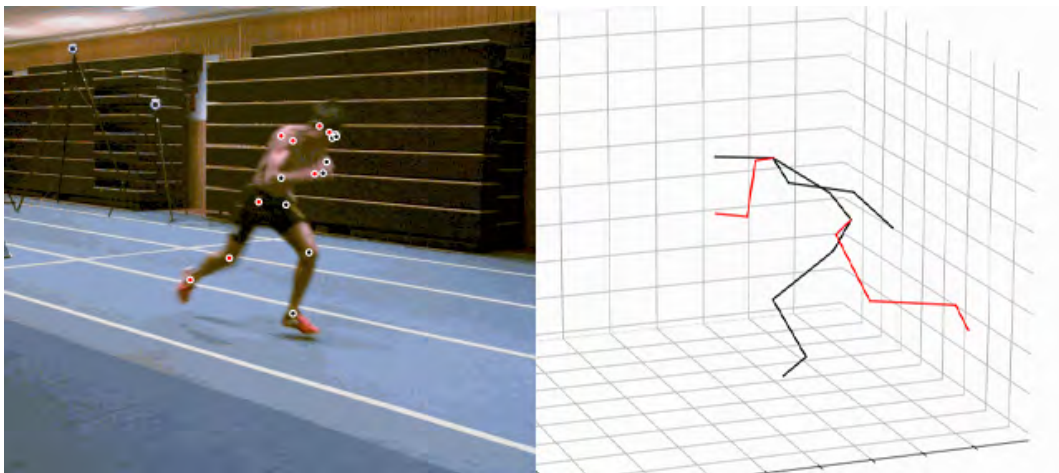




CHALMERS



Biomekanisk 3D-visualisering för löpare

Markörlös optisk rörelsefångst genom artificiell intelligens

TIFX04-22-10

Kandidatarbete vid institutionen för fysik

Tijana Barac, Alexander Boström, Lisa Lövgren, Erik Ryde-
hag, Richard Svensson, Alvin Ånestrand

Institutionen för fysik

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2022
www.chalmers.se

KANDIDATARBETE 2022

Biomekanisk 3D-visualisering för löpare

Markörlös optisk rörelsefångst genom artificiell intelligens

Författare

Tijana Barac, IT
tijanab@student.chalmers.se

Alexander Boström, D
alebost@student.chalmers.se

Lisa Lövgren, F
lisalov@student.chalmers.se

Erik Rydehag, M
rydehag@student.chalmers.se

Richard Svensson, TM
ricsven@student.chalmers.se

Alvin Ånestrand, TM
aalvin@student.chalmers.se

Handledare

Magnus Karlsteen
Docent
Chalmers Tekniska Högskola

Darko Sarovic
Bihandledare
Leg läk, fil kand, doktorand
Göteborgs Universitet

Uppdragsgivare

Jeremy Pryce
Verksamhetsansvarig
IFK Göteborg Friidrott

Janne Wanhainen
Utbildningsansvarig
Svensk Friidrott



CHALMERS

Institutionen för fysik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2022

Biomekanisk 3D-visualisering för löpare

Markörlös optisk rörelsefångst genom artificiell intelligens

Tijana Barac, Alexander Boström, Lisa Lövgren, Erik Rydehag, Richard Svensson, Alvin Ånestrand

© Tijana Barac, Alexander Boström, Lisa Lövgren, Erik Rydehag, Richard Svensson, Alvin Ånestrand, 2022.

Handledare: Magnus Karlsteen, Institutionen för fysik. Darko Sarovic, Institutionen för Neurovetenskap och Fysiologi, Göteborgs Universitet.

Uppdragsgivare: Janne Wanhainen, Svenska Friidrottsförbundet. Jeremy Pryce, IFK Göteborg Friidrott

Examinator: Lena Falk, Institutionen för fysik

Kandidatarbete 2022

Institutionen för fysik

Chalmers tekniska högskola

SE-412 96 Göteborg

Telephone +46 31 772 1000

Omslagsbild: Uppritad 3D-modell över AI-förutspådd biomekanisk modell utifrån videomaterial.

Typsättning i L^AT_EX

Göteborg, Sverige 2022

Biomekanisk 3D-visualisering för löpare

Markörlös optisk rörelsefångst genom artificiell intelligens

Tijana Barac, Alexander Boström, Lisa Lövgren, Erik Rydehag, Richard Svensson, Alvin Ånestrand

Institutionen för fysik

Chalmers Tekniska Högskola

Sammandrag

I löpning är löparens hållning och ledvinklar avgörande för prestation och biomekanisk modellering är därför användbart för rörelseanalys. Då en tränare bedömer en löpares teknik kan tränaren spela in löparen för att sedan analysera videon. Utan externa verktyg begränsas tränarens analys av videon då bedömning av många ledvinklar är svåra eftersom videon är tvådimensionell. I arbetet har data på personers rörelsemönster, i form av koordinater för utvalda nyckelpunkter, samlats in och artificiella neuron nätverk har nyttjas för att genomföra rörelsefångst baserat enbart på videodata. Utifrån en video på en löpande person kan nyckelpunkters positioner förutsägas i 3D med en genomsnittlig avvikelse på 115 millimeter i samtliga led, och datan kan användas för beräkning av ledvinklar. Löpningen visualiseras i en webbapplikation skapad för användarvänlighet och som utnyttjar neuron nätverket. Webbapplikationen har en interaktiv design och genom att ladda upp en video över en löpsekvens fås en intuitiv förståelse för löparens rörelse i varje tidpunkt. Det finns även möjligheter för vidareutveckling av produkten gällande applicering på och analys av andra typer av rörelser.

Nyckelord: Markörlös optisk rörelsefångst, gånganalys, maskininlärning, artificiellt neuron nätverk, poseuppskattning, webbapplikation

Abstract

The runner's posture and joint angles are crucial for running performance and biomechanical modeling is therefore helpful for movement analysis. To examine a runner's technique, a coach can film the runner and then analyze the video. Without external tools, the coach's analysis in terms of some joint angles will be limited as the video is two-dimensional. In this project, data of running person's movement patterns has been collected in the form of coordinates for selected keypoints, and artificial neural networks have been used to perform motion capture based only on video data. Using video footage of a running person, the keypoint positions can be predicted in 3D with a deviation of 115 millimeters in all dimensions, and the data can be used to calculate joint angles. The run was visualized in an web application developed for ease of use and that uses the neural network. The web application has an interactive design and by uploading a video over a running sequence, an intuitive understanding of the runners movement at every time through the run can be obtained. There are also opportunities for further development of the product for application and analysis of other types of movements.

Keywords: Markerless optical motion capture, gait analysis, machine learning, artificial neural network, pose estimation, web application

Författarnas tack

När vi till en början fick uppdraget som ligger till grund för detta kandidatarbete från vår uppdragsgivare, Svenska Friidrottsförbundet, var det ett mycket öppet projekt. Det huvudsakliga syftet med arbetet var att utveckla en produkt för att optimera träning och prestationsutveckling för friidrottare. Vi kommer från varierande program och har olika intressen, vilket vi sett som en styrka i arbetsprocessen och bidragande faktor till resultatet. Under den 16 veckor långa arbetsperioden har vi samarbetat med olika partners och mottagit hjälp från flera håll, vilket möjliggjort vår slutprodukt och underlättat resan dit.

Först och främst vill vi uttrycka vår tacksamhet till vår handledare och bihandledare, Magnus Karlsteen respektive Darko Sarovic. Karlsteen har genom hela arbetets gång varit en stöttepelare med oerhört mycket erfarenhet, samt varit närvarande från början till slut. Sarovic har genom sitt engagemang och intresse för projektet och för vad vi kan skapa, varit en nyckelperson vid utformningen av problemformuleringen och den riktning kandidatarbetet tagit. Genom sina kunskaper inom både biomekanik och löpning har hans idéer och kritik varit otroligt värdefulla.

Vi vill också tacka Janne Wanhainen och Jeremy Pryce från Svenska friidrottsförbundet för sin visade passion, optimism och framförallt den tilltro de givit oss. Extra stort tack till löparna som ställde upp och som utgör grunden för datan.

Slutligen vill vi rikta ett enormt stort tack till gänget från Qualisys. Vi tackar Patrik Almström och Martin Skoger som deltog vid mättillfällena, liksom Magnus Sjölin som dessutom varit kontaktperson och organisatör. Sist men inte minst, vill vi tacka Katarina Hjältman som varit huvudansvarig för insamlingstillfällena i Friidrottens Hus, samt den som svarat på våra påföljande frågor. Utan Qualisys hade datan som ligger till grunden för hela projektet inte varit möjlig att samla in med nuvarande precision. Tack för att ni tog er tid och visade intresse för vad vi arbetat med!

Kandidatgruppen, Göteborg, maj 2022

Innehåll

1	Introduktion	1
1.1	Syfte	1
1.2	Problemformulering	2
1.2.1	Delpproblem och översiktlig arbetsprocess	2
1.3	Avgränsningar	3
2	Teori	4
2.1	Biomekanisk data	4
2.1.1	Gånganalys	4
2.1.2	Biomekanisk data för idrott	4
2.1.3	Rörelsefångst	5
2.1.4	Qualisys	6
2.2	Maskininlärning och artificiella neuronnätverk	6
2.2.1	Data	7
2.2.2	Djupinlärning	7
2.2.3	Överföringsinlärning	10
2.3	Webbutveckling	10
2.3.1	Front-end	11
2.3.2	Back-end	11
3	Metod	12
3.1	Skapande av dataset för träning av artificiellt neuronnätverk	12
3.1.1	Datainsamling av biomekanisk rörelse	12
3.1.1.1	Mätuppställning	12
3.1.1.2	Markörer och nyckelpunkter	14
3.1.1.3	Löparna	17
3.1.1.4	Genomförande av testförsöken	17
3.1.2	Behandling av insamlad data	18
3.2	Djupinlärning	18
3.2.1	Litteraturöversikt	19
3.2.2	Bedömning av nätverk	19
3.2.2.1	Val av djupinlärningsbibliotek	19
3.2.3	Skapandet av ett basnätverk	20
3.2.4	Överföringsinlärning	20
3.2.4.1	Detectron2	20
3.2.4.2	Temporalt faltningsnätverk	21

3.3	Utveckling av webbapplikation	22
3.3.1	Funktionalitet	22
3.3.2	Vinkelberäkning	23
4	Resultat	24
4.1	Skapande av dataset för träning av artificiell intelligens	24
4.1.1	Bildresultat för datainsamlingen	24
4.2	Djupinlärning	26
4.2.1	Detectron2	26
4.2.2	Temporalt faltningsnätverk	26
4.3	Webbapplikation	27
5	Diskussion	28
5.1	Skapande av dataset	28
5.1.1	Felanalys av det första mättillfället	28
5.1.2	Förbättringsområden	29
5.2	Djupinlärning	29
5.2.1	Detectron2	29
5.2.2	Temporalt faltningsnätverk	30
5.2.3	Bias	30
5.3	Webbapplikation	31
5.4	Framtida utvecklingsmöjligheter	31
5.5	Samhälleliga och etiska aspekter	32
6	Slutsats	33
	Referenser	34
A	Faltningsnätverk	I
A.1	Faltning	I
A.2	Pooling	II
B	GDPR-avtal	IV
C	QTM och mocap-data	V
D	Matlabkod för konvertering av QTM-data till textfil	VI
E	Webbapplikation	VII
F	Exploderande kostnad	VIII

Introduktion

Sport och träning är aktiviteter som en stor andel av världens befolkning ägnar mycket tid åt. Det kan exempelvis vara för att umgås, förbättra hälsan, koppla av eller för att förbättras inom en idrott. Det finns dock mycket som kan gå fel. Allt från värre hälsorelaterade händelser som skador från överansträngningar eller fel teknik, till mildare sportsliga problem som att fel träningsmetodik används och förbättrade resultat uteblir. I en idrottsrelaterad tränar och adept relation är det oftast tränaren som har huvudansvaret över hur träningen ska läggas upp och utvärderas för att optimera adeptsens resultat och utveckling. En fråga som kan ställas i denna situation är vilket stöd teknologiska lösningar bland annat kan ge för att underlätta utvärderingen av adeptsens prestationer.

Svenska Friidrottsförbundet är uppdragsgivaren till detta kandidatarbete. De söker en produkt som möjliggör automatiserad registrering av biomekanisk data. Friidrottstränaren Darko Sarovic, bihandledare för detta projekt, har beskrivit att det finns ett stort behov av att mäta vinklar i leder för att veta vad som kan förbättras gällande tekniken i en sport eller idrott. Sarovic problematiserar exempelvis att han idag vid analys av utväxling ur startblock för löpare är tvungen att filma löparen från flera olika vinklar och sedermera beräkna löparens ledvinklar utifrån videomaterialet manuellt. Detta är en arbetsprocess som både är tids- och arbetskrävande. Om det skulle finnas en produkt som möjliggör automatiserad biomekanisk rörelseanalys utifrån en tagen videovinkel skulle friidrottstränare likt Sarovic kunna spara mycket tid på liknande arbetsprocesser och deras tränarroll tidseffektiviseras. Behovet av en produkt som automatiserar registreringen finns både hos elitidrottare som vill nå ännu högre, och för motionären som vill förbättra sin träning.

Metoden som användes i detta projekt för att ta fram den biomekaniska datan är optisk rörelsefångst, vilket innebär att rörelsen hos ett föremål eller person registreras genom analys av videoinspelning. För de som tränar finns det ett behov av att kunna göra detta med endast en kamera, och utan markörer som sätts på kroppen. Inspejningen bör också kunna göras från ett stort antal vinklar och varierande avstånd.

1.1 Syfte

Syftet med projektet är att skapa en produkt som under löpträning kan stödja tränare och de som tränar i att analysera rörelser och därigenom öka atleters prestation. Syftet är också att möjliggöra en vidareutveckling av arbetet, där den biomekaniska datan används för vidare analys. Exempelvis kan registrering genomföras av hur rörelser bör se ut för ett stort antal sporter och idrotter, analysera kraftutveckling under sprint, eller löpeekonomi för långdistanslöpare. Mer specifikt är målet med kandidatarbetet att skapa en produkt som nyttjar maskininlärning för att kvantifiera och återskapa biomekanisk rörelse.

1.2 Problemformulering

Projektets huvudproblem är att med hjälp av tillgänglig hårdvara såsom kameror, utveckla och optimera en löparens rörelsemönster. Slutprodukten ska bestå av en *minimum viable product* (MVP), en produkt med precis tillräckliga funktioner för att vara användbar.

Valt fokusområde är att återskapa och kvantifiera biomekanisk rörelse. Uppgiften blir således att skapa mjukvara som utifrån kamerainspelat videomaterial över en löparens utväxling ur startblock och delar av efterföljande accelererande löpsekvens. Återskapar en digital 3D-modell av löparen som återspeglas av den inspelade löpsekvensen. Från 3D-modellen ska olika ledvinklar hos löparen kunna extraheras och avläsas i en skapad applikation, som genom analys i ett senare skede kan bidra till atletens prestationsutveckling.

1.2.1 Delproblem och översiktlig arbetsprocess

Projektet kan delas upp i tre större delproblem, vardera bestående av några konkreta steg. De tre större delproblemen bearbetas parallellt under arbetets gång. Rapportens struktur är skriven på ett sådant sätt som motsvarar den parallella arbetsgången. Nedan listas dessa tre större delproblem och deras ingående steg.

1. Skapa ett dataset för att kunna skapa en biomekanisk 3D-modell.

1. Undersöka vilka nyckelpunkter och följande beräkningar som krävs för att nå de behov uppdragsgivaren och användaren har av vår slutgiltiga produkt.
2. Genomföra datainsamling via löpare från IFK Göteborg Friidrott.
3. Hämta information från tidigare arbeten och vetenskapliga studier med problemet att representera en 3D-modell utifrån flera 2D-bilder.
4. Undersök vilken mätutrustning som finns tillgänglig eller går att införskaffa.
5. Ta fram ett standardiserat arbetssätt för att samla in data.
6. Dela upp dataset i tre delar: träning, validering och testning. Proportionen mellan delarna beror på den totala mängden av insamlad data. Större datamängd tillåter större proportion av träningsdata för maskininlärning.

2. Träna ett artificiellt neuron nätverk.

1. Undersöka publikt tillgängliga dataset som använts för liknande arbeten.
2. Välja passande arkitektur för artificiella neuron nätverket.
3. Välja programmeringsspråk och djup-inlärningsbibliotek för träning av artificiella neuron nätverket.
4. Träna, utvärdera och finjustera det artificiella neuron nätverket.

3. Skapa ett grafiskt användargränssnitt av löparens videodata.

- Undersöka vilka språk och verktyg är bäst lämpat för webbutveckling.
- Möjliggöra att användaren kan skicka in en video och få den analyserad med hjälp av det artificiella neuron nätverket.
- Visa resultatet från analysen i form utav en grafisk representation med nyckelpunkterna.

- Möjliggöra att användaren kan manipulera resultatet som att backa, spela upp långsamt, etc.
- Göra webbapplikationen så användarvänlig som möjligt.

1.3 Avgränsningar

För att arbetet realistiskt ska vara genomförbart under tiden som det fortskrider, behöver ett antal avgränsningar göras. En av dessa är uteslutande av hårdvaruutveckling. Stora delar av den hårdvara som ämnas nyttjas i arbetet uppskattas av arbetsgruppen kräva kunskaper utöver dess kunnande för att framställas på egen hand. Olika typer av videokameror är exempel på hårdvaror som detta berör. Arbetsgruppen väljer därav att förhålla sig till befintligt framtagna och åtkomlig hårdvara för att lägga ökat fokus på arbete med utveckling av mjukvara.

Ytterligare en avgränsning som tillämpas berör antalet individer i bild som det artificiella neuronätverket avser analysera. Här avgränsar sig arbetet till att enbart analysera en individ åt gången, d.v.s. en löpare i bild, med ändamålet att förenkla och öka precisionen i nätverkets förutsägelser. Om flera löpare finns i bild så kommer inte applikationen att fungera som tänkt.

Vidare avgränsas projektet till att använda ett artificiellt neuronät för monokulär vy, där endast en kamera används, eftersom våra användare och slutprodukt behöver poseuppskattning med endast en kamera.

2

Teori

I detta avsnitt behandlas teoretisk bakgrund för detta projekt. Teorin ämnar ge en förståelse för projektets metoddel, samt ge en helhetsbild av projektets bakgrund. Först behandlas teori gällande biomekanik och databehandling. Därefter behandlas maskininläring och artificiella neuronnätverk. I sista avsnittet behandlas utveckling av webbapplikation.

2.1 Biomekanisk data

Biomekanik är studien om strukturen, funktionen och rörelsen hos en levande kropp utifrån mekaniska aspekter. Biomekanisk data är data över de mekaniska aspekterna, och kan exempelvis vara ledvinklar, lemmars hastighet, och belastning. Här behandlas för detta projekt relevanta områden inom biomekanik, vilka är gånganalys, biomekanisk data för idrott, samt rörelsefångst.

2.1.1 Gånganalys

Ett viktigt användningsområde av biomekanik är *gait analysis*, d.v.s. gånganalys, vilket är studien av lokomotion, djurs förflyttning. Studien innefattar att kvantifiera storheter som lemmars hastighet och vinklar i leder under förflyttning, samt härleda egenskaper om djuret utifrån dessa storheter. Hädanefter syftar gånganalys på analys av människors förflyttning.

I detta projekt används gånganalys för atleter, men det finns många andra användningsområden som visar på vad en vidareutveckling av projektet kan vara. Genom gånganalys kan en person identifieras enbart på sin gångstil, vilket kallas *gait recognition*, gångigenkänning. Varje person har en unik gångstil, vilket gör detta möjligt [1]. Gånganalys kan även användas i medicinska sammanhang för bland annat diagnostisering och mätning av effekter av medicinska behandlingar. Ett exempel på medicinsk applikation av gånganalys är en studie där det användes under behandling av barn med en sjukdom kallad *crouching cerebral palsy*, hukande cerebral pares [2].

2.1.2 Biomekanisk data för idrott

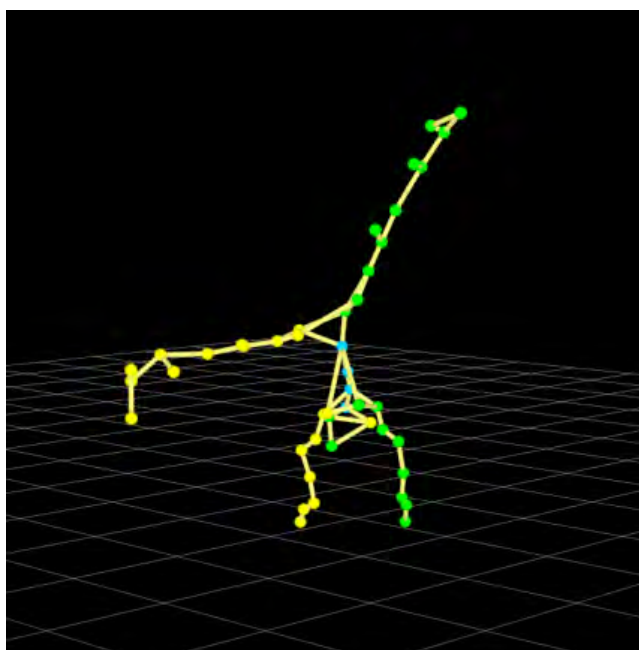
Ett område inom biomekanik är *sports biomechanics*, sportbiomekanik, vilket är den biomekaniska studien av atleter och sportaktiviteter. I detta område kan gånganalys appliceras för träningsoptimering och för att undvika skador. I och med att löpning är vad som analyseras i detta projekt är denna användning av biomekanik och gånganalys i särskilt fokus. Studier som undersöker relationen mellan biomekaniska parametrar och löpningsrelaterade skador har dock inkonsekventa resultat [3] i det att sambanden mellan parametrar och prestation är oklara, vilket försvårar användandet av gånganalys för att undvika skador.

Löpekonomi är ett mått på hur mycket energi eller syre som förbrukas under löpning. Gällande användning av biomekanik vid träningsoptimering för löpning visas användbarheten exempelvis

av en studie där relation mellan löpekonomi och ett antal biomekaniska parametrar undersöktes [4]. Bra löpekonomi innebär en effektiv användning av kroppens resurser under löpning. De biomekaniska parametrarna som undersöktes var bland annat stegvinkel, svängtid, steglängd och stegfrekvens.

2.1.3 Rörelsefångst

Processen i vilken verkliga rörelser för utvalda nyckelpunkter hos ett objekt kartläggs kallas *motion capture* (mocap), rörelsefångst. Då mocap genomförs med hjälp av kameror kallas det *optical motion capture*, optisk rörelsefångst [5]. Då inga markörer används för optisk rörelsefångst kallas det markörlös optisk rörelsefångst, vilket är typen av mocap projektets produkt ska göra. Objekten kan exempelvis vara ett djur så som en människa, eller ett mer maskinärt system så som en industrirobot. Oftast genomförs kartläggningen av nyckelpunkterna i kartesiska koordinater (x,y,z) och översätts sedan till en digital representation i antingen två- eller tre-dimensioner. Kortfattat kan man sammanfatta mocap som en teknik som möjliggör översättning av verkliga objekts rörelser till digitala återspeglings av rörelserna. Ett exempel på detta ges i Figur 2.1.



Figur 2.1: Exempel på hur mocap-data visualiseras. Här är en person som hjular [6].

Valet av objektets nyckelpunkter görs utifrån motivet av att bäst kunna återspegla objektet på ett önskvärt sätt under genomförandet av rörelserna. Exempelvis i en situation där en löpars rörelser ska undersökas skulle valet av kartlagda nyckelpunkter göras utifrån målbilden av att återge löparen på ett optimalt sätt för vidare analys av dess löprörelser. Vanligtvis återfinns dessa nyckelpunkter vid förbindelser av stela delar. För en person skulle dessa nyckelpunkter exempelvis kunna vara olika leder som sammanbinder skelettdelar som önskas kartläggas under de verkliga rörelserna av intresse. Uppskattning av en människas hållning i 3D kallas *3D human*

pose estimation, mänsklig poseuppskattning i 3D. Mocap för människor är i princip mänsklig poseuppskattning under rörelser.

2.1.4 Qualisys

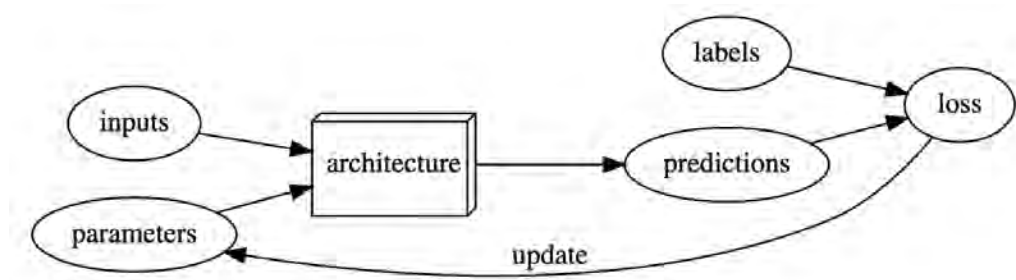
Qualisys är ett företag med huvudkontor i Göteborg som beskriver sig själva som en ledande leverantör av mocap och 3D-positionsspårningssystem. Företaget har kontor i Europa, Nordamerika samt Asien och har över 30 års erfarenhet av att förse olika industrier med högteknologiska kamerasystem och expertis [7]. Företaget specialiserar sig i sportrelaterade mätningar och gånganalys där egen framtagen mjuk- och hårdvara används för optimal spårning av rörelser.

QTM (*Qualisys Track Manager*) är Qualisys eget framtagna mjukvara som används vid datainsamling och senare för att återge datan. Från QTM kan datan även exporteras till diverse format för vidarebehandling, däribland formaten *.mat* för MatLab samt *.tsv* för Microsoft Excel. Bland utvecklad hårdvara finns kameran Arqus som kan detektera och spåra markörer i det tredimensionella rummet med hög precision.

2.2 Maskininlärning och artificiella neuronnätverk

För att analysera biomekanisk data från videoinspelningar används maskininlärning. Det innebär att träningsdata används för att uppdatera parametrar hos en som används för att göra förutsägelser. Genom uppdatering av arkitekturs parametrar förbättras arkitekturs förmåga att göra förutsägelser. En modell av hur maskininlärning fungerar har skapats av Arthur Samuel och sammanfattas i Figur 2.2.

För maskininlärning används en arkitektur, ofta kallad modell. Arkitekturen tar in indata, och använder parametrar för att skapa *predictions*, förutsägelser, om indatan. Ett exempel är objektigenkänning, då en bild ges som indata och arkitekturen förutsäger om bilden innehåller ett visst objekt. Om indatan har *labels*, etiketter, kan en förutsägelser korrekthet utvärderas i form av *loss*, förlust, enligt en *loss function*, förlustfunktion, och arkitekturs parametrar uppdateras utifrån förlusten. Etiketterna är *ground truth*, grundsanning, och förutsägelserna är bättre desto närmare grundsanningen de är. För objektigenkänning visar etiketten om bilden faktiskt innehåller det sökta objektet eller ej. Är en förutsägelse bra har den låg förlust. Det finns även en annan funktion kallad *cost function*, kostnadsfunktion. Medan förlustfunktionen bedömer prestation för en indata används kostnadsfunktionen för utvärdering av arkitekturs prestation på ett dataset, det vill säga ett stort antal indata. När parametrarna är uppdaterade har arkitekturen tränats. Då den tränade arkitekturen används för att göra förutsägelser på indata utan etikett kallas det *inference*, inferens. Syftet med att träna arkitekturen är just att kunna göra inferens, då man vill att arkitekturen ska kunna förutsäga något om indatan.



Figur 2.2: Samuels modell av maskininlärning [8].

2.2.1 Data

En mängd data kallas för dataset. Det behövs vanligtvis tre dataset för maskininlärning. Det är träningsdata, valideringsdata, samt testdata. Träningsdata används för att uppdatera arkitekturens parametrar enligt beskrivningen ovan. Därefter används valideringsdata för att avgöra hur bra arkitekturen presterar på ny data, data som ej använts för träning, vilket visar om parametrarna har överanpassats. En kostnadsfunktion används för denna utvärdering. Överanpassning innebär att parametrarna anpassats för mycket efter träningsdatan, vilket gör att arkitekturen presterar dåligt på ny data. Den tredje datamängden, testdata, används för att kontrollera överanpassning av arkitekturen. Då skaparna av mjukvaran anpassar arkitektursstrukturen för att förbättra dess prestation på valideringsdatan kan återigen en överanpassning ske, med resultatet att arkitekturen presterar dåligt på data som varken är med i träningsdatan eller valideringsdatan.

Existerande data kan ofta användas för att skapa mer data genom *data augmentation*, dataökning. Ett exempel på detta är då data är i form av bilder, då en bild kan vändas horisontellt för att få en ny bild.

Beroende av vilken data som använts under utveckling kan det uppstå bias. Det innebär att arkitektursföresägelser ej motsvarar verkligheten på ett bra sätt, utan exempelvis påverkas av mänskliga förutfattade meningar. Det finns tre olika sorters bias som kan uppstå under utvecklingen av ett artificiellt neuronnätverk [9]:

- *Sample bias* - uppstår p.g.a. egenskaper hos datapopulationen.
- *Annotator bias* - uppstår p.g.a. fördomar hos skapare av dataset.
- *Temporal bias* - uppstår p.g.a. tidpunkten när maskininlärningen utförs.

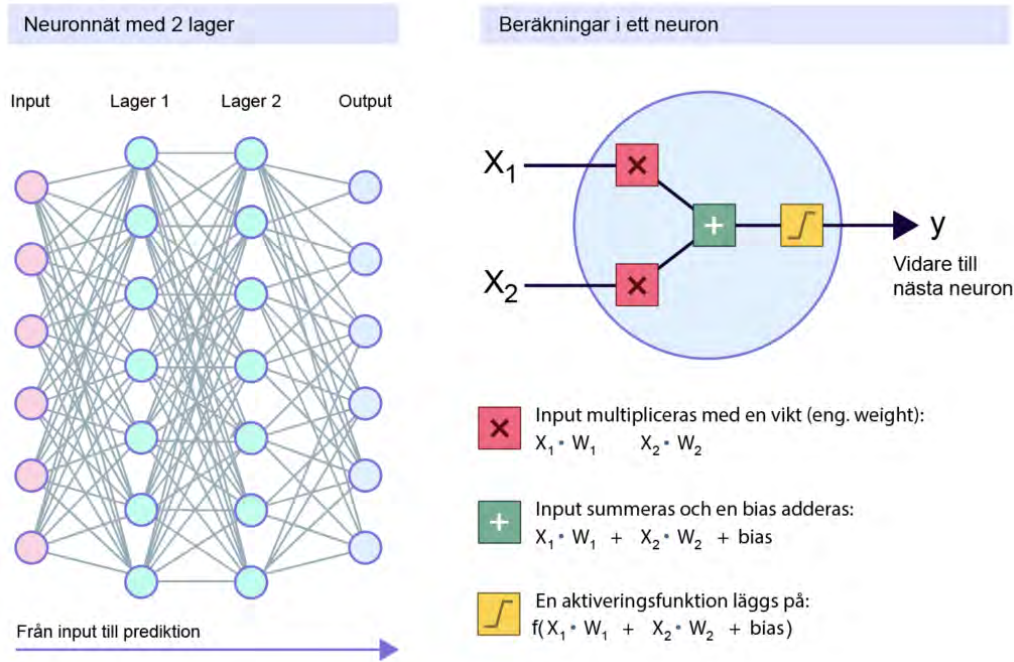
Det viktiga vid produktutveckling är att vara medveten om att det kan uppstå bias under olika stadier av utvecklingen för kunna undvika det.

2.2.2 Djupinlärning

En form av maskininlärning är djupinlärning. För djupinlärning används ett artificiellt neuronnätverk, förkortat ANN. ANN består av en samling noder, kallade neuroner, sammankopplade med viktade kanter, och är inspirerat av strukturen hos ett biologiskt nervsystem.

Strukturen hos ett ANN illustreras i Figur 2.3, tillsammans med en översikt över hur signaler propagerar genom nätverket. Neuronerna är ordnade i tre typer av lager. Det finns ett indata-

lager, ett utdata-lager, och dolda lager mellan dem. Det finns ingen övre gräns på antal dolda lager i ett ANN. Det första lagret, indata-lagret, hanterar obehandlad indata. Signaler skickas därefter från lager till lager tills det kommer till utdata-lagret. I figuren skickas därmed signaler från vänster till höger. Djupinlärning syftar på att ett nätverk med flera dolda lager används för maskininlärning, därav är det ”djupt”.



Figur 2.3: Illustration av struktur hos ANN [10].

Då indata tas emot av indata-lagret aktiveras neuroner i det lagret. Neuroners aktivering i ett lager påverkar aktivering av neuroner i nästa lager, vilket kallas framåtpropagering. På detta sätt leder aktivering av neuroner i indata-lagret till aktivering av neuroner i utdata-lagret, vilket ger arkitekturs förutsägelse.

En neuron j får som indata $p_j(t)$ vid en tidpunkt t . Indatan till arkitekturen utgör indatan till neuronerna i indata-lagret. För neuroner i ett annat lager bestäms indatan av det föregående lagrets neuroner. Aktiveringen hos neuron j vid tiden $t + 1$ visas i ekvation 2.1,

$$a_j(t + 1) = f(a_j(t), p_j(t)). \quad (2.1)$$

$a_j(t)$ är neuron j :s aktivering vid tidpunkt t , f är en aktiveringsfunktionen, och $p_j(t)$ är indata. En neuron har även en utdatafunktion, f_{ut} , som visas i ekvation 2.2. Denna bestämmer neuronens utdata, $o_j(t)$, som senare används under beräkning av andra neuroners indata.

$$o_j(t) = f_{ut}(a_j(t)) \quad (2.2)$$

Beräkning av indata vid tidpunkt t för neuron j visas i ekvation 2.3.

$$p_j(t) = \sum_i o_i(t)w_{ij} + b_j \quad (2.3)$$

$p_j(t)$ och $o_i(t)$ är definierade enligt beskrivningen ovan. i betecknar en neuron kopplad med en riktad kant till neuron j . w_{ij} är vikten för kanten mellan neuron i och j , och b_j betecknar en neurons bias. Bias är i detta sammanhang något annat än brister i arkitekturs förutsägelser, vilket också kallas bias och behandlas i avsnitt 2.2.1. Det är en parameter som behövs för att möjliggöra större flexibilitet för aktivering av en neuron. Genom bias är aktiveringen inte endast beroende av andra neuroners utdata.

Som beskrivet ovan utvärderas det artificiella neuronnätverkets förutsägelse för en indata med en förlustfunktion. Den betecknas som $(w, b) \rightarrow E(w, b)$, där w är alla vikter och b är alla bias. w och b utgör tillsammans nätverkets parametrar. Förlustfunktionen för en indata används för att minska kostnadsfunktionen för en hel datamängd. För minimering av kostnadsfunktionen uppdateras arkitekturs parametrar, det vill säga vikter och bias. En process kallad bakåtpropagering kan användas för detta. För ett ANN görs bakåtpropagering genom optimeringsmetoden *gradient descent*, lutande nedstigning. Uppdatering av parametrar genom lutande nedstigning är beskrivet i ekvation 2.4 och 2.5.

$$w_{ij,ny} = w_{ij,gammal} - \eta \frac{\delta E}{\delta w_{ij}} \quad (2.4)$$

$$b_{j,ny} = b_{j,gammal} - \eta \frac{\delta E}{\delta b_j} \quad (2.5)$$

Parametern η är arkitekturs inlärningstakt. Arkitekten tar emot en indata, gör en förutsägelse som utvärderas med förlustfunktionen, och parametrarna uppdateras på ett sätt som minskar förlustfunktionen och därmed ökar arkitekturs förmåga att göra en förutsägelse utifrån indatan. Genom att uppdatera parametrarna för varje indata i träningsdata minskas förlusten hos arkitekturs förutsägelse för varje data i träningsdatan. Om förlustfunktionen och kostnadsfunktionen är valda på ett bra sätt kommer detta leda till en minskning av kostnadsfunktionen för träningsdatan. Då det artificiella neuronnätverkets parametrar uppdaterats för träningsdatan är det tränat, och kan utvärderas genom att beräkna kostnaden för nätverkets förutsägelser på valideringsdata.

Det finns flera typer av ANN. Den typ av nätverk som används i detta projekt är *convolutional neural network* (CNN), faltningsnätverk. CNN används vanligtvis då bilder ges som indata. Ett av nätverken för detta projekt är en typ av CNN kallat *temporal convolutional network* (TCN), temporalt faltningsnätverk. Se Appendix A för mer information om CNN och TCN.

Det finns ett stort antal begrepp relaterade till djupinlärning och maskininlärning. Här ges en översikt över begrepp som är relevanta för detta projekt.

Slumptalsfrö Då datorer genererar slumpässiga tal används ett *random seed*, slumptalsfrö.

Om slumptalsfröet är samma vid generering av två slumpal kommer talen vara identiska.

Regularisering Då åtgärder tas för att minska tendensen att ett nätverk överanpassas kallas det regularisering.

Bortfall Under träning innebär *dropout*, bortfall, att vissa noder tillfälligt och slumpmässigt ignoreras. Detta är en regulariseringsmetod.

Epok En *epoch*, epok, är träning av ett nätverk på all data i träningsdatasetet. Vanligtvis tränas nätverket i fler än en epok, vilket innebär att nätverket tränas på all träningsdata flera gånger.

Kanal Djupet hos en tensor behandlad under faltning är antalet *channels*, kanaler, för tensorerna (se Appendix A).

Filterstorlek Vid faltning ges en kernels dimensioner av *filter size*, filterstorlek (se Appendix A).

Sats En *batch*, sats, är en mindre del av ett dataset.

Satsnormalisering För att göra ANN snabbare används *batch normalization*, satsnormalisering, vilket bland annat görs genom skalning av lagrets indata. Satsnormalisering tillåter en högre inlärningstakt [11].

Inlärningstaktsförfall Träningen av nätverket optimeras genom att starta med en hög inlärningstakt som sedan minskas, vilket kallas *learning rate decay*, inlärningstaktsförfall. Hastigheten för förfallet kallas *decay factor*, förfallsfaktor.

Varians Ett mått på hur mycket en arkitektur kan anpassas efter träningsdata är varians. Hög varians ger att nätverket kan få stor skillnad i kostnad gällande träningsdata, valideringsdata och testdata.

Blandad precision Nätverket tränas med *mixed precision*, blandad precision, då nätverkets parameterdata sparas genom både 16-bitars och 32-bitars representation för högre hastighet och mindre minnesåtgång. [Bounding box] En *bounding box* markerar var ett objekt är i en bild genom att rama in det i en rektangel. [Instanssegmentering] Då en bild delas upp för att markera vilka delar i en bild som föreställer vissa objekt kallas det *instance segmentation*, instanssegmentering.

Lifting Ett nätverk kan användas för att först förutsäga nyckelpunkters 2D-positioner. Då ytterligare ett nätverk används för att förutsäga 3D-positioner utifrån det första nätverkets utdata kallas det *lifting*.

2.2.3 Överföringsinlärning

Överföringsinlärning är ett område inom maskininlärning som behandlar hur ett program som tränats för att lösa en uppgift kan användas för att utveckla ett nytt program som kan lösa en annan, liknande uppgift. Materialet som det första programmet tränats på kallas för källdomän, och uppgiften programmet utför kallas för källuppgift. Materialet och uppgiften för det nya programmet är måldomän respektive måluppgift.

Överföringsinlärning används bland annat då det är brist på data med etiketter för att träna ett ANN. Parametrarna hos redan tränat ANN, som tränats på data från källdomänen, tränas på nytt. Man tränar då om parametrar hos vissa av lagren på data från måldomänen för att få ett ANN som presterar bra trots bristen på data. Resterande parametrar lämnas som de är. Då ett lagrets parametrar ej tränas på nytt är det ett fryst lager.

2.3 Webbutveckling

För att kunna projektets ANN behövdes ett verktyg för att visualisera och göra produkten användbar där användaren kan ladda upp en video på sin löpning och få den analyserad. Webbutveckling är paraplybegreppet för utvecklingen av webbapplikationer och dess delar. Det innefattar bl.a. webbdesign, hantering av data, säkerhet- och klient-server hantering. De två större delarna är front-end och back-end.

2.3.1 Front-end

Front-end kan sammanfattas som det en användare kan se och interagera med på en hemsida. Knappar, bilder och text är några exempel på vad som förknippas med front-end. För utvecklingen av detta projekt var dessa språk och verktyg relevanta i front-enden:

HTML *HyperText Markup Language* är ett märkspråk som berättar hur innehållet på en hemsida ska visas och ser till att rubriker, paragrafer, menyer, bilder etc. hamnar på rätt plats.

CSS *Cascading Style Sheets* utgör en hemsidas design och kan användas för att bestämma storlek, form och färg på olika element. CSS är väsentligt för att en hemsida ska kunna visas rätt på flera olika plattformar.

Javascript Det som ska hända när en användare interagerar med hemsidan, till exempel trycker på en knapp eller skrollar, bestäms med språket *Javascript*.

React.js Ett Javascript-bibliotek som underlättar byggandet av interaktiva användargränssnitt är *React.js*. År 2021 var React.js det populäraste bibliotek för webbutveckling bland utvecklare enligt en undersökning gjord av StackOverflow [12].

2.3.2 Back-end

Back-end sköter webapplikationens logik och består av kod och databaser på en server. Det är därifrån front-enden hämtar sin information. Till skillnad från front-end är back-end mer flexibelt när det kommer till val av programmeringsspråk. För utvecklingen av detta projekt var dessa språk och verktyg relevanta i back-enden:

Python Programmeringsspråket som användes för back-enden var *Python*.

Flask Ett microramverk som använder sig av Python är *Flask*. Flask används för kommunikation mellan back-end och front-end.

Lokal server En server där den lokala värddatorn omdirigerar till den privata IP-adressen 127.0.0.1 [13] kallas *localhost*, lokal server. En webapplikation som körs på en privat IP-adress kan endast nås från värddatorn.

3

Metod

Nedan behandlas projektets metod bestående av tre delar; datainsamling och skapande av ett dataset för träning av artificiellt neuronnätverk, djupinlärning med artificiella neuronnätverk innefattande en litteraturöversikt och skapande av ett basnätverk för utveckling av nätverket samt dess inlärning, och slutligen framtagning av en webbapplikation.

Nyckelpunkterna produkten har i uppgift att uppskatta valdes efter diskussion med bihandlaren Sarovic, som är friidrottstränare och läkare. Nyckelpunkterna är kopplade till leder som används och är viktiga inom löpning och genom att utnyttja deras position kunde ledvinklar i löpsteget beräknas. De utvalda lederna visas nedan i Tabell 3.1.

Tabell 3.1: Vinklar i leder som beräknades med framtagna nyckelpunkterna.

1. Nacke	2. Rygggrad
3. Vänster armbåge	4. Höger armbåge
5. Vänster knä	6. Höger knä
7. Vänster fotled	8. Höger fotled

3.1 Skapande av dataset för träning av artificiellt neuronnätverk

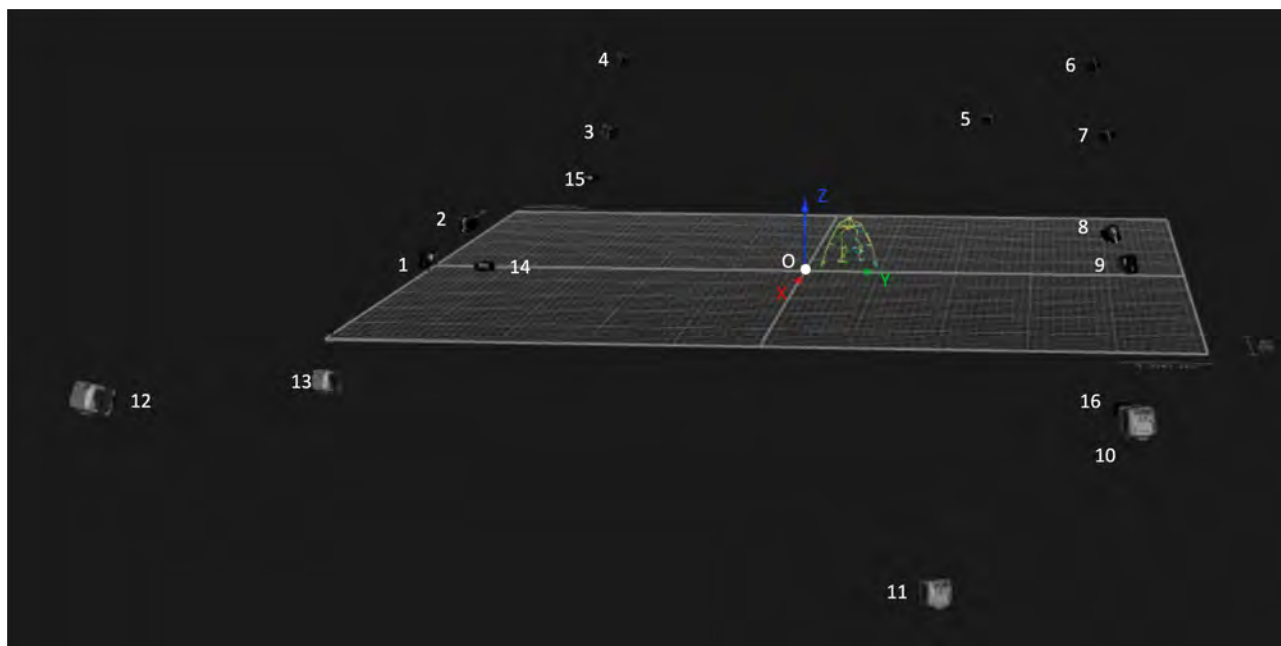
Datainsamling är ofta ett viktigt försteg till djupinlärning. Det artificiella neuronnätverket kräver data representativ för rörelsen det ska tränas på för att göra pålitliga uppskattningar, vilket i detta fall var data för utväxling ur startblock. Arbetet kring skapandet av ett dataset bestod av två huvudsakliga delar, datainsamling av biomekanisk rörelse och behandling av insamlad data.

3.1.1 Datainsamling av biomekanisk rörelse

Mätningarna genomfördes i Friidrottens hus i Göteborg för att samla data i en miljö i vilken slutprodukten troligen kommer att appliceras. Representanterna från Qualisys var de som praktiskt genomförde mätningarna och ställde upp och kalibrerade kamerorna. Totalt genomfördes två datainsamlingstillfällen. Orsaken till varför två tillfällen behövdes, grundas i att den insamlade mocap-datan från det första tillfället var ofullständig och därigenom oanvändbar. Det krävdes därav ett andra mättillfälle där misstagen korrigerades.

3.1.1.1 Mätuppställning

På en löparbana i Friidrottens hus utnyttjades totalt 16 kameror, 3 videokameror samt 13 mocap-kameror av två olika modeller. Kamerorna placerades ut enligt Figur 3.1 där videokamerorna ställdes på sidan av löparen, snett bakom samt snett framför. Resterande kameror placerades i varierande höjd runt omkring löpbanan, se Tabell 3.2 för kameran-specifikation.



Figur 3.1: Kamerauppställning bestående av 3 videokameror och 13 mocap-kameror med löpare bakom origo O. Kamerainformation och placering hittas i Tabell 3.2

Tabell 3.2: Specificerad kameratyp samt koordinater i förhållande till origo O angivet i enheter av millimeter på använda kameror, se Figur 3.1.

Numrering	Kameratyp	X	Y	Z
1	Oqus 700+	8669	-2819	1591
2	Oqus 700+	6600	-2922	1624
3	Arqus A12	-2502	-3070	1566
4	Arqus A12	-4362	-3188	2430
5	Arqus A12	-4612	2307	1564
6	Arqus A12	-4647	3871	2379
7	Arqus A12	-2805	4040	1540
8	Oqus 700+	6521	3876	1595
9	Oqus 700+	8351	4004	1588
10	Arqus A12	18514	3723	2393
11	Arqus A12	17560	2574	1107
12	Arqus A12	18647	-2125	2454
13	Arqus A12	13999	-2212	1534
14	Miqus Video	6643	-2730	1154
15	Miqus Video	-697	-2969	1122
16	Miqus Video	14351	3796	1505

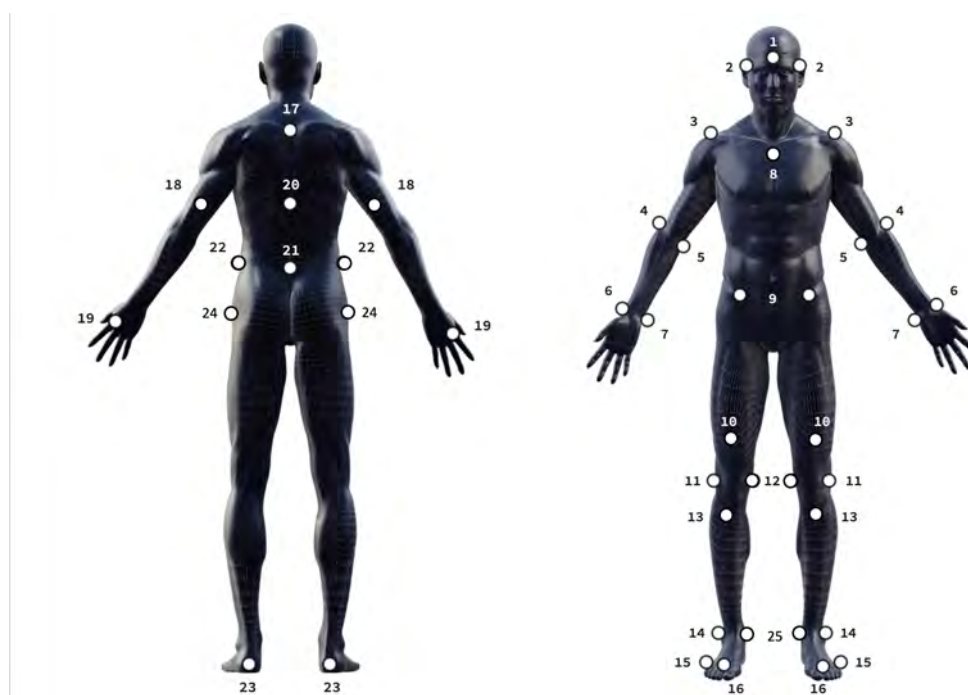
Efter att kamerorna placerades ut och kopplades samman gjordes förberedelser och kalibrering av systemet av Qualisys enligt följande:

1. Riktning ställdes in för att täcka in startblock och löpbana.

2. *Exposure-and-flashtime*, exponeringstid, och *threshold*, ljusinsläpp, anpassades efter rådande förhållanden.
3. Frekvensen för bildinsamling sattes till 85 Hz för videokamerorna Miquis Video och till 170 Hz för rörelsefångstkamerorna Oqus 700+ respektive Arqus A12.
4. Kalibrering gjordes genom att lägga ut en s.k. *L-frame*, en L-formad metallskena med specifika mått, precis framför startblocket som fungerar som referens och även origo för koordinaterna i mätdata. En T-formad stav rörde därefter i samtliga plan i hela den volym där mätningen skulle ske.
5. Mätbarheten, d.v.s. hur bra markörerna syns i volymen, kontrollerades och kamerornas position justerades därefter följt av en kalibrering (steg 4). Därefter upprepades denna process ytterligare en gång.

3.1.1.2 Markörer och nyckelpunkter

För att kunna träna en artificiellt neuronätverk och sedermera extrahera vinklar i en rörelse fastställdes ett antal nyckelpunkter, vilka skulle spåras i mätningarna. Dessa punkter valdes efter Qualisys standardiserade modell [14] för uppsättning av sportmarkörer (se markörnummer 1 till 23 i Tabell 3.3) och kompletterades med ytterligare två punkter (se markörnummer 24 och 25 i Tabell 3.3) efter överläggning med bihandledare Sarovic.



Figur 3.2: Bild över marköruppsättningen för mätningarna. Markör 1-23 utefter Qualisys standardmodell för sportmarkörer och markör 24 och 25 som komplement (se även Tabell 3.2). Bilden är från Qualisys [14], och har omarbetats med tillstånd.

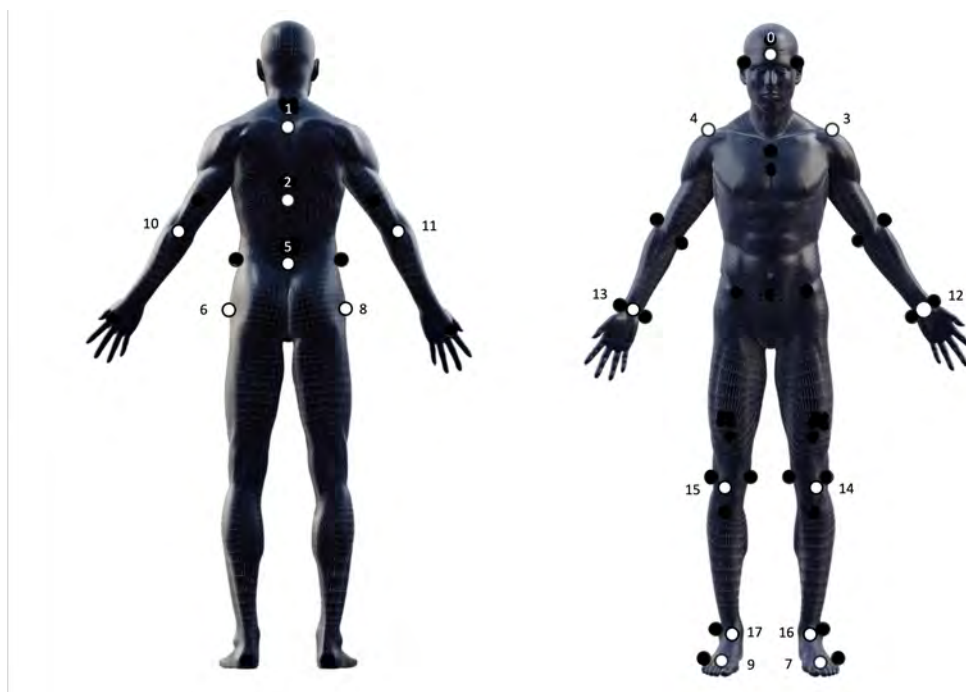
Genom att kombinera punkt 25 med 14 kunde en punkt för ankeln erhållas, varför punkt 25 lades till. Vidare utökades markörsetet med punkt 24 placerad över *Trochanter Major*, en del

av benet *femur*, för att täcka in rörelsen i höftleden.

Tabell 3.3: Nyckelpunkter som användes för placering av markörer på löparna, motsvarande markörnumrering samt deras positionering, se även Figur 3.2. Punkt 24 och 25 har adderats till den annars Qualisys standardiserade modell för sportmarkörer.

Markörnummer	Namn	Placering
1	HeadFront	Panna, ovanför näsan.
2	HeadL, HeadR	Precis ovanför örat.
3	LShoulderTop, RShoulderTop	På axeln (benet).
4	LElbowOut, RElbowOut	På utsidan av armbågen (benet).
5	LElbowIn, RElbowIn	På insidan av armbågen (benet).
6	LWristIn, RWristIn	På insidan av handleden (benet på tumsidan).
7	LWristOut, RWristOut	På utsidan av handleden (benet på lillfingersidan).
8	Chest	Övre delen av bröstbenet.
9	WaistFrontLow	På framsidan av bäcken (benet).
10	LThighFrontLow, RThighFrontLow	Över knäskålen.
11	LKneeOut, RKneeOut	På utsidan av knäet (benet).
12	LKneeIn, RKneeIn	På insidan av knäet (benet).
13	LShinFrontHigh, RShinFrontHigh	På framsida skenben.
14	LAnkleOut, RAnkleOut	På utsidan av ankeln.
15	LForefoot5, RForefoot5	På basen av den femte tån.
16	LForefoot2, RForefoot2	På basen av den andra tån
17	SpineThoracic2	Vid det andra utstickande benet under den största kotan i ryggraden.
18	LArm, RArm	På baksidan av överarmen.
19	LHand2, RHand2	På baksida hand vid basen av pekfingeret.
20	SpineThoracic12	Några centimeter nedanför mitten av nedre delen av skulderbladen.
21	WaistBack	Vid mittpunkten av benen vid baksida bäcken.
22	WaistL, WaistR	På sidorna av bäcken.
23	LHeelBack, RHeelBack	Baksida häl.
24	LThigh2, RThigh2	Trochanter Major.
25	LAnkleIn, RAnkleIn	På insidan av ankeln.

Efter att data samlades för originaluppsättningen av nyckelpunkter enligt Figur 3.2 konverterades dessa punkter till en ny uppsättning nyckelpunkter, vilket det skapade datasetet baserades på. Positioneringen av nyckelpunkter återfinns i Figur 3.3 vilka beskrivs i Tabell 3.4.



Figur 3.3: Bild över den nya nyckelpunktsuppsättningen för datasetet. Bilden är från Qualisys [14], och har omarbetats med tillstånd.

Tabell 3.4: Tabell över de nya utvalda nyckelpunkterna som används för datasetet, se Figur 3.3, samt dess relation till den gamla markörmodellen i Tabell 3.3.

Markörnummer	Namn	Markörnummer i gammal modell
0	HeadFront	1
1	SpineThoracic2	17
2	SpineThoracic12	20
3	LShoulderTop	3
4	RShoulderTop	3
5	WaistBack	21
6	LTrochanterMajor	24
7	LForefoot	16
8	RTrochanterMajor	24
9	RForefoot	16
10	LElbow	4 & 5
11	RElbow	4 & 5
12	LWrist	6 & 7
13	RWrist	6 & 7
14	LKnee	11 & 12
15	RKnee	11 & 12
16	LAnkle	14 & 25
17	RAnkle	14 & 25

3.1.1.3 Löparna

IFK Göteborg Friidrott bistod med 4 aktiva friidrottare som agerade testpersoner för datainsamlingen. Information om löparna ges i Tabell 3.5.

Tabell 3.5: Löparna som deltog vid mätningstillfället.

Löpare	Ålder	Kön	Längd [cm]	Vikt [kg]
Löpare 1	30	Kvinna	170	59
Löpare 2	17	Kvinna	160	52
Löpare 3	25	Kvinna	175	57
Löpare 4	18	Man	165	52

Informationen samlades in i samtycke med löparna och utefter utformat GDPR-avtal (se Appendix B). Avtalet behandlade utnyttjandet av persondata, bild- och videobehandling under arbetsprocessen för projektet samt träningen av det artificiella neuron nätverket. Datan lagrades i Google Drive och raderas innan 1:a Juni 2022 enligt avtalet.

3.1.1.4 Genomförande av testförsöken

Den praktiska datainsamlingen för projektet ägde rum onsdagen den 23:e februari samt torsdagen den 7:e april, 2022 på plast i Friidrottens Hus.

Efter kamerakalibrering utrustades de fyra löparna med ett komplett sportmarkörset enligt Figur 3.2. Reflexmarkörerna som användes fästes på hud och kläder med hjälp av dubbelhäftad tejp och sporttejp. I tur och ordning fick de genomföra mätningarna genom att först ställa sig statiskt framför startblocken, då en statisk mätning togs med kamerorna för att se samtliga punkter och deras förhållande till varandra. Därefter gjordes en utväxling från startblocket inkluderat löpning en sträcka på 10 meter vilket kamerorna samlade mocap- samt videodata för (se Figur 3.4). Det gjordes först 3 mätförsök och efter att datan kontrollerats kunde eventuella kompletterande starter göras för att få heltäckande resultat utan för många saknade nyckelpunkter i mocap-datan.



(a) Bild från mocap-data.



(b) Bild från videofilm.

Figur 3.4: Löpare uppställd i startblocket under mocap-mätning (a) och i videobild (b).

3.1.2 Behandling av insamlad data

Qualisys egenutvecklade mjukvara, QTM, möjliggör bearbetning och visualisering av insamlad data från deras spårningssystem. QTM nyttjades primärt för att extrahera nyckelpunktsdatan till ett passande dataformat för vidare efterbearbetning. Det huvudsakliga och slutgiltiga målet med bearbetningen av spårningsdatan var att konvertera den till ett hanterbart format för träning av ett artificiellt neuronnätverk.

Processen för databearbetningen från start till mål bröts ner i följande 3 huvudsteg samt tillhörande delsteg:

1. Exporterade QTM-datan till Matlab-format, se Appendix C för exempeldata.
2. Fann den datafil (video eller mocap) som begränsade överlappet av dessa.
 - 2.1 Undersökte för vilken bild i mocap-datan första markör tappades och jämförde med tiden när löparen i videon försvinner ur bildramen för varje videokamerorna.
 - 2.2 Klippte videodatan och mocap-datan efter tidpunkt och bild för det första data-tappet. Konverterade mocap-datan från en Matlab-fil till en textfil, se Appendix D.
 - 2.3 Beskära videon från den tredje videokameran, Miquis Video 16, utefter dimensioner för att undvika andra personer i bilden.
3. Importerade textfilen till Python och bearbetar med tillhörande kod [15].
 - 3.1 Beräknade medelvärden mellan vissa specifika nyckelpunkter för att skapa nya nyckelpunkter.
 - 3.2 Exkluderade de nyckelpunkter som nyttjades vid medelvärdesberäkningarna och önskade nyckelpunkter från datan.
 - 3.3 Skalade ner mocap-datan till 85 Hz från 170 Hz.
 - 3.4 Konverterade mocap-koordinaterna från 3D till respektive kameras pixelkoordinater.
 - 3.5 Dela upp utdatan i 2D och 3D och exportera som csv-filer.
 - 3.6 Nyckelpunkten *WaistBack* sattes till rot för datan, vilket innebär att kamerarelativ data i 3D omvandlades till koordinater med nyckelpunkten *WaistBack* som origo. Det ger data som är oberoende av löparens plats i rummet.
 - 3.7 Omvandlade all 3D-data från millimeter till meter.

Koordinatsystemet omvandlas till rotrelativa kamerakoordinater genom att *WaistBack*, roten för datan, sätts som nytt origo, (0,0,0). För varje videobildruta subtraheras nyckelpunkternas koordinatdata med koordinatdatan för *WaistBack* i respektive bildruta. Det är sedan dessa rotrelativa kamerakoordinater som det temporala faltningsnätverket gör förutsägelser i.

3.2 Djupinlärning

För att göra den mänskliga poseuppskattningen i 3D gjordes en litteräröversikt. De undersöktes vad som gjorts tidigare gällande djupinlärning för mänsklig poseuppskattning. Därefter skapades ett basnätverk, ett nätverk att utgå från för att utveckla det slutgiltiga nätverket, vilket gjordes med överföringsinlärning.

3.2.1 Litteraturöversikt

I början av projektet gjordes en litteraturstudie. Det var viktigt att få en förståelse och överblick över djupinlärning och mänsklig poseuppskattning för att kunna fatta viktiga beslut om projektets riktning.

Hemsidan *Paperswithcode* [16] användes som utgångspunkt för litteraturstudien. Där publiceras vetenskapliga rapporter inom maskininlärning som rankas efter prestation på väl etablerade publika dataset för olika typer av maskininlärningsproblem. Ibland publiceras tillhörande kod som öppen källkod. *Human3.6M* är det största och mest använda dataset för att göra bedömningar av ANN för mänsklig poseuppskattning och mocap. Det innehåller 3.6 miljoner videobilder inspelade från 4 olika vinklar och kameror med 50 Hz. Det är elva personer som utför 15 olika handlingar däribland att sitta, äta och gå [17].

3.2.2 Bedömning av nätverk

Det gjordes en bedömning av befintliga nätverk som publicerats på *Paperswithcode*. Denna bedömning ligger till grund för det vidarearbete som gjordes utifrån de erfarenheter och kunskaper som fanns publicerade i dessa rapporter. De kriterier nätverken bedömdes på för detta arbete var låg *mean per joint estimation error* (MPJPE), genomsnittligt uppskattningsfel per led, samt användning av monokulär vy. MPJPE är den kostnadsfunktion som vanligtvis används för nyckelpunktsuppskattning och monokulär vy innebär att endast en kamera används för poseuppskattning. Olika nätverks prestation på *Human3.6M* visas i Tabell 3.6.

Tabell 3.6: Utvärdering av artificiella neuronnätverk med monokulär vy. Felet som visas i tabellen är det genomsnittliga felet av de 15 olika handlingarna i *Human3.6M* under protokoll 1 (MPJPE) [17].

Nätverk	MPJPE [mm]	MPJPE* [mm]
Tesse Track	44,6	-
MixSTE	40,9	21,6
U-CondDGConv@GT 2D Pose	41,1	22,7
VideoPose3D	46,8	37,2
RIE	44,3	30,1

*MPJPE när nätverket använder sig av sanningsenliga 2D punkter och inte ytterligare ett nätverk.

Fyra artificiella neuronnätverk undersöktes, nämligen *U-CondDGConv@GT 2D Pose* [18], *RIE* [19], *VideoPose3D* [20] och *MixSTE* [21]. Dessa fyra nätverk använder sig alla av lifting. RIE kan ses som en vidareutveckling av *VideoPose3D*. Båda använder sig av ett *temporal convolutional network* (TCN), temporalt faltningsnätverk, för att göra poseuppskattning i 3D.

3.2.2.1 Val av djupinlärningsbibliotek

Djupinlärningsbibliotek är verktyg som används för djupinlärning. Majoriteten (58%) av all kod som publiceras på *Paperswithcode* är skriven med hjälp av biblioteket *PyTorch* [22]. För

de ovan nämnda nätverken där kod fanns publicerad var allt skrivet i PyTorch. Anledningen till att PyTorch användes även för detta projekt är att det fanns tidigare kompetens och erfarenhet med just detta bibliotek inom projektgruppen. Vidare underlättar tillgängligheten av tidigare nätverk utvecklingen av projektets eget nätverk.

3.2.3 Skapandet av ett basnätverk

Idén bakom att skapa ett basnätverk var att det gav en grund att arbeta utifrån och utveckla vidare. Basnätverket utvecklades efter att i högre grad uppfylla de mål som satts upp. Eftersom nätverk tränade på datasetet Human3.6M fanns publikt tillgängliga etablerades ett basnätverk med hjälp av företaget *Meta Research* och deras nätverk VideoPose3D [20]. VideoPose3D valdes som basnätverk för att det fanns väldokumenterade instruktioner och kod för att göra inferens med fullgoda resultat.

VideoPose3D klonades först från GitHub. Koden skrevs om i *Jupyter Notebooks*, en webbaserad interaktiv programmeringsmiljö, för att få en ökad förståelse över koden. Pythonbiblioteket *nbdev* användes för att omvandla koden från Jupyter Notebooks tillbaka till Pythonfiler, samt skapa dokumentation och kontinuerlig integration d.v.s. att kodbidrag vid flera tillfällen sammanläs till en central version.

Därpå gjordes inferens på egna videor enligt instruktionerna för VideoPose3D [23] för att testa kodens funktionalitet. Nätverket använder sig av ett annat nätverk, *Detectron2* som är skapat av Meta Research, för att göra förutsägelser i 2D. Nätverket VideoPose3D är ett temporalt faltningsnätverk, och gör förutsägelser i 3D baserat på utdata från Detectron2. Notera att de förutsägelser som görs av de båda nätverken inte matchar de exakta nyckelpunkter vi tog fram i Figur 3.3. För att projektets nätverk skulle förutsäga dessa krävdes ett nytt dataset och överföringsinlärning på det etablerade basnätverket.

3.2.4 Överföringsinlärning

Överföringsinlärning användes för Detectron2 som gör förutsägelser i 2D från bilder samt för det temporala faltningsnätverket som gör förutsägelser i 3D från nyckelpunktsdata i 2D.

3.2.4.1 Detectron2

Fyra olika Detectron2-nätverk kunde användas för överföringsinlärning i detta projekt: R50-FPN, R50-FPN-3x, R101-FPN-3x, och X101-FPN-3x. Det som skiljer dem åt är framförallt storleken på respektive nätverk samt hur länge de tränats. Dessa nätverk är tränade på dataset *MS-COCO* [24]. Användning av överföringsinlärning på dessa nätverk krävde att 2D-dataset som skapades för detta arbete var på samma format som MS-COCO, *COCO-keypoint format* [25]. Formatet krävde bounding boxes och instanssegmentering för varje videobildruta i vårt dataset, något datasetet saknade.

VideoPose3D använder ett tränat Detectron2-nätverk som är fritt att använda. Anledningen till att ett annat nätverk behövs för detta projekt är på grund av att det inte finns ett tränat nätverk som förutsäger någon nyckelpunkt på fötterna. Det kan också vara en fördel om nät-

verket förutsäger exakt samma nyckelpunkter som sedan uppskattas i 3D.

För att använda datan som samlats in i samarbete med Qualisys måste varje videobildruta annoteras med mocap-data i pixelkoordinater. Ett skript skapades i Python för detta. Skriptet läser in en videofil och använder sig sedan av ett redan tränat Detectron2-nätverk, *Mask-RCNN(R101-FPN-3x)*, för att förutsäga bounding box och *binary masks* för varje bildruta. Binary masks är en typ av instanssegmentering, och konverteras till *COCO polygonrepresentation* som är den typ av instanssegmentering som behövs för COCO-keypoint format. För att få pixelkoordinater för nyckelpunkterna användes ett annat skript för omvandling av mocap-koordinater från 3D till pixelkoordinater. Respektive kameras kameraparametrar användes för detta. Skriptet kombinerar sedan all information och returnerar en *JSON*-fil med informationen i *COCO-keypoint format* för varje videobildruta. Varje bildruta sparas också i en mapp för bilder som också behövs för träning av Detectron2.

Ett alternativ till att använda datan som samlats in för projektet var att använda *COCO-WholeBody* [26], vilket också testades. Det är ett dataset som bland annat inkluderar tre nyckelpunkter för varje fot. Att använda exakt samma nyckelpunkter för Detectron2 och det temporala faltningsnätverket leder eventuellt till större precision för förutsägelserna, vilket inte går om COCO-WholeBody används. Däremot använder VidoPose3D ej ett Detectron2-nätverk med nyckelpunkter som exakt motsvarar nyckelpunkterna i datasetet Human3.6M. Detta innebär att exakt samma nyckelpunkter för förutsägelser i 2D respektive 3D inte är ett krav för att få bra resultat. COCO-WholeBody är tillgängligt i JSON-filer med data i COCO-format, och har tillhörande bilder.

3.2.4.2 Temporalt faltningsnätverk

För att träna det temporala faltningsnätverket som gör förutsägelser i 3D användes förutsägelser i 2D från Detectron2 som indata, och datan i 3D som samlats in med Qualisys användes som grundsanning. Datan delades sedan upp i tre delar: datan från löpare 3 och 4 bestod av 6424 videobildrutor och användes för träning, datan från löpare 2 bestod av 2414 videobildrutor och användes för validering och datan från löpare 1 bestod av 884 videobildrutor och användes för test.

Indatan normaliserades så att den hamnade mellan $[-1, 1]$ samtidigt som bildförhållande bibehölls. Låt X vara indatan, x vara den normaliserade indatan och w, h vara bredden respektive höjden för en bild. Formeln för att beräkna den normaliserade indatan ges i ekvation 3.1.

$$x = 2 * (X/2) - (1, h/w). \quad (3.1)$$

Indatan normaliseras för att öka inlärningshastigheten, då det ger en snabbare konvergens av nätverket. Vidare laddades nätverket med parametrar i form av vikter och bias från ett tränat nätverk som VideoPose3D gjort tillgängligt. Parametrarna hade tränats med indata från Detectron2 på Human3.6M. För att nätverket ska kunna göra förutsägelser enligt vårt nya nätverk med 18 nyckelpunkter krävdes ett nytt utdatalager för nätverket. Neuronlagret initialiserades med slumpmässiga vikter, och bias initialiserades genom att ta medelvärde av grundsanningen till träningsdatan för varje videobildruta.

Steg som togs för att få ett vältränat och fungerande nätverk:

1. Fixerade slumpvalsfröet för att garantera samma resultat varje gång nätverket tränades.
2. Tränade en omgång med all indata lika med noll för att få en baslinje oberoende av indata att jämföra mot.
3. Överanpassade nätverket på 3 bilder för att kontrollera att nätverket kan nå noll i förlust på dessa bilder. Annars har nätverket ovälkomna buggar.
4. Visualiserade de förutsägelser nätverket gjorde för att kontrollera den utdata nätverket gav.
5. Överanpassade på hela träningsdatasetet för att få en baslinje och för att verifiera att nätverket kan nå en tillräckligt låg acceptabel förlustnivå på datan.
6. Regulariserade nätverket med hjälp av bortfall, minskad satsstorlek och dataökning.

Den befintliga träningsdatan utökades genom att spegla nyckelpunkter till höger och vänster av kroppen, vilket dubblade datan för nätverket att träna på. Nätverket tränades med blandad precision, och satsnormalisering användes. Det temporala faltningsnätverket tränades med följande hyperparameterar:

- Bortfall: 0.125
- Kanaler: 1024
- Satsstorlek: 128
- Epoker: 230
- Filterstorlek: [3, 3, 3, 3, 3]
- Förfallfaktor för inlärningstaktsförfall: 0.97

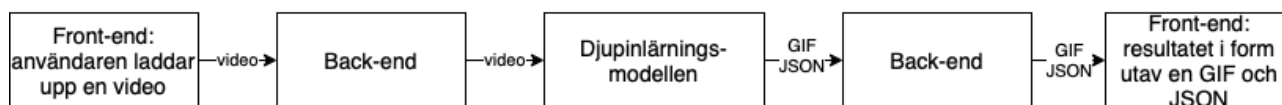
När nätverket tränades frystes alla lager förutom det sista lagret i fyra epoker med inlärningsfaktor $\eta = 0.003$. Detta gjordes för att anpassa det sista lagret till det förtränade nätverket och till datan. Nätverket tränades sedan ytterligare 96 epoker med en inlärningsfaktor på $\eta = 0.0003$. För vidare information se [15].

3.3 Utveckling av webbapplikation

Första stadiet för att utveckla en webbapplikation bestod utav att undersöka vilka språk och ramverk som skulle användas. Det fanns viss erfarenhet i projektgruppen inom webbutveckling, men inte att skapa en hel webbapplikation från grunden. Beslutet för vilka språk och ramverk grundades mest på intresse och erfarenhet i projektgruppen. Utvecklingen av applikationen delades upp i front-end och back-end för att effektivisera arbetet genom att arbeta med dessa delar parallellt.

3.3.1 Funktionalitet

Webbapplikationens primära syfte var att användaren skulle kunna ladda upp en video på sin löpsekvens, få den analyserad av nätverket och sedan få resultatet på hur nyckelpunkterna rör sig samt värdena på de önskade ledvinklarna. Webbapplikationens front-end innehöll användarens interaktion i form utav videouppladdning och även visualiseringen av analysen. Kommunikation med nätverket gjordes via back-end som även skötte hanteringen av de olika filformaten. Användaren laddade upp en video och nätverkets resultat var i bildformatet *GIF* och dataformatet *JSON*. Back-end omvandlade bilden från nätverket till flera bilder för att användaren skulle kunna stega igenom bildruta för bildruta och se hur nyckelpunkterna rör sig samt hur ledvinklarna utvecklas. I Figur 3.5 återges flödesschemat för webbapplikationen.



Figur 3.5: Flödesschema för webbapplikationen.

Front-end använde sig av ramverket React.js (se avsnitt 2.3.1) och back-end använde sig av Python och Flask (se avsnitt 2.3.2). Python valdes för att nätverket är skapat med Python, således undveks det att spendera tid vid konvertering mellan olika programmeringsspråk. Flask användes för att möjliggöra webbutveckling med Python och React.js.

I det initiala stadiet var planen att göra en webbapplikation som är på en fjärrserver, men efter att ha gjort efterforskning beslutades det att det hade tagit för mycket tid att utveckla och därmed bestämdes det att webbapplikationen skulle sättas upp med localhost (se avsnitt 2.3.2).

3.3.2 Vinkelberäkning

Vinklarna i Tabell 3.1 beräknades med den framtagna datan för nyckelpunkterna i Tabell 3.4 med den förutspådda datan och visualiserades därefter i webbapplikationen. En vinkel beräknas från tre nyckelpunkters koordinater. Relevanta nyckelpunkter för varje vinkel visas i Tabell 3.7.

Tabell 3.7: Tabell över de nya utvalda nyckelpunkterna som används för datasetet (se Figur 3.3), samt dess relation till den gamla markörmodellen i Tabell 3.7.

Ledvinkel	Nyckelpunkter (i,j,k)
1. Nacke	(0,1,2)
2. Ryggrad	(1,2,3)
3. Vänster armbåge	(3,10,12)
4. Höger armbåge	(4,11,13)
5. Vänster knä	(6,14,16)
7. Höger knä	(8,15,17)
7. Vänster fotled	(14,16,7)
8. Höger fotled	(15,17,9)

För bestämmande av en ledvinkel sammankopplad av nyckelpunkterna (i, j, k), och där vinkeln är vid punkt j. Beräknas vinkeln utifrån nyckelpunkternas koordinater med hjälp av formeln:

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos \theta.$$

$\vec{a}$ och $\vec{b}$ är de tredimensionella vektorerna som utgörs av skillnaden mellan koordinaterna i och j respektive j och k. θ är vinkeln mellan vektor $\vec{a}$ och $\vec{b}$ (vid punkt j) [15].

Resultat

För att göra poseuppskattning har ett ANN tränats på data som samlats in i samarbete med Qualisys och Svensk Friidrott. Nätverket används sedan för att utföra inferens på videor som laddas upp i en webbapplikation. Resultaten av datainsamlingstillfällena behandlas i första delen av detta avsnitt. Därefter redogörs resultaten för träning av det temporala faltningsnätverket och Detectron2. En beskrivning av webbapplikationen presenteras i sista delen.

4.1 Skapande av dataset för träning av artificiell intelligens

Som en konsekvens av att den insamlade mocap-datan från första insamlingstillfället var bristande i sin kvalitet, kunde den ej nyttjas för träning av artificiella neuronnätverk. Samtliga av de insamlade löpsekvenserna innehöll en stor andel hål ifråga om nyckelpunkternas positioner (x,y,z). Det andra datainsamlingstillfället resulterade däremot i ett flertal kvalitativa löpsekvenser ur mocap-data hänsyn, som sedermera kunde nyttjas för att träna upp artificiella neuronnätverket. Kopplat till detta tas enbart resultaten ifrån det andra datainsamlingstillfället upp i denna resultatdel.

4.1.1 Bildresultat för datainsamlingen

Från det andra datainsamlingstillfället erhöles elva lyckade löpsekvenser av sammanlagt femton stycken. De lyckade löpsekvenserna var framgångsrika utifrån perspektivet att mocap-datan var hundra procentig. Den innehöll inte några hål ifråga om nyckelpunkternas positioner igitom löpningarna under en sammanhängande tid om minst två sekunder från det att löparen lämnat startblocken. Samtliga av de fyra misslyckade löpsekvenserna ingick i de fem första löpförsöken då den första löparen sprang.

Av de elva lyckade löpningarna stod löpare 1 för en av dessa, löpare 2 och 3 för tre vardera, samt löpare 4 för fyra (se Tabell 3.5 för information om löparna). Nedan i Tabell 4.1-4.4 återges begränsningen av antalet bilder för mocap-datan och de tre videokamerorna för respektive löpare och tillhörande lyckade mätningar. De gråmarkerade rutorna i tabellerna representerar den begränsande faktorn mellan mocap-datan och de tre videokamerorna. Med andra ord är det den part som innehar minst antal användbara bilder per löpning för träning av neuronnätverket.

Tabell 4.1: Bildbegränsning för mocap-datan och videofilmerna för löpare 1.

Löpare 1	Kamera	Mocap bild interval (170 Hz)	Video bild interval (170 Hz)
Mätning 2	14	1-658	1-646
	15	1-658	1-476
	16	1-658	1-646

Tabell 4.2: Bildbegränsning för mocap-datan och videofilmerna för löpare 2.

Löpare 2	Kamera	Mocap bild interval (170 Hz)	Video bild interval (170 Hz)
Mätning 1	14	1-685	1-578
	15	1-685	1-408
	16	1-685	1-748
Mätning 2	14	1-678	1-544
	15	1-678	1-408
	16	1-678	1-714
Mätning 3	14	1-665	1-544
	15	1-665	1-374
	16	1-665	1-714

Tabell 4.3: Bildbegränsning för mocap-datan och videofilmerna för löpare 3.

Löpare 3	Kamera	Mocap bild interval (170 Hz)	Video bild interval (170 Hz)
Mätning 1	14	1-889	1-646
	15	1-889	1-476
	16	1-889	1-816
Mätning 2	14	37-774	37-476
	15	37-774	37-544
	16	37-774	37-374
Mätning 3	14	1-816	1-748
	15	1-816	1-578
	16	1-816	1-782

Tabell 4.4: Bildbegränsning för mocap-datan och videofilmerna för löpare 4.

Löpare 4	Kamera	Mocap bild interval (170 Hz)	Video bild interval (170 Hz)
Mätning 1	14	1-699	1-476
	15	1-699	1-340
	16	1-699	1-646
Mätning 2	14	1-779	1-578
	15	1-779	1-442
	16	1-779	1-816
Mätning 3	14	1-776	1-612
	15	1-776	1-442
	16	1-776	1-816
Mätning 4	14	1-889	1-714
	15	1-889	1-544
	16	1-889	1-850

4.2 Djupinlärning

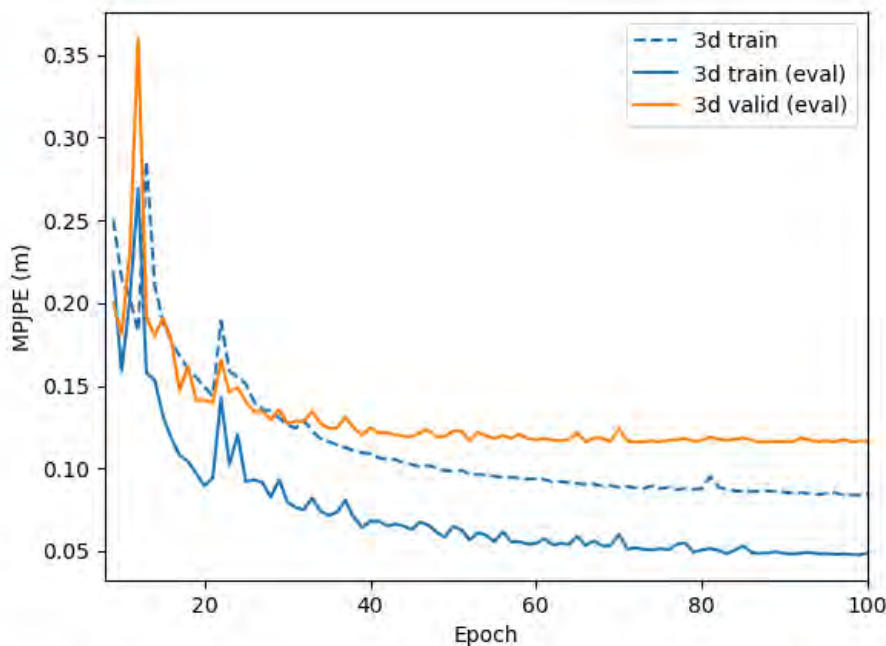
Det var blandade resultat för förutsägelser i 2D och 3D. Träning av Detectron2, både med projektets dataset och med COCO-WholeBody, gick ej att genomföra under projektets tidsgränser. Ett redan tränade Detectron2-nätverket, utan nyckelpunkter på fötterna, användes därmed för träning av det temporala faltningsnätverket. Källkoden finns på GitHub [15].

4.2.1 Detectron2

Träning av Detectron2 med data från insamlingen med Qualisys misslyckades. Detectron2 gav inte felmeddelanden under träning, vilket tyder på att datasetet för träning hade rätt format. Visualisering av resultatet från träningen visade däremot inte förutsagda nyckelpunkter på bilder som användes för inferens. Under granskning av träningsdatan framstod att nyckelpunkternas pixelkoordinater hade stora värden i jämförelse med videobildrutornas pixelhöjd och pixelbredd. Genom att undersöka en nyckelpunkts pixelkoordinater i x-led framstod även att nyckelpunkten rörde sig till vänster, medan det är tydligt att nyckelpunkten rör sig åt höger från videodata. Omvandlingen från grundsanning i 3D till pixelkoordinater har därmed misslyckats. Även träning av Detectron2 på COCO-WholeBody har ännu inte lyckats. COCO-WholeBody har data på COCO format, men Detectron2 kan inte hantera de tillagda nyckelpunkterna. Datan behövde därmed formateras om för att registraras korrekt av Detectron2.

4.2.2 Temporalt faltningsnätverk

Det slutgiltiga temporala faltningsnätverket för vår rapport tränades 100 epoker. Resultatet, som mättes med kostnadsfunktionen MPJPE (se avsnitt 3.2.2), var 45.11 mm för träningsdatan, 115.03 mm för valideringsdatan samt 90.09 mm fel för testdatan. Trots försök till regularisering genom bortfall, minskad satsstorlek och datautökning kan man notera i Figur 4.1 att det temporala faltningsnätverket har en hög varians. Det är en förhållandevis låg kostnad för träningsdatan medan valideringsdatans kostnad är 2.5 gånger större. Positivt är dock att överanpassning till träningsdatan inte ökar kostnaden för valideringsdatan.



Figur 4.1: MPJPE efter varje epok av det temporalala faltningsnätverket. I figuren innebär *eval* utvärdering av träningsdatan samt valideringsdatan utan till exempel bortfall och dataökning. Testdatan kördes efter träning och är därav inte med på bild.

4.3 Webbapplikation

Målet var att skapa en webbapplikation i form av en MVP som användaren kan interagera med. Detta mål har uppnåtts. Användaren kan att ladda upp en video på sin löpning i webappen, få den analyserad av modellen och sedan se resultatet (se Appendix E). Resultatet kan även manipuleras av användaren genom att stega igenom videobildruta för videobildruta för att tydligt kunna se poseuppskattning och ledvinklar. Källkoden för webbapplikationen finns på GitHub [27].

Diskussion

I detta avsnitt diskuteras vad som har uppnåtts inom projektets delar, och jämförs med syftet att skapa en produkt som kan användas för analys gällande löprörelser. Felkällor och förbättringsområden redogörs också här. I första delen behandlas skapandet av dataset, där fokus ligger på analys av varför det första insamlingstillfället hade bristfälliga resultat. I andra delen behandlas djupinlärning, och därefter diskuteras utvecklingen av webapplikationen. Sist behandlas samhälleliga och etiska aspekter för projektet.

Detta projekt skiljer sig från tidigare projekt inom nyckelpunktsdetektering eftersom den nya nyckelpunktsuppsättningen i Figur 3.3 tar hänsyn till metatarsofalangealleden alltså rörelsen i fotleden samt rörelse i höftleden (trochanter major). Lederna är vitala vid modellering av rörelse och specifikt vid löpning. Således blir denna modell unik och väl användbar i diverse sport- och rörelserelaterade analyser.

Här valdes en ledmodell framför exempelvis en muskuloskeletal modell då den är enklare att visualisera och anpassa baserat på de nyckelpunkter som valts ut. Det var naturligt att välja en ledmodell då uppdragsgivaren vill analysera vinklar mellan leder. Dock skulle en muskoskeletal modell tillåta modellering av kraftutveckling och därigenom möjliggöra vidare analys om vilka muskler som behöver stärkas i relation till varandra, något som inte är möjligt med en ledmodell. Det skulle emellertid kunna vara en framtida utvecklingsidé.

5.1 Skapande av dataset

Det första insamlingstillfället genererade inte något resultat som gick att använda och ytterligare ett mättillfälle behövde göras. Genom det andra mättillfället, 1,5 månader senare, erhöles däremot data för markörerna med tillräckligt hög procentuell täckning för att användas till träning av artificiella neuronnätverk. Däremot skulle somliga åtaganden som listas nedan i avsnitt 5.1.2 kunna göras för smidigare databearbetning.

5.1.1 Felanalys av det första mättillfället

Det finns några framträdande faktorer som bidrog till det dåliga resultatet och den bristande täckningen av mätpunkter. Den första orsaken handlar om solen och ljusinsläppet. En solig mättdag samt stora fönster som släpper in ljus från flera riktningar gör att reflexmarkörerna inte kan identifieras av kamerorna från lika stort avstånd som i en laboriemiljö. För att hantera problemet bör tröskelvärde för ljusinsläpp höjas, dock sker det på bekostnad av ljusintensiteten som markörerna reflekterar och således minskar det möjliga avståndet markörerna kan mätas ifrån.

Det missades också att analysera vardera mätning och täckningen av antalet spårade mätpunkter, liksom att kolla och eventuellt uppdatera nödvändig kalibrering av kamerorna.

Allmänt viktigt att tänka på vid preparation av systemet var att L-ramen placerades nära mätområdet för att undvika skalningsfel i mätvolymen, att kamerorna omgav volymen med full sikt över L-ramen samt vikten av att kalibreringsstaven rördes i alla tre plan över hela volymen.

5.1.2 Förbättringsområden

Insamlingssprocessen och bearbetningen av data efter det andra mättillfället skulle kunna underlättas genom att följande områden förbättrades:

- Antalet lyckade mätningar. Den antagligen största möjliga förbättringen som kan göras är att genomföra fler mätningar med fler olika personer. På så vis får det temporala faltningsnätverket mer material att träna samt validera med.
- Kompatibiliteten video- och mocap-datan sinsemellan. Fördelaktigt hade det varit om datan samlades in med samma bildfrekvens då de behöver vara av samma storlek för att kunna nyttjas vid träning. På samma sätt hade det varit önskvärt med lika långa och synkade mätningar för att underlätta bearbetningen av erhållen data.
- Omgivningens påverkan. Vid mätning är det av vikt för Detectron2 att endast löparen syns i bild. Genom att noggrant utesluta andra personer i bild skulle tid sparas då vissa videofilmer behövdes beskäras.
- Upplösningen av videofilmerna från videokamerorna skulle kunna förbättras. Det hade underlättat för Detectron2 att finna punkter vilket i sin tur ökar noggrannhet i 3D.

5.2 Djupinlärning

Målet med träning av Detectron2 var att kunna träna Detectron2 för att förutsäga nyckelpunkter på fötterna. Tyvärr uppnåddes inte detta. Det skedde fel någonstans i processen för de båda metoderna som användes, och det fanns otillräckligt med tid för att rätta till dem. Gällande det temporala faltningsnätverket var målet att nätverket skulle fungera och prestera tillräckligt bra för att kunna användas för en MVP. Detta uppnåddes med ett redan tränat Detectron2-nätverk, och nätverket kunde därmed användas i webapplikationen.

5.2.1 Detectron2

Det hade varit bra om överföringsinlärningen för Detectron2 påbörjats tidigare. Istället för att vänta på data från Qualisys hade vi kunnat träna på COCO-WholeBody. Den möjligheten förbisågs, vilket bidrog till att träningen inte fungerade i tid. Användning av COCO-WholeBody var dessutom enklare eftersom all data redan var i rätt format. När vi fick data för träning hade det ändå varit en bra idé att börja med träning av Detectron2 med COCO-WholeBody. Om det därefter funnits tid över hade vi kunnat använda runningpose-datasetet för träning. Det datasetet motsvarar i högre grad data som kommer användas under inferens, då datan är för löpande personer. Användning av exakt samma nyckelpunkter för Detectron2 och det temporala faltningsnätverket hade också potentiellt kunnat ge en högre precision.

Ett problem med Detectron2 var att nätverket ej är anpassat för att hantera flera typer av nyckelpunkter. COCO-WholeBody hade fem typer av nyckelpunkter, varav två av typerna var relevanta för detta projekt. Det innebar att extra steg behövdes tas för att registrera datan på

rätt sätt. Ett alternativ för detta var att använda verktyg för att registrera *custom dataset* egna dataset. Ett annat var att formatera om datan så att det bara fanns en typ av nyckelpunkter.

5.2.2 Temporalt faltningsnätverk

Resultat med det temporala faltningsnätverket visade på hög varians. Vad som kan göras för att minska denna varians är mer data eller en annan nätverksarkitektur. Under träning av det temporala faltningsnätverket påträffades det ibland att nätverket fick en exploderande kostnad mitt i träningen, se ett exempel i Appendix F. Det kan bero på vilken satsstorlek som användes tillsammans med optimeringsalgoritmen. En enkel men betydelsefull implementation är att nätverket använder sig av RIE [19], som nämndes under nätverksbedömningen (se avsnitt 3.2.2).

Nätverket som i nuläget rankas etta på mänsklig poseuppskattning i 3D på Human3.6M är *TesseTrack* [16], som kan använda sig av en monokulär vy med MPJPE på $44.6mm$ [28]. Nätverket kan dessutom hantera mer än en person i bild. *TesseTrack* skiljer sig från andra nätverk i att inte använda sig av lifting. Författarna menar att detta gör nätverket mer robust och att MPJPE minskar. Genom att använda sig av ett nätverk likt *TesseTrack* hade produkten kunnat hantera mer än en person i bild och troligtvis kunnat göra mer precisa uppskattningar av nyckelpunkters koordinater.

Vidare hade det varit intressant att öka antalet nyckelpunkter som ges av Detectron2 som indatas till det temporala faltningsnätverket. Intuitivt borde fler punkter vara en fördel, exempelvis då en punkt ej syns i bild skulle en närliggande punkt kunna hjälpa till att uppskatta var den täckta punkten är i rummet.

Även om felet hos de temporala faltningsnätverket var relativt högt jämfört med de nätverk vi såg i litteraturstudien, så bör det faktiska felet vid användning nätverket vara mindre, då dagens moderna mobiltelefoner har en bättre upplösning för videor än vad nätverket tränades på. Moderna mobiltelefoner har även ett höghastighetsläge runt cirka 240Hz vilket är en fördel då nätverket är testat och validerat på endast 85Hz allt annat lika borde därför nätverket prestera bättre.

De brister som i nuläget existerar hos nätverket är att den är endast tränad på en begränsad mängd videodata (75.6s) från endast 3 stationära vinklar på en förhållandevis homogen grupp löpare. En användare av den tänkta slutprodukten kan tänkas placera sig på fler vinklar och även ha en rörelse av kameran under inspelning, men även vara av en helt annan kroppstyp.

5.2.3 Bias

Bias som kan ha uppstått för det temporala faltningsnätverket har inte undersökts eftersom det låg utanför projektets omfattning, dock kan ett resonemang föras över vilka bias som med sannolikt uppstått. Det temporala faltningsnätverket använder sig av koordinater i 2D från Detectron2, så bias för nätverket innebär brister i dess förmåga att göra förutsägelser utifrån olika typer av koordinatdata från Detectron2. Om produkten som helhet uppvisar andra typer av bias beror det snarare på brister hos det redan tränade Detectron2.

Gällande sample bias bör det låga antalet löpare orsaka problem. Nätverket är förmodligen sämre på att analysera personer som skiljer sig i storlek eller sätt att springa, från löparna under insamlingen. Kanske kan det vara problematiskt om en person har extra långa ben i förhållande till överkropp, eller är väldigt ovan att springa. Ett annat problem är att nätverket förmodligen presterar sämre på vissa typer av videor. Det inkluderar videor som är inspelade från vissa vinklar i förhållande till löparen och löparens riktning, och videor där kameran rör sig eller vrids runt. Datainsamlingen gjordes med videoinspelning från endast tre vinklar, där kamerorna inte rörde sig alls. En annotator bias som kan ha uppstått gäller utplacering av markörer. Det var flera personer som gjorde det under det andra datainsamlingstillfället. Om olika personer placerade markörer på lite olika ställen försämrades förmodligen nätverkets förutsägelser.

5.3 Webbapplikation

Fokuset låg på att skapa en MVP. Hela projektet delades upp i olika delar för att effektivisera utvecklingen med parallellt arbete, men det var omöjligt att helt eliminera beroende. I projektgruppen fanns inte heller all kunskap för att skapa en fungerande webbapplikation, mycket förstudier behövdes. Därför var målet från början en MVP och det uppnåddes med webbapplikationen.

För framtida utveckling finns det flera förbättringsmöjligheter:

- Använda fjärrserver istället för lokal server för att möjliggöra användning för utomstående.
- Säker databas för datalagring.
- Anpassa applikation till mobil användning.
- Uppdatera designen i förmån av användarupplevelse.

5.4 Framtida utvecklingsmöjligheter

En utvecklingsmöjlighet är att utöka antalet ledvinklar som kan beräknas med nuvarande nyckelpunkter, exempelvis vänster axel (frontalplanet), höger axel (frontalplanet), vänster axel (sagittalplanet), höger axel (sagittalplanet) samt vinklar i de båda höfterna. Dessa skulle relativt enkelt kunna extraheras från modellen. Det som krävs är ett smart sätt att definiera hur frontal- och sagittalplanet ska representeras med befintliga koordinater och hur dessa relateras till vinklarna som är intressanta, något som inte hanns med inom tidsramen för projektet. All data och kod finns för att utöka antalet ledvinklar.

Ett nästa steg skulle kunna vara att utveckla så att produkten utöver att beräkna vinklar, även ska kunna genomföra en analys och bedöma huruvida vinklarna och således rörelserna bör förändras för optimering av prestationen. Nätverket skulle alltså, i viss mån, kunna ersätta tränaren och ge motsvarande instruktioner.

Det skulle också vara relativt enkelt att utöka analysen för andra typer av rörelser utöver startblocksutväxlingar. Förbättring av dataset hade kunnat göras genom skapande av ett mer omfattande dataset med fler videoinspelningar och ett större antal vinklar och individer, bestående av fler typer av rörelser. Fler nyckelpunkter hade även kunnat spåras under insamlingen.

Ett experiment gjordes där modellen fick en video där en person gjorde ett golfsving, dock var resultaten dåliga då modellen försökte utläsa ett löpsteg ur rörelsen. Det handlar alltså främst om träningsprocessen för det artificiella neuron nätverket och att nätverket behöver träna på den typ av rörelse som nätverket ska göra förutsägelser för. Produkten skulle således kunna appliceras på andra typer rörelsemönster där ledvinklar är av intresse, exempelvis inom golf eller konståkning.

5.5 Samhälleliga och etiska aspekter

Produkten samlade in data som sedan behandlas och analyseras. En praktisk aspekt som fanns i åtanke vid utvecklingen av produkten var hur datainsamlingen skulle gå tillväga och att det skulle finnas ett samtycke. I Sverige finns GDPR (dataskyddsförordningen) som reglerar datainsamling och det är avgörande att följa den. Datan innefattade personliga uppgifter, bilder och videos på testpersonerna och för att produkten ska vara godkänd, skrev personerna på ett GDPR-avtal. I avtalet stod det att datan skulle lagras till senast 1a juni 2022 och detta kommer att fullföljas. Vid en potentiell vidareutveckling av webbapplikationen, borde lagring av data behandlas och undersökas hur det skulle göras på säkraste sätt. Om webbapplikationen skulle resultera i en som lagrar icke-nödvändiga cookies på användarnas datorer skulle det behöva implementeras ett samtycke för att följa GDPR.

6

Slutsats

Målet med detta kandidatarbete var att skapa en MVP för markörlös optisk rörelsefångst genom maskininlärning. För att uppnå målet har tre delproblem lösts, vilka var datainsamling och skapande av ett dataset, träning av ett artificiellt neuron nätverk, samt utveckling av en webbapplikation. Ett dataset av hög kvalité skapades som det temporala neuron nätverket framgångsrikt tränades mot, vilket syftar på att det kan göra uppskattningar av nyckelpunkternas koordinater om än med 12 centimeters avvikelse. Via webbapplikationen kan användaren därav enkelt analysera rörelsen genom att visualisera vinklar i utvalda leder genom att ladda upp en videofil på en löpare. Vidare finns utrymme för framtida utvecklingsmöjligheter såsom att beräkna fler ledvinklar, göra en mer avancerad analys av löpning eller analysera fler typer av rörelsemönster.

Avslutningsvis är förhoppningen att projektet kommer kunna användas och att det lever upp till uppdragsgivarnas, men också idrottstränares och löpares, önskemål och behov. Önskvärt och spännande vore det om en vidareutvecklingsprocess initierades för ytterligare precision och funktion i produkten.

Referenser

- [1] J. N. Carter och M. S. Nixon, “Advances in Automatic Gait Recognition”, i *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, IEEE Computer Society, 2004, s. 139. DOI: 10.1109/AFGR.2004.1301521.
- [2] J. Amen, M. ElGebeily, D. M. E. El-Mikkawy, A. H. Yousry och T. A. El-Sobky, “Single-event multilevel surgery for crouching cerebral palsy children: Correlations with quality of life and functional mobility”, 2018. DOI: 10.4103/jmsr.jmsr_48_18.
- [3] L. Ceyssens, R. Vanelderen, C. Barton, P. Malliaras och B. Dingenen, “Biomechanical Risk Factors Associated with Running-Related Injuries: A Systematic Review”, *Sports Medicine*, årg. 49, s. 1095–1115, 2019. DOI: 10.1007/s40279-019-01110-z.
- [4] J. Santos-Concejero et.al., “Stride Angle as a Novel Indicator of Running Economy in Well-Trained Runners”, *Journal of Strength and Conditioning Research*, årg. 28, nr 7, s. 1889–1895, 2014. DOI: 10.1519/JSC.0000000000000325.
- [5] G. Guerra-Filho, “Optical Motion Capture: Theory and Implementation.”, *RITA*, årg. 12, nr 2, s. 61–90, 2005.
- [6] Qualisys. (2022). “Motion Capture”, URL: <https://www.qualisys.com/cameras/5-6-7/> (hämtad 2022-05-01).
- [7] Qualisys. (2022). “Motion Capture”, URL: <https://www.qualisys.com/about/> (hämtad 2022-05-10).
- [8] D. Littlefield, “Fastai Course Chapter 1 Q & A on Mac”, *Mac O’Clock*, 2021.
- [9] B. Gaonkar, K. Cook och L. Macyszyn, “Ethical Issues Arising Due to Bias in Training A.I. Algorithms in Healthcare and Data Sharing as a Potential Solution”, *The AI Ethics Journal*, årg. 1, nr 1, 2020. DOI: <https://doi.org/10.47289/AIEJ20200916>.
- [10] science. (u. å). “Lesson: Neuronnät & Djupinläring (Deep Learning)”, URL: <https://science.nu/courses/artificiell-intelligens-kurs/lektion/neuronnat-deep-learning/> (hämtad 2022-05-10).
- [11] S. Ioffe och C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, *CoRR*, årg. abs/1502.03167, 2015. arXiv: 1502.03167. URL: <http://arxiv.org/abs/1502.03167>.
- [12] StackOverflow. (2021). “2021 Developer Survey”, URL: https://insights.stackoverflow.com/survey/2021?_ga=2.80851666.411358645.1651742190-1058753271.1651742190#section-most-popular-technologies-web-frameworks (hämtad 2022-05-11).
- [13] S. Cheshire och M. Krochmal. (2013). “RFC 6761 - Special-Use Domain Names”, URL: <https://datatracker.ietf.org/doc/html/rfc6761> (hämtad 2022-05-12).

-
- [14] *Qualisys Sports Marker Set*, Kvarnbergsgatan 2, Göteborg, 2021. URL: <https://cdn-content.qualisys.com/2021/11/Sports-Marker-Set.pdf>.
- [15] R. Svensson och A. Ånestrand. (2022). “RunningPose”, URL: <https://github.com/TIFX04-22-10/runningpose> (hämtad 2022-05-11).
- [16] Paperswithcode. (2022). “3D Human Pose Estimation on Human3.6M”, URL: <https://paperswithcode.com/sota/3d-human-pose-estimation-on-human36m> (hämtad 2022-03-18).
- [17] C. Ionescu, D. Papava, V. Olaru och C. Sminchisescu, “Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, årg. 36, nr 7, s. 1325–1339, juli 2014.
- [18] W. Hu, C. Zhang, F. Zhan, L. Zhang och T.-T. Wong, “Conditional Directed Graph Convolution for 3D Human Pose Estimation”, 2021. DOI: 10.48550/ARXIV.2107.07797. URL: <https://arxiv.org/abs/2107.07797>.
- [19] W. Shan, H. Lu, S. Wang, X. Zhang och W. Gao, “Improving Robustness and Accuracy via Relative Information Encoding in 3D Human Pose Estimation”, i *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, s. 3446–3454. DOI: 10.48550/arXiv.2107.13994.
- [20] D. Pavlo, C. Feichtenhofer, D. Grangier och M. Auli, “3D human pose estimation in video with temporal convolutions and semi-supervised training”, i *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. DOI: 10.48550/arXiv.1811.11742.
- [21] J. Zhang, Z. Tu, J. Yang, Y. Chen och J. Yuan, *MixSTE: Seq2seq Mixed Spatio-Temporal Encoder for 3D Human Pose Estimation in Video*, mars 2022. DOI: 10.48550/ARXIV.2203.00859.
- [22] Paperswithcode. (2022). “Trends”, URL: <https://paperswithcode.com/trends> (hämtad 2022-03-18).
- [23] Meta research. (2022). “Inference in the wild”, URL: <https://github.com/facebookresearch/VideoPose3D/blob/main/INFERENCE.md> (hämtad 2022-03-28).
- [24] COCO. (2020). “COCO 2020 Keypoint Detection Task”, URL: <https://cocodataset.org/#keypoints-2020> (hämtad 2022-03-28).
- [25] A. Kelly. (2019). “Create COCO Annotations From Scratch”, URL: <https://www.immersivelimit.com/tutorials/create-coco-annotations-from-scratch> (hämtad 2022-03-28).
- [26] J. Sheng et.al., “Whole-Body Human Pose Estimation in the Wild”, i *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
- [27] T. Barac och A. Boström. (2022). “Web-app”, URL: <https://github.com/TIFX04-22-10/web-app> (hämtad 2022-05-12).
- [28] N. D. Reddy, L. Guigues, L. Pishchulin, J. Eledath och S. G. Narasimhan, “TesseTrack: End-to-End Learnable Multi-Person Articulated 3D Pose Tracking”, i *Proceedings of (CVPR) Computer Vision and Pattern Recognition*, IEEE, juni 2021.

A

Faltningsnätverk

En typ av ANN som vanligtvis används då bilder ges som indata är *convolutional neural network* (CNN), faltningsnätverk. Definitionen av CNN är att en operation kallad faltning görs i minst ett lager av nätverket. En typ av faltningsnätverk är *temporal convolutional network* (TCN), temporalt faltningsnätverk. Detta används då en förutsägelse kan förbättras genom att ta hänsyn till data från flera tidpunkter. Exempelvis kan man använda TCN då arkitekturen ska göra förutsägelser för bilder i en video. För att göra en förutsägelse för en bild används då även andra bilder i videon.

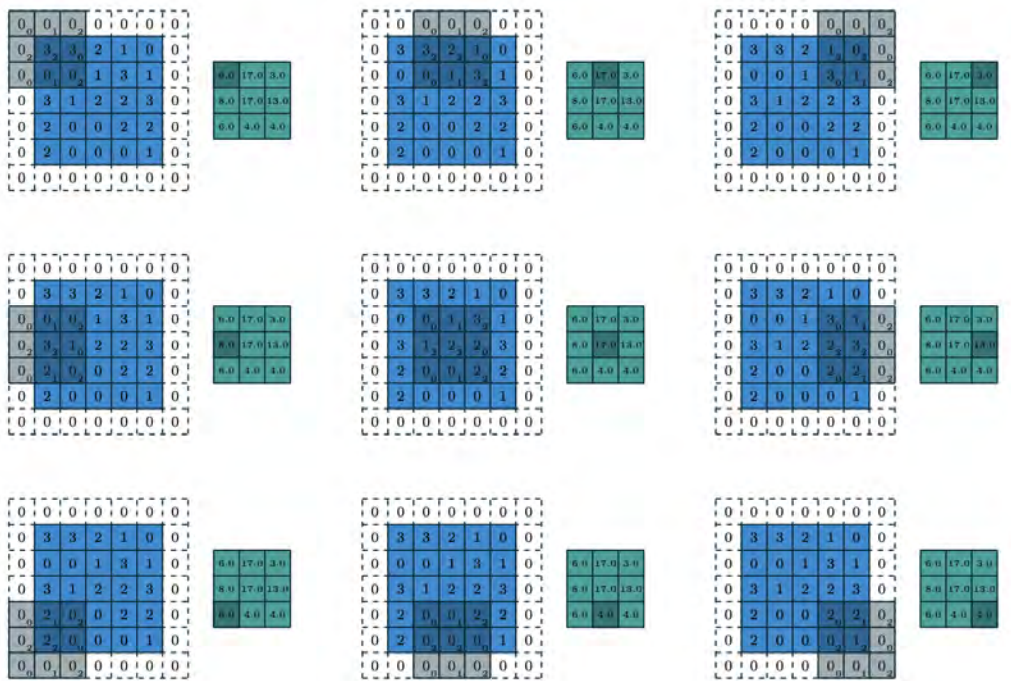
A.1 Faltning

Ett lager i ett CNN där faltning sker är ett faltningslager. Indata till lagret är en tensor, ett matematiskt objekt som ordnar vektorer i ett vektorrum. Det är vanligtvis en multidimensionell matris med skalärer. I faltning används en annan multidimensionell matris kallad kernel. En del av tensorn som är indata, hädanefter indatatensorn, används tillsammans med kerneln för att ge ett element i utdata kallad *activation map*, aktiveringskarta. För att underlätta beskrivningen betraktar vi ett exempel av tvådimensionell faltning av en exempelmatis som indatatensor. En kernel visas i Figur A.1.

0	1	2
2	2	0
0	1	2

Figur A.1: Tvådimensionell kernel som används för faltning

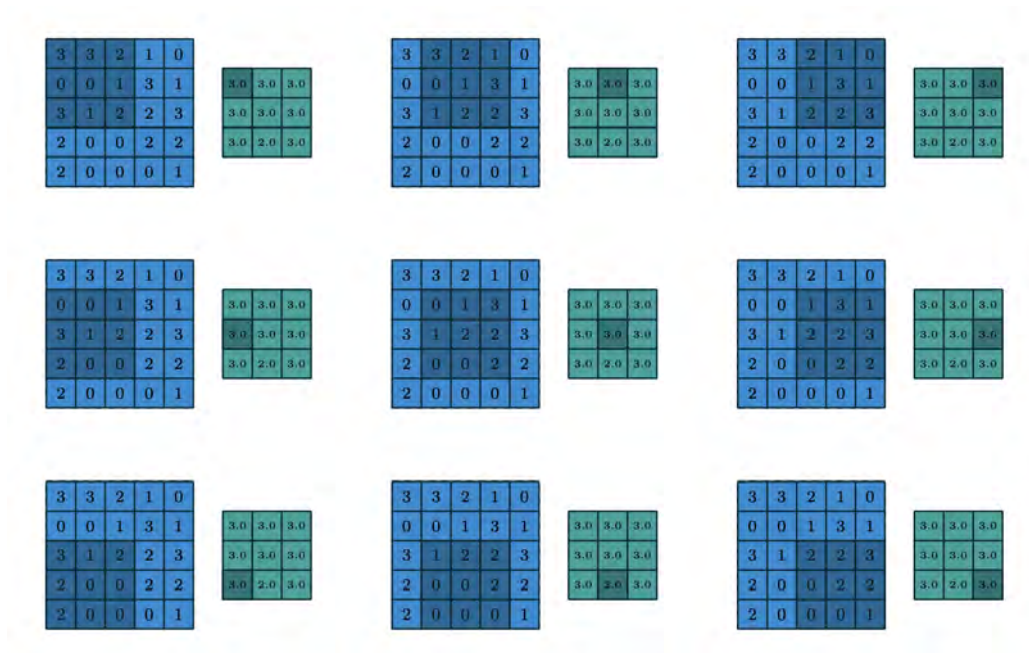
Framställandet av en aktiveringskarta i två dimensioner visas i Figur A.2. Blå rutor är exempelmatisen, och gröna rutor är aktiveringskartan. I exemplet har *bit padding*, bitutfyllning, genomförts. Bitutfyllning innebär att man lagt till nollor runt exempelmatisen. Bitutfyllning används för att information om indatan ej ska gå förlorad. Varje par av exempelmatis och aktiveringskarta i figuren beskriver hur ett värde beräknas i aktiveringskartan. Matisen med bit har ett skuggat område, som tillsammans med kerneln ger det skuggade elementet i aktiveringskartan. Varje element ur det skuggade området multipliceras med ett element från kerneln med samma position. Exempelvis multipliceras elementet uppe i vänstra hörnet av det skuggade området med elementet uppe i vänstra hörnet av kerneln. Detta görs med alla element ur det skuggade området, och produkterna adderas. Detta ger det skuggade elementet i aktiveringskartan.



Figur A.2: Illustration av faltning samt bitutfyllning

A.2 Pooling

Pooling används för att reducera datas dimensioner. Precis som under faltning behandlas en del av en indata tensor i taget. I Figur A.3 visas max pooling av samma matris som i exemplet för faltning. Max pooling ger det högsta värdet av elementen i det skuggade området av exempelmatrixen som värde i den gröna matrisen för utdata.



Figur A.3: Illustration av max pooling

B

GDPR-avtal

GDPR avtal

Syftet med detta avtal är att för att personen ska godkänna användning av sina personuppgifter för oss i projektgruppen för kandidatarbetet TIFX04-22-10 vid Chalmers Tekniska Högskola och även för Qualisys. Vi behandlar personens personuppgifter för att kunna slutföra vårt kandidatarbete och skapa en produkt som idrottare och Svenska Friidrott kan förhoppningsvis ha användning av. Genom att skriva på detta avtal, ger personen sitt samtycke till att vi får behandla och lagra sina personuppgifter. Vi samlar in personuppgifter som bara är absolut nödvändiga för oss.

Vi behandlar dessa personuppgifter från personen:

- Namn
- Ålder
- Kön
- Längd
- Vikt
- Bilder & videos

I enlighet med GDPR (EU:s dataskyddsförordning) så har personen ett antal rättigheter avseende behandlingen av dina personuppgifter. Personen kan alltid vända sig till oss för att utöva sina rättigheter. Vi lagrar personens data på ett så säkert sätt som möjligt och de som har tillgång till den är projektgruppen och även Qualisys. I slutrapporten anonymiseras datan som innebär att personens namn och bilder kommer inte finnas med.

Person har rätt att begära rättelse eller radering av sina personuppgifter. Personen har rätt att när som helst återkalla sitt samtycke enligt detta avtal. Personen måste göra en undertecknad ansökan för att återkalla sitt samtycke enligt detta avtal. När samtycke har återkallats så upphör detta avtal att gälla.

Vi sparar personens personuppgifter till absolut senast **1a juni 2022**.

Namn

Underskrift

Ort & datum

C

QTM och mocap-data

NO_OF_FRAMES	699										
NO_OF_CAMERAS	16										
NO_OF_MARKERS	45										
FREQUENCY	170										
NO_OF_ANALOG	0										
ANALOG_FREQUENCY	0										
DESCRIPTION	--										
TIME_STAMP	2022-04-07 626700.37880610										
DATA_INCLUDED	3D										
MARKER_NAMES		HeadL			HeadR				HeadFront		
COORDINATE	x	y	z		x	y	z		x	y	z
	33.877	631.862	565.916		47.549	469.478	569.404		44.405	545.182	467.627
	33.966	631.641	565.631		47.443	469.237	569.146		44.280	544.918	467.356
	34.079	631.411	565.408		47.250	468.977	568.857		44.118	544.698	467.097
	34.341	630.760	565.014		46.776	468.333	568.750		43.898	543.873	466.808
	34.508	630.387	564.710		46.528	467.926	568.465		43.720	543.453	466.513
	34.622	630.254	564.478		46.520	467.785	568.220		43.707	543.317	466.272
	34.530	629.879	563.927		46.344	467.446	568.040		43.407	542.730	465.890
	34.842	629.758	563.894		46.331	467.294	567.834		43.574	542.695	465.768
	34.973	629.721	563.741		46.444	467.251	567.616		43.677	542.692	465.581
	34.891	629.482	563.290		46.441	467.055	567.514		43.544	542.265	465.293
	35.328	629.370	563.358		46.483	466.929	567.394		43.882	542.269	465.249
	35.456	629.274	563.231		46.592	466.836	567.321		44.029	542.144	465.149
	35.190	629.647	563.055		47.182	467.189	567.140		44.164	542.492	465.032
	35.716	629.882	563.484		47.546	467.357	567.061		44.834	543.005	465.228
	35.255	630.026	563.076		47.891	467.566	567.033		44.692	542.953	465.024

Figur C.1: Exempelbild över extraherad mocap-data från QTM.

D

Matlabkod för konvertering av QTM-data till textfil

```
clear all;close all;clc

% Ladda in specifik QTM-data
QTMDData = load('Namn_på_extraherad_QTM_fil');

% Klipp parametrar
antalDimensioner = #; % [1-3]
startFrame = #;
slutFrame = #;

% Extrahera önskad QTM-data samt relaterade nyckelpunkter.
data = QTMDData."Namn_på_extraherad_QTM_fil".Trajectories.Labeled.Data(:,[1:...
antalDimensioner],[startFrame:slutFrame]);
nyckelpunkter = string(data."Namn_på_extraherad_QTM_fil".Trajectories.Labeled...
.Labels);

% Skapa txt-filer för QTM-data och nyckelpunkter.
csvwrite('Namn_på_QTM_txt_filen.txt', Data)

filePh = fopen('Namn_på_nyckelpunkt_txt_filen.txt','w');
fprintf(filePh,'%s\n',labels);
fclose(filePh);
```

E

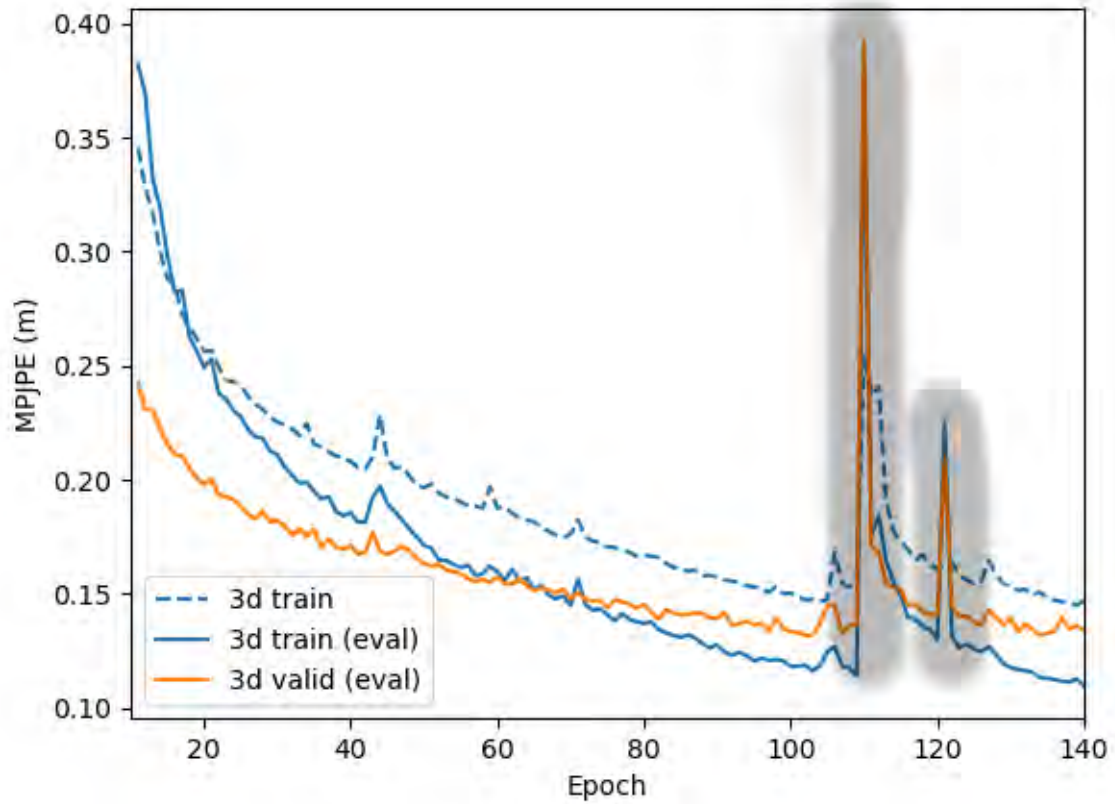
Webbapplikation



Figur E.1: Webbapplikationens sida där användaren kan ladda upp en video och se resultatet.

F

Exploderande kostnad



Figur F.1: Exploderande kostnad under träning ses kring 110 epoker.



CHALMERS