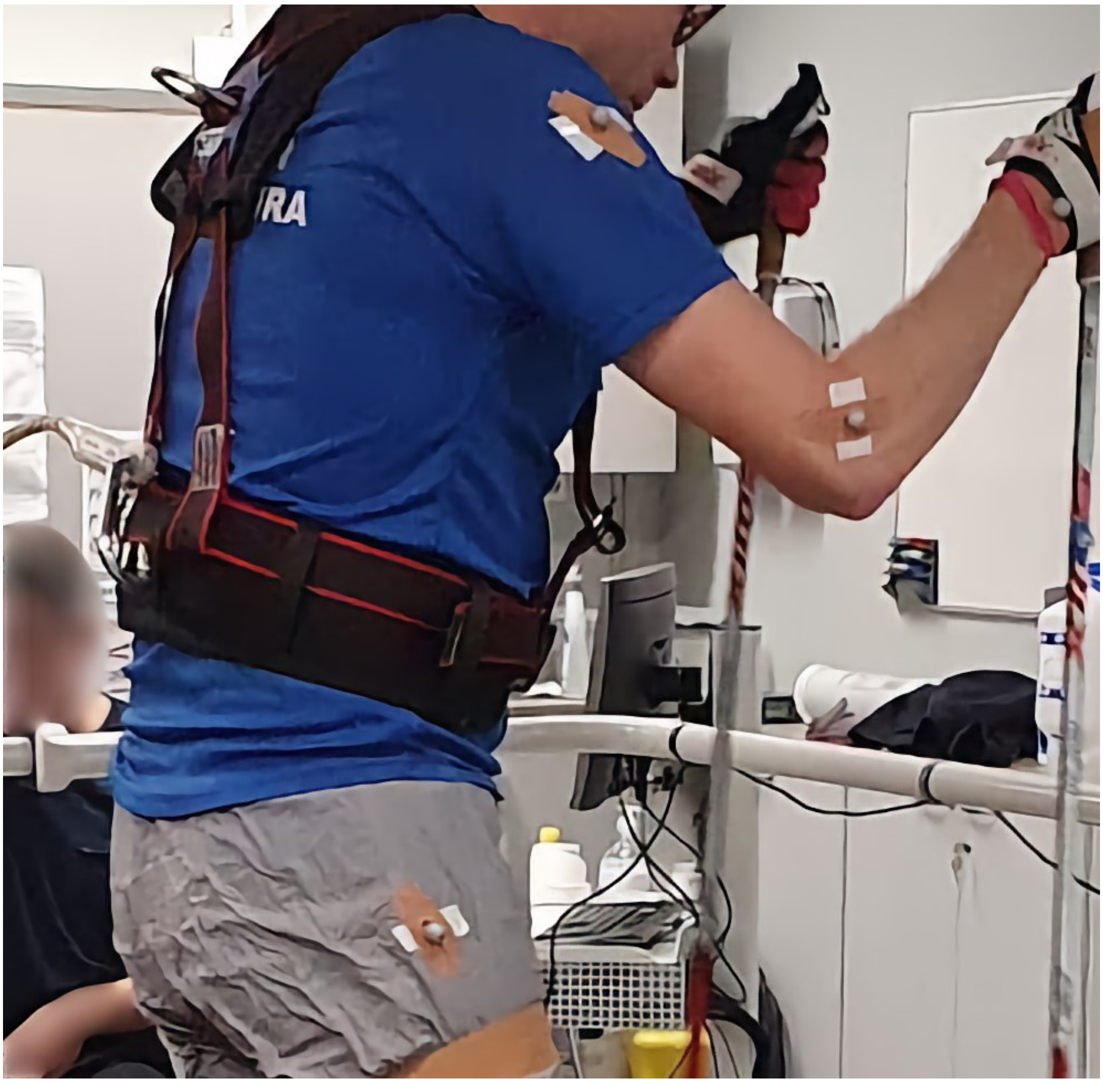


Figure 5: Example showing markers attached to the subject. Some markers are attached to skin and others to clothes. This subject is wearing loose clothes so there might be movement in the marker's position.

3.3 Single and multi-person pose estimation with machine vision

Detecting human pose from images or video is a fundamental challenge for machine vision. The image from a camera is a two dimensional representation of the three dimensional real world. When detecting human pose, there are unique characteristics and challenges, e.g. body positions can cause self-occlusions and body shapes vary depending on different clothes. Complex environments may cause foreground occlusions or occlusion from nearby persons. The camera view may impose limitations by occluding certain parts of the person. (Chen et al. 2020)

Two dimensional (2D) pose estimation utilises a single camera to record images or video from one side of the subject. Occlusion from the foreground objects or the subject's own body can significantly limit the performance. Three dimensional (3D) pose estimation on the other hand is not as prone to occlusion errors than 2D, but it is more difficult to achieve with a monocular camera. The need to also detect the depth of the joint increases the complexity of 3D pose estimation. There is also significantly less annotated material to train 3D models compared to single camera 2D material. (Chen et al. 2020)

There are two categories of human pose estimation algorithms for different purposes based on knowledge about the number of persons in the image: single person detection and multi-person detection. (Fang et al. 2017)

Single person detection is not sufficient for many real-world cases. Photographs often include more than one person and it is not clear that a single person detector can generalise well enough to handle this. Pishchulin et al. (2016) argue that there is a need for more attention towards multi-person detection because of its importance in real-world tasks. They list partial visibility of persons, overlapping bounding box regions around people and unknown number of people in an image as key challenges in multi-person detection. (Pishchulin et al. 2016)

All of the challenges described above are present in this work. Figure 6 shows one frame where the subject is skiing in the Vuokatti Skiing laboratory and there are a total of three persons in the frame even though we are only interested in the one in the foreground and the two partially visible persons should be ignored. The bounding boxes are overlapping and in some frames one of the background persons is behind the skier so there are only two bounding boxes in that frame. There are practical reasons why the video cannot be shot without the persons in the background. There has to be an operator controlling the treadmill and when performing real skiing practices, there is always a coach moving around the skier monitoring the performance. Room dimensions and the placement of the control desks do not allow changing camera to the other side of the room.

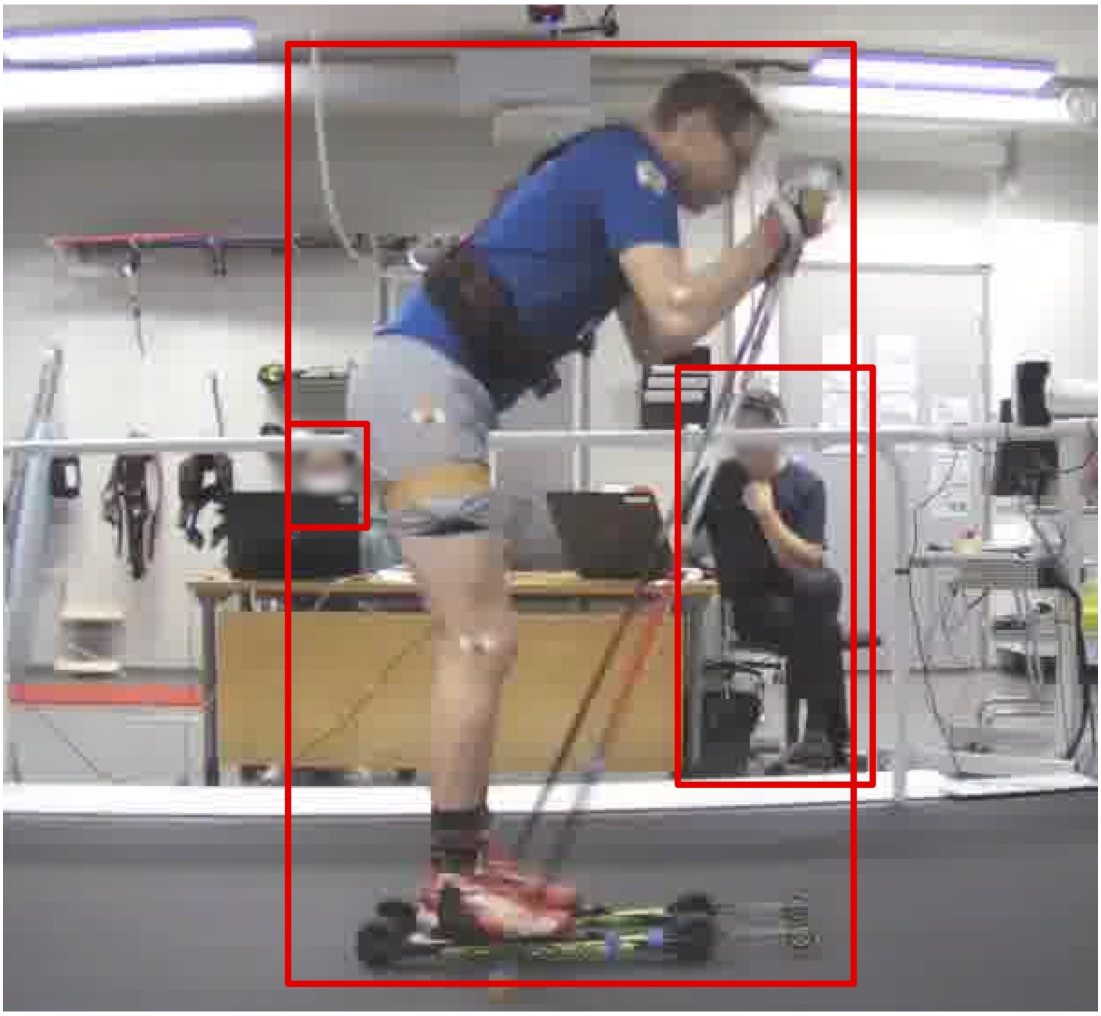


Figure 6: Three persons detected from image. Bounding boxes overlap and there are partially visible persons. Detected persons in the background are not relevant for the studied task and have to be ignored.

Recognizing multiple persons in an image is a lot more difficult problem than recognizing the pose of a single person. There are two approaches for this problem. One approach is to first detect the number of persons with bounding boxes and then estimate the poses for each person. This approach is known as the *two-step framework*. Pishchulin et al. (2016) argue that estimating bounding boxes first does not suit situations where there are many people close to each other. The other approach is to detect all parts of the human bodies separately and then connect the parts to form human body poses. Detecting bounding boxes correctly is a key task for accurate pose estimation in the two-step framework. A downside of the part-based approach is that if persons are too close to each other, it is difficult to decide which body each part belongs to. (Fang et al. 2017)

Only the relevant human pose should be detected in the task studied in this thesis. Figure 7 shows a frame from a video where there are three humans. The joints are marked and the skeleton is drawn only for the main subject.

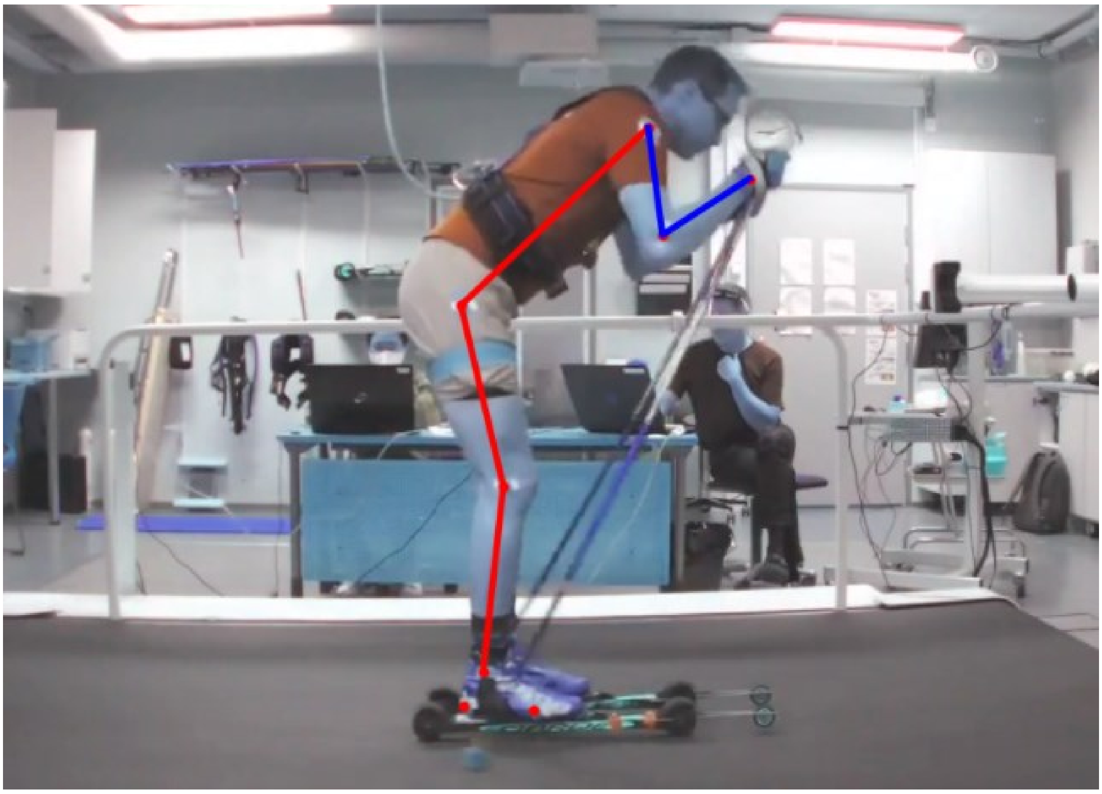


Figure 7: Joints are marked and connected only for the main subject in the frame. Humans in the background are ignored.

Tree models and random forest models have demonstrated to be very efficient in single person pose estimation tasks. Recent progress in deep learning techniques have yielded great improvements in human pose estimation as well as all object detection tasks. (Fang et al. 2017)

In multi-person pose estimation, so-called *k*-poselets are used by Gkioxari et al. (2014) to detect human joints and predict locations of human poses. Poselets are generalisations of poses like bigrams and trigrams are in natural language processing. K-poselets are based on spatial relations between parts the algorithm has learned.

Pishchulin et al. (2016) used integer linear programming to label and assemble body parts detected by their proposed DeepCut method. They used a *convolutional neural* *network* (CNN) to detect body parts. Their method uses integer linear programming to perform non-maximum suppression on the part candidate sets and forms a configuration of the body parts based on the candidates. CNN is explained in the next section.

Insafutdinov et al. (2016) used a *ResNet*-based stronger part detector and an incremental optimisation strategy in their method. They used pairwise terms between the body part hypotheses to group those into a valid human pose configuration. They say earlier models benefit more from the pairwise terms, but even for recent models this seems useful, as they still see the benefit for them due to better grouping.

In this work, the AlphaPose algorithm is used. It is built using the *Regional Multi-person Pose Estimation* (RMPE) algorithm by Fang et al. (2017), which is based on convolutional neural networks (CNN). It utilises a two-step framework to first detect bounding boxes and then to detect the joints. This approach and the AlphaPose algorithm were chosen for the task because of good performance and relatively easy usage of the available implementation. AlphaPose will be discussed in more detail in Section 4.2.

3.4 Machine Learning and Machine Vision

Machine learning is a form of applied statistics. It emphasises the statistical estimation of complicated functions with computers. The most fundamental characteristic of machine learning is that it can estimate or predict results for a task by learning the rules from the data, instead of using hard coded rules. Machine learning can be used for tasks too complicated to solve using traditional rule-based logic. (Goodfellow et al. 2016. p. 95-96)

In traditional computing, the programmer is crafting logic for the application by hand based on some rules and then feeding data into that application to get the result. In machine learning humans are feeding algorithms with data and correct answers and let algorithms figure out the rules. These rules can then be applied to new data to get answers. That leads into the reason why the paradigm is called machine *learning*. The algorithm is indeed trained with many iterations of the data and answers set so it can find statistical structures to form a rule that can be applied to new data. (Chollet. 2018. p. 4-5)

Machine vision is a broad field of study that by definition of Myler (1999) is “an implementation of systems that allow machines to recognize objects from acquired image data and perform useful tasks from that recognition.” Machine vision can also be referred to as computer vision, image understanding, scene analysis and robot vision, among others. The used name depends on the field of study where it is practised. Myler defines the term machine vision in his book by including both hardware and software into it.

Deep learning is the fast-growing subfield of machine learning. It is based on multi-layer artificial neural networks. The word ‘deep’ comes from the fact that the increasing computing powers from massively parallel graphic processing units (GPU) have enabled researchers to combine multiple layers of artificial neural networks, leading to substantial advancements. The increased computing power and massive growth of the available training datasets have increased the application areas of deep learning. As a concrete example, Cao et al. (2018) argue that the advancements in object recognition, localization and segmentation have brought machine learning into medical image analysis, for example to detect tumour tissue from the images of a patient.

Convolutional neural networks (CNN) are a special kind of neural network for processing images and other data that has grid-like topology. Image data can be thought of as a two-dimensional grid of pixels whereas e.g. time series data is forming a one-dimensional grid. The name “convolutional neural network” comes from the mathematical *convolution* operation. CNNs are neural networks that have at least one layer containing convolutional operations. CNNs have achieved significant success in several practical applications. (Goodfellow et al. 2016. p. 321)

4 Relevant algorithms for pose estimation

Pose estimation problems can be approached with several algorithms with different strengths and weaknesses. One open-source algorithm, AlphaPose, has been taken into deeper inspection in this work. AlphaPose was found to be the most promising algorithm in an earlier study in this topic and for that reason we selected it to be used in this work. The other reasons for this selection are the open source licence, somewhat easy usage and the possibility to train models further.

In Section 4.1 we discuss some relevant datasets for human pose estimation and which datasets we are using in this work. The AlphaPose algorithm is described in Section 4.2. We explain how it is used and what has been done in order to improve the results during this work. Some other relevant algorithms, like *DeepLabCut*, are discussed briefly in Section 4.3.

4.1 Relevant datasets in human pose estimation

The existing datasets described in this chapter are commonly used in machine vision research and they are publicly available on the internet. In addition to the selected theme-specific image files, the datasets consist of labels for object detection, and keypoints for pose estimation. In the scope of skier pose detection, we are interested in labelled body keypoints. There are some significant differences in keypoint formats that are very essential to understand. In skiing the foot keypoints are essential, because they are needed for the propulsion force calculations that are currently done based on motion capture system data. The foot keypoints are missing from *Microsoft Common Objects in Context dataset* (MS COCO) (Lin et al. 2014) but are present in *HALPE* (Li 2020).

The MS COCO dataset is a collection of images of common objects in context. It consists of 2.5 million objects in 328 000 human labelled images about complex everyday scenes containing common objects. The dataset is collected by Microsoft and it is created with the goal of developing object recognition algorithms for machine vision. The MS COCO dataset is used as validation data in annual machine vision contests.[11]

We used MS COCO keypoint format in this work. MS COCO format was selected because it was most commonly used in the existing AlphaPose models and thus the easiest one to approach. The MS COCO keypoints are shown in Figure 8 where on the left-hand side there are all the 17 keypoints with the appropriate numbering and the skeleton model connecting the keypoints. On the right-hand side there are keypoints and the skeleton model drawn on top of a tennis player.

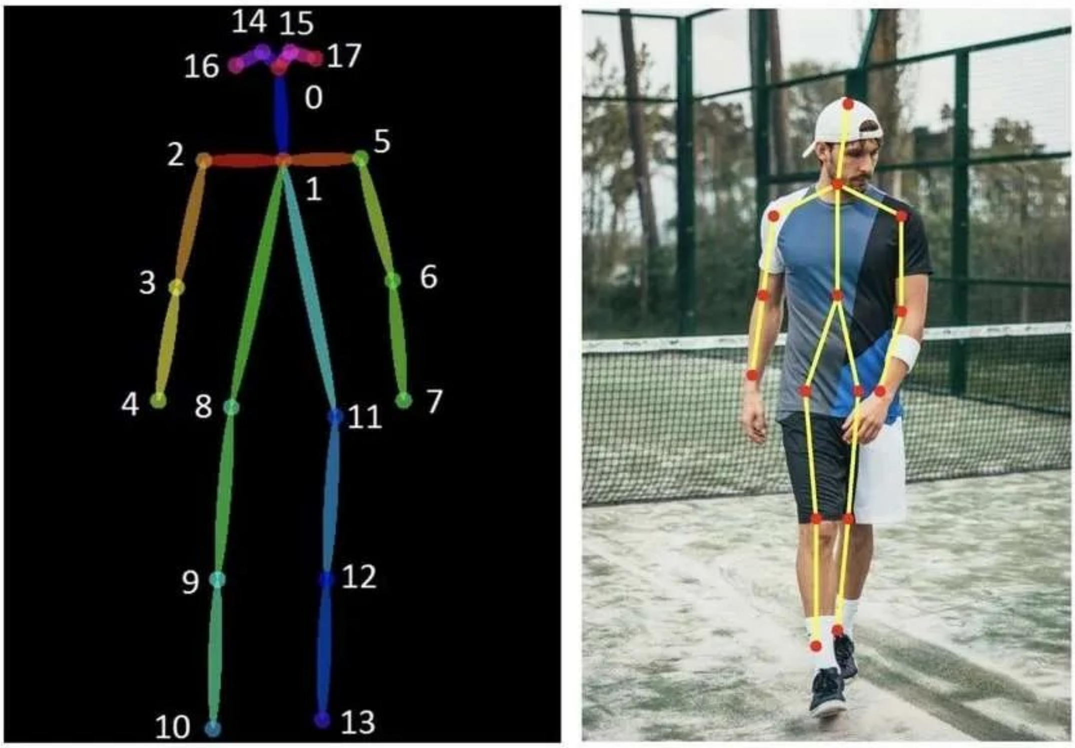


Figure 8: The MS COCO keypoints connected with the skeleton model (left) and the keypoints and skeleton drawn on top of a photograph of a tennis player (right).[12]

The HALPE dataset (Li 2020) is a joint project of AlphaPose and *Human Activity Knowledge Engine* (HAKE). It provides annotations of 136 human keypoints where 26 keypoints are of body, 68 of face, 21 of left hand and 21 of right hand.[13]

[11] https://cocodataset.org (30.1.2022)

[12] https://viso.ai/deep-learning/openpose/ (30.1.2022)

The main difference between MS COCO and HALPE in the context of this work is that HALPE includes keypoints for feet. These keypoints are essential for skier pose estimation, because knowing the angle between foot and leg is important for analysis and force production calculations. In this work we are aiming to use only part of the body keypoints that are shown in Figure 9. We focused only on keypoints on the right side of the body, because we were recording video only from the right side of the skier. The keypoints we were tracking were 6, 8, 10, 12, 14, 16, 23 and 25.

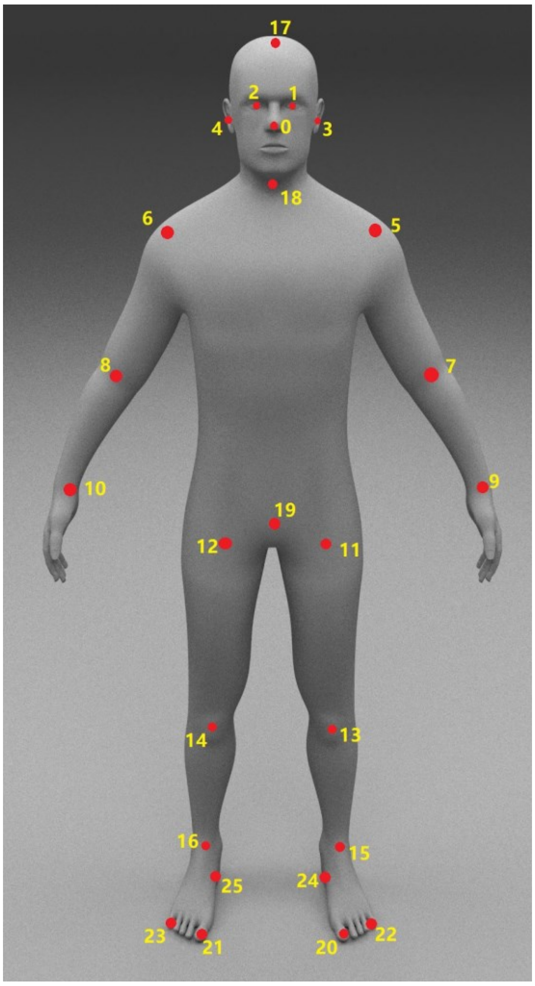


Figure 9: HALPE 26 body keypoints drawn on top of a human body shape.[14]

The *MPII dataset* (Andriluka 2014) contains around 25 000 images of 40 000 humans. The images are collected from *YouTube* videos and every image is

[13] https://github.com/Fang-Haoshu/Halpe-FullBody (30.1.2022)
[14] https://github.com/Fang-Haoshu/Halpe-FullBody (30.1.2022)

annotated with the body joint positions. The dataset consists of images of humans in everyday activities. A total of 410 annotated human activities are covered.[15]

Our plan was to start by formatting our training dataset into the MS COCO format and as soon as we would have functioning Python scripts for data preparation, fine-tuning the model and comparison, move on to the HALPE format that includes foot keypoints that are essential to our purpose. We started with MS COCO even though it did not include foot keypoints, because most of the available configuration files and models were using the MS COCO format and we assumed it would be easier to start the fine tuning of existing models with this format.

4.2 AlphaPose

AlphaPose (Fang et al. 2017) is a publicly available solution for multi-person pose estimation based on *Regional Multi-person Pose Estimation* (RMPE) algorithm. It is said to be the first open-source system to achieve 70+ *mean average precision* (mAP) on MS COCO and 80+ mAP on MPII dataset.[16]

The RMPE algorithm follows a two-step framework. It is aimed at detecting accurate human poses even when the bounding boxes obtained by body detection are inaccurate. It addresses problems in *single-person pose estimators* (SPPE). However, it can detect human poses incorrectly even when the human is correctly detected and the bounding box placement is correct. Another problem with SPPEs are the redundant detections. If the human detector detects the body incorrectly, a redundant pose is then also produced.

Fang et al. (2017) proposed their RMPE algorithm version 1 in 2016 and they have been improving it since then. The latest version 5 is from 2018. Their solution is based on three components: *Symmetric Spatial Transformer Network* (SSTN), *Parametric Pose Non-Maximum-Suppression* (NMS) and *Posed-Guided Proposals*

[15] http://human-pose.mpi-inf.mpg.de/ (30.1.2022)

[16] https://www.mvig.org/research/alphapose.html (4.1.2021)

Generator (PGPG). In their solution the SSTN is attached to the CNN-based SPPE for extracting a high-quality region for a single person even from an inaccurate bounding box. To tackle the pose redundancy issue, they introduce parametric pose NMS. It tries to eliminate redundant poses with a novel pose distance metric. PGPG is a novel way to augment training samples to produce large samples of data for training.

RMPE is a general solution and can be applied to both single-person and multi-person detection. The algorithm was able to achieve 76,7 mAP on the MPII dataset and it was able to handle inaccurate bounding boxes and redundant detections.

4.3 Other relevant pose estimation algorithms

In addition to AlphaPose, there are also other human pose estimation algorithms like *OpenPose*, *DensePose* and *HRNet*.[17] OpenPose is an open source algorithm for real time multi-person keypoint detection in two dimensional space. It can also be used in 3D as a real-time single-person keypoint detector. OpenPose has several keypoint configurations and it has a highly experimental and not production ready training repository for training own models.[18] (Cao et al. 2019; Hidalgo 2019)

DensePose (Güler 2018) is an algorithm to map all human pixels of an RGB image to the 3D surface based representation of the body. Since there has not been a labelled dataset for dense surface based pose estimation, one part of the DensePose project has been to create such a dataset. A MS COCO based training dataset has beens created by human labellers. DensePose is open source and it is based on convolutional neural network architecture.

Sun et al. (2019) introduced the novel *High-Resolution Network* (HRNet) algorithm which is based on high-resolution representations through the whole network architecture.

[17] https://www.analyticsvidhya.com/blog/2022/01/a-comprehensive-guide-on-human-pose-estimation/ (13.3.2022)

[18] https://github.com/CMU-Perceptual-Computing-Lab/openpose_train (13.3.2022)

Human pose estimation functionalities have also been added to general machine learning libraries. For example in *TensorFlow* there is the *Pose Detection* package which provides three state-of-the-art models that can be used to run real-time pose detection tasks. Pose detection models are accurate and so fast that they can be run on laptops and smartphones.[19]

There are also more complete solutions for pose estimation like DeepLabCut (Mathis 2018), which is a markerless pose estimation toolbox originally developed for animal pose estimation. DeepLabCut is not just an algorithm, but it combines tools to handle the whole process from data manipulation and training dataset labelling. It uses ResNets and redout layers as feature detectors and *DeeperCut* algorithm for human pose estimation from Insafutdinov et al. (2016). The algorithm needs to be trained with a custom training dataset before use. Researchers in Vuokatti are studying DeepLabCut in the topic of skier pose estimation.

[19] https://github.com/tensorflow/tfjs-models/tree/master/pose-detection (13.3.2022)

5 Data acquisition

This Chapter describes the methods used to collect training data to train skier pose tracking models. Section 5.1. describes the steps done in the practical work of this study. These steps are discussed in more detail in the forthcoming chapters. In Section 5.2. data collection methods and participants are described, Section 5.3 describes the testing protocol.

5.1 Work structure

The work started with a data collection event in the Vuokatti skiing laboratory. The VICON system and a video camera were used to collect the data from which ground truth and training dataset were constructed.

Next we started the data preprocessing which included three steps in the first part of the work: first we converted, trimmed and transformed the videos to sets of single frame images sorted into directories. Extra frames were removed from the beginning of the video and each set of frames started with the pole slam point which was used as the synchronisation mark. One of the videos was converted to *Audio Video Interleave* (AVI) format for the next step.

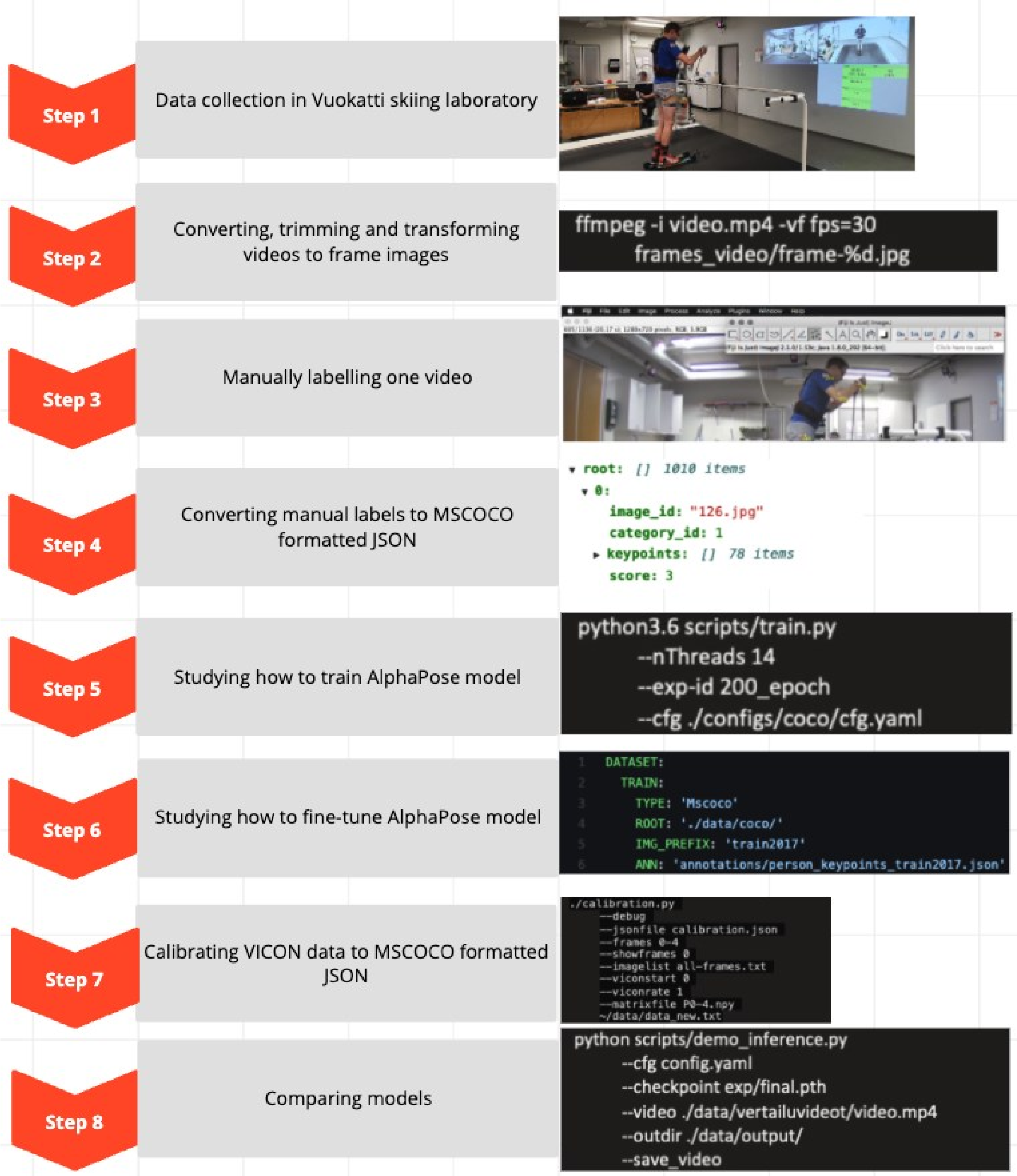
The second step of preprocessing was to manually label the AVI formatted file. Each frame of the video was processed and each joint was manually pointed in a labelling application to collect its coordinates.

The third step was to convert the manual labels into a MS COCO formatted *JSON* file. A *Python* script was created for this purpose.

With the manually-created JSON labels and the image dataset we were able to start the AlphaPose training and fine tuning. First we experimented with training models from scratch and then we switched to fine tuning existing models.

Then we returned to preprocessing the VICON data. The data had to be calibrated to match the video camera image coordinates. *The Direct Linear Transform* (DLT) algorithm was used for this task. The calibrated data was then also formatted as MS COCO into JSON files.

The last step was to run AlphaPose inference with different models and to compare the model outputs. All these steps are illustrated in Figure 10.



Step 1 Data collection in Vuokatti skiing laboratory

Step 2 Converting, trimming and transforming videos to frame images

```
ffmpeg -i video.mp4 -vf fps=30
          frames_video/frame-%d.jpg
```

Step 3 Manually labelling one video

Step 4 Converting manual labels to MSCOCO formatted JSON

```
▾ root: [] 1010 items
  ▾ 0:
      image_id: "126.jpg"
      category_id: 1
    ▸ keypoints: [] 78 items
      score: 3
```

Step 5 Studying how to train AlphaPose model

```
python3.6 scripts/train.py
          --nThreads 14
          --exp-id 200_epoch
          --cfg ./configs/coco/cfg.yaml
```

Step 6 Studying how to fine-tune AlphaPose model

```
DATASET:
  TRAIN:
    TYPE: 'Mscoco'
    ROOT: './data/coco/'
    IMG_PREFIX: 'train2017'
    ANN: 'annotations/person_keypoints_train2017.json'
```

Step 7 Calibrating VICON data to MSCOCO formatted JSON

```
./calibration.py
    --debug
    --jsonfile calibration.json
    --frames 0-4
    --showframes 0
    --imagelist all-frames.txt
    --viconstart 0
    --viconrate 1
    --matrixfile P0-4.npy
    ~/data/data_new.txt
```

Step 8 Comparing models

```
python scripts/demo_inference.py
        --cfg config.yaml
        --checkpoint exp/final.pth
        --video ./data/vertailuvideot/video.mp4
        --outdir ./data/output/
        --save_video
```

Figure 10: All steps with short descriptions and example snippets of the step.

5.2 Collecting training data

There were four participants in the video recording session. There were one female and three males, and there was considerable variation in the body sizes of the participants. The participants were representing two categories: skiing enthusiasts, who are former competitive skiers, and active competitive national level skiers. The skiers were selected among volunteers to represent various body sizes and shapes and performance levels. The idea was to get more heterogeneous data by selecting skiers from two categories.

Every participant had read through a document where the reason for the video recording session was stated. The participants were also asked to sign a paper to declare they are perfectly healthy.

The videos were recorded using a Lilin UFG1122ex3 camera. Figure 11 is showing the used camera in its attachment position. The camera is attached to a movable arm and it was easy to move either on purpose or by accident. This may cause issues when calibrating the VICON point cloud data for training the dataset. The calibration process is discussed in Section 6.4.

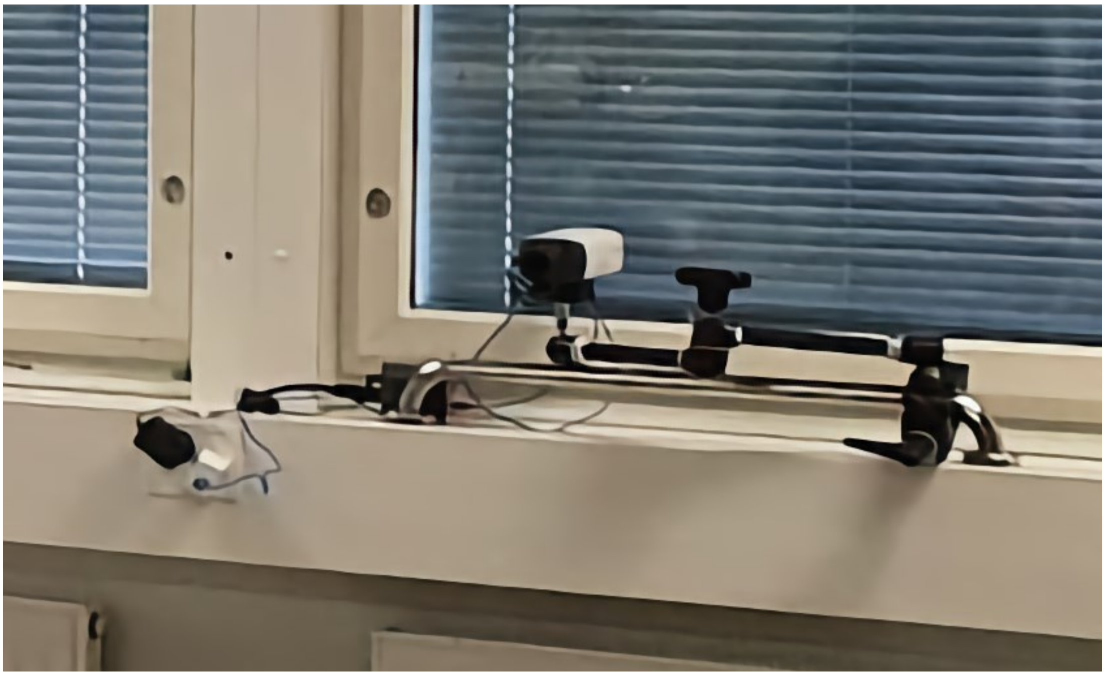


Figure 11: Lilin UFG1122ex3 camera attached to its movable arm. The easily movable arm can cause problems with calibrating VICON and the camera, since the location of the camera needs to be fixed when calibrating point cloud data to video.

5.3 Test protocol

As the first step, the reflective markers for the VICON motion capture system were attached to the participants. Since the video is recorded from the side, there was no need to attach markers to both sides of the skier. Figure 12 is illustrating the marker positions and the attachment using sports tape with a whole body image and more precisely in the elbow.

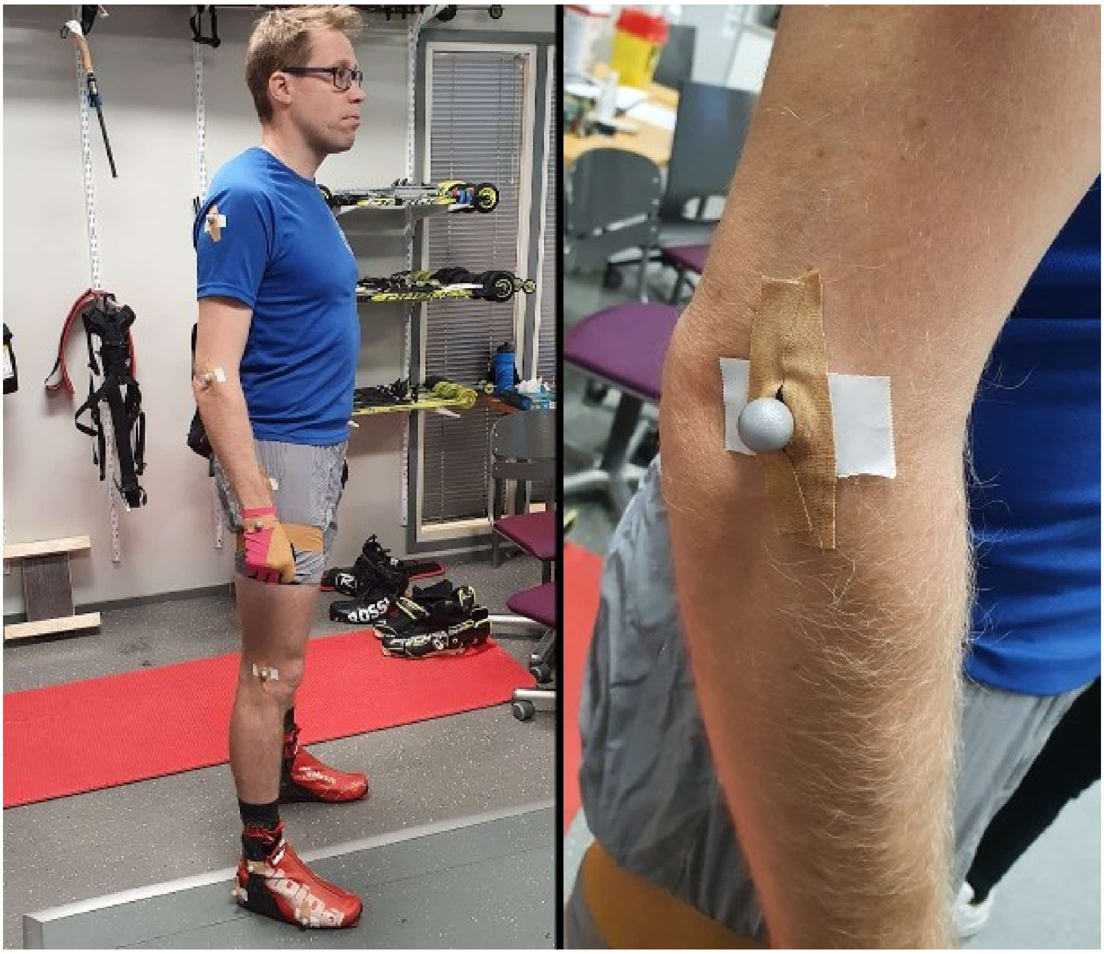


Figure 12: There were a total of 8 markers attached to the right side of the subject: in the wrist, elbow, shoulder, hip, knee, ankle, heel and base of the fifth toe. The markers are attached using sports tape.

The reflective markers are tracked by 8 infrared cameras of the VICON system that are located around the skiing treadmill. Figure 13 is illustrating the locations of the four infrared cameras in the other side of the room.

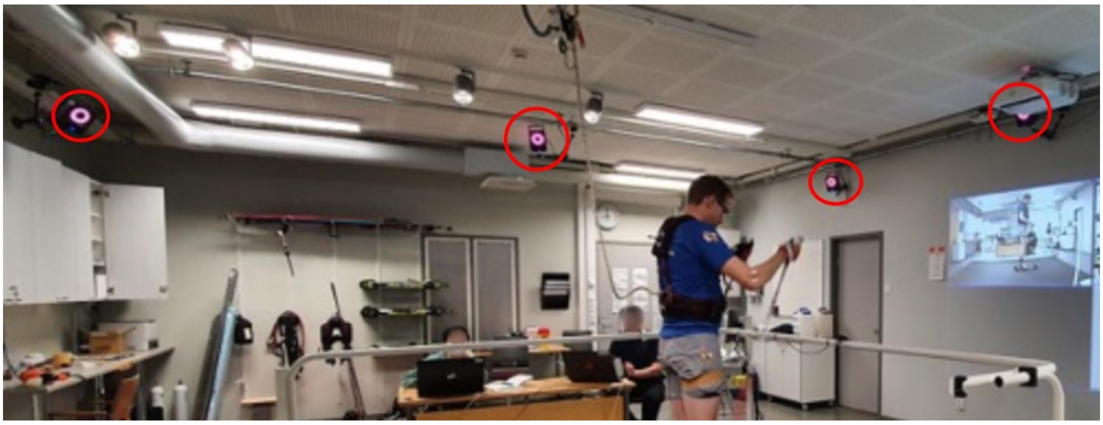


Figure 13: VICON infrared cameras surrounded with red circles are installed to the ceiling of the laboratory room around the skiing treadmill.

The recording events started for each subject with a warm-up period where the skiers had the possibility to ski on the treadmill using various settings for speed and inclination, until he/she was ready to perform the actual test runs with the specified speeds according to the test protocol. The testing protocol consists of the following steps:

All participants were supposed to perform about 10 full skiing cycles in four speed levels and with two different classic skiing styles: *double poling*, and *diagonal skiing*.

A *cycle* in skiing means body movement from the rest position through propulsion generation movement back to the rest position. Figure 14 illustrates the one double poling cycle with four intermediate positions between the beginning and end positions. (Danielsen 2018)

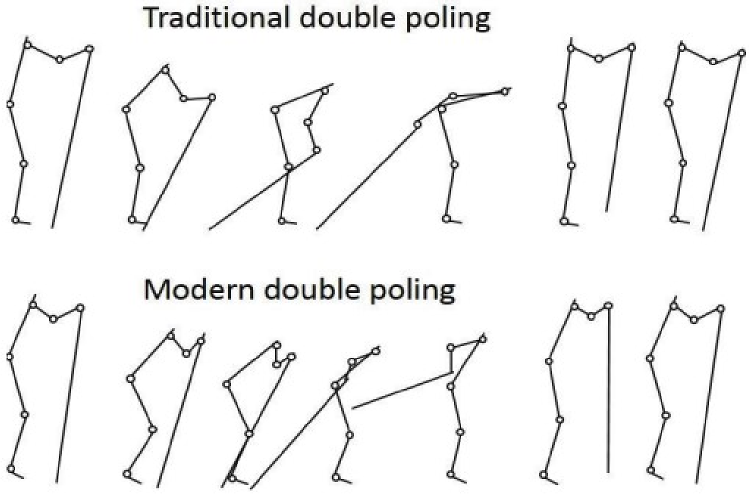


Figure 14: Illustration of a cycle in double poling. There are differences between the traditional and modern versions, but these differences are not relevant in this study. (Danielsen 2018 - page 5)

Double poling is a skiing technique in which all propulsive force is applied through the poles and where the skis glide continuously forward parallel to track. It is mostly used on low declination downhill and flat track parts. Strong competitive skiers can even use double poling in steep uphills. Most of the propulsion output power is produced with upper extremities but also lower extremities play a significant role in power output by extending and raising the body before the swing phase. (Danielsen 2018)

While double poling is mostly used in the flat parts of the track and lower inclines, diagonal stride is more of a steeper incline technique. The propulsion power is produced with both the poles and the skis. One cycle of diagonal stride skiing is illustrated in figure 15. From a kinematic perspective diagonal stride can be compared to running since the arms and legs are moving in anti-symmetrical synchronous fashion. (Danielsen 2018)

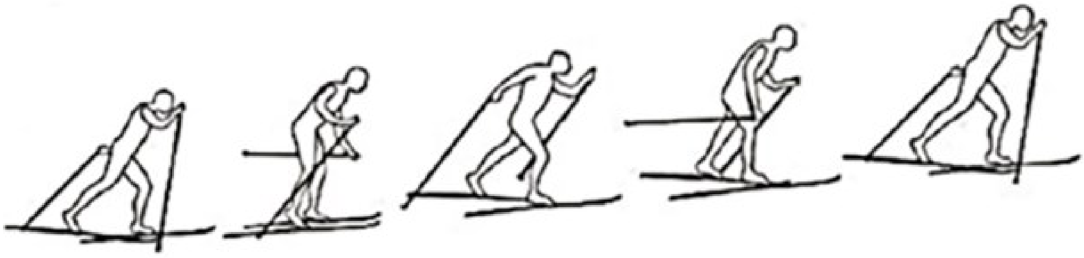


Figure 15: One cycle of the diagonal stride skiing technique illustrated with three intermediate body positions between the beginning and end positions. The figure is modified from the original. (Welde 2017)[20]

Figure 16 visualises the difference between treadmill inclination angles. A 1° inclination angle can be considered flat, since it is commonly used in treadmill training to compensate for missing resistance from air flow. An 8° inclination was chosen for uphill skiing.

[20] http://dx.doi.org/10.1371/journal.pone.0187111 (18.4.2022)

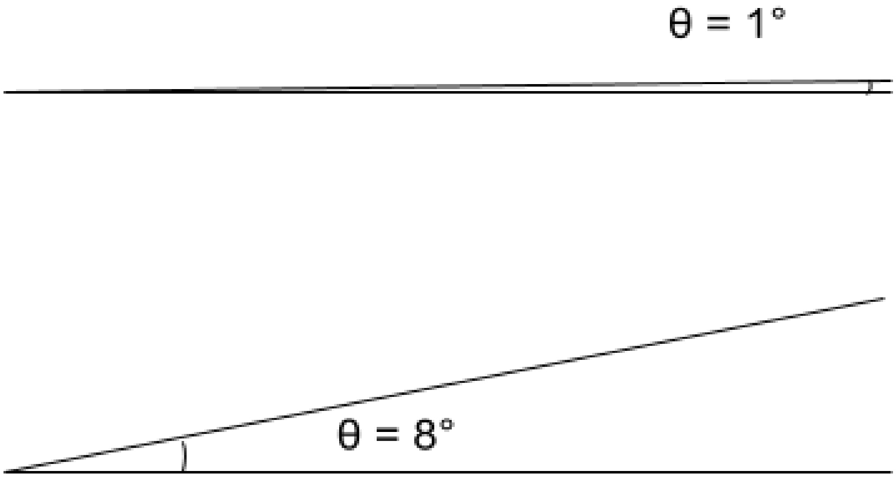


Figure 16: A visualisation of the difference between 1° and 8° inclinations used in the videos for this work. A 1° inclination is commonly used in indoor workouts to compensate for missing air resistance from outdoor training. 8° was used for uphill skiing.

Double poling was done on 1-degree inclination and diagonal skiing was performed on an 8-degree inclination. Styles are shown in Figure 17 demonstrating differences in inclination angles and skiing techniques used.

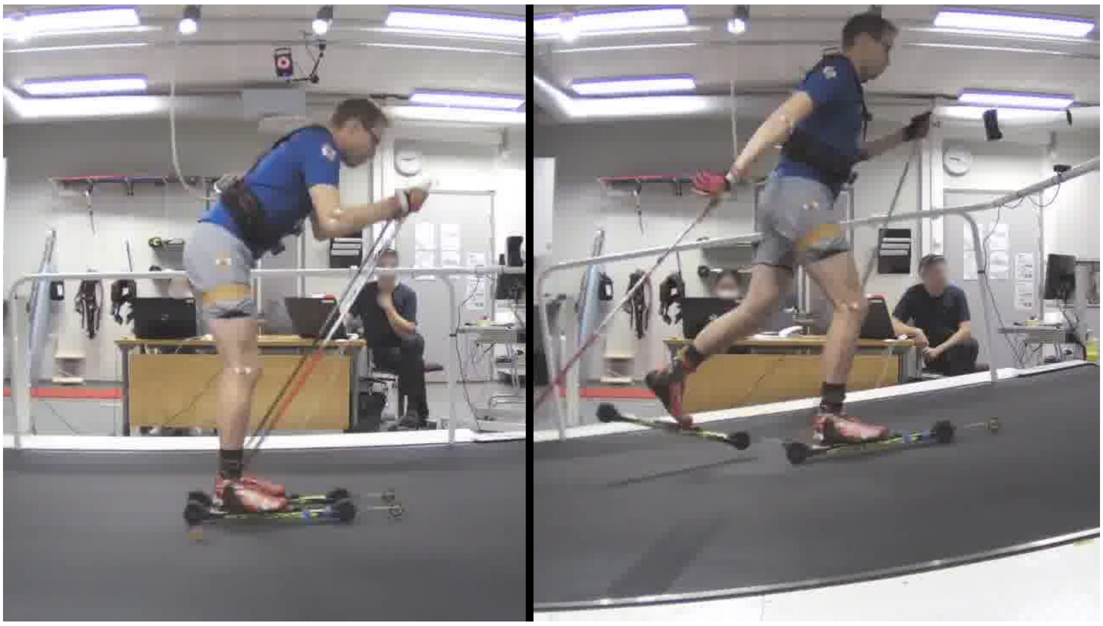


Figure 17: Double poling on 1° inclination shown on the left figure, diagonal skiing on 8° inclination on the right figure.

In the first session double poling was used. For skiing enthusiasts the treadmill speeds were 10 km/h, 15 km/h, 20 km/h and 25 km/h and for competitive skiers the speeds were 15 km/h, 20 km/h, 25 km/h and 30 km/h. The speeds are presented in Table 1. The skiers had the possibility to take breaks as needed between runs. The breaks lasted a few minutes. The breaks were shorter between the early runs but as the speed of the treadmill increased and therefore the runs got more demanding, the breaks became longer. In the second session with diagonal skiing, the treadmill speeds for enthusiasts were 6 km/h, 8 km/h, 10 km/h and 12 km/h. For competitive skiers the speeds were a bit higher: 8 km/h, 10 km/h, 12 km/h and 14 km/h. (Table 1)

Table 1: Treadmill speeds in double poling and diagonal skiing. The treadmill was set on 1° inclination.

	Run 1	Run 2	Run 3	Run 4
Double poling				
Skiing enthusiasts	10 km/h	15 km/h	20 km/h	25 km/h
Competitive skiers	15 km/h	20 km/h	25 km/h	30 km/h
Diagonal skiing				
Skiing enthusiasts	6 km/h	8 km/h	10 km/h	12 km/h
Competitive skiers	8 km/h	10 km/h	12 km/h	14 km/h

At the beginning of each run, the camera and VICON were started, and participants slammed the right pole to the treadmill for a synchronisation mark. Then the treadmill was started, and the skier performed about 10 cycles with the desired skiing style. This corresponds to about 20 to 40 seconds of video material with 30 frames per second.

Some of the recorded runs had to be discarded because of failures in following the test protocol, i.e. the synchronisation slam was forgotten or the recording was not started at the correct time. The total number of collected frames after trimming the beginning of the videos was 34 333. Unfortunately we were not able to utilise all of the collected frames because of video quality issues. From Subject 1 we collected 5344 frames from the five successful runs. From Subject 2 we managed to collect data from all eight runs with a total of 9607 frames. From Subject 3 seven successful runs were recorded but none of the recorded 8473 frames were used in the end. Most of the frames of Subject 3 suffered from flickering and artefacts leading to us discarding those frames. From Subject 4 we managed to capture nine runs with a total of 10909 frames after trimming but because of the same quality issues as with Subject 3 only 10 of those were used in the end. We manually picked 10 frames for the calibration dataset. All datasets are described in the next chapter. Subject 4 tested the highest speed twice which explains the one extra run.

It is noteworthy that the low number of used frames from Subject 4 was because we were unable to get the VICON calibration to work sufficiently, which caused us to have to rely on manually labelled data only. We manually selected 10 frames that were used for the calibration and in the end for the model performance evaluation dataset. These issues are discussed further in Section 6.5.

6 Data pre-processing

This chapter describes the steps in collecting the video files and pre-processing the VICON point cloud data before it could be utilised in the training of the machine vision models. In Section 6.1 we discuss how the data from two different sources, video camera and VICON motion capture system was synchronised for calibration. The method for synchronisation had to be decided and added to the test protocol before the data collection event so the athletes could prepare for it. Section 6.2. describes the dataset structure. In Section 6.3 we explain how the data was labelled for training and Section 6.4 explains the process to calibrate VICON point cloud data points to match the video camera view and the problems encountered.

To address the research question "How to create a training dataset by utilising accurate VICON data", a Python script was created to modify VICON output to the MS COCO format for algorithms to understand as labels for video. All recorded material was supposed to be labelled so that different parts of it could be used for training and validation in different experiments. During the work we found out that calibrating the VICON data to match the two-dimensional space of the video camera view was more difficult than anticipated and in the end only small portions of the VICON data got labelled. The calibration process and issues with it turned out to be a significant part of the work and are discussed in detail in the Section 6.4.

In Section 6.5 we explain the encountered challenges in the pre-processing phase and how different the pre-processing turned out to be than initially thought. Python source codes for dataset creation can be found from *Hyteli-scripts* GitHub repository.[21]

[21] https://github.com/juhahu/hyteli-scripts (9.4.2022)

6.1 Synchronising and converting the videos

The *Handbrake* open source video transcoder application was used to process the videos. When we started to work with the video files, we noticed that the camera was set to use a variable frame rate. *Variable frame rate* (VFR) is a feature of some video containers where the frame rate of parts of the video where there is less movement is decreased to reduce the size of the video container (Waggoner 2013 p. 134). This frame rate inconsistency caused problems when we started to synchronise the data.

VFR also prevented AlphaPose algorithm from working, since when framerate started to fluctuate, the AlphaPose run failed. This would have also caused synchronisation problems later. Therefore the framerate was changed to constant 30 with Handbrake. The videos were also re-coded to H.264 for better usability.

For training, the videos needed to be converted as frame images. The *ffmpeg* application was used as follows:

```
ffmpeg -I input_video_name.mp4 -vf fps=30 frame_directory/frame-%d.jpg
```

Ffmpeg application is a multimedia framework which can be used to decode, encode, transcode or play multimedia files in a wide variety of operating systems. Figure 18 shows the converted frame from the video.

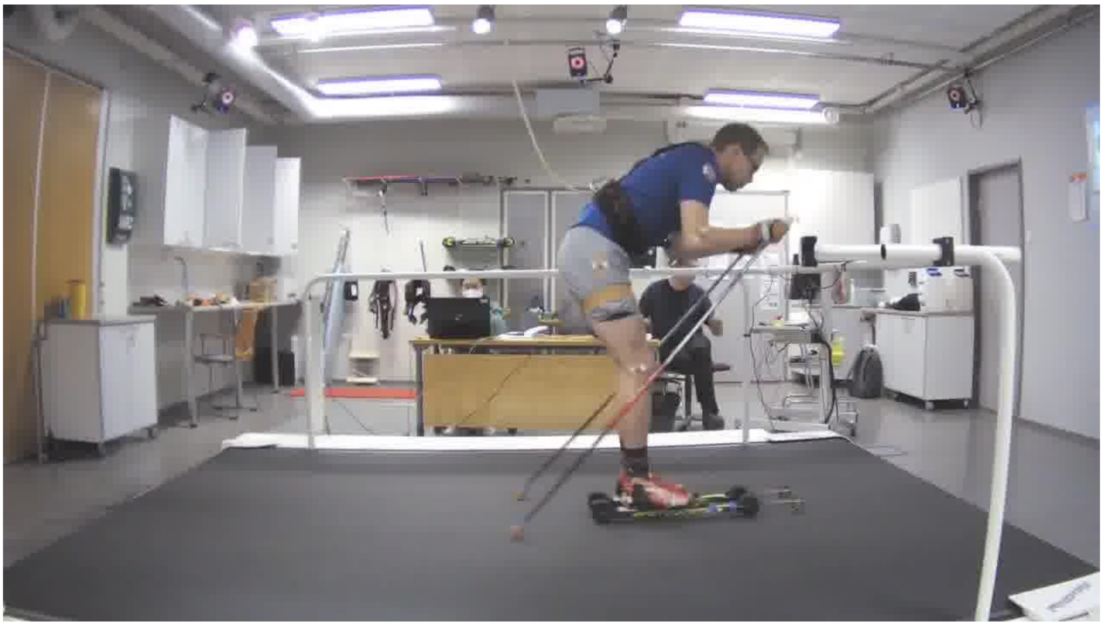


Figure 18: A frame from the processed video.

One video had to be converted to Audio Video Interleave (AVI) format for manual labelling, as the *Fiji*[22] image processing application (Schindelin 2012) used for manual labelling did not work with QuickTime (MOV) or MPEG4 (MP4) file formats. ffmpeg was used for the conversion.

```
ffmpeg -I input_vide_name.mp4 -codec:v rawvideo
avi_directory/video.avi
```

Finally, the unnecessary waiting times from the start of the videos were trimmed. The test protocol included a pole slam as a synchronisation mark to synchronise video with VICON data and this mark was used to trim the videos. All frames before the synchronisation mark were removed. The trimmed parts were frames where skiers stood still while waiting for the treadmill to start rolling. As a result, the total

[22] https://imagej.net/software/fiji/ (18.4.2022)

amount of collected data was reduced, but no frames with actual skiing movement were lost during trimming.

6.2 Datasets for skier pose estimation

The dataset creation was an important part of the work. It was a requirement that the data is of good quality in order for the training to succeed and the accuracy of the model to be sufficient for the targeted application. In total, we created three MS COCO formatted datasets for skier pose estimation. The first one was the manually labelled dataset for the first training experiment and the second the VICON dataset for the next experiment. The last dataset was the calibration dataset used for VICON calibration and later for the validation of the models. The manually labelled dataset was called the *handpicked dataset*.

For MS COCO, the correct file hierarchy is important. The data should be located in a dataset specific directory where it is splitted into three subdirectories: *train2017, val2017* and *annotations*. For the first dataset we used 1000 frames. The first 700 frames were used for training and were saved into the train2017 directory. The last 300 frames were saved into the val2017 directory. All the datasets with subject counts, total image counts and the train / val -splits are presented in Table 2. The JSON formatted manually picked labels were saved into the annotations directory. For the second, the *VICON dataset* we collected the 700 and 300 frames from five videos so in total the second dataset had 3500 frames in the train2017 directory and 1500 frames in the val2017 directory. In total there were 5000 frames. The annotation file to this dataset was created from calibrated VICON data.

The naming of the frame files turned out to be important as well. We renamed the frame files of each directory to be six digit index numbers with leading zeros starting from 000001. Renaming functionality from *Mac OS Finder* was used for renaming and creating the file hierarchy.

The third dataset, i.e. the calibration dataset, was a bit different. It contained a total of 30 frames, 10 from three subjects each, with five frames each from two different videos. The videos were selected to represent different speeds and skiing techniques. No images from Subject 3 were used due to issues with the video quality. With these data splits the aim was to get as heterogeneous dataset for calibration and validation as possible from the collected material. We selected the 100th, 200th, 300th, 400th and 500th frame from each of the selected videos. The file hierarchy was also slightly different than in the two previous datasets since all 30 images were saved into the train2017 directory and val2017 was left empty. The labels were saved into the annotations directory like with the previous datasets.

Table 2: Details of the datasets created: the count of subjects presented, the total image count, train / val -split used and the origin of labels.

Dataset	**Subjects**	**Images**	**Train**	**Val**	**Labels**
Handpicked	1	1000	700	300	Manually labelled from the frames of 1 video
VICON	2	5000	3500	1500	Calibrated VICON point cloud data from 5 videos with DLT algorithm
Calibration	3	30	30	0	Manually labelled from the frames of 6 videos

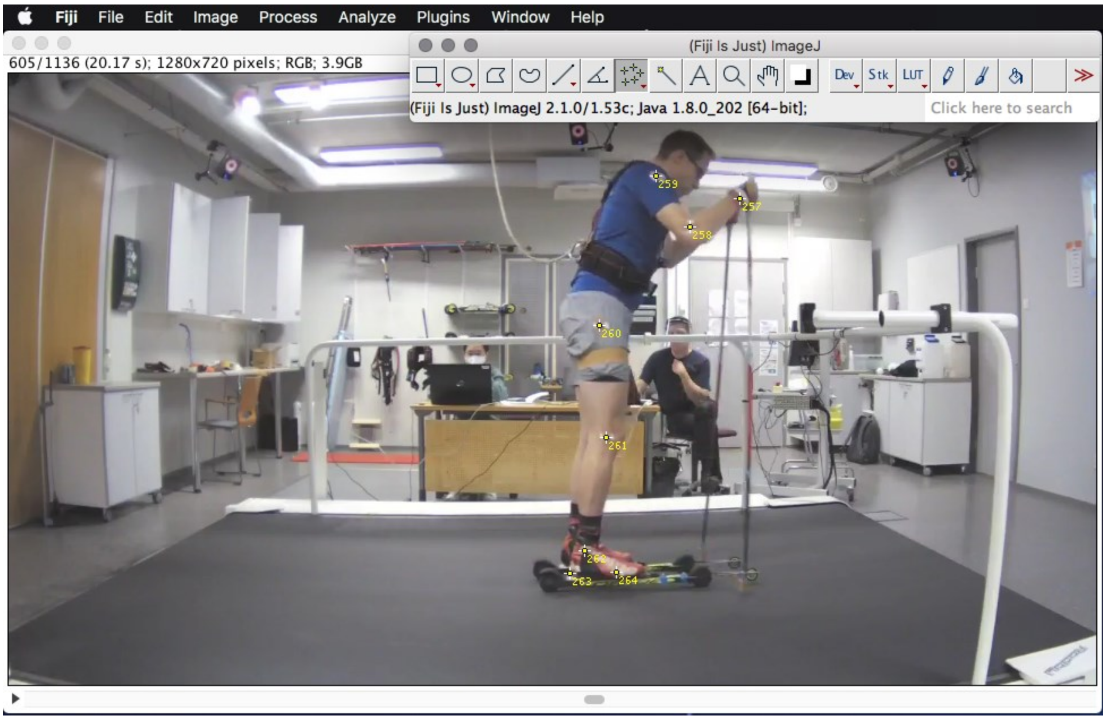
There were many issues within this part of the work. For example, there was uncertainty with the label file format because of missing documentation, causing pre-processing to take us more time than anticipated. These issues are discussed in more detail in Section 6.5.

We used the *Pandas*[23] library to manipulate data when pre-processing the label information into JSON formatted annotation files. Pandas is an open source Python library used in data manipulation and analysis.

6.3 Labelling the training material

The recording of the sessions was done using a Lilin UFG1122ex3 camera and the VICON motion capture system. One of the recorded videos was then labelled manually using the Fiji application. Figure 19 illustrates the manually picked labels in the Fiji user interface. Fiji is an open-source application for rapid prototyping of image-processing algorithms. It was originally developed for biological image analysis. Fiji is based on open-source software *ImageJ* and uses ImageJ plugin format to enable easy sharing of new algorithms between users. (Schindelin 2012). With Fiji, each frame of the video was processed and 8 joints were marked with mouse clicks to collect the corresponding location coordinates. A total of 9088 precisely marked points were marked for that one video. Labelling data manually proved to be a very time-consuming task that could not be done for very long at a time. It also turned out that creating the markings precisely was very difficult. The frames had to be switched back and forth and earlier markings had to be edited because the motion blur effect caused markers in some of the frames to be more difficult to detect. At the end, Fiji processed the video and created a CSV output file containing one data row per each marked point.

[23] https://pandas.pydata.org/ (18.4.2022)

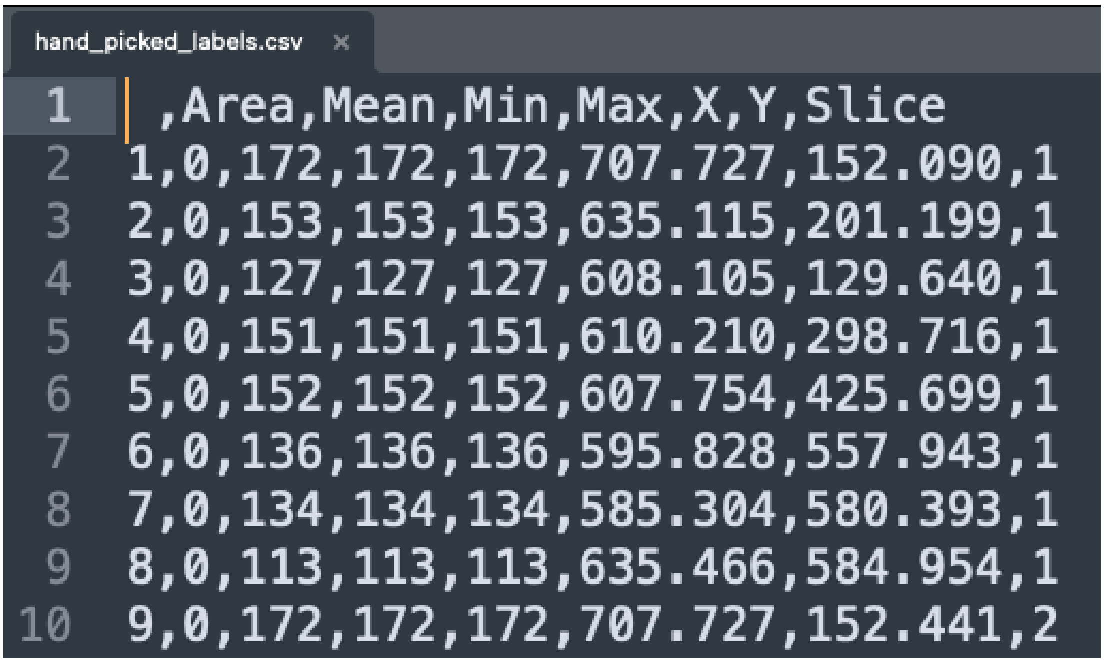


605/1136 (20.17 s); 1280x720 pixels; RGB; 3.9GB

(Fiji Is Just) ImageJ 2.1.0/1.53c; Java 1.8.0_202 [64-bit];

Figure 19: Manual labelling with Fiji.

The CSV file created by Fiji contained details of each marked joint (Figure 20). The relevant data columns for this work were the X and Y coordinates and Slice. Slice indicated which frame the row was about. This was used first in the script to determine if data for all 8 joints were present for each frame.



```
 ,Area,Mean,Min,Max,X,Y,Slice
1,0,172,172,172,707.727,152.090,1
2,0,153,153,153,635.115,201.199,1
3,0,127,127,127,608.105,129.640,1
4,0,151,151,151,610.210,298.716,1
5,0,152,152,152,607.754,425.699,1
6,0,136,136,136,595.828,557.943,1
7,0,134,134,134,585.304,580.393,1
8,0,113,113,113,635.466,584.954,1
9,0,172,172,172,707.727,152.441,2
```

Figure 20: CSV file containing manually labelled points.

There are also other supportive scripts that had to be crafted. One script was for detecting synchronisation frames from the label files so that the start of the label data and the video could be trimmed precisely correctly. Another script was to draw images and skeleton models from labels to verify that the frame synchronisation was correct for manually labelled and transformed VICON data. All of the scripts are written in the Python language and are stored in the Hyteli-scripts GitHub repository.[24]

6.4 VICON data calibration

One research question this work was supposed to answer was: is it possible to to use the point cloud data from the VICON motion capture system to create large amounts of training data. Since VICON collects coordinates of reflective markers (Figure 12, in Section 5.3) in 3D space with eight cameras and the video camera is filming a skier only on one side and in 2D, the coordinates have to be calibrated.

The calibration was done using a modified version of a script denoted as *Camera Calibration* available with an open MIT licence in a Github repository.[25] The Camera Calibration script is created for calibrating real-world 3D coordinates to match a 2D view. The script first calculates the camera projection matrix P using the *Discrete Linear Transform* (DLT) algorithm (Abdel-Aziz & Karara 1971; Zhang 2000) from the given calibration 3D-2D point correspondences. We used the point correspondences from 5 manually labelled frames. With the estimated projection matrix P, we can then calibrate the 3D coordinates from any VICON point cloud data frame to 2D image plane to match the camera field of view.

DLT is a linear algorithm and as such is not capable of handling optical lens distortions. One possibility is to extend the standard 12 element 3D DLT with 5 additional parameters, also the optical distortion and decentering distortion can be

[24] https://github.com/juhahu/hyteli-scripts (9.4.2022)
[25] https://github.com/sreenithy/Camera-Calibration (18.3.2022)

addressed. 8 control points are needed in this case.[26] Other more advanced algorithms include Tsai's calibration method (Tsai 1987) and the *calibrateCamera()* function implemented in *OpenCV*.[27] In this work we used the standard 12 element 3D DLT which can be used with a minimum of 6 control points. The control points in this work are the joint coordinates.

We used the original DLT functions from the Camera Calibration script but created our own script to preprocess the data and to call those functions. Usually the DLT calibration is done using some object whose dimensions are known such as the Rubik's Cube or a Chess board pattern. We were not able to use an approach like that but instead created a script to utilise the location coordinates from VICON point cloud data. Python functions from the Pandas library were used for manipulation.

The projection matrix *P* is calculated with the DLT algorithm by providing the known 2D image plane coordinates as *x* and the known coordinates of the 3D world from VICON data as *X*. The projection for the *i*th coordinates is shown in Equation 1.

$$x_i = PX_i \tag{1}$$

where

$$x_i = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad X_i = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

[26] http://www.kwon3d.com/theory/dlt/dlt.html (18.3.2022)
[27] https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html (25.3.2022)

The elements x and y in the vector x_i are the 2D image control points and the elements X, Y and Z in the matrix X_i are the 3D world coordinates. Both vectors are represented in homogeneous coordinates.

The resulting P is a 3x4 projection matrix containing 12 unknowns, or actually 11 as the scale cannot be derived from a 2D image. We get two equations from each 2D point and since the $11/2 = 5.5$ we need at least 6 known joint locations to estimate P. In our data we had eight joints manually labelled so we had enough coordinates for the calibration even on a single frame. More parameters can yield better performance though as the sensitivity for the errors would decrease. The longer the distance between the coordinate points of the joint between calibration images, the better the calibration result would be. For that reason we did not pick consecutive frames for the calibration dataset, but instead used every 100th frame starting from the frame 100, for a total of five calibration frames.

From P we calculated the *RQ* decomposition which gave us the intrinsic values K and extrinsic values R and t, which are the rotation matrix and the translation vector, respectively. The intrinsic values are the internal features of the camera i.e. focal length and principal points. Extrinsic values describe the environment, that is the location and orientation of the camera with respect to an external coordinate system.

$$P \cong K(R|t)$$

$$P \Longrightarrow K, R, t \tag{3}$$

We did not have to focus on calculating the intrinsic and extrinsic values and only used them for debugging purposes. Instead, we utilised the P matrix to translate VICON data to the 2D image plane and saved that data as MS COCO labels into JSON files to be used in the model training. The calibration step was an essential part in this work and there were major problems with it. The issues are discussed in the next section.

6.5 Challenges with data pre-processing

Data pre-processing turned out to be a much more time-consuming and difficult task than we assumed when scheduled the work. There were lots of things to learn in data manipulation to modify data to correct format for processing with the used algorithms.

Some of the videos got corrupted when saving to a file which was not noticed until we started pre-processing the data. There were flickering and artefacts in all videos of the two latter subjects. These issues were not visible in the real time video stream projected to the laboratory screen during the event but appeared after saving the stream as video files. For this reason most of the data from the latter subjects had to be discarded.

We used the ffmpeg application to convert one video file into the AVI format for manual labelling. That conversion decreased the image quality which made the detection of the exact joint locations difficult. The reflective markers helped in detecting the correct locations. We claim that with better image quality the labelling process would have been easier and faster and probably the results would have been more accurate.

When we started to pre-process the data for the training dataset, we renamed the images with the subject name and an index number but we found out that AlphaPose was unable to find images with that name pattern. After renaming the images to six digit index numbers with leading zeros, the images were found.

We created a directory structure containing the images used for the training dataset and the test dataset and the labels in separate subdirectories. The labels are included in a MS COCO formatted annotation file. The path to the image location and to the annotation file were entered into the AlphaPose configuration file. Even though all the images were stored locally in the file system, the script also needed the URL of the images. Otherwise the training failed with the error message about missing data. However, the URL was not actually needed, since the issue was solved by adding any URL string with the correct directory structure, e.g.. "*http://xxx/train2017/000001.jpg*", to the appropriate field in the label file.

Converting the keypoints to the correct format was a relatively easy task with the Python script but there were some things to note in creating the annotation file. After creating the file with the keypoint and file location information, there were still multiple issues before we managed to get the training to start. The algorithm appeared not to be completely ready for users to train models with custom data. It did not provide a clear error message about what was preventing the training from starting, and as a result it took some effort to figure all requirements out.

In addition to the keypoint labels, we had to have bounding boxes of the persons in the images. The YOLO v3 algorithm was used to detect persons and to write the bounding box coordinates to a file. This file was then read by the script and the coordinates were written among the other data to the MS COCO formatted annotation file. The need to provide the bounding box information with the training labels was not known before the actual work and since the documentation did not contain anything about the dataset creation, this had to be figured out by ourselves. When reflecting back, this appears reasonable since AlphaPose is based on a two-step framework where the bounding box detection comes before the joint detection. Therefore, it seems clear that the bounding boxes have to be provided with the training annotations.

When detecting the bounding boxes with YOLO, some kind of file path length limitations were encountered. YOLO could not detect images from paths longer than some threshold, but after changing the hierarchy to be one level shallower the images were found.

Difficult problem with shaping data from VICON to be used as training data for machine vision with a normal video camera was something we were not prepared for at all. Managing different aspect ratios and camera distances posed a significant challenge. Our initial thoughts about that problem were just to export, scale and crop x and y -axis data from VICON 3D-point cloud to match video camera field of view but it soon appeared that it would not be as easy as that at all.

When creating the scripts to calibrate VICON data to match the video camera view, we noticed that during the recording session, the VICON-operator had to change the sample rate of VICON from 150 samples per second to 100. That did not sound a big deal during the recording session but when creating the scripts that difference in videos caused yet another problem to solve. Since videos were recorded with 30 frames per second and 100 samples per second is not divisible by 30, we were unable to get the exactly correct match between the VICON data and the video frames. We used a Python script to pick the nearest VICON data frame for each video frame. Of course this was not the perfect solution since the data was not from the exact same moment. We should have thought about what possible drawbacks there might be when the VICON operator said she changed the frequency but we did not. These frame rates are one issue to consider in the forthcoming studies.

Variable frame rate (VFR) in video files was another major issue with the data quality when considering the VICON calibration. The frame rate of the video files was changed to a constant one in the pre-processing phase but the data integrity was suffering from the conversion as some of the original frames had to be duplicated to increase the number of frames to match the desired constant frame rate. That causes mismatches between the video camera frames and the VICON point cloud coordinates. We should have reviewed the camera settings before recording. We suspect that the data inconsistency caused by converting the VFR from the video saffected the calibration performance.

Pre-processing phase took by far most of the time of the practical part of this work and that confirmed the known fact that data wrangling is the most time-consuming phase of data science workflow.

7 Model training

In this chapter we go through the phases of training the machine vision model for skier pose estimation with AlphaPose. We go through how models are trained and how existing models were utilised in this work as a base model for our own tailored dataset.

7.1 Training model from scratch

Training models with existing algorithms such as AlphaPose and existing pre-processed data can be very straightforward. This involves downloading a dataset consisting of relevant labelled images, like MS COCO, setting up a configuration file provided by algorithm developers and starting the training. We tested the training of one model from scratch with the full MS COCO dataset even though that was not necessary for this work since a similar pretrained model is provided in the AlphaPose model zoo. We used the configuration file *Fast Pose (DCN)*[28] by the AlphaPose developer team. It uses a *ResNet50*[29] - *DCN*[30] backbone and *YOLO* version 3[31] as the detector. ResNet50 is a 50-layer deep convolutional neural network trained with the ImageNet database and used for image classification. DCN refers to a deformable convolutional network. In deformable convolutional networks, each grid node is moved by a learnable offset, compared to the fixed grid used in regular convolutional networks. Convolutional neural networks are discussed in Section 3.4. YOLOv3 is an algorithm created for real-time object detection in videos. It utilises deep convolutional neural networks and is implemented using the *Keras* or OpenCV

[28] https://github.com/MVIG-SJTU/AlphaPose/blob/master/configs/coco/resnet/256x192_res50_lr1e-3_2x-dcn.yaml (18.4.2022)

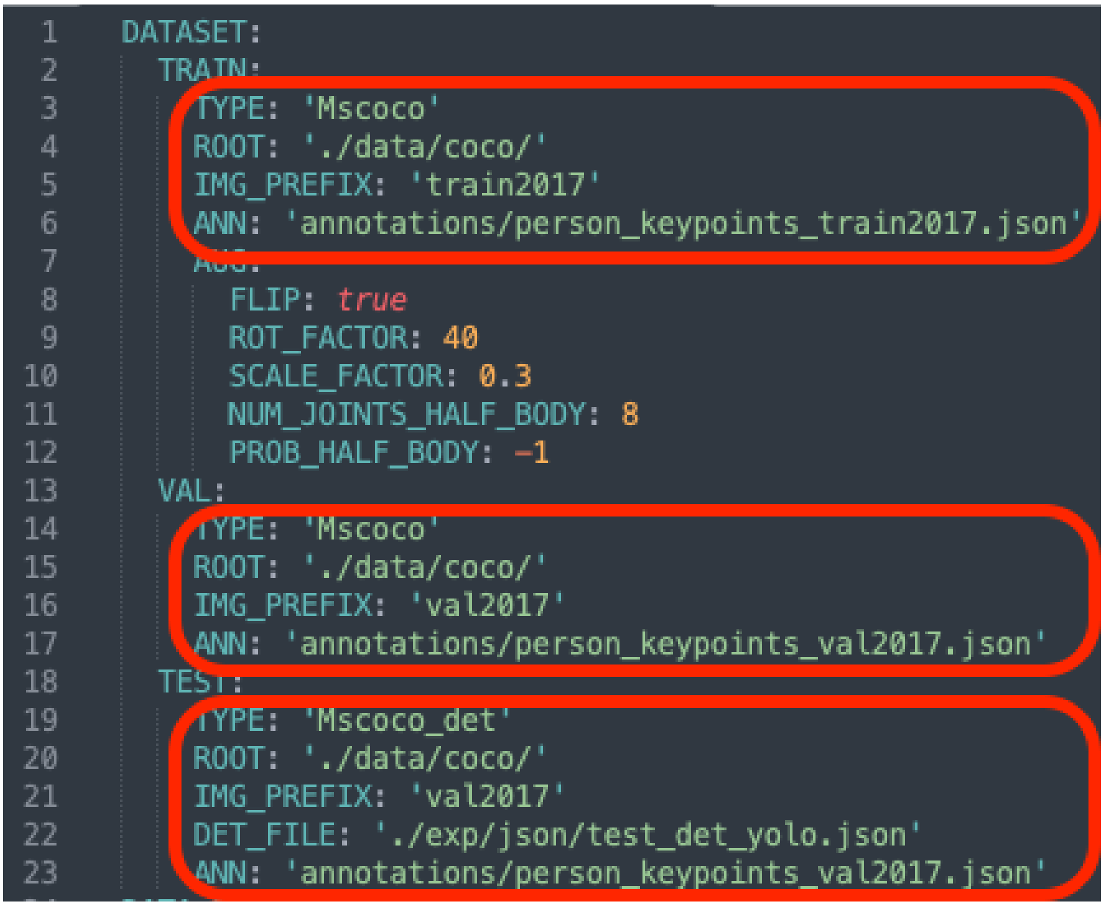
[29] https://se.mathworks.com/help/deeplearning/ref/resnet50.html;jsessionid=b7688a0fbe35f65703efa222f862 (18.4.2022)

[30] https://towardsdatascience.com/review-dcn-deformable-convolutional-networks-2nd-runner-up-in-2017-coco-detection-object-14e488efce44 (18.4.2022)

[31] https://viso.ai/deep-learning/yolov3-overview/ (18.4.2022)

libraries. The model is trained with keypoints from 64115 images of MS COCO 2017 training set and validated with 5000 images of MS COCO 2017 validation set.

Configuration files are formatted as *YAML*, as configuration files commonly are. YAML is a digestible data serialisation language.[32] The YAML acronym originates from the sentence: "Yet Another Markup Language". Figure 21 shows the example part of the configuration file containing the paths to one dataset.



```yaml
DATASET:
  TRAIN:
    TYPE: 'Mscoco'
    ROOT: './data/coco/'
    IMG_PREFIX: 'train2017'
    ANN: 'annotations/person_keypoints_train2017.json'
    AUG:
      FLIP: true
      ROT_FACTOR: 40
      SCALE_FACTOR: 0.3
      NUM_JOINTS_HALF_BODY: 8
      PROB_HALF_BODY: -1
  VAL:
    TYPE: 'Mscoco'
    ROOT: './data/coco/'
    IMG_PREFIX: 'val2017'
    ANN: 'annotations/person_keypoints_val2017.json'
  TEST:
    TYPE: 'Mscoco_det'
    ROOT: './data/coco/'
    IMG_PREFIX: 'val2017'
    DET_FILE: './exp/json/test_det_yolo.json'
    ANN: 'annotations/person_keypoints_val2017.json'
```

Figure 21: Snippet of the original configuration file used to train the model with the MS COCO dataset. The only things to verify when training with the existing dataset and configuration file were the paths to data.

Training this kind of neural network models with big image datasets is computationally very heavy. A powerful virtual machine from CSC *cPouta* cloud (Table 3) was used for the training but even with this kind of powerful GPU-

[32] https://circleci.com/blog/what-is-yaml-a-beginner-s-guide/ (18.4.2022)

accelerated machine it took two and half days to complete. More powerful and faster computing systems should be used if the aim is to train the models with multiple datasets and each model multiple times with different hyperparameters.

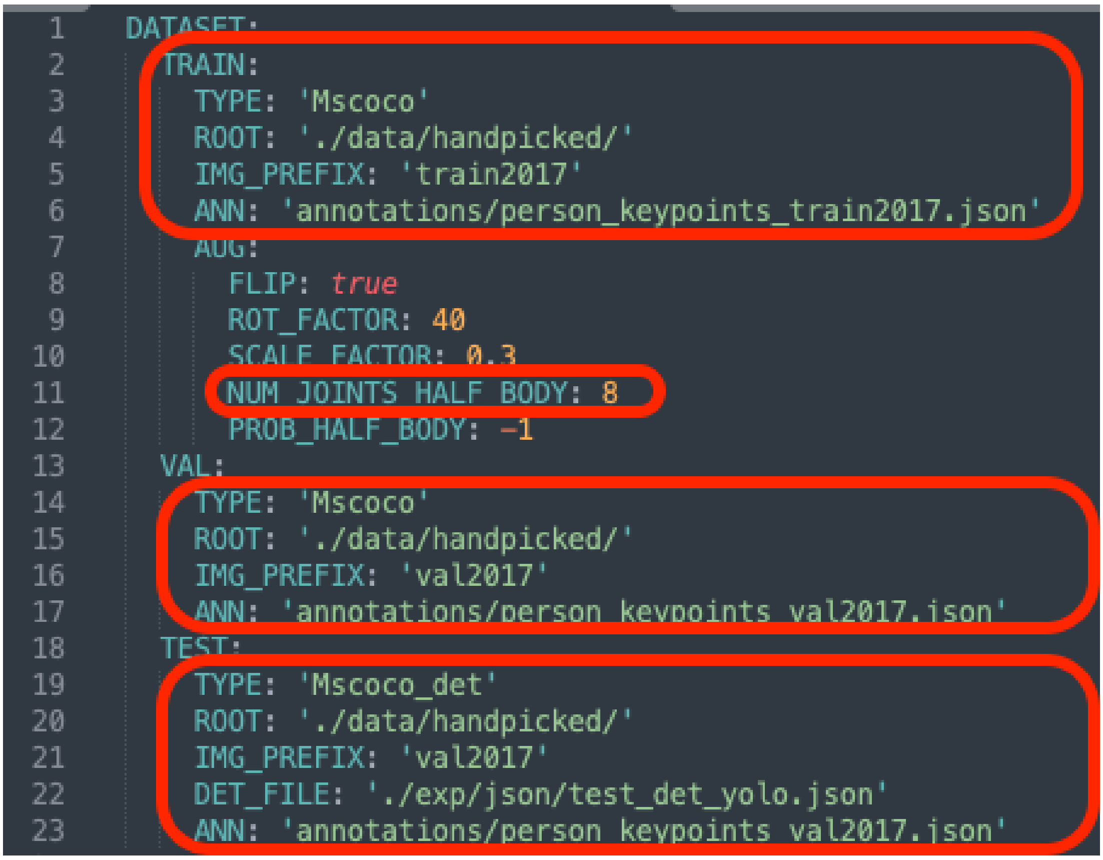
Table 3: Virtual machine specifications used in this study. The virtual machine is provisioned from CSC cPouta cloud and it is running Ubuntu 18.04.

Flavor	**Cores**	**GPUs**	**Memory (GiB)**	**Total disk (GB)**	**Memory/core (GiB)**
gpu.1.1gpu	14	1	112	1080	8

The AI partition of either the CSC *Puhti* supercluster or *Mahti* supercomputer[33] was planned to be used for further training and hyperparameter optimisation because of the far superior computing power and parallelism than of a single virtual machine. This was not needed in the end because of problems in getting enough training material. The fine tuning of existing models did not take long enough that the supercomputers would have been needed. While the time for training from scratch was calculated in days, the fine tuning with as small datasets as we used took just minutes or hours.

When training the model from scratch with our own custom dataset, only the same dataset paths and the number of half body keypoints to detect had to be changed. These changes are illustrated in Figure 22 with example paths used when training the first dataset. We used *handpicked* as the name of the directory and the model name for the first custom dataset described in Section 6.2. Changing the configuration file to work with the custom dataset was not complicated but to get the dataset to the correct format was a more complex task. The challenges with that were discussed in Section 6.5.

[33] https://research.csc.fi/csc-s-servers (30.1.2022)



```
DATASET:
  TRAIN:
    TYPE: 'Mscoco'
    ROOT: './data/handpicked/'
    IMG_PREFIX: 'train2017'
    ANN: 'annotations/person_keypoints_train2017.json'
    AUG:
      FLIP: true
      ROT_FACTOR: 40
      SCALE_FACTOR: 0.3
      NUM_JOINTS_HALF_BODY: 8
      PROB_HALF_BODY: -1
  VAL:
    TYPE: 'Mscoco'
    ROOT: './data/handpicked/'
    IMG_PREFIX: 'val2017'
    ANN: 'annotations/person_keypoints_val2017.json'
  TEST:
    TYPE: 'Mscoco_det'
    ROOT: './data/handpicked/'
    IMG_PREFIX: 'val2017'
    DET_FILE: './exp/json/test_det_yolo.json'
    ANN: 'annotations/person_keypoints_val2017.json'
```

Figure 22: Snippet from the beginning of the configuration file used for training a completely new model with our own manually labelled data. The notable parts are the paths for the data and the number of used joint keypoints.

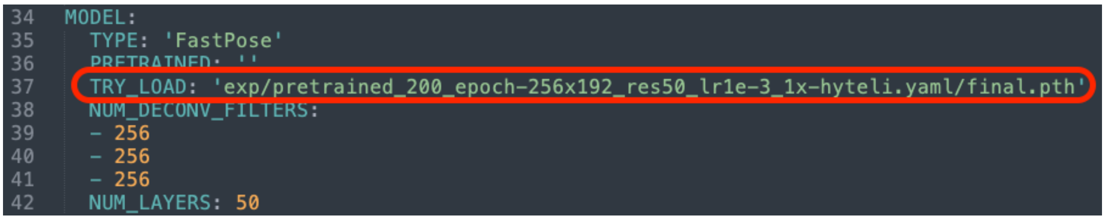
Since the Handpicked dataset consisted of only 1000 frame images and the second dataset VICON dataset was just 5000 frame images, they were not as heavy to train than with the full MS COCO dataset and therefore the supercomputers were not needed for the task. These trained models are later referred to as *base models* and the test results of those in skier pose estimation are compared in the Section 8.2.

7.2 Fine-tuning existing models

To modify existing solutions to be better in one specific task there often is no need to start the model training from scratch. There are many existing models available in algorithm developers' model zoos that can be used as starting points and for further fine-tuning with own data. AlphaPose provides pre-trained models such as Simple *Baseline*, *Fast Pose* and *HRnet* and config files on how to utilise those.[34] Fine-tuning existing models was done in this work. The model used as a baseline was the same Fast Pose (DCN) by AlphaPose developer team described in the previous section.

Fine tuning was in the end a very easy task, but figuring out how to do it correctly needed some research. The documentation about fine-tuning on the AlphaPose GitHub page was lacking and apparently the code was supposed to be used with pre-made models. As a result, there were hard-coded variables and values in the code and in the configuration files that needed to be discovered. The error messages from the AlphaPose training script were generic and the debugging of the errors was difficult.

When fine tuning an existing model, the same configuration file was used than when training a completely new model. Only the dataset paths had to be modified and the used original model path had to be added to the 'TRY LOAD' field in the MODEL section. (Figure 23)



```
MODEL:
  TYPE: 'FastPose'
  PRETRAINED: ''
  TRY_LOAD: 'exp/pretrained_200_epoch-256x192_res50_lr1e-3_1x-hyteli.yaml/final.pth'
  NUM_DECONV_FILTERS:
  - 256
  - 256
  - 256
  NUM_LAYERS: 50
```

Figure 23: Example snipped from the configuration file where the existing model was set for loading. The self trained model is used in that example case.

In this work we trained two base models from scratch, one with the Handpicked dataset and one with the VICON dataset. These are described in Section 7.1. In addition to those, we created three *fine tuned models*. The first two fine tuned models were based on the original MS COCO model from the AlphaPose repository which was fine tuned with our custom datasets, the first one with the Handpicked dataset and the second with the VICON dataset. The last fine tuned model was the Handpicked model which was fine tuned with the VICON dataset. The performance of these models is discussed in Section 8.2.

[34] https://github.com/MVIG-SJTU/AlphaPose/blob/master/docs/MODEL_ZOO.md (18.4.2022)

8 Experiments

This chapter describes what experiments were performed, how they were measured and what are the results gained. Section 8.1 describes the used performance metric, *Mean square error* (MSE), which was used to compare models. Section 8.2 introduces the results we got when we tested the models against a comparison dataset.

8.1 Performance evaluation

Performance evaluation was one of the things to study in this work. The existing pose estimation algorithms use their existing cost functions to measure accuracy and error during their training and validation processes. The performance is calculated against the given ground truth data and it is reported on dataset or batch level during training and in the end. Since in our work we wanted to verify the model performance in real world use cases, skier pose estimation in this case, a domain-specific performance measurement method was developed. For the evaluation of the center of mass and the force component calculations, the joint specific errors would be needed. For this, the existing batch or dataset level accuracy and error metrics were not sufficient.

The used metric to evaluate performance for comparison of models was the *mean squared error* (MSE) of the joints. It was calculated with the *sklearn.metrics.mean_squared_error* function from the *ScikitLearn* library. More specifically, the MSE value is the mean of summed and squared pixel distance of joint coordinates from the ground truth value. Ground truth in this context is the manually labelled coordinate. The function is run for every image, and the values from every image is summed and the result is divided with the total joint count from the test dataset (30 * 6 = 180 joints). The MSE error values reported in Table 5 are averages of all the joints in the test dataset.

The comparisons were made between the original COCO model from the AlphaPose GitHub repository and self-trained models. VICON synchronisation was not good enough for VICON models to be considered as a success, but those results were still accepted for the comparison table in Section 8.2.

For measuring model performance, a small test dataset was used. This dataset was the same 30 frame calibration dataset described in Section 6.2 which was used when calibrating the VICON data to the 2D image plane. This dataset contains 5 images from 3 skiers with both skiing styles and different speeds, 30 images in total. These images were excluded from the training set, so the model had not seen them before testing. The images were labelled manually with the Fiji application as described in Section 6.2.

8.2 Results

The models created in the work were the three base models described in Section 7.1, and three fine tuned models described in Section 7.2. Table 4 shows the models created. The first base model which was used as a reference in the comparison, was the original MS COCO model by the AlphaPose developers. It was trained from the MS COCO dataset collected by Microsoft for object recognition algorithm development. The dataset is described in Section 4.1. We compared the other two base models, the VICON model which was trained with the 5000 calibrated VICON data frames and the Handpicked model, which was trained with the 1000 manually labelled images of the one skier, against that reference model. The fine tuned models, MS COCO fine tuned with VICON data, MS COCO fine tuned with Handpicked data and Handpicked model fine tuned with VICON data. All the datasets used for training are described in Table 2 in Section 6.2.

Table 4: The Original dataset column tells which dataset was used for training the base model. The Fine tuning dataset column tells which dataset was used to fine tune the model.

Name	**Type**	**Original dataset**	**Fine tuning dataset**
MS COCO model	Base model	MS COCO dataset from Microsoft	-
VICON model	Base model	VICON dataset	-
Handpicked model	Base model	Handpicked dataset	-
MS COCO model fine tuned with VICON dataset	Fine tuned model	MS COCO dataset from Microsoft	VICON dataset
MS COCO model fine tuned with Handpicked dataset	Fine tuned model	MS COCO dataset from Microsoft	Handpicked dataset
Handpicked model fine tuned with VICON dataset	Fine tuned model	Handpicked dataset	VICON dataset

Table 5 shows a comparison of the models. Comparison metric is the MSE described in Section 8.1. Lower MSE value is better. We started testing with the three *base* models, and progressed to fine tuned models. The first tested model, MS COCO, yielded quite good results with all subjects. With the second, the VICON model, the results were worse than expected. The inferior performance was even clearly visible from the output images. The third model, the so-called *Handpicked model*, which was trained from scratch by using manually labelled material from one subject. This model showed great performance when tested with the other videos of that same subject, but yielded poor performance with the other subjects. By comparing these three base models we noticed that the original COCO model was clearly the best one. However the Handpicked model performed better than COCO for the other videos of the same subject which it was trained from. For the other subjects the performance was however worse than with the COCO. For forthcoming studies, more heterogeneous manually labelled dataset should be used for the training. The poor performance of the VICON model was not expected and that indicated that the calibration process was not successful.

After comparing the base models we studied the performance of the fine tuned models. We used the COCO model as a base for fine tuning two models. Fine tuning the base with the VICON model decreased the results significantly but the results were slightly better than with the original VICON model. The result was still far from usable. Next we fine tuned the base model with the manually labelled Handpicked dataset. This model was the key indicator to observe how fine tuning can improve performance. A MSE of 49 was better than the original COCO model's 144. The handpicked dataset was collected from one person only, but still the fine-tuned model worked better also for other persons than the original COCO model. Figure 24 shows a comparison of two frames from videos created with the original COCO model, and the best-performing fine-tuned model.

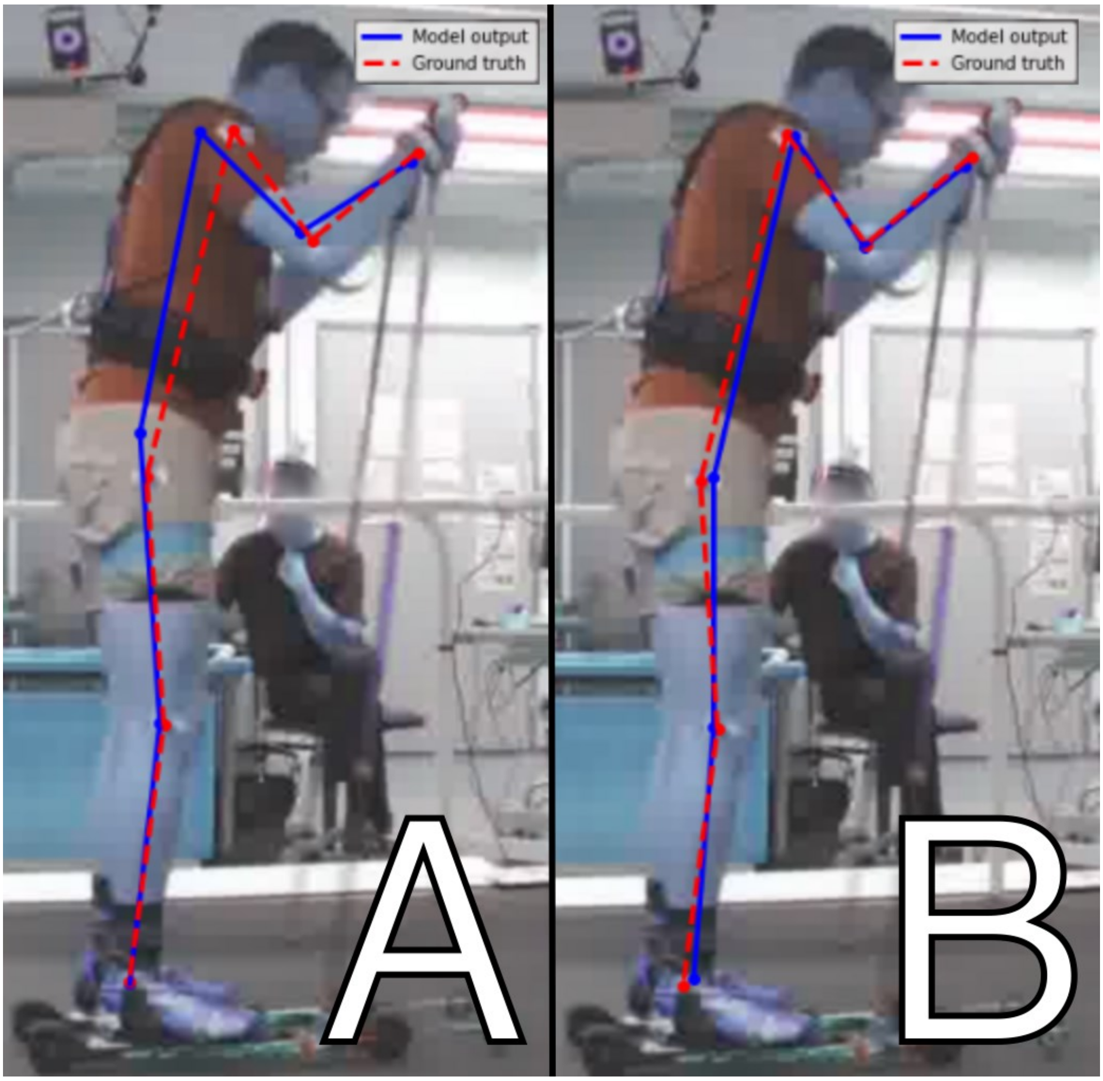


Figure 24: Comparison between A) Original COCO model from AlphaPose (MSE 143) and the best-performing B) fine-tuned COCO model (MSE 48). The ground truth is marked with a dashed red line and the output of the model is marked with a solid blue line.

Figure 25 demonstrates the difference between the best-performing model, the COCO model fine tuned with the Handpicked dataset and the worst-performing VICON model which was trained from scratch. At last we tried to fine tune the Handpicked model with the VICON data. When the result with this base model was also much inferior after fine tuning with the VICON data, we came to the conclusion that the quality of the calibrated VICON data was not good enough and using it for the training will reduce the quality of the models. As a result of this observation we decided not to try fine tuning the VICON model with other datasets.

All six compared models were tested with the Calibration dataset, which is described in detail in Section 8.1. It was a 30-frame datasets which was more heterogeneous than the larger training datasets. It included 10 frame images from each of a total of three skiers. Five images were from the double poling technique and five from diagonal skiing for each skier.

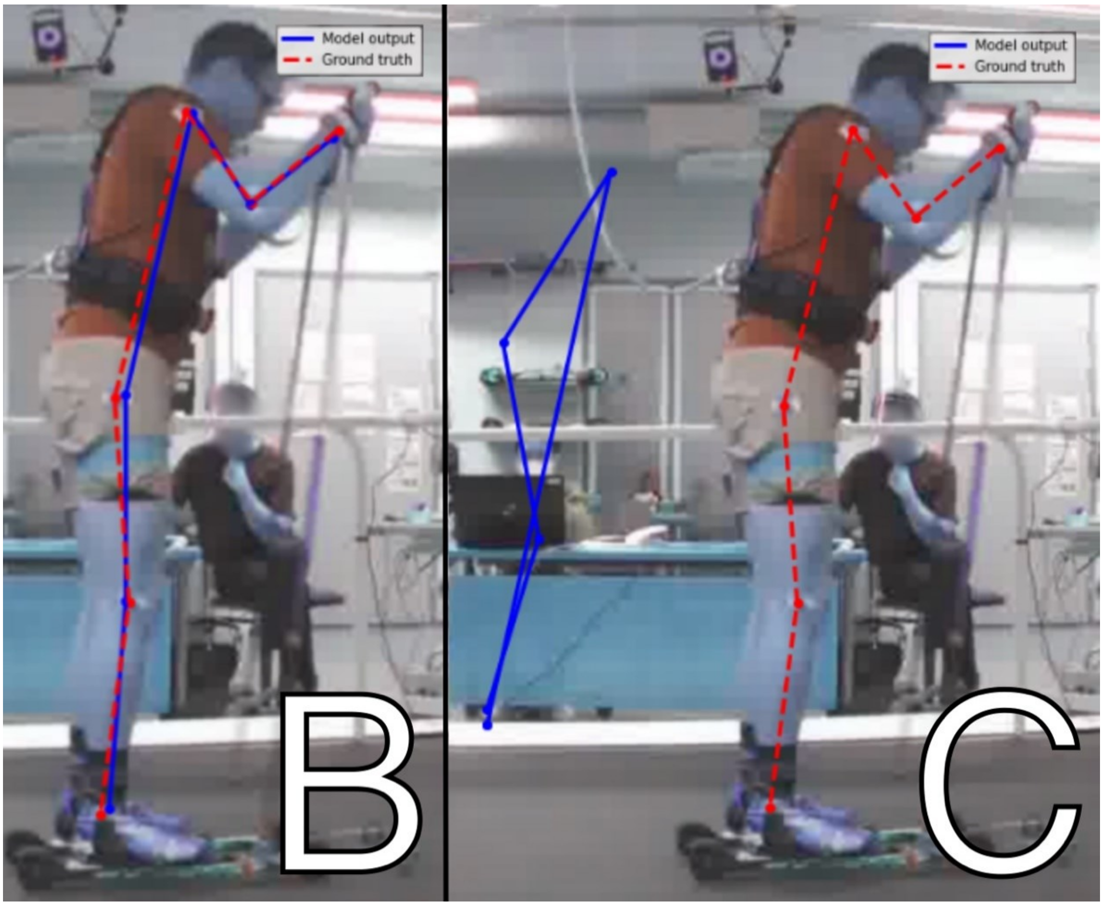


Figure 25: Demonstration of the significance of the differences between the best-performing Original COCO model fine tuned with the Handpicked dataset (B) and the worst VICON model which was trained from scratch (C). The skeleton in image C is in the wrong location and the points are not even close to the pose of the skier. The ground truth is marked with a dashed red line and the output of the model is marked with a solid blue line.

Table 5: Model comparison table with the training dataset information. For fine tuned models both original dataset and fine tuning dataset are included. The table also includes the model descriptions and values of the used performance metric (Mean Squared Error). MSE is calculated by summing and squaring all coordinate errors as pixel distances across all images and calculating the mean value. A lower MSE value indicates more accurate detection of the skier joints. The calibration dataset is used for evaluating the performance. The datasets are described in detail in Section 6.2.

Type	**Name**	**MSE**
Base	MS COCO model	143
Base	VICON model	8345
Base	Handpicked model	904
Fine tuned	MS COCO model fine tuned with VICON dataset	7920
Fine tuned	MS COCO model fine tuned with handpicked dataset	49
Fine tuned	Handpicked model fine tuned with VICON dataset	7272

According to this comparison, fine tuning existing models with the carefully manually labelled domain-specific training data leads to better performance. The body joint detection accuracy of the existing general models might not be precise enough for the skier pose estimation task, but the accuracy can be improved by fine tuning the model with a self-created domain-specific dataset. For future research we suggest that the COCO or other relevant general purpose pose estimation model is used as a base, and it is fine tuned with carefully manually labelled domain-specific dataset which is crafted from the image frames of the multiple subjects.

9 Discussion

The main research question in this work was to find out if there is a way to replace the VICON motion capture system with an ordinary video camera and machine vision algorithms in the studied application setting. The results suggest that skier pose estimation can indeed be done with an ordinary video camera and machine vision, but the joint detection accuracy is not good enough for customers needs with the existing pre-trained models. The accuracy can be improved by fine-tuning the models even with relatively small amounts of self-created domain-specific training data. In conclusion, VICON can probably be replaced but further research is still needed to increase the model accuracy.

While studying the topic for the main research question also the additional questions got answered. Fine tuning AlphaPose was challenging due to lacking documentation and some hard-coded values in code and configuration files that had to be discovered when debugging runtime errors. Additional data to improve the results in skier pose estimation were gathered by arranging an event where four skiers were recorded while performing different skiing styles and speeds according to a test protocol created by a skiing coach. These videos were used to create the datasets to fine tune existing pre-trained pose estimation models. Datasets were created both with manual labelling and by calibrating VICON point cloud data. The Fiji image processing application was used for manual labelling and a developed tool based on an open source implementation of Direct Linear Transform was used to calibrate point cloud data to match the video camera field of view. The conversion from point cloud data turned out to be much more difficult than originally estimated. As a result, even though we were able to create a dataset from VICON data, the dataset could not be properly utilised for skier pose estimation and we could not use it as ground truth to compare the other models against. Instead, the comparisons were made against manually labelled data. Mean squared error in the joint level was used to compare all results to the manually-labelled ground truth.

The final research question was about how to decide whether the model was good enough to replace VICON. We managed to get better results with our own trained model compared to the original models provided by the AlphaPose team, but due to the problems in getting enough training data, more research is needed before that question can be fully answered. In the following we will discuss in more detail the problems encountered during the study and lessons learned.

Dataset quality was a major problem during the work. To get better results we would have needed a bigger and more heterogeneous training dataset. More material should have been labelled manually and from multiple skiers and multiple videos instead of just the one video from one skier. This also caused the model comparison to lack a variety of the validation data. The reason to manually label all frames of a single video was the expectation of getting more accurate VICON-based data. Since the automatic labelling from the VICON data turned out to be not usable because of imprecise calibration, the resulting validation data was too homogenous. We noticed the need for more hand-labelled data too late in the project, so there was no time to do more manual labelling. There were also issues with video quality. The videos of the two latter skiers had been corrupted when saving to files and there were flickering in the frames as a result. The flickering was not present in the live view of the camera during the recording. So half of the material had to be discarded. We argue that a better quality dataset would have yielded better results in model performance.

We used the MS COCO model in the work even though it was not what originally had been requested. The MS COCO pose model does not include feet, which is essential for the targeted skier pose estimation application. MS COCO was selected for the first experiments of the training because of the easier approach due to multiple existing models, datasets and annotation files. The aim was to develop scripts and processes with MS COCO and then progress into the HALPE model which would have better suited the need by including the foot keypoints. We had prepared the training data and scripts to work with MS COCO and as time was running low, we decided to focus on testing with MS COCO to determine whether this kind of training is even a possible option with the AlphaPose algorithm. Had we started to refactor our codes and to prepare the training data again, the remaining time budget in the project would have run low. MS COCO was easier to start with, because it had more pretrained models available, but since it lacked foot detection, it was not suitable for the targeted application after all. In further studies of this topic it would be reasonable to start straight with the HALPE model to include foot detection from the start of the work. In that case the startup would need more work, but the extra steps and time to refactor the codes would be avoided.

The image quality decreased when converting the video into single frame images for labelling. That caused some problems with hand picking the correct spots for the markers. Finding a better tool for converting video to images would have been worth spending some time. Image quality might have affected the training results as well. One way for creating a better training set could have been to manually pick the best frames from the calibrated VICON data and use those for training. In forthcoming studies, getting better tools for data manipulation and preparation would be worth spending more time and focus.

VICON point cloud data can be used to create training data, but calibrating three-dimensional point cloud data to two-dimensional image plane is a challenging task and that task was not solved to a sufficient level during the project. Probably the biggest issues in calibration were the video camera used. The camera was set to capture varied frame rate video and that had to be transformed to a fixed 30 frames per second framerate. This transformation caused some issues since it resulted in duplicate frames. Decreased image quality from the video to images conversion might also have negatively affected the calibration.

The DLT algorithm on the basic level does not take lens distortions into account and those can affect the performance of the calibration. Some other algorithm could have performed better with the camera used. Using more calibration frames could have provided better results and if the calibration would have been approached iteratively, by calibrating first with a small set and then introducing more calibration frames, the results might have been better. More research is needed to improve the calibration in this context. During the data collection event the VICON frame rate had been changed from 150 fps to 100 fps and that also caused problems. Synchronising 100 fps point cloud data to 30 fps video did not match very well. For some reason the calibration for one person succeeded much better than for the others, even though the number of synchronisation images was the same. The reason for this was not studied in the scope of this work.

In the forthcoming studies the quality and validity of the collected data should be verified immediately to avoid missing parts of valuable data because of corrupted video files or problems with data pre-processing because of changed settings during the collection event.

10 Conclusion

In this work we showed that machine vision can be used for human pose estimation when skiing on a treadmill. Fine tuning an existing AlphaPose model with custom training data is possible and it can increase model accuracy in skier pose estimation problems. The results indicate that even though machine vision can work in skier pose estimation, this approach in practice requires very careful work in creating the training dataset for fine-tuning the existing models. The training data can be created manually by picking desired joint positions from every frame by hand but this is very slow and time consuming.

The point cloud data from the VICON motion capture system can be calibrated to match the video data, but future research is needed to get the calibration to work better. The calibration method needs to be an iterative process and it requires careful fine tuning to find out the best synchronising point for every frame of the video. The synchronisation is possible to achieve, but there is much practical work to get it precise enough. The dataset used in the calibration might have been too small for the task and a bigger dataset would perhaps have yielded better results. Still, this would not have fixed issues with the frame rates.

Even though the work did not provide a model that could replace VICON, we showed that existing general-purpose pose estimation models can be improved in detecting skier pose by fine tuning with a custom made training dataset. VICON data can be calibrated to be used as training data, but the calibration process needs more research. Research with this topic continues in Vuokatti supported by CSC.

Data preprocessing or cleaning is an important and time-consuming part of any data analytics or machine learning project. Often the preprocessing phase is underestimated and it is the necessary evil before the tasks that are considered more important and interesting can be started. Andrew Ng (2021)[35] has stated that the time

[35] https://www.youtube.com/watch?v=06-AZXmwHjo (7.4.2022)

spent in increasing the data quality in the preprocessing phase can be equally effective than doubling the training dataset size. According to our experiences from this thesis work, the preprocessing phase is really time consuming and important. It definitely should not be treated lightly.

References

Abdel-Aziz, Y.I. and Karara, H.M. (1971). Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry, *Proceedings of the Symposium on Close-Range Photogrammetry*, 26–29 January 1971, Urbana, Illinois, pp. 1—18.

Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition* (pp. 3686-3693).

Cao, C., Liu, F., Tan, H., Song, D., Shu, W., Li, W., ... & Xie, Z. (2018). Deep learning and its applications in biomedicine. *Genomics, proteomics & bioinformatics,* 16(1), 17-32.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291-7299).

Chen, Y., Tian, Y., & He, M. (2020). Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding,* 192, 102897.

Chollet F. (2018). *Deep Learning with Python*. Manning Publications Co. New York, The United States of America.

Corazza, S., Muendermann, L., Chaudhari, A. M., Demattio, T., Cobelli, C., & Andriacchi, T. P. (2006). A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach. *Annals of biomedical engineering,* 34(6), 1019-1029.

Danielsen, J. (2018). Energetics and dynamics of double poling cross-country skiing. *Doctoral theses at NTNU*, 2018: 365.

Fang, H. S., Xie, S., Tai, Y. W., & Lu, C. (2017). Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2334-2343).

Gkioxari, G., Hariharan, B., Girshick, R., & Malik, J. (2014). Using k-poselets for detecting people and localizing their keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3582-3589).

Goodfellow I., Bengio, J., Courville A. (2016): *Deep learning.* The MIT Press Cambridge, Massachusetts London, England.

Güler, R. A., Neverova, N., & Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7297-7306).

Göpfert, C., Pohjola, M. V., Linnamo, V., Ohtonen, O., Rapp, W., & Lindinger, S. J. (2017). Forward acceleration of the centre of mass during ski skating calculated from force and motion capture data. *Sports Engineering, 20(2)*, 141-153.

Hidalgo, G., Raaj, Y., Idrees, H., Xiang, D., Joo, H., Simon, T., & Sheikh, Y. (2019). Single-network whole-body pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6982-6991).

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., & Schiele, B. (2016, October). Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European conference on computer vision* (pp. 34-50). Springer, Cham.

Li, Y. L., Xu, L., Liu, X., Huang, X., Xu, Y., Wang, S., ... & Lu, C. (2020). Pastanet: Toward human activity knowledge engine. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 382-391).

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9), 1281-1289.

Myler, H. R. (1999): *Fundamentals of machine vision.* Vol. 33. Spie Press.

Ohtonen, O. (2019). Biomechanics in cross-country skiing skating technique and measurement techniques of force production. *JYU dissertations.*

Ohtonen, O., Linnamo, V., Göpfert, C., Lindinger, S.J. (2020). Effect of 20 km simulated race load on propulsive forces during ski skating. *International Journal of Performance Analysis in Sport*.

Ohtonen, O., Ruotsalainen, K., Mikkonen, P., Heikkinen, T., Hakkarainen, A., Leppävuori, A., Linnamo, V. (2016). Online feedback system for athletes and coaches. In *A. Hakkarainen, V. Linnamo & S. Lindinger (Eds.) Science and Nordic Skiing III*. Jyväskylä University Printing House, 53-60.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., & Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4929-4937).

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., ... & Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods, 9*(7), 676-682.

Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5693-5703).

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation, 3*(4), 323-344.

Waggoner, B. (2013). *Compression for great video and audio: master tips and common sense.* Routledge.

Welde, B., Stöggl, T. L., Mathisen, G. E., Supej, M., Zoppirolli, C., Winther, A. K., ... & Holmberg, H. C. (2017). The pacing strategy and technique of male cross-country skiers with different levels of performance during a 15-km classical race. *PLoS One, 12*(11), e0187111.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence, 22*(11), 1330-1334.