

# **Koneoppimisen hyödyntäminen pituushyppysuorituksen analysoinnissa**

Tekoäly

Tietojenkäsittelytieteiden tutkinto-ohjelma  
Tietotekniikan laitos, Teknillinen tiedekunta  
Pro gradu -tutkielma

Laatija:

Kalle Lehtonen

Ohjaajat:

Erikoistutkija Petra Virjonen  
Professori Jukka Heikkonen

Huhtikuu 2024

## **Pro gradu -tutkielma**

**Tietojenkäsittelytieteiden tutkinto-ohjelma, Tietotekniikan laitos, Teknillinen tiedekunta  
Turun yliopisto**

**Oppiaine:** Tekoäly

**Tutkinto-ohjelma:** Tietojenkäsittelytiede

**Tekijä:** Kalle Lehtonen

**Otsikko:** Koneoppimisen hyödyntäminen pituushyppysuoritusten analysoinnissa

**Sivumäärä:** 88 sivua, 7 liitesivua

**Päivämäärä:** Huhtikuu 2024

Tämän pro gradu -tutkielman tavoitteena oli selvittää, onko mahdollista luoda koneoppimista hyödyntävä järjestelmä, jonka avulla olisi mahdollista määrittää edullisella laitteistolla kuvatuilta pituushyppyvideoilta erilaisia hyppysuoritukseen liittyviä suureita. Ihmiskehon avainpisteiden (esimerkiksi jalkojen nivelten) xy-koordinaatit määritettiin hyppyvideoiden kuvaruuduista koneoppimista hyödyntävän ihmiskehon asennontunnistusalgoritmin avulla. Xy-koordinaattien sijaintien perusteella älypuhelimella kuvatuista kahdeksasta hyppyvideosta määritettiin hyppysuorituksen ponnistusvaiheeseen ja sitä edeltävään vauhdinottoaskeleeseen liittyviä suureita, kuten kontakti-/lentoaikoja, erilaisia kulmia/nopeuksia ja viimeisen askeleen pituus. Työssä tehdyllä järjestelmällä hypyistä määritettyjen arvojen suuruuksia pystyttiin arvioimaan myös avoimeen lähdekoodiin perustuvalla Kinovea-sovelluksella, jota käytetään liikeanalysointiin. Työssä määritettyjen arvojen suuruudet olivat samassa suuruusluokassa Kinovea-sovelluksen avulla määritettyjen arvojen kanssa seuraavilla suureilla: kontakti- ja lentoaika, osa ponnistusvaiheen kulmista ja viimeisen askeleen pituus. Kansainvälisiin pituushyppääjien arvoihin verrattuna työssä määritetyt: kontakti- ja lentoajat, kulmat (ponnistavan/heilahtavan jalan kulmat, kehon kaltevuuskulmat ja hyppykulma), viimeisen askeleen pituudet, ponnistavan jalan polvikulman muutosnopeudet ja heilahtavan jalan keskimääräiset kulmanopeudet, olivat keskenään samassa suuruusluokassa. Työn tuloksena syntyi koneoppimista hyödyntävä järjestelmä, jolla voidaan määrittää pituushyppysuorituksen ponnistusvaiheeseen ja sitä edeltävään askeleeseen liittyviä suureita älykameralla kuvatuilta hyppyvideoilta. Työn tuloksena syntynyt järjestelmä voitaisiin käyttää apuna pituushyppääjien valmennuksessa esimerkiksi vuosittain tapahtuvassa hyppytekniikan kehityksen seuraamisessa.

**Asiasanat:** pituushyppy, ihmiskehon asennon arviointi, Google MediaPipe.

## **KIITOKSET**

Olen suorittanut opintoja vuoden 2017 syksystä alkaen työohessa osa-aikaisesti ja opintovapaalla, kiitos esimiehelleni suopeasta suhtautumisesta opintojen suorittamiseen. Kurssien ja pro gradu -tutkielman tekeminen on ollut iso ponnistus, kiitos perheelleni ja erityisesti vaimolleni kestämisestä.

## **LYHENTEET JA TERMIT**

Alle on kuvattu työssä käytetyt lyhenteet ja termit.

### **Lyhenteet**

**AI** Artificial intelligence - tekoäly

**HPE** Human pose estimation - ihmisen asennon arviointi

**IAAF** International Association of Athletics Federations - kansainvälinen yleisurheiluliitto

**FPS** Frames per second – kuvaruutua sekunnissa

### **Termit**

#### **Algoritmi**

Yleisesti määritettynä algoritmi on yksityiskohtainen kuvaus toimenpiteistä, joilla jokin tietty tehtävä saadaan suoritettua. Tehtävän ratkaisemiseksi tietokoneella pitää ohjelmoida algoritmi. Algoritmi laskee/muodostaa tuloksen sille annettujen syötteiden perusteella. Tietokonesovellukset koostuvat yleensä useista algoritmeista, joista jokainen ratkaisee jonkin tehtävän sille annettujen syötearvojen perusteella.

#### **Ihmisen asennon arviointi (Human pose estimation)**

Ihmisen asennon arvioinnissa pyritään määrittämään kuvasta tai videon kuvaruuduista ihmiskehon avainpisteiden kuten nivelien, kantapään ja isovarpaan sijainti kuvassa tai kuvaruudussa.

#### **Kapea tekoäly (Narrow AI)**

Kapean tekoälyn avulla pystyy ratkaisemaan vain tiettyyn tehtävään liittyvän asian/ongelman, jonka ratkaisemiseksi tekoälyä käyttävä sovellus on tehty.

#### **Klusterointi (Clustering)**

Tekoälyyn liittyvä tietojen ryhmittelytapa, jossa pyritään ryhmittelemään aineiston samankaltaiset näytteet samoihin ryhmiin. Ryhmään kuuluvat ovat keskenään enemmän samankaltaisia, kuin muihin ryhmiin kuuluvat näytteet.

## **Koneoppiminen (Machine Learning)**

Koneoppiminen on tekoälyn osa-alue, jossa tietokoneella ohjelmoitu sovellus pystyy oppimaan asioita saamistaan tiedoista. Koneoppiminen jäljittelee tapaa, jolla ihmiset oppivat asioita. Koneoppimista käyttävä sovellus pystyy vähitellen parantamaan suorituksen tarkkuutta.

## **Kovakoodaus**

Sovellukseen kirjoitettu arvo, jota ei pysty syöttämään, muuttamaan tai hakemaan muualta sovelluksen suorittamisen aikana.

## **Kuvaruutua sekunnissa (Frames per second, FPS)**

Kuvataajuuden yksikkö kuinka monta kuvaruutua esimerkiksi videolle tallennetaan yhden sekunnin aikana.

## **Lineaarinen suhde**

Kahden suureen välinen suhde on lineaarinen, jos yhden yksikön muutos muuttujassa  $x$  aiheuttaa muuttujassa  $y$  muutoksen, joka ei ole riippuvainen  $x$ :n alkuperäisestä arvosta.

## **Luokittelu (Classification)**

Tekoälyyn liittyvä tietojen luokittelutapa, jossa malli opetetaan sellaisella aineistolla, jossa näytteiden luokat ovat tiedossa. Opetetun mallin avulla ennustetaan mihin luokkaan ennen näkemätön syöte kuuluu.

## **Ohjaamaton oppiminen (Unsupervised learning)**

Ohjaamaton koneoppiminen on koneoppimisen osa-alue. Ohjaamattomassa koneoppimisessa opetettavalle koneoppimisalgoritmille annetaan vain syötteet (tulot). Algoritmille ei ole annettu oikeita vastauksia/luokkia, johon sen tulisi päästä. Algoritmi pyrkii etsimään tulojen välisiä samankaltaisuuksia ja ryhmittelemään samankaltaiset samaan ryhmään.

## **Ohjattu oppiminen (Supervised learning)**

Ohjattu koneoppiminen on koneoppimisen osa-alue. Ohjatussa koneoppimisessa opetettavalle koneoppimisalgoritmille annetaan opetusvaiheessa tiedot, jotka pitävät

sisällään syötteet (tulot) ja kutakin syötettä vastaavan oikean vastauksen (lähtö/luokka). Opetusvaiheessa käytettävien tietojen perusteella algoritmi opettaa mallin, jonka avulla uusille syötteille (tuloille) voidaan ennustaa (lähtö/luokka).

### **Syväoppiminen (Deep learning, DL)**

Syväoppiminen on koneoppimisen osa-alue, jonka mallissa käytetään kerroksittain järjestettyjä keinotekoisia neuroneja. Syvissä neuroverkoissa sisään- ja ulostulokerroksien välissä on useampia piilokerroksia.

### **Tekoäly (Artificial intelligence, AI)**

Tekoäly tai keinoäly on yleensä tietokoneella ohjelmoitu sovellus, joka tekee älykkäinä pidettäviä toimintoja.

### **Vahvistusoppiminen (Reinforcement learning)**

Vahvistusoppiminen on koneoppimisen osa-alue, jossa koneoppimismallia opetetaan yrityksen ja erehdyksen avulla. Mallin toiminnasta eri tilanteissa annetaan opetusvaiheessa positiivista ja negatiivista palautetta. Mallin oppimista muokkaa sen saama palaute, tämän vuoksi se pyrkii saamaan mahdollisimman paljon positiivista palautetta.

### **Yleinen tekoäly (General AI)**

Yleisen (tai vahvan) tekoälyn avulla pystyttäisiin jäljittelemään inhimillistä älykkyyttä ja ratkaisemaan useita erilaisia ongelmia. Tämän kaltaista tekoälyä ei ole vielä saatu toteutettua.

# Sisällysluettelo

<b>1</b>	<b>JOHDANTO</b> .....	<b>1</b>
<b>2</b>	<b>TUTKIMUSKYSYMYS</b> .....	<b>4</b>
<b>3</b>	<b>TYÖHÖN LIITTYVÄ TEORIA JA TAUSTAT</b> .....	<b>6</b>
3.1	Pituushypyn lajianalyysi.....	6
3.2	Koneoppimisen soveltaminen urheilussa.....	7
3.3	Koneoppiminen.....	8
3.3.1	Tekoäly.....	9
3.3.2	Koneoppimisen taustat ja osa-alueet.....	9
3.3.3	Tietokonenäkö.....	11
3.3.4	Koneoppimisessa tarvittava aineisto.....	11
3.3.5	Koneoppimisen vaiheet.....	13
3.3.6	Ali- ja ylisovittaminen.....	14
3.4	Keinotekoiset neuroverkot.....	15
3.4.1	Keinotekoinen neuroni.....	16
3.4.2	Neuronin opettaminen.....	18
3.4.3	Gradienttialgoritmi/-menetelmä.....	19
3.4.4	Epookki.....	19
3.4.5	Keinotekoinen neuroverkko ja sen opettaminen.....	20
3.4.6	Vastavirtamenetelmä/-algoritmi.....	21
3.4.7	Aktivaatiofunktio.....	21
3.4.8	Ali- ja ylisovittaminen neuroverkoissa.....	24
3.5	Neuroverkot kuvan tunnistuksessa.....	26
3.6	Konvoluutioneuroverkko.....	27
3.6.1	Suodattimen liu'uttaminen ja nollien lisääminen.....	28
3.6.2	Konvoluutiovaihe.....	29
3.6.3	Yhdistämisvaihe.....	32
3.6.4	Piirrekarttojen muuttaminen vektoriksi.....	33
3.7	Ihmisen asennon arviointi.....	33
3.7.1	Ihmiskehon mallit.....	34
3.7.2	Ihmisen asennon arvioinnin haasteet.....	34
3.7.3	Asennon arviointimenetelmät.....	37
<b>4</b>	<b>MATERIAALIT</b> .....	<b>41</b>
4.1	Hyppyvideot.....	41

4.2	Hyppyvideoiden esikäsittely.....	45
<b>5</b>	<b>TOTEUTUS.....</b>	<b>46</b>
<b>5.1</b>	<b>Toteutuksen menetelmät.....</b>	<b>46</b>
5.1.1	Hampel-suodatin.....	46
5.1.2	Perspektiivin muuttaminen .....	48
5.1.3	Kulmien laskeminen .....	49
5.1.4	Skaalaaminen.....	49
<b>5.2</b>	<b>Vaihe 1 - avainpisteiden etsintä.....</b>	<b>51</b>
<b>5.3</b>	<b>Vaihe 2 – mittausarvojen käsittely ja kontaktikohtien määrittäminen.....</b>	<b>52</b>
5.3.1	Mittausarvojen käsittely .....	52
5.3.2	Kontaktikohtien määrittäminen .....	56
<b>5.4</b>	<b>Vaihe 3 – hyppyjen analysointi.....</b>	<b>59</b>
5.4.1	Kontaktiajat ja viimeisen askeleen lentoaika .....	59
5.4.2	Viimeisen askeleen pituus .....	60
5.4.3	Lonkan pystysuuntainen sijainnin muutos.....	63
5.4.4	Keskimääräinen nopeus viimeisen askeleen aikana.....	64
5.4.5	Ponnistusvaiheen nopeudet .....	65
5.4.6	Hyppykulma .....	65
5.4.7	Ponnistavan jalan polven kulmat.....	65
5.4.8	Ponnistavan jalan polvikulman muutosnopeus .....	66
5.4.9	Lonkan kaltevuuskulmat (kantapää - lonkan).....	66
5.4.10	Kehon kaltevuuskulmat (lonkka - olkapää).....	67
5.4.11	Heilahtavan jalan kulma (lonkka - polvi).....	67
5.4.12	Heilahtavan jalan keskimääräinen kulmanopeus (astetta/sekunti) .....	68
<b>5.5</b>	<b>Vaihe 4 – tulosten kokoaminen.....</b>	<b>68</b>
<b>6</b>	<b>TULOKSET JA NIIDEN TARKASTELU.....</b>	<b>70</b>
<b>6.1</b>	<b>Kontakti- ja lentoajat .....</b>	<b>71</b>
<b>6.2</b>	<b>Kulmat .....</b>	<b>72</b>
<b>6.3</b>	<b>Viimeisen askeleen pituus .....</b>	<b>75</b>
<b>6.4</b>	<b>Nopeudet.....</b>	<b>76</b>
<b>6.5</b>	<b>Muut suureet.....</b>	<b>78</b>
<b>7</b>	<b>MENETELMIEN ARVIOINTI.....</b>	<b>79</b>
<b>7.1</b>	<b>Avainpisteiden tunnistaminen .....</b>	<b>79</b>

7.2	Avainpisteiden tunnistaminen videon toistaminen .....	81
7.3	Avainpisteiden puolten vaihtuminen.....	81
7.4	Kyynärpään avainpisteet.....	82
7.5	Hyppyvideoiden suurennos .....	82
7.6	Skaalaus ja hyppyyn liittyvät nopeudet .....	83
7.7	Massakeskipisteen/lonkan alenema, lonkan/kehon kaltevuuskulmat .....	85
8	YHTEENVETO.....	87
	Lähteet.....	89
	Liitteet.....	97
	Liite 1. Havainnekuva välineistön sijoittelusta kentälle .....	97
	Liite 2. Hypystä tallennetut tiedot.....	98
	Liite 3. Hypyistä tallennetut tiedot (hyppääjä noin 16...17-vuotias).....	99
	Liite 4. Hypyistä tallennetut tiedot (kansallisen tason top-10) .....	100
	Liite 5. Avainpisteiden tunnistaminen (mittaustietoja ei muokattu).....	101

# 1 JOHDANTO

Asennontunnistusalgoritmeihin perustuvia menetelmiä on käytetty monien eri joukkue- ja yksilölajien analysointiin. Tässä työssä toteutettiin pituushyppysuoritukseen asennontunnistusalgoritmia hyödyntävä sovellus, jonka avulla määritettiin erilaisia hyppyyn liittyviä suureita hyppyvideoilta vauhdinottovaiheen loppuosasta ja ponnistusvaiheesta. Kirjallisuudesta ei löytänyt vastaavanlaista tutkimusta, jossa asennontunnistusalgoritmia olisi käytetty pituushyppyyn liittyvien suureiden määrittämisessä.

Huippu-urheilijoiden suorituksien analysointiin käytetään usein yhtä tai useampaa kallista kameraa, vastaavien kameroiden käyttäminen ei kuitenkaan ole mahdollista kustannuksien vuoksi esimerkiksi urheiluseuroissa. Useilla ihmisillä on käytössä älypuhelin, jota käyttämällä urheilusuorituksen kuvaaminen onnistuu ilman merkittäviä lisäkustannuksia. Tämän työn tavoitteena oli tutkia, onko edullisella laitteistolla (älypuhelimella) kuvatuista pituushyppysuorituksesta kuvatuista videoista mahdollista määrittää hyppyyn liittyviä suureita, käyttämällä ihmiskehon asennontunnistusalgoritmiin perustuvaa sovellusta.

Tähän työhön liittyvässä erikoistyössä [1] kuvattiin videoita kahden pituushyppääjän hyppysuorituksista. Hyppysuorituksia kuvattiin kahdella eri suurennos arvolla (1.0x ja 0.5x). Erikoistyössä hyppyvideoille suoritettiin muun muassa kalibrointi (korjaa puhelimen linssin aiheuttaman vääristymän) ja videoiden leikkaaminen lyhyemmäksi sekä stabilointi (poistaa kameran huojuntaa).

Pituushyppysuoritukseen liittyvien suureiden määrittämisen kannalta oleellisia ihmiskehon avainpisteitä ovat lonkan ja jalan (polvi ja nilkka) nivelet, sekä jalan ääripisteet (isovarvas ja kantapää). Avainpisteiden xy-koordinaattien määrittäminen olisi mahdollista tehdä manuaalisesti hyppyvideon kuvaruuduilta, mutta tämä olisi varsin työläs ja aikaa vievä toimenpide. Tässä työssä ihmiskehon avainpisteiden määrittämiseen hyppyvideoilta käytettiin Googlen MediaPipe Pose-asennontunnistusalgoritmia. Saatujen avainpisteiden xy-koordinaattien sijaintien perusteella arvioitiin hyppysuorituksista hyppyyn liittyviä suureita, kuten erilaisia kulmia, nopeuksia ja viimeisen askeleen pituus.

Tämän työn tekemisen aikana kokeiltiin käyttää suurennoksella 0.5x kuvattuja hyppyvideoita, mutta asennontunnistusalgoritmilla oli vaikeuksia määrittää ihmiskehon avainpisteiden xy-koordinaattien sijainnit kuvaruudusta, koska hyppääjän koko kuvaruudulla oli liian pieni. Tässä

työssä hyppyihin liittyviä suureita määritettiin vain suurennoksella 1.0x kuvatuilta hyppyvideoilta, joilta asennontunnistusalgoritmi pystyi määrittämään kehon avainpisteet.

Pituushypyssä vauhdinotto- ja ponnistusvaiheilla (muut vaiheet ilmalento ja alastulo) on suurin merkitys hypyn pituuteen, tässä työssä pyritään analysoimaan erilaisia muuttujia pituushypyn vauhdinottovaiheen loppuosasta ja ponnistusvaiheesta. Ilmalentoon ja alastuloon liittyviä asioita ei pyritty määrittämään tässä työssä. Kansainväliselle yleisurheiluliitolle tehdystä vuoden 2018 hallimaailmanmestaruuskisoissa pituushypyn biomekaniikkaraportissa [2] määritetään noin 15 vauhdinotto- ja ponnistusvaiheeseen liittyvää suurta. Työn edetessä yhdeksi tavoitteeksi asetettiin, että hyppyvideoilta pyritään määrittämään samat suuret, kuin kansainvälisen yleisurheiluliiton raportissa oli määritetty.

Suurennoksella 1.0x kuvatuilta hyppyvideoilla näkyi kaksi viimeistä kontaktikohtaa (ponnistusta edeltävä kontakti ja ponnistus). Suureiden määrittämisen alkukohta oli kohta, jossa hyppääjän kantapää osuu vauhdinotto radan pintaan viimeisessä vauhdinottoaskeleessa (ponnistusta edeltävä kontakti alkaa). Suureiden määrittämisen loppukohta oli kohta, jossa hyppääjän varvas irtoaa ponnistuksessa hyppylankulta.

Työssä käytetty Googlen MediaPipe Pose-asennontunnistusalgoritmi tunnisti kehon avainpisteiden sijainnit varsin hyvin suurennoksella 1.0x kuvattujen hyppyvideoiden kuvaruuduista. Tämän työn aikana tehdyllä sovelluksella eri suureille määritettyjä tuloksia verrattiin kansainvälisen yleisurheiluliiton biomekaniikkaraportin tuloksiin. Työssä määritetyt tulokset ovat samaa suuruusluokkaa kontakti-/lentoajoilla, viimeisen askeleen pituuksille, hypystä määritellyille kulmille ja muutosnopeuksille. Lonkan/massakeskipisteen ja nopeuksien osalta ei päästy samanlaisiin tuloksiin, kuin kansainvälisen yleisurheiluliiton biomekaniikkaraportissa.

Työn toisessa luvussa käydään läpi työhön liittyvät tutkimuskysymykset. Kolmannessa luvussa käydään läpi työhön liittyvää teoriaa ja taustoja, kuten pituushypyn teoriaa ja miten ihmiskehon asennon arviointia on käytetty urheilun analysointiin. Koneoppimista, keinotekoisia neuroverkkoja ja neuroverkkojen käyttämistä kuvantunnistuksessa käsittelevissä kappaleissa käydään läpi teoriaa, jotka luovat perustan ihmiskehon asennon arvioinnille. Kolmannen luvun viimeisessä kappaleessa käsitellään asennon arviointiin liittyviä asioita ja käydään läpi erilaisia asennon arviointimenetelmiä. Teoriaosuuden jälkeen neljännessä luvussa luodaan katsaus analysoinnissa käytettävään materiaaliin eli hyppyvideoihin ja miten niitä on esikäsitelty ennen tämän työn tekemistä. Viidennessä luvussa käsitellään työn toteutukseen liittyvää teoriaa ja

käydään läpi tässä työssä tehtyjen sovellusten toimintaa, luvun kappaleissa käydään läpi myös hyppysuorituksista laskettavat suureet. Kuudennessa luvussa tarkastellaan työssä saavutettuja tuloksia. Seitsemännessä luvussa arvioidaan työssä käytettyjä menetelmiä. Kahdeksannessa luvussa mietitään vielä kehitettäviä asioita. Työn viimeinen luku on yhteenveto.

## 2 TUTKIMUSKYSYMYS

Tämän työn tarkoituksena oli tutkia, pystytäänkö ilman kalliita kameroita kuvatuilta pituushyppyvideoilta määrittämään hyppyyn liittyviä suureita ihmiskehon asennontunnistusalgoritmia käyttämällä. Pituushyppysuoritukset kuvattiin erikoistyössä [1] älypuhelimella.

Erilaisten pituushyppysuoritukseen liittyvien suureiden määrittämiseksi hyppyvideoilta pitää etsiä ihmiskehon avainpisteitä, joille on määritettävä hyppyvideon kuvaruuduista xy-koordinaattien sijainnit. Pituushyppysuorituksesta arvioitavien suureiden määrittämisessä lantion ja jalkojen nivelien sekä isovarpaan ja kantapään avainpisteiden sijainnit ovat tärkeitä, esimerkiksi nilkan/polven/lantion xy-koordinaattien sijaintien avulla voidaan laskea polvikulma. Avainpisteiden xy-koordinaattien sijainnit voitaisiin määrittää manuaalisesti kuvaruuduista, mutta tämä olisi varsin työlästä toteuttaa. Asennontunnistusalgoritmia käyttämällä avainpisteiden xy-koordinaattien sijainteja ei tarvitse määrittää manuaalisesti, vaan algoritmi määrittää ihmiskehon avainpisteiden xy-koordinaattien sijainnit kuvaruuduista. Työssä käytetyksi ihmiskehon asennon arviointialgoritmiksi valikoitui Googlen kehittämä MediaPipe-tuoteperheen Pose-kirjasto.

Tämän työn tärkeimmät tutkimuskysymykset, ovat:

1. Onko mahdollista tehdä sovellus tai sovelluksia, jolla voidaan määrittää pituushyppysuoritukseen liittyviä suureita edullisella laitteistolla kuvatuilta hyppyvideoilta? Soveltuuko tekoälyyn perustuva esikoulutettu ihmiskehon asennontunnistusalgoritmi, ihmiskehon avainpisteiden xy-koordinaattien sijaintien määrittämiseen pituushyppyvideon kuvaruuduilta?
2. Tunnistaako asennontunnistusalgoritmi ihmiskehon avainpisteet riittävän tarkasti hyppyvideoilta, jotta hyppysuoritukseen liittyviä suureita voidaan määrittää? Löytyykö määritetyistä avainpisteiden xy-koordinaateista esimerkiksi virheellisiä arvoja (outliereita) tai esiintyykö niitä hyppyvideolla kohdissa, jotka haittaavat hyppyyn liittyvien suureiden määrittämistä? Onko virheellisten xy-koordinaattiarvojen tunnistaminen ja korjaaminen mahdollista? Suoriutuuko asennontunnistusalgoritmi avainpisteiden tunnistamisesta pituushyppyvideon kuvaruuduilta järkevässä ajassa?

3. Ovatko sovelluksella tai sovelluksilla määritettyjen suureiden arvot riittävän tarkkoja, jotta niiden avulla on mahdollista esimerkiksi seurata hyppääjän kehittymistä kauden aikana?

### 3 TYÖHÖN LIITTYVÄ TEORIA JA TAUSTAT

Koneoppimisen ja tietokonenäön avulla voidaan suorittaa ihmisen asennon arviointia esimerkiksi kuvasta tai videolta. Tämän luvun kappaleissa käydään lyhyesti läpi ihmisen asennon arviointiin liittyviä teorioita ja käsitteitä. Käsiteltäviä käsitteitä ovat tekoäly ja koneoppiminen. Lisäksi käydään läpi ihmisen asennon arvioinnin perusteita ja sen soveltamista urheilussa, sekä pituushypyssä. Seuraavassa kappaleessa käydään läpi pituushyppysuoritukseen liittyviä asioita.

#### 3.1 Pituushyppyn lajianalyysi

Pituushyppy on laji, jossa pituushyppääjä yrittää vauhdinoton ja ponnistuksen avulla lennättää itsensä mahdollisimman pitkälle hiekkakasaan. Pituushyppysuoritus voidaan jakaa eri vaiheisiin: vauhdinotto, ponnistus, ilmalento ja alastulo. Vauhdinotossa pyritään saavuttamaan paras mahdollinen ponnistusasento mahdollisimman suurella nopeudella, josta kontrolloitu ponnistus on mahdollinen. [3] Ponnistusvaiheen tavoitteena on muuttaa riittävän suuri osa vaakasuuntaisesta nopeudesta pystysuuntaiseksi nopeudeksi, kuitenkin siten, että vaakasuuntaista nopeutta menetettäisiin mahdollisimman vähän. [4] Vauhdinotto- ja ponnistusvaiheet vaikuttavat merkittävimmin (yli 95 %) hypyn pituuteen. Ilmalennon ja alastulon päätehtäviä ovat pituushyppääjän kehon tasapainon ylläpitäminen ja hypyn pituuden maksimointi. Ilmalennolla ja alastulolla on vain minimaalinen vaikutus hypyn pituuteen, koska hyppääjän massakeskipisteen lentorata on määrätynyt jo ponnistavan jalan irrotessa hyppylankusta. [5]

Viimeisten askelten tarkoitus on ponnistukseen valmistautuminen, tämä tehdään muuttamalla askelten rytmiä ja tiheyttä. Mitä suurempi vauhdinotossa saavuttu nopeus, niin sitä vähemmän jää aikaa suorittaa viimeiset ponnistuksen valmistavat askeleet oikein. Rytmin muutos alkaa jo 4...6 viimeisellä askeleella, merkittävin muutos tapahtuu vasta kahdella viimeisellä askeleella. Toiseksi viimeisellä askeleella hyppääjän tulisi laskea massakeskipistettä selvästi alaspäin. Tämä mahdollistaa itse ponnistuksessa muutoksen sen kiihtyvyyden suunnassa ylöspäin. Jotta massakeskipiste laskisi ilman suurempaa pienenemistä horisontaalinopeudessa, tulisi toiseksi viimeistä askelta pidentää. Sen sijaan viimeinen askel tulisi olla nopea ja lyhyt. [5]

Ponnistus on pituushyppyn tärkein vaihe, mutta ilman edellä kuvattua oikeanlaista valmistautumista ponnistus jää kuitenkin tehottomaksi. Viimeiset askeleet on suoritettava niin, että vaakasuuntaisen nopeuden menetys on mahdollisimman pieni. [5] Hyppääjän

massakeskipisteen tulisi saavuttaa mahdollisimman alhainen kohta ponnistavan jalan iskeytyessä lankkuun ja hyppääjän vartalon tulisi olla pienessä takakenossa. Tämän seurauksena ponnistuksen edetessä massakeskipisteen kiihtyvyyssmatka on mahdollisimman pitkä ja sen suunta on ylöspäin. Ponnistava jalka toimii ponnistuksen aikana vipuvartena tai niin sanottuna pivot -mekanismina, joka antaa hypylle pystynopeutta vaakanopeuden kustannuksella. Ponnistuksen edetessä vartalo pysyy hiukan takakenossa tai lähes pystysuorassa asennossa. Ponnistuksessa heilahtava jalka tulee tuoda polvi edellä (kantapää pakaran kautta) mahdollisimman nopeasti eteen, tämän jälkeen reiden tulisi olla suunnilleen vaakatasossa. Samalla hyppyjalan puoleinen käsi pitää nostaa tehokkaasti ylös ja eteenpäin. Heilahtavan jalan ja hyppyjalan puoleisen käden heilahtavat liikkeet tehostavat ponnistusnopeutta. Heilahtavan jalan ja käsien liikkeillä ohjataan hyppy ylös ja eteenpäin. [3, 5]

Tässä työssä pyritään analysoimaan erilaisia muuttujia vauhdinottovaiheen loppuosasta ja ponnistusvaiheesta. Kansainväliselle yleisurheiluliitolle tehdystä vuoden 2018 hallimaailmanmestaruuskisoissa pituushypyn biomekaniikkaraportissa [2] analysoidaan noin 15 suuretta, jotka liittyvät vauhdinottovaiheeseen ja ponnistusvaiheeseen. Tässä työssä pyrittiin analysoimaan erikoistyössä [1] kuvatuilta hyppyvideoilta samat suuret, kuin kansainvälisen yleisurheiluliiton raportissa oli analysoitu.

### **3.2 Koneoppimisen soveltaminen urheilussa**

Urheilijoiden suorituksia analysoidaan usein biomekaniikan avulla, jossa tutkitaan ihmiskehon liikettä suorituksen aikana. [6] Usein biomekaniikkaan liittyvissä tutkimuksissa urheilusuurituksen videokuvaamiseen on käytetty useita erikoiskameroita (monikamerakuvausjärjestelmä), joiden tiedot yhdistetään kolmiulotteiseksi malliksi (työläs prosessi). Kameroiden tietojen yhdistämiseksi, on kameroiden tarkat sijainnit, suuntaukset ja kalibroinnit oltava tiedossa/kunnossa. Suorituksen tekevään henkilöön on mahdollisesti myös kiinnitetty seurantapisteitä (markkereita, marker), jotka helpottavat liikkeen (esimerkiksi polven) seuraamista. [7] Monikamerajärjestelmällä kuvattujen suoritusten prosessointi on yleensä niin työlästä, että se rajoittaa esimerkiksi tutkimukseen osallistuvien henkilöiden määrän pieneksi. [8]

Koneoppimista soveltavan ihmisen asennon arviointisovelluksen käyttäminen mahdollistaa esimerkiksi älypuhelimella kuvattujen videoiden käytön urheilusuuritusten analysoinnissa. Tämä pienentää kustannuksia ja vähentää tarvittavan työn määrää, esimerkiksi: ei tarvita useita

erikoiskameroita (hinta, sijoittelu, kalibrointi) ja useiden eri videoiden tietoja ei tarvitse yhdistää keskenään. Asennon arviointisovellus tunnistaa ihmiskehon avainpisteet videolta, joten niitä ei tarvitse etsiä ja merkitä manuaalisesti kuvaruuduista.

Suoritusten analysointi harjoituksissa ja kilpailuissa/otteluissa on ammattiurheilussa varsin yleistä. Joukkuelajeissa seurataan pelivälineen (esimerkiksi pallon) ja pelaajien liikkeitä pelikentällä, tämän perusteella voidaan analysoida esimerkiksi yksittäisen pelaajan liikkumista kentällä tai joukkueen pelaajien sijoittumista pelikentällä. Joukkuelajeissa koneoppimiselle haasteita aiheuttavat usean pelaajan samanaikainen seuraaminen, okklusio (pelaaja ei ole kuvaruudulla näkyvä kokonaan esimerkiksi toisen pelaajan taakse jäävä pelaaja), pelaajat ovat välillä kuvaruudun ulkopuolella sekä lähekkäin tai toisiinsa kontaktissa olevat pelaajat. [9, 10] Yksilölajeissa kuvasta tai videon kuvaruudusta koneoppimisen avulla arvioidaan yleensä ihmisen asentoa kehon avainpisteiden (kuten nivelien ja muiden kehonosien esimerkiksi kantapäähän tai isovarpaan) avulla, pisteiden perusteella voidaan esimerkiksi arvioida nivelten kulmia ja tunnistaa lajiin liittyviä tapahtumia. Ihmisen asennon arviointia on käytetty monenlaisissa sovelluksissa esimerkiksi pöytätenniksessä lyöntitavan tunnistamiseen [11], sulkapallon pelaajan suorittaman lyönnin tunnistamiseen (määritetään sulkapallon lentorata) [12], mäkihypyn tyyliä laskevassa sovelluksessa [13], jujutsussa asennon määrittämisessä ja pistelaskennassa [14] ja jooga-asentojen määrittämisessä [15].

Yleisurheilussa ihmisen asennon arvioivaa sovellusta on käytetty aitajuoksun analysointiin [16]. En löytänyt tutkimusta, jossa olisi käytetty koneoppimiseen perustuvaa ihmisen asennon arviointisovellusta pituushyppäsuoritusten analysoinnissa. Kansainvälinen yleisurheiluliitto on tehnyt vuosien 2017 ja 2018 hallimaailmanmestaruuskisoista biomekaniikkaraportteja eri lajeista (kuvattu useilla kameroilla) [17].

### **3.3 Koneoppiminen**

Ihmisen asennon arviointiin perustuvissa menetelmissä käytetään hyväksi tekoälyä, koneoppimista ja tietokonenäköä, tämän luvun kappaleissa käydään läpi niiden taustoja ja toimintaa. Edellä mainitut menetelmät ovat laajoja aihealueita, joten ne käsitellään tämän työn kannalta tarvittavassa laajuudessa. Syväoppimisen perusteiden läpi käynnin lisäksi työssä käsitellään keinotekoisien neuroverkkojen, konvoluutioneuroverkkojen ja takaisinkytkettyjen neuroverkkojen rakennetta ja toimintaa.

### 3.3.1 Tekoäly

Tekoälyn pyrkimyksenä on tietokonetta käyttäen automatisoida älyllisiä tehtäviä, joiden on perinteisesti ajateltu vaativan ihmisälyä. Tekoälyä käyttävä tietokonesovellus pystyy tehtävän suorittamisen aikana itsenäisesti mukauttamaan toimintaansa.

Tekoälyn yhteydessä käytetään termejä kapea tekoäly ja yleinen tekoäly. Kaikki nykyisin käytössä oleva tekoäly on heikkoa tai kapeaa tekoälyä (weak artificial intelligence, narrow artificial intelligence), jota voidaan käyttää vain tietyn tarkkaan määritetyn tehtävän ratkaisemiseen. Yleisimpiä kapean tekoälyn käyttökohteita ovat: ennustaminen, tietojen luokittelu (classification), tietojen ryhmittely (clustering) ja poikkeamien tunnistaminen (anomaly detection). Kapeaan tekoälyyn liittyviä sovelluksia käytetään esimerkiksi sään ennustamisessa, roskapostiviestien suodattamisessa, itseohjautuvissa autoissa, kuvien/kasvojen tunnistamisessa, puheen tunnistamisessa, tekstin lukemisessa tai kääntämisessä, pankin lainapäätösten tekemisessä, lääketieteellisten diagnoosien teossa ja verkkosivujen mainosten kohdentamisessa. Yleinen tekoäly (general artificial intelligence) pystyisi puolestaan oppimaan ja soveltamaan oppimaansa erilaisiin tehtäviin, joita sillä ei ole aiemmin ratkaissut. Yleisellä tekoälyllä olisi tietoisuus ja se pystyisi ratkaisemaan itsenäisesti ihmisen kaltaisesti kaikenlaisia tehtäviä, tämän kaltaista tekoälyä ei ole vielä pystytty luomaan. [18, 19, 20]

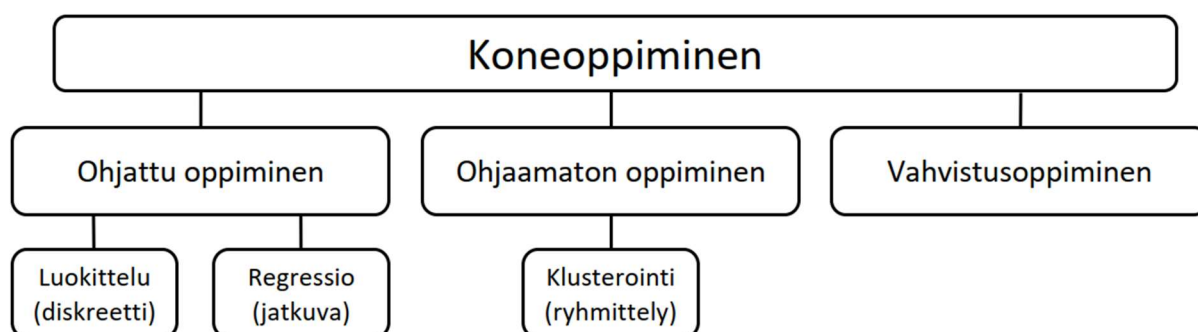
Monissa tänä päivänä käytössä olevissa järjestelmissä ja sovelluksissa käytetään hyväksi tekoälyyn liittyviä menetelmiä ja aihealueita, jotka voidaan luokitella monella eri tavalla. Tämän työn kannalta oleellisia tekoälyyn liittyviä menetelmiä ja aihealueita ovat: koneoppiminen, tietokonenäkö, neuroverkot ja syväoppiminen, näiden toimintaa käydään läpi seuraavissa luvuissa.

### 3.3.2 Koneoppimisen taustat ja osa-alueet

Cholletin mukaan koneoppiminen syntymiseen liittyi seuraavanlaisia kysymyksiä (vapaasti käännettynä):

Pystyisikö tietokoneella tehty sovellus oppimaan itse tietyn tehtävä suorittamisen ja ratkaisemaan sen ilman, että sovellus suorittaisi tehtävän ennalta määrättyllä tavalla? Voisiko tietokone oppia ratkaisuun tarvittavat säännöt automaattisesti tarkastelemalla sille annettuja tietoja? [21]

Koneoppimisen avulla pyritään ratkaisemaan tiedonjalostustehtävä. Tehtävä voi olla esimerkiksi auton hinnan ennustaminen, käsin kirjoitetun tekstin tulkitseminen tai luottokorttipetoksen havaitseminen. Koneoppimisalgoritmin tehtävänä on tuottaa käytettävään opetusaineistoon perustuva malli, joka ennustaa mahdollisimman hyvin ennennäkemättömillä syötteillä. [22] Usein koneoppimista käyttävä sovellus opetetaan käyttämällä opetusaineistoa, jonka jälkeen luotua mallia voidaan käyttää ennustamiseen. Opetusvaiheen aikana sovellus muuttaa mallin sisäisiä parametrejaan. Ennustamisessa mallille annetaan ennennäkemättömiä syötteitä, joille malli ennustaa arvon. Mikäli mallia on tarvetta päivittää (esimerkiksi mallin suorituskykyä halutaan parantaa), niin mallia voidaan opettaa käyttämällä uutta opetusaineistoa käyttäen.



Kuva 1. Koneoppimisen osa-alueet. Kuva: Kalle Lehtonen

Helsingin yliopiston verkkokurssin materiaalissa [23] koneoppiminen jaetaan kolmeen osa-alueeseen (kuva 1): ohjattu oppiminen, ohjaamaton oppiminen, vahvistusoppiminen. Eri osa-alueiden väliset rajat ovat kuitenkin epämääräisiä ja lisäksi osa koneoppimismenetelmistä eivät sovi tähän yksinkertaistetun jaotteluun, vaan ovat yhdistelmiä tai muunnelmia näistä. [23] Edellisten lisäksi koneoppimisen yhtenä osa-alueena voidaan pitää myös keinotekoisia neuroverkkoja [25], joita sovelletaan ihmisen asennon arvioinnissa. Keinotekoisien neuronien ja neuroverkkojen toimintaa käydään läpi seuraavassa luvussa.

Ohjatussa oppimisessa (supervised learning) koneoppimismallin opettaminen suoritetaan käyttämällä opetusaineistoa, joka pitää sisällään syötteen ja tuloksen eli tiedon oikeasta luokasta tai lopputuloksesta (aineisto koostuu: syöte-tulos pareista). [26] Ohjattu oppiminen voidaan jakaa kahteen luokkaan (kuva 1): luokittelu ja regressio. Luokittelussa malli ennustaa mihin luokkaan syötteenä annettu näyte kuuluu. Regressiossa mallin tuottamat ennusteet ovat puolestaan jatkuvia esimerkiksi arvio ulkolämpötilasta. [27] Tässä työssä keinotekoisien neuroverkkojen opettamista käsittelevissä osuuksissa opettaminen on ohjattua oppimista.

Ohjaamattomassa oppimisessa (unsupervised learning) opetusaineisto ei pidä sisällään tulosta eli oikeaa vastausta (aineistoa ei siis ole luokiteltu). Ohjatussa oppimisessa tavoitteena on löytää aineistosta samankaltaisia rakenteita. Klusterointi on yksi ohjaamattoman oppimisen menetelmä, jossa tavoitteena on ryhmitellä samankaltainen aineisto samoihin ryhmiin. [22, 28]

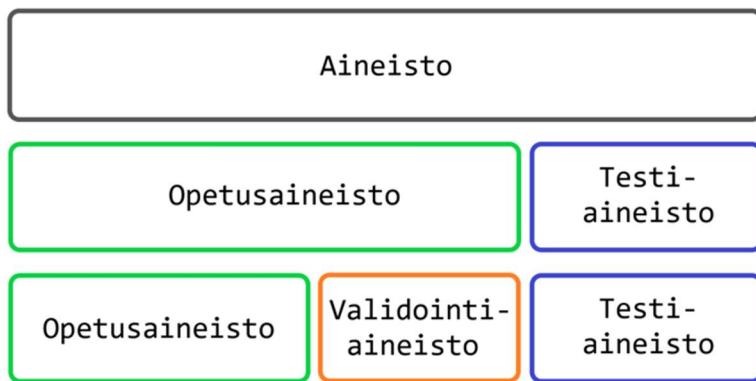
Vahvistusoppiminen (reinforcement learning) on yrityksen ja erehdyksen avulla oppimista, sovellukselle ei anneta oikeita tuloksia. Vahvistusoppimisessa sovellukselle annetaan positiivista ja negatiivista palautetta siitä, kuinka hyvin se on toiminut eri tilanteissa. Palautteen perusteella sovellus pyrkii parantamaan toimintaansa. [22, 26]

### 3.3.3 Tietokonenäkö

Tietokonenäkö (computer vision) on tekoälyyn liittyvä osa-alue, jossa yritetään opettaa tietokoneita tunnistamaan visuaalisia asioita (digitaalisista kuvista tai videoilta) samalla tavalla kuin ihmiset. Tietokonenäön avulla voidaan saada merkityksellistä tietoa kuvista tai videoista, joiden perusteella käyttäjä tai sovellus voi tehdä toimenpiteitä tai antaa suosituksia tehtävistä toimenpiteistä. Tietokonenäköä käytetään esimerkiksi kasvojentunnistussovelluksissa, itse ajavissa autoissa, urheilusuoritusten analysoinnissa ja teollisuudessa valmistusvirheiden havaitsemiseen. [29, 30]

### 3.3.4 Koneoppimisessa tarvittava aineisto

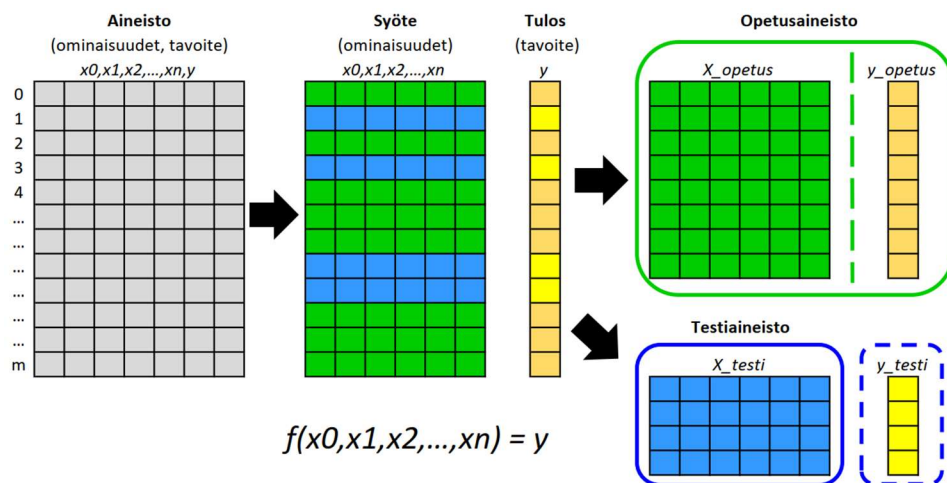
Koneoppimissovelluksen ohjelmointivaiheessa tarvitaan tietoja eli aineistoa, jota käyttäen valittu koneoppimismalli opetetaan. Aineiston laatu määrittelee mitä siitä voidaan oppia, sillä koneoppimismalli ei pysty päättämään asioita, joita aineistossa ei löydy [31]. Koneoppimismallin opetusvaiheessa käytettävän aineiston pitää olla kattava tutkittavan ongelman kannalta, tällöin malli oppii mahdollisimman tehokkaasti ja ennustusten tarkkuus paranee. Mikäli opetuksessa käytettävä aineisto on huonolaatuista, niin tarkasti ennustavan koneoppimismallin toteuttaminen on epätodennäköistä. Epätarkan ja/tai vinoutuneen aineiston käyttäminen opetusvaiheessa johtaa siihen, että myös mallin tuottamat ennusteet ovat epätarkkoja ja/tai vinoutuneita. Hyvälaatuisella opetusaineistolla opetettu koneoppimismalli pystyy tuottamaan tarkempia ennustuksia ennen näkemättömän aineistoa käytettäessä. [32] Opetetun koneoppimismallin testaamisessa tarvitaan lisäksi aineistoa, jota malli ei ole aiemmin nähnyt (eli sitä ei ole käytetty mallin opetusvaiheessa).



Kuva 2. Aineiston jakaminen opetus-, testi- ja validointiaineistoiksi Kuva: Kalle Lehtonen

Usein käytettävissä oleva aineisto jaetaan kahdeksi toisistaan erilliseksi joukoksi/ryhmäksi (kuvan 2 keskimmäinen rivi): opetus- ja testausaineistoksi. Aineisto voidaan jakaa myös kolmeen ryhmään (kuvan 2 alimmainen rivi), mikäli opetettavassa tekoälymallissa on esimerkiksi malliin liittyviä käyttäjän muutettavissa olevia hyperparametreja. [33]

Opetusaineistoa käytetään koneoppimismallin opettamisessa, jossa mallin sisäiset parametriarvot lasketaan/päivitetään (eli opetetaan) aineistoa käyttäen. Testiaineistolla puolestaan testataan, miten hyvin opetettu koneoppimismalli toimii näytteillä, jotka ovat sille ennen näkemättömiä/uusia syötteitä. Validointiaineiston avulla voidaan tarkastella esimerkiksi millä hyperparametrien arvoilla opetettu malli toimii parhaiten ja valita kyseiset parametriarvot lopulliseen malliin.



Kuva 3. Esimerkki aineiston jakaminen syötteeksi ja tulokseksi sekä opetus- ja testiaineistoksi. Kuva: Kalle Lehtonen

Opetus-, validointi- ja testausaineistoa käytetään vain kyseisessä vaiheessa. Mikäli aineisto on jaettu kolmeen osaan, niin validointivaiheen jälkeen voidaan valittu malli opettaa uudelleen opetus- ja validointiaineistoa käyttäen ja testata testiaineistolla. Oli aineisto sitten jaettu kahteen

(kuva 3) tai kolmeen osaan, niin lopullinen (käyttöön tai tuotantoon otettava) malli opetetaan koko aineistoa käyttäen.

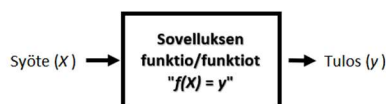
Asennontunnistusalgoritmin opetusvaiheessa opetus- ja testiaineistoksi tarvitaan kuvia tai kuvaruutuja (videolta), joissa on kuvattu ihmisiä (miehiä, naisia, lapsia ja aikuisia) erilaisissa asennoissa, vaatteissa, valaistuksessa, kulmissa ja niin edelleen. Mikäli asennontunnistusalgoritmia käytetään ihmiskehon avainpisteiden sijainnin ennustamiseen kuvasta/kuvaruudusta, niin mallin opetusvaiheessa käytettävän aineiston eli kuvien/kuvaruutujen on lisäksi oltava annotoituja, eli jokaisesta kuvasta/kuvaruudusta on tiedossa kehon avainpisteiden (esimerkiksi polven tai nilkan) sijainnit kuvaruudulla (kaksiulotteisissa asennon arviointimenetelmissä xy-koordinaatit ja kolmiulotteisissa xyz-koordinaatit). Ilmaisia asennontunnistusalgoritmien opetukseen ja testaamiseen käytettäviä aineistoja on saatavilla useita muun muassa MPII, COCO ja Human3.6M [34]. MPII tietokannassa on yksittäisen ihmisen asennon arviointiin saatavilla kuvia noin 25000 kappaletta (opetukseen: 15000, validointiin: 3000 ja testaukseen: 7000 kuvaa), kuvat on kerätty YouTube videoilta, kattavat noin 410 erilaista aktiviteettia ja kuviin on käsin merkitty 16 kehon avainpistettä [35]. COCO (Common Objects in Context) tietokannassa on ihmisen asennon arviointiin saatavilla kuvia noin 250 000 kappaletta, joihin on merkitty 17 kehon avainpistettä [36]. Human3.6M tietokannassa on noin 3,6 miljoonaa kolmiulotteista kuvaa ihmisen asennon arviointiin, kuvissa on 11 näyttelijää, kattavat 17 erilaista aktiviteettia ja niihin on merkitty 24 kehon avainpistettä [34].

Asennontunnistusalgoritmin mallin opetusvaiheessa käytetään opetusaineistoa, vaiheen suorittamisen aikana mallin sisäisiä parametrejä muutetaan. Opetusvaiheen jälkeen mallin toimivuutta voidaan testata testiaineistolla, jota ei ole käytetty mallin opetusvaiheessa.

### 3.3.5 Koneoppimisen vaiheet

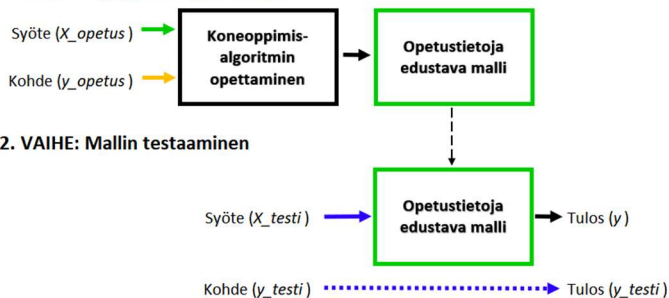
Perinteisessä sovelluksessa sovelluksen tekijä on erikseen ohjelmoinut sovellukseen tarvittavat toiminnot (yksittäisen funktion tai useampia funktiota) [21], joiden perusteella syötteelle ( $X$ ) muodostetaan/lasketaan tulos ( $y$ ) (kuvan 4 vasemmalla puolella). Koneoppimisen (kuva 4 oikealla puolella) ensimmäisessä vaiheessa koneoppimisalgoritmi opetetaan opetustietoja käyttäen (ohjatussa oppimisessa:  $X_{opetus}$  ja  $y_{opetus}$  tai ohjaamattomassa oppimisessa pelkästään  $X_{opetus}$ ), tämän vaiheen tuloksena syntyy opetustietoja edustava malli. Mallia käytetään varsinaisen tuloksen ( $y$ ) muodostamisessa/laskemisessa. Mallin testausvaiheessa mallin syötteeksi annetaan  $X_{testi}$ , jolle malli muodostaa/laskee varsinaisen tuloksen ( $y$ ).

### Perinteinen sovellus



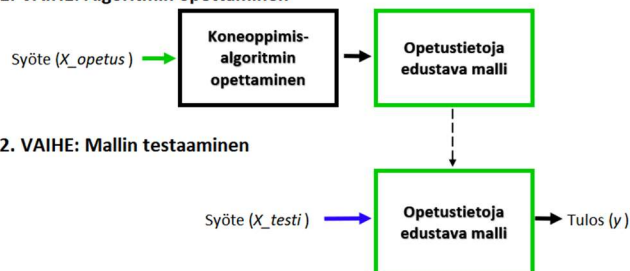
### Koneoppiminen - Ohjattu oppiminen

#### 1. VAIHE: Algoritmin opettaminen



### Koneoppiminen - Ohjaamaton oppiminen

#### 1. VAIHE: Algoritmin opettaminen

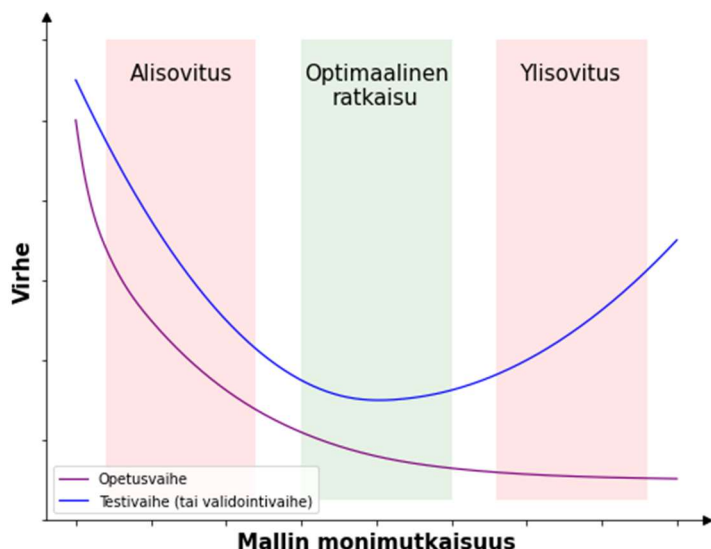


Kuva 4. Perinteinen sovellus ja ohjatun/ohjaamattoman oppimisen koneoppimismallien opettaminen sekä testaaminen Kuva: Kalle Lehtonen

Asennontunnistusalgoritmin opetusvaiheessa (ohjattu oppiminen) koneoppimisalgoritmile syötetään kuvia, joihin on liitetty tieto kehon avainpisteen (esimerkiksi polvi, nilkka ja niin edelleen) sijainnista kuvaruudussa. Syötettyjen kuvien perusteella mallin parametreja muutetaan ja opetusvaiheen jälkeen syntyneitä malleja voidaan testata. Testausvaiheessa mallille syötetään ennennäkemättömiä/uusia kuvia, joille malli ennustaa avainpisteen sijainnin kuvassa. Testivaiheessa käytetyissä kuvista on tiedossa myös avainpisteen todellinen sijainti, johon ennustettua avainpisteen sijaintia voidaan verrata.

### 3.3.6 Ali- ja ylisovittaminen

Koneoppimismallin opettamisen tavoitteena on opetustietoja edustava malli, joka suoriutuu mahdollisimman hyvin, kun mallille annetaan ennen näkemättömiä/uusia syötteitä (esimerkiksi testivaiheessa). Mikäli koneoppimismalli suoriutuu huonosti näkemättömillä/uusilla syötteillä, niin tämä voi johtua ali- tai ylisovittamisesta. Alisovittaminen (underfitting) johtuu siitä, että opetusaineiston perusteella muodostettu malli on liian yksinkertainen opetusaineistoon verrattuna, esimerkiksi lineaarisella mallilla ei pysty mallintamaan epälineaarisia suhteita. Ylisovittamisessa (overfitting) opetusvaiheessa luotu koneoppimismalli on puolestaan liian monimutkainen, jolloin se on oppinut opetusaineistossa olevan kohinan, mutta ei ole oppinut sen ominaisuuksia, tällöin malli tuottaa virheellisiä tuloksia. [37]



Kuva 5. Mallin ali-/ylisovitus ja optimaalinen ratkaisu Kuva: Kalle Lehtonen

Liian yksinkertainen malli aiheuttaa alisovittamisen, tällöin mallin opetuksen aikana ja testivaiheessa (kuvan 5 vasemmanpuoleinen alue) suuri osa mallin tuottamista tuloksista ovat luokitteluun liittyvissä tehtävissä virheellisiä tai regressioon liittyvissä tehtävissä mallin tuottaman tuloksen virhe on suuri. Ylisovittavalla mallilla (kuvan 5 oikeanpuoleinen alue), opetusvaiheessa mallin tuottamat tulokset ovat lähes virheettömiä, mutta esimerkiksi testivaiheessa annetuilla mallille ennen näkemättömillä/uusilla syötteillä tuottamat tulokset ovat suurelta osin virheellisiä. Mallin opetusvaiheessa tulisi löytää optimaalinen malli (kuvan 5 keskimäinen alue). [37] Keinotekkoisten neuroverkkojen ali-/ylisovittaminen käydään läpi, keinotekkoisten neuroverkkoihin liittyvässä osuudessa.

### 3.4 Keinotekoiset neuroverkot

Keinotekoisilla neuroverkoilla (Artificial Neural Network, ANN) jäljitellään ihmisen hermoston/aivojen neuroneiden ja synapsien toimintaa. Neuroverkkojen tutkimus katsotaan alkaneen 1940-luvulla, jolloin tutkijat mallinsivat ensimmäisen kerran neuronin toimintaa. [38]. Ihmisen hermostossa/aivoissa on erittäin suuri määrä prosessointiyksikköinä toimivia neuroneita, joista jokaisella on suuri määrä synapseiksi nimitettyjä yhteyksiä toisiin neuroneihin. Keinotekoisissa hermoverkoissa simuloidaan neuronien ja synapsien toimintamekanismia. Neuronit toimivat rinnakkain ja välittävät viestejä synapsien välityksellä toisille neuroneille. Neuronien lähettämät viestit eivät ole diskreettejä (eli 1 tai 0) tiloja, vaan neuronin aktivoituessa se laukeaa taajuudella, jonka arvo on jatkuva. Nykyään uskotaan, että neuronit suorittavat prosessoinnin ja synapsit toimivat puolestaan muistina. [28].

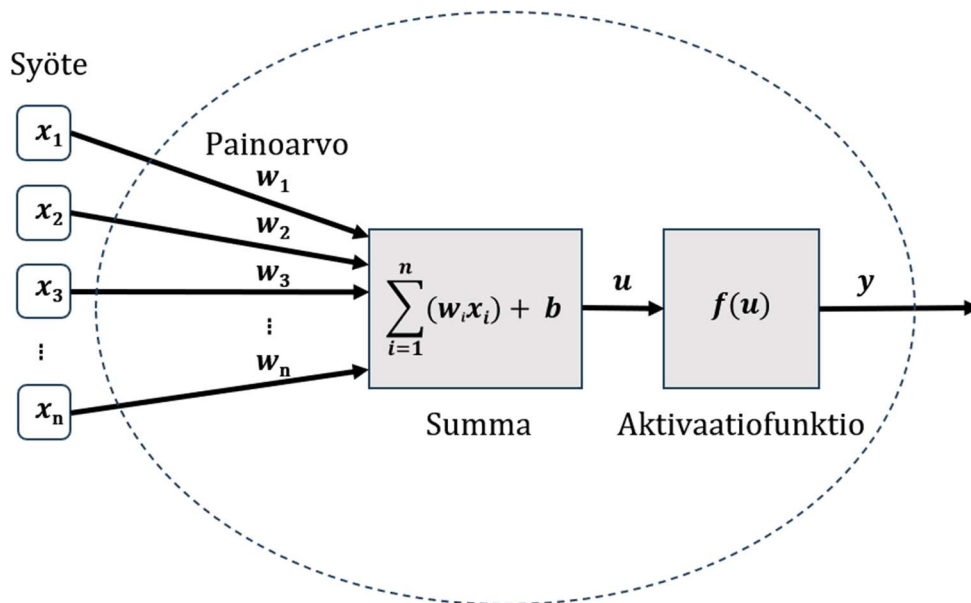
Mielenkiinto neuroverkkoihin räjähti 2010-luvulla, kun kuvantunnistamisessa saavutettiin parannuksia aiempiin tuloksiin verrattuna ja syvien neuroverkkojen käyttäminen yleistyi konenäköön liittyvissä tehtävissä. Neuroverkkojen läpimurron mahdollistivat tarjolla olevien tietomäärien kasvaminen (neuroverkon opetusvaiheessa tarvitaan paljon aineistoa), internetin myötä aineiston kerääminen ja jakelu helpottui, tietokoneiden laskentakapasiteetti kasvoi merkittävästi kehitettyjen grafiikkasuorittimien ansiosta, menetelmien kehittyminen mahdollisti erittäin (tuhansia kerroksia) syvien neuroverkkojen opettamisen, merkittävien tulosten ansiosta yritysten kiinnostus ja rahoitus kasvoivat merkittävästi. [21, 39] Syvien neuroverkkojen avulla on saavutettu erittäin hyviä tuloksia myös ihmisen asennon arvioinnissa ja niitä käytetään monissa asennon arviointimenetelmissä [40].

Keinotekoisia neuroverkkoja on monenlaisia, kuten monikerroksiset (eteenpäin kytketyt) neuroverkot, konvoluutioneuroverkot ja takaisinkytketyt neuroverkot. Monikerroksiset neuroverkot ovat yleisimpiä syviä neuroverkkoja ja ne muodostavat perustan muille neuroverkoille. Konvoluutioneuroverkon avulla pystytään tunnistamaan kuvista kuvioita, niitä käytetään tietokonenäköön liittyvissä kuvan-/hahmontunnistussovelluksissa. Takaisinkytketyissä neuroverkoissa on eteenpäin kytkettyjen signaalien lisäksi myös kytkentöjä takaisinpäin neuroverkkoon, ja ne kykenevät muistamaan aiemmin tapahtuneita asioita. Takaisinkytkettyjä neuroverkkoja käytetään ensisijaisesti aikasarjallisen aineiston kanssa, kuten puheentunnistuksessa, kielenkääntämisessä ja osakemarkkinoiden ennustamisessa. [41, 42]

Tämän luvun kappaleissa käydään läpi keinotekoisien neuronien rakenteeseen, opettamiseen ja gradienttimenetelmään/-algoritmiin liittyviä asioita. Lisäksi seuraavissa luvuissa käydään läpi monikerroksisen neuroverkon rakenne, opettaminen ja opettamiseen liittyvä vastavirtamenetelmä/-algoritmi, sekä konvoluutioneuroverkkojen rakenne ja toiminta.

### 3.4.1 Keinotekoinen neuroni

Keinotekoinen neuroverkko koostuu keinotekoisista neuroneista (jatkossa työssä käytetään nimitystä neuroni) ja niiden välisistä yhteyksistä. Neuroni on prosessointia suorittava osa, joka käsittelee ja yhdistää syötteenä saadut arvot toisiinsa sekä muodostaa syötteiden arvojen perusteella lähtöarvon. Neuronin (kuva 6) parametreja ovat syötteiden painoarvot/kertoimet ja kynnyksarvon/vakiotermin ( $b$ ) arvo, joiden arvoja muutetaan neuronien muodostaman keinotekoisien neuroverkon opetusvaiheessa.



Kuva 6. Keinotekoisien neuronien rakenne. Kuva: Kalle Lehtonen

Neuroni koostuu seuraavista osista (kuva 6): syöte, syötearvoon liittyvä painoarvo/kerroin, lineaarinen summausfunktio, kynnsarvo/vakiotermi, aktivaatiopotentiali, aktivaatiofunktio ja aktivaatioarvoista eli neuronin lähtöarvoista. Syöte sisältää yleensä useita syötearvoja ( $x_1, x_2, x_3, \dots, x_n$ ), jotka ovat neuronille tulevia tuloja/signaaleja esimerkiksi muuttujien arvoja. Kuvankäsittelyssä pikselien väriarvoja tai muiden neuronien lähettämiä arvoja. Jokaista syötettä joko vahvistetaan tai heikennetään painoarvoa ( $w_1, w_2, w_3, \dots, w_n$ ) käyttäen, painoarvon avulla määritetään kunkin syötteen merkittävyys suoritettavassa tehtävässä, esimerkiksi luokittelussa niiden syötteiden painoarvojen arvot ovat suuria, jotka antavat virheettömmiä tuloksia. Syötteiden painottamisen jälkeen painotetut arvot summataan keskenään (lineaarinen summa) ja tulokseen lisätään kynnsarvo ( $b$ ), joka vaikuttaa neuronin aktivoitumisherkkyyteen. Kynnsarvo vastaa suoraan yhtälön vakiotermiä (siirtää suoraa ylös/alas y-akselilla), sitä käyttäen siirretään päätösraja pois origosta. Painotettujen syötearvojen summan ja kynnsarvon yhteenlaskun tulosta ( $u$ ) kutsutaan aktivaatiopotentialiksi, joka on aktivaatiofunktion ( $f(\cdot)$ ) syötearvo. Aktivaatiofunktion ulostulo on neuronin lähtöarvo ( $y$ ). [43, 44, 45] Aktivaatiofunktioiden ominaisuuksia käydään tarkemmin läpi myöhemmin, niitä käsittelevässä kappaleessa.

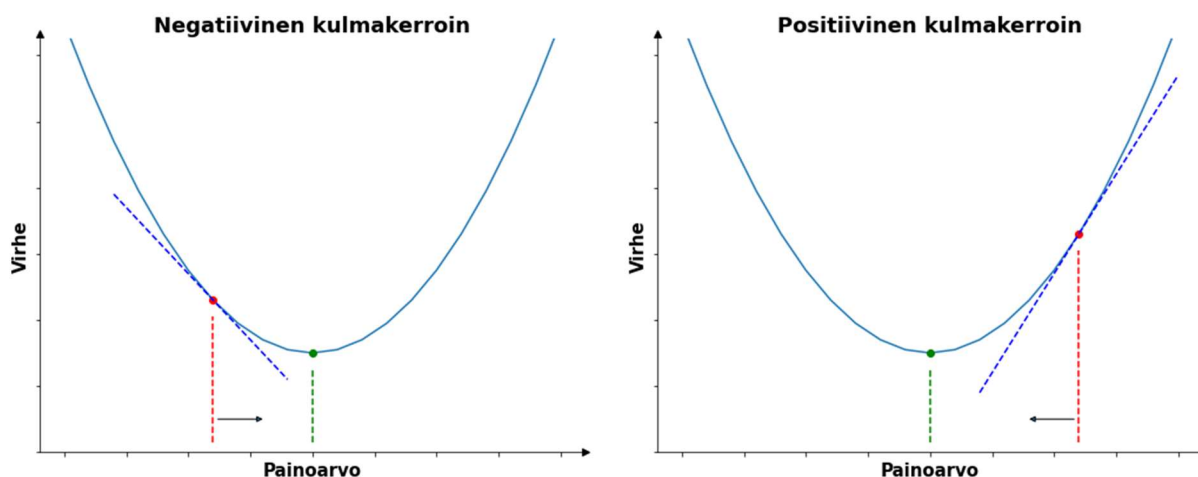
$$y = f \left( \sum_{i=1}^n (w_i x_i) + b \right) \Rightarrow y = f \left( \sum_{i=0}^n (w_i x_i) \right) \quad (1)$$

Edellä esitettiin, että neuronin lineaariseen summaan lisätään kynnsarvo ( $b$ ) (kaavassa 1 vasemmalla). Kynnsarvoksi voitaisiin antaa vakioarvo käyttäjän toimesta, mutta siitä voidaan

tehdä opittava arvo. Tämä voidaan tehdä muuttamalla (kaavassa 1 oikealla) kynnsarvo ( $b$ ) painoarvoksi ( $w_0$ ) ja siihen liittyvän syötteen arvo vakioarvoksi yksi ( $x_0 = 1$ ), lisäksi summan indeksi muutetaan alkamaan nolasta ( $i = 0$ ). [45] Kynnsarvon sisällyttäminen summaan painoarvona nopeuttaa neuroneista koostuvan keinotekoisien neuroverkon opettamista, koska tällöin painotetun summan laskeminen voidaan suorittaa kahden vektorin kertolaskuna. [45] Jatkossa kynnsarvon laskentaa käsitellään tässä työssä niin, että se on sisällytetty painoarvojen laskentaan.

### 3.4.2 Neuronin opettaminen

Ilman opettamista neuroni ei pysty ratkaisemaan minkäänlaista tehtävää, ennen opettamista neuronien syötteiden painoarvoille annetaan satunnaiset arvot. Mikäli neuronille syötetään tässä vaiheessa esimerkiksi kaksi muuttujaa sisältävä syötepari ( $x_1, x_2$ ), niin neuronin muodostama ulostulo on satunnainen. Opetuksen aikana (ohjatussa oppimisessa) voidaan neuronin ( $x_1, x_2$ ) syöteparille laskea, kuinka paljon neuronin lähtöarvo ( $y$ ) poikkeaa halutusta lähtöarvosta. Tätä poikkeamaa kutsutaan virheeksi. Opetuksen aikana (jokaisella syöteparilla) neuronin syötteiden painoarvoja päivitetään/sovitetaan niin, että neuronin tekemän virheen määrä minimoituu. Neuronia voidaan ajatella funktiona, jonka muutettavia parametreja ovat neuronin syötteiden painoarvot, funktion tavoitteena on minimoida virheen arvo.



Kuva 7. Virhe painoarvon funktiona (tangentin negatiivinen ja positiivinen kulmakerroin). Kuva: Kalle Lehtonen

Mikäli esimerkin vuoksi kynnsarvoon liittyvä painoarvo pidettäisiin neuronin opettamisen aikana vakiona ja neuronilla olisi vain yksi painoarvo, niin virhe painoarvon funktiona voisi olla kuvan 7 mukainen (vasemman- ja oikeanpuoleisissa kuvissa neuronilla on eri syötteet). Kuvan 7 kuviin on merkitty punaisella pisteellä tietyn syötteen painoarvon aloitusarvo, joka aiheuttaa virheen (virheen suuruus y-akselilla). Virheen pienimmän arvon (vihreä piste)

saavuttamiseksi, neuronin painoarvoa pitäisi kasvattaa tai pienentää. Neuron ei kuitenkaan tiedä missä kohtaa virheen minimikohta on ja ei osaa valita muutoksen suuntaa. Tämän selvittämiseksi neuronin etsii käyrän tangentin (kuva 7 sininen katkoviiva) kyseisessä pisteessä (kuva 7 punainen piste) ja laskee sen kulmakertoimen, joka kertoo funktion muutosnopeuden ja suunnan. Mikäli kulmakerroin on negatiivinen (kuvan 7 vasemmanpuoleinen kuva) painoarvon arvoa kasvatetaan (nuoli kuvassa), mikäli kulmakerroin on positiivinen (kuvan 7 oikeanpuoleinen kuva) painoarvon arvoa pienennetään (nuoli kuvassa). Funktion kulmakerroin tietyssä pisteessä voidaan laskea derivoimalla, virheen pienentämiseksi painoarvoa on muutettava derivaatan verrattuna vastakkaiseen suuntaan. [44] Mikäli neuronin syöte sisältää useita eri syötearvoja/muuttujia, tällöin edellä esitettyä derivaatan laskemista kutsutaan gradienttialgoritmiksi.

### 3.4.3 Gradienttialgoritmi/-menetelmä

Normaalisti neuronilla on useita syötteisiin liittyviä painoarvoja, neuronin sovittaa niiden painoarvoja edellä esitetyn yhden painoarvon esimerkin mukaisesti. Derivaatan sijasta neuronin laskee virheen niin sanotun osittaisderivaatan kunkin yksittäisen painoarvon suhteen. Kun neuronissa on  $n$  kappaletta painoarvoja, niin derivaattoja on yhteensä  $n + 1$  kappaletta. Kun funktiossa on monta muuttujaa, niin funktion kaikki osittaisderivaatat sisältävää vektoria kutsutaan gradientiksi (vastaa kulmakerrointa). Pienentääkseen virheen arvoa on neuronin päivitettävä/sovitettava kutakin painoarvoa suuntaan, joka on vastakkainen kuin gradientin muodostamien osittaisderivaattojen osoittama suunta. Neuronin ulostuloarvon virheen minimoivaa painoarvojen päivittämistä gradientin avulla kutsutaan gradienttialgoritmiksi. [44]

Yksinkertaistettuna gradienttimenetelmän oppimisprosessia voidaan kuvata kaksivaiheisella algoritmilla, joka käy läpi aineiston syöte-ulostulo parit. Ensimmäisessä vaiheessa lasketaan neuronin ulostulo ja virhe. Toisessa vaiheessa päivitetään neuronin painoarvot siten, että virhe minimoituu. [44] Keinotekoisien neuroverkkojen opettamista ja neuroverkkojen opettamisessa käytettävää vastavirtamenetelmää/-algoritmia käsitellään tarkemmin tämän luvun kappaleissa.

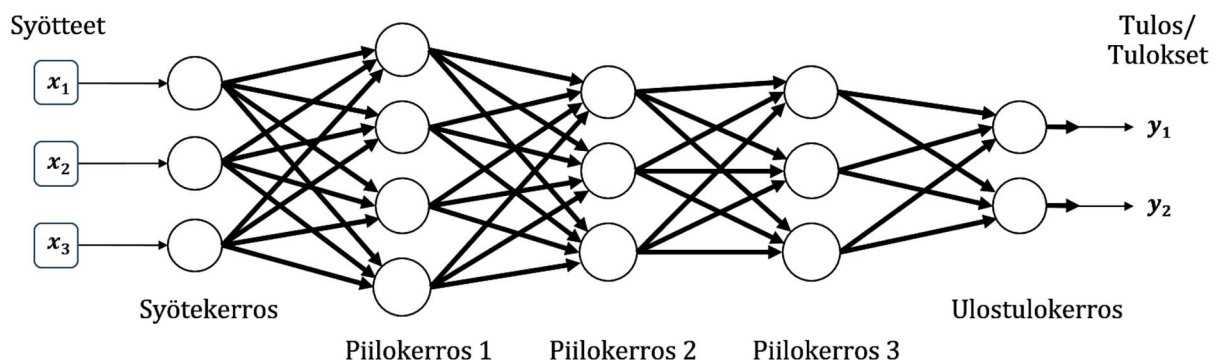
### 3.4.4 Epookki

Opetusvaiheessa keinotekoiselle neuroverkolle syötetään suuri määrä opetusaineistoa. Kun opetusvaiheessa opetusaineisto on kokonaisuudessaan käyty läpi, niin tällöin sanotaan, että on kulunut yksi epookki. Yleensä opetusvaihetta ei lopeteta tähän vaan prosessi toistetaan useita kertoja (useita epookkeja) peräkkäin. Tällöin keinotekoisien neuroverkkojen neuronit ovat jo

oppineet aiempien epookkien aikana neuronien painoarvoja, jolloin tuloksen pitäisi parantua edelliseen epookkiin verrattuna. [44] Jokaisen epookin suorittamisen aikana, neuronien painoarvoihin tehdään pieni muutos (kohti virheen minimikohtaa).

### 3.4.5 Keinotekoinen neuroverkko ja sen opettaminen

Yhden neuronin avulla voidaan ratkaista vain varsin yksinkertaisia tehtäviä. Monimutkaisten tehtävien ratkaisemiseksi neuroneista muodostetaan keinotekoinen neuroverkko (jatkossa keinotekoisesta neuroverkosta käytetään nimitystä neuroverkko), jossa neuronit on järjestetty kerroksiksi ja neuronien lähtöarvot voivat olla muiden neuroneiden syötearvoja. Neuroverkkojen kyky mallintaa monimutkaisia suhteita, perustuu yksinkertaisten neuronien suureen määrään ja keskinäiseen vuorovaikutukseen. Yleensä neuroverkon yhden kerroksen neuronit on kytketty seuraavan kerroksen kaikkiin neuroneihin (kuten kuvassa 8), tällöin neuroverkon sanotaan olevan täysin kytketty (fully connected). Kuvassa 8 esitellyssä neuroverkossa on viisi kerrosta: syötekerros, kolme piilokerrosta ja ulostulokerros. Ensimmäinen kerros eli syötekerros vastaanottaa neuroverkolle annettavat syötteet, syötekerros toimii muistipaikkana ja siinä ei muokata syötettä. Viimeisen kerroksen eli ulostulokerroksen neuronien lähtöarvot ovat puolestaan neuroverkon muodostamat tulokset. Mikäli kerros ei ole syöte- tai ulostulokerros, niin sen on piilokerros. Neuroverkon sanotaan olevan syvä neuroverkko (deep neural network, DNN), mikäli siinä on syöte- tai ulostulokerroksen lisäksi vähintään kaksi piilokerrosta. Useimmissa syväoppivissa neuroverkoissa piilokerrosten on huomattavasti tätä suurempi. Verkon syvyyden mitta on piilokerrosten ja ulostulokerroksen summa. [44, 45]



Kuva 8. Eteenpäin kytketyn neuroverkon rakenne. Kuva: Kalle Lehtonen

Neuroverkon opettaminen tapahtuu samalla tavalla kuin yksittäisen neuronin opettaminen. Ennen neuroverkon opettamista, neuronien painoarvoille annetaan satunnaiset arvot. Opetusvaiheessa neuroverkolle syötetään opetusaineistosta arvoja syöte kerrallaan ja opetuksen

aikana neuronien painoarvojen arvoja päivitetään siten, että neuroverkosta tulee asteittain paremmin toimiva ja se alkaa vastata opetusaineistoa yhä paremmin. Neuroverkossa signaalit etenevät neuroverkossa nuolien suuntaisesti (kuvassa 8 vasemmalta oikealle) syötekerroksesta piilokerroksien kautta ulostulokerrokseen. Ulostulokerroksen neuronien lähtöjen arvot ovat neuroverkon laskemat/muodostamat tulokset.

Neuroverkko on rakenteeltaan monikerroksinen perseptroniverkko (Multi-Layer Perceptrons, MLP), mikäli se on: eteenpäin kytketty, täysin kytketty ja siinä on syötekerros, (yksi tai useampi) piilokerros ja ulostulokerros. Eteenpäin kytketty tarkoittaa sitä, että syötteenä saadut signaalit kulkevat vain kohti ulostulokerrosta (signaaleja ei ole takaisinkytketty). Kuvan 8 neuroverkon rakenne täyttää monikerroksisen perseptroniverkon määritelmän. [46, 47] Jatkossa työssä monikerroksisesta perseptroniverkosta käytetään nimitystä neuroverkko. Neuroverkkojen opettamisessa käytettävää vastavirtamenetelmää/-algoritmia käsitellään seuraavassa luvussa.

### 3.4.6 Vastavirtamenetelmä/-algoritmi

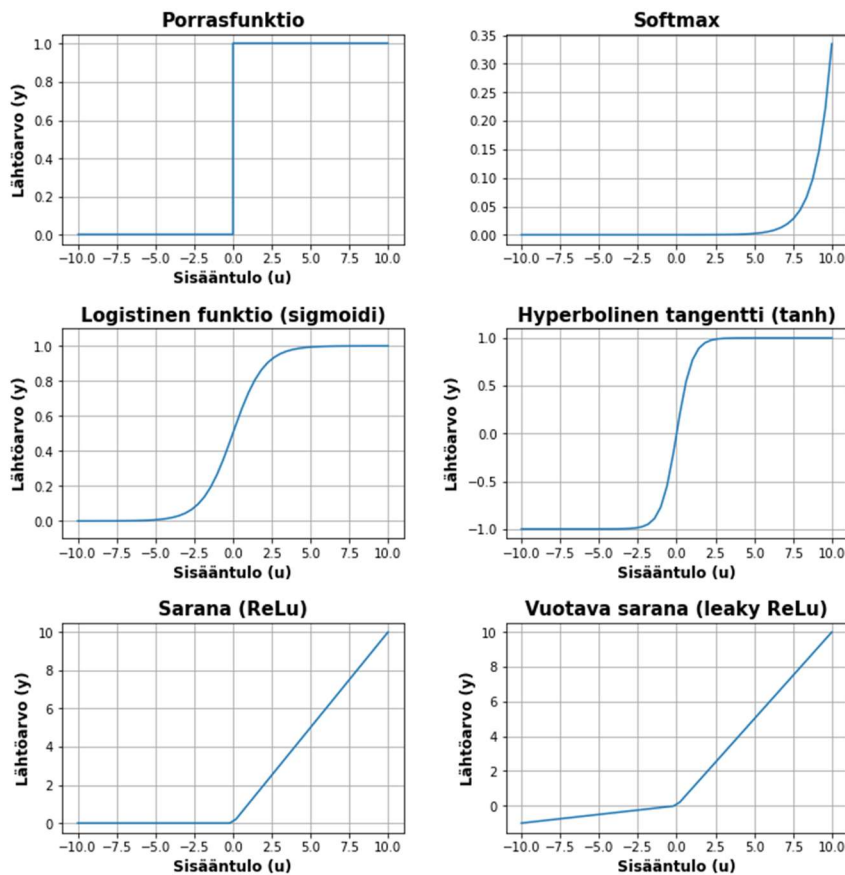
Vastavirtamenetelmän/-algoritmin (backpropagation) avulla eteenpäin kytketyn neuroverkon painoarvot päivitetään/sovitetaan neuroverkon opetusvaiheessa. Vastavirtamenetelmän ensimmäisessä vaiheessa neuroverkko laskee syötearvoille (kuvassa 8:  $x_1$ ,  $x_2$  ja  $x_3$ ) tulokset (kuvassa 8:  $y_1$  ja  $y_2$ ), jonka jälkeen päivitetään/sovitetaan neuroverkon kaikkien neuronien painoarvojen arvot. Päivittämisen tarkoituksena on muuttaa neuronien painoarvoja niin, että neuronin ulostulon virhe minimoituu (esimerkiksi gradientiaalgoritmia käyttäen). Vastavirtamenetelmän toisessa vaiheessa painoarvojen päivittäminen/sovittaminen tehdään kerroksittain, kulkemalla neuroverkossa vastakkaiseen suuntaan eli vastavirtaan, siis ulostulokerroksesta kohti syötekerrosta (kuvassa 8 oikealta vasemmalle). Ulostulokerroksen neuronien painoarvot päivitetään ensimmäiseksi, jonka jälkeen päivitetään viimeisen piilokerroksen painoarvot. Tämän jälkeen päivitetään viimeistä piilokerrosta edeltävän piilokerroksen painoarvot, piilokerroksien neuronien painoarvojen päivittäminen etenee näin piilokerros kerrallaan ja viimeisenä päivitetään ensimmäisen piilokerroksen neuronien painoarvot. [44]

### 3.4.7 Aktivaatiofunktio

Aktivaatiofunktio mainittiin aiemmin keinotekoisia neuroneja käsittelevässä tekstissä. Aktivaatiofunktiolla (kuvassa 6  $f(u)$ ) on ratkaiseva merkitys neuronin käyttäytymiseen. Mikäli

keinotekoisien neuroverkon neuroneissa ei olisi aktivaatiofunktioita, niin neuronien toiminta rajoittuisi lineaariseksi kuvaukseksi (painotetun summan laskemiseen) ja neuroverkko pystyisi mallintamaan vain muuttujien välisiä lineaarisia suhteita. Epälineaaristen suhteiden mallintaminen ei kuitenkaan olisi kovin tarkkaa tai onnistunutta (liian yksinkertainen neuronin rakenne johtaa alisovittamiseen). Periaatteessa minkä tahansa epälineaarisen funktion käyttäminen aktivaatiofunktiona mahdollistaisi neuroverkon oppia syötteen/syötteiden ja ulostulon välisiä (epälineaarisia) suhteita. Aktivaatiofunktioina käytetään kuitenkin sellaisia funktioita, joiden matemaattisista ominaisuuksista on hyötyä neuroverkon käyttämisessä ja opettamisessa. [45]

Neuroverkon eri kerroksissa (piilokerroksissa ja ulostulokerroksessa) käytetään erilaisia aktivaatiofunktioita, yleensä neuroverkon tietyssä kerroksessa käytetään vain yhdenlaisia neuroneja (eli niiden aktivaatiofunktiot ovat samoja). [45] Erilaisia aktivaatiofunktioita on olemassa monenlaisia, funktioiden valinta onkin tärkeä vaihe keinotekoisien neuroverkon suunnitteluvaiheessa. Nettisivuilla [48] on listattu yli 60 erilaisia aktivaatiofunktioita. Aktivaatiofunktiot voidaan kahteen ryhmään: neuroverkon ulostulokerroksessa ja piilokerroksissa käytettäviin funktioihin. Ulostulokerroksen aktivaatiofunktiot rajaavat lähdön arvot tietylle alueelle (kuten välille 0...1). Piilokerrosten aktivaatiofunktioiden on tarjottava epälineaarisuus, ilman epälineaarisuutta keinotekoiset neuroverkot eivät voisi mallintaa epälineaarisia suhteita. [49]



Kuva 9. Erilaisia aktivaatiofunktioita. Kuva: Kalle Lehtonen

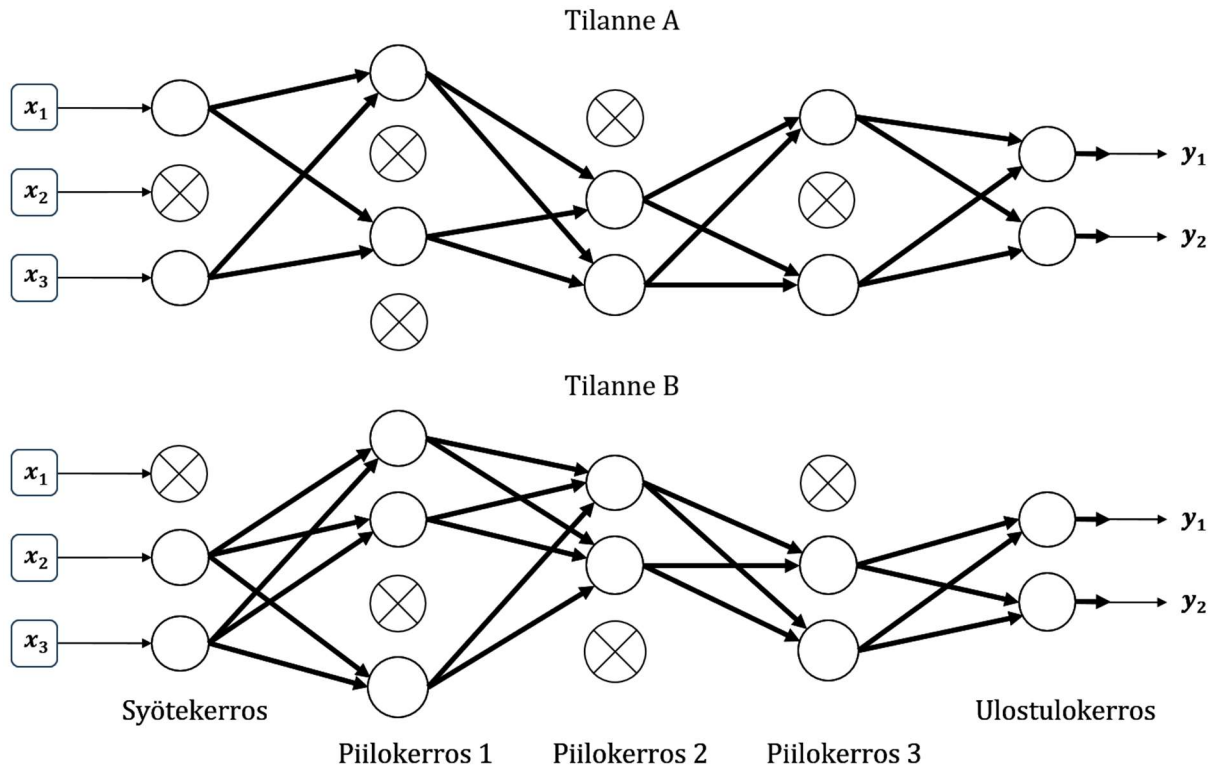
Kuvaan 9 on koottu kuuden aktivaatiofunktion kuvat: porras-/askelfunktio, softmax, logistinen funktio (sigmoidi), hyperbolinen tangenti (tanh) ja saranafunktioita (Rectified Linear Unit, ReLU). Erilaisia aktivaatiofunktioita käytetään siksi, että käytännössä jotkin niistä soveltuvat paremmin tiettyihin asioihin kuin toiset. [44] Porras-/askelfunktion lähtöarvo on yksi, kun funktion sisääntulon arvo on suurempi kuin nolla (muuten ulostulon arvo on nolla). Porrasfunktiota käytetään binäärisissä (tosi/epätosi) luokittelutehtävissä. Porrasfunktiota ei voi käyttää vastavirtamenetelmässä, koska se ei ole porraskohdan epäjatkuvuuden vuoksi derivoituva [45]. Softmax funktiota käytetään ulostulokerroksessa, kun luokkia on enemmän kuin kaksi. Softmax funktio laskee todennäköisyyden luokkiin kuulumisesta ja pakottaa ulostulokerroksen lähtöarvojen summaksi yksi. Logistinen funktio (sigmoidi) puristaa ulostulonarvon välille  $0 \dots 1$ . Logistista funktiota käytetään usein tehtävissä, joissa arvioidaan todennäköisyys kahdelle eri luokalle (lasketaan todennäköisyys molemmille luokille siihen, että neuroverkon syöte kuuluu luokkaan). Hyperbolinen tangenti (tanh) on logistisen funktion kaltainen funktio, joka puristaa ulostulonarvon välille  $-1 \dots 1$ . Neuroverkon piilokerroksissa hyperbolinen tangenti toimii lähes aina paremmin kuin logistinen funktio, koska siinä aineiston keskiarvo keskittyy  $0,5:n$  sijasta nollan lähelle. Saranafunktion (Rectified Linear Unit, ReLU) arvo on nolla, jos sisääntulon arvo on pienempi kuin nolla ja muuten lähtöarvo seuraa

lineaarisesti sisääntuloarvoa. Saranafunktion ongelmana on, että negatiivisilla sisääntulo arvoilla derivaatan arvo on nolla. Saranafunktio toimii hyvin erilaisissa tilanteissa ja sitä käyttämällä saavutetaan parempia tuloksia kuin logistisella tai hyperbolisella tangentti aktivointifunktioilla. Saranafunktion kehittyneempi versio on vuotava saranafunktio, siinä negatiiviset sisääntulojen arvot kerrotaan arvolla 0,01 ja muuten lähtöarvo on sisääntuloarvon suuruinen. Vuotava saranafunktio lieventää saranafunktioon liittyvää derivaattaan liittyvää ongelmaa. Neuroverkon ulostulokerroksen neuronien aktivointifunktioina käytetään usein logistista funktiota (binäärisissä luokittelutehtävissä) ja softmax funktiota (luokkia on enemmän kuin kaksi). Neuroverkon piilokerroksissa aktivointifunktioina käytetään puolestaan yleensä sarana- tai vuotavaa saranafunktioita. [49, 50]

### 3.4.8 Ali- ja ylisovittaminen neuroverkoissa

Neuroverkoissa alisovittaminen ja ylisovittaminen ovat neuroverkon tiloja, jotka voivat syntyä, kun neuronien määrä on liian pieni tai suuri. Alisovittavassa neuroverkossa on liian vähän (neuroneja) painoarvoja, jolloin neuroverkko ei pysty opetusvaiheessa oppimaan opetusaineistosta neuroverkon painoarvoja. Liian pieni neuroverkko ei ole tarpeeksi joustava oppiakseen painoarvoja opetusaineistosta, tällöin neuroverkon tuloksien virhe on suuri opetus- ja testausaineistolla. Ylisovittavassa neuroverkossa on puolestaan liikaa (neuroneja) painoarvoja ja neuroverkko on liian joustava, jolloin neuroverkko oppii opetusaineistossa esiintyvän kohinan (opetusaineiston erityisominaisuudet ja häiriöt). Tällainen neuroverkko pystyy tuottamaan oikeita tuloksia (virhe on hyvin pieni) opetusaineistolla. Annettaessa ylisovittavalle neuroverkolle ennen näkemättömiä/uusia syötteitä, niin sen tuottamat tulokset eivät ole tarkkoja (tulokset ovat virheellisiä). [27, 43]

Neuroverkon riskiä ylisovittamiseen voidaan vähentää esimerkiksi pienentämällä neuroverkon neuronien määrää ja neuroverkon opettamiseen liittyvillä säännöstelymenetelmillä. Neuronien määrän vähentäminen neuroverkossa yksinkertaistaa opetettavaa neuroverkkomallia, jolloin opettavien painoarvojen määrä vähenee, tämän seurauksena neuroverkon kyky oppia opetusaineiston kohinaa vähenee. Neuroverkon pienentäminen voidaan tehdä poistamalla neuroverkosta piilokerroksia ja/tai vähentämällä yksittäisten piilokerrosten neuronien määrää. [51]



Kuva 10. Kaksi esimerkitilannetta neuroverkon opetuksen aikaisesta neuronien väliaikaisesta poistamisesta (rastitetut neuronit eivät ole käytössä). Kuva: Kalle Lehtonen

Neuroverkon ylisovittamisen riskiä voidaan opetusvaiheessa vähentää säännöstelemällä/rajoittamalla neuroverkon opettamista opetusvaiheessa, usein käytettyjä menetelmiä ovat esimerkiksi: neuronien poistaminen, opetusvaiheen pysäyttäminen aikaisemmin sekä L1 ja L2 säännöstely. Neuronien poistamisessa (dropout) osa neuroverkon syöte-/piilokerroksen neuroneista poistetaan opetusvaiheessa väliaikaisesti pois neuroverkosta, tällöin neuronin liittynät edelliseen ja/tai seuraavaan kerrokseen katkaistaan, jolloin neuronin painoarvoja ei päivitetä. Neuronien tiputtaminen tehdään satunnaisesti esimerkiksi seuraavasti: annetaan jokaiselle syöte-/piilokerroksen neuroneille tiputtamiseen liittyvä todennäköisyys, jonka perusteella neuroni tiputetaan pois opetusvaiheessa. [51, 52] Kuvassa 10 on esitetty kaksi eri tilannetta neuroverkon rakenteesta opetusvaiheessa, molemmissa tapauksissa jokaisesta neuroverkon syöte- ja piilokerroksista neuroneja on poistettu. Poistaminen voidaan toteuttaa myös niin, että se kohdistuu vain esimerkiksi yhteen piilokerrokseen. Neuronien määrän vähentäminen neuroverkossa yksinkertaistaa neuroverkkoa, jolloin sen kyky oppia opetusaineiston kohinaa vähenee ja riski ylisovittamiseen pienenee.

Neuroverkon opetusvaiheessa liian useiden epookkien suorittamisesta (eli liian runsaasta opettamisesta) johtuva ylisovittaminen voidaan estää lopettamalla opetusvaihe ennakoitua aikaisemmin (early stopping). Neuroverkon opetusvaiheessa jokaisen epookin jälkeen

lasketaan (ennen näkemätöntä) validointiaineistoa käyttämällä validointivirheen arvo. Kun validointivirhe ei enää parane (eli sen arvo alkaa nousta) edelliseen epookkiin verrattuna, niin neuroverkon opettaminen lopetetaan. Mikäli neuroverkon opettamista jatkettaisiin tämän jälkeen, niin riski neuroverkon ylisovittamiseen kasvaisi. [44, 50, 53] Kuvassa 5 neuroverkon opettaminen lopetettaisiin, kun testi-/validointivaiheen kuvaaja saavuttaa minimikohdan.

L1 ja L2 säännöstelyt (regulazation) toteutetaan lisäämällä säännöstelytermi painoarvon virhefunktion laskentaan, tämän tuloksena neuroverkon neuronien painoarvojen arvot pienevät. Neuroverkon opetusvaiheessa L2 (ridge regression) ja L1 (lasso regression) säännöstelyt pakottavat neuronien painoarvot lähelle nollaa, lisäksi L1 säännöstely voi pakottaa arvon nolnaan (tällöin neuronin toiminta lakkaa). Neuronien toiminnan lakkauttamisen vuoksi, L1 säännöstelyn voi ajatella toteuttavan ominaisuuksien valintaa (feature selection). [53, 54]

### 3.5 Neuroverkot kuvan tunnistuksessa

Täysin yhdistetyssä neuroverkossa kerroksen neuronit on kytketty kaikkiin seuraavan kerroksen neuroneihin ja jokaisella kytkennällä on painoarvo, jonka arvoa päivitetään opetusvaiheessa. Mikäli täysin kytketylle neuroverkolle annettaisiin käsiteltäväksi pieni mustavalkoinen  $64 \times 64$  pikselin kokoinen kuva, niin neuroverkon syötekerroksen koko olisi  $64 \times 64 \times 1 = 4096$  neuronia. (Mustavalkoisessa kuvassa harmaan erisävyt ilmoitetaan väriskanava muuttujaa käyttäen, arvot ovat yleensä välillä  $0 \dots 255$ , arvo 0 on musta ja arvo 255 on valkoinen.) Mikäli ensimmäisessä piilokerroksessa olisi esimerkiksi 512 neuronia, niin syöte- ja piilokerroksien neuronien välisiä kytkentöjä olisi noin 2,1 miljoonaa. Värillisellä kuvalla syötekerroksen koko olisi  $64 \times 64 \times 3 = 12\,288$  (kolme väriskanavaa) ja vastaavalla piilokerroksen koolla, syöte- ja piilokerroksen välisiä kytkentöjä olisi noin 6,3 miljoonaa. Mikäli neuroverkossa olisi useampia piilokerroksia, niin kytkentöjen/painoarvojen määrä kasvaa vielä suuremmaksi. Oppimisvaiheessa neuroverkon painoarvojen päivittäminen vaikeutuu ja vaatii paljon aikaa, kun päivitettäviä painoarvoja on miljoonia. [50, 55] Tässä työssä käytettyjen hyppyvideoiden kuvaruudun koko oli  $1920 \times 1080$  pikseliä, mikäli kuvaruutu muutetaan mustavalkoiseksi, niin neuroverkon syötekerroksen koko olisi  $(1920 \times 1080 \times 1)$  hieman alle 2,1 miljoonaa neuronina.

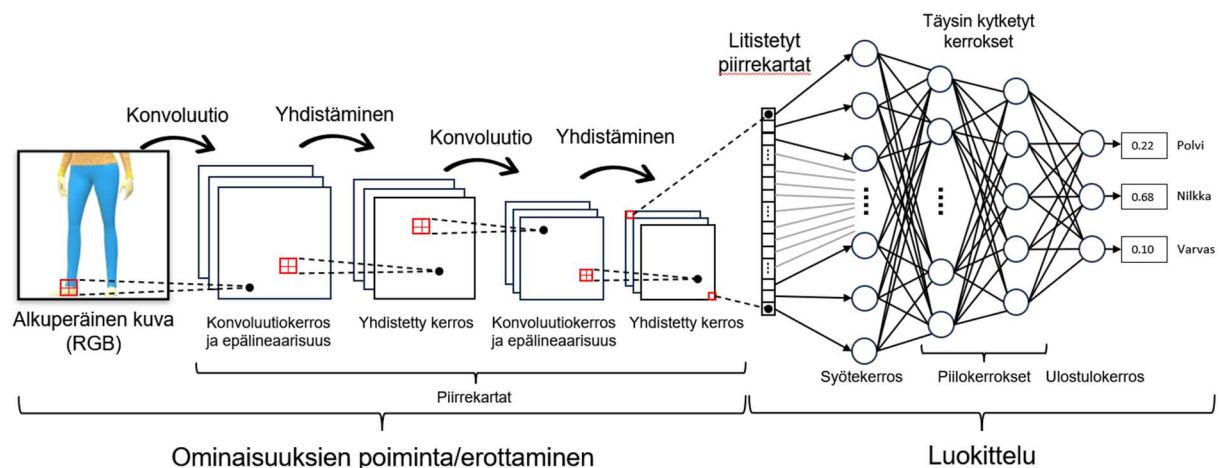
Toinen ongelma neuroverkon käyttämisessä kuvien kanssa on tapa, jolla kuvan pikselien arvot syötetään neuroverkon sisääntulokerrokseen. Kuvan yksittäisten pikselien arvot on syötettävä neuroverkkoon (vektorina) peräkkäisinä arvoina, esimerkiksi lukemalla kuvasta pikselien arvot ylhäältä alaspäin rivi kerrallaan. Tämä johtaa kuitenkin siihen, että alkuperäinen pikselien

välinen yhteys menetetään. Neuroverkolla ei ole tietoja toisiinsa liittyvistä pikseleistä, jotka esimerkiksi muodostavat alkuperäisessä kuvassa jonkin kuvion. [50] Matemaattisesti tätä voisi ajatella, matriisin (kaksiulotteinen) litistämistä (yksiulotteiseksi) vektoriksi, matriisin arvot luetaan rivi kerrallaan vektorin peräkkäisiksi arvoiksi.

Seuraavassa kappaleessa käsiteltävässä konvoluutioneuroverkossa ei tarvitse suorittaa kuvan litistämistä. Konvoluutioneuroverkolle alkuperäinen kuva syötetään ilman muokkaamista ja verkko pystyy säilyttämään tiedon toisiinsa liittyvistä pikseleistä.

### 3.6 Konvoluutioneuroverkko

Konvoluutioneuroverkko (Convolutional Neural Networks, CNN) on syvä neuroverkko rakenne, joka suunniteltiin alun perin kuvantunnistustehtäviin [45] Konvoluutioneuroverkko koostuu kuvan ominaisuuksien poiminnasta/erottamisesta ja luokitteluun liittyvistä osioista (kuva 11). Kuvasta ominaisuuksia poimivan/erottelevan osion muodostavat peräkkäiset kuvan yli suoritettavat konvoluutio-, epälineaarisuus- ja yhdistämisvaiheet, joita toistetaan yleensä useamman kerran peräkkäin, esimerkiksi kuvassa 11 peräkkäin vaiheet: konvoluutio-epälineaarisuus-yhdistäminen-konvoluutio-epälineaarisuus-yhdistäminen. Konvoluutio- ja yhdistämisvaiheiden matriisit tallennetaan vastaavasti nimetyille kerroksille, näitä matriiseja kutsutaan piirrekartoiksi/ominaisuuskartoiksi (feature map). Epälineaarisuuden lisäävänä funktiona käytetään yleensä saranafunktiota (Rectified Linear Unit, ReLU), jota käytetään/sovelletaan konvoluutiokerroksen piirrekarttaan. [50]



Kuva 11. Yksinkertaisen konvoluutioneuroverkon rakenne (konvoluutiovaiheissa käytössä yksi suodin). Kuva: Kalle Lehtonen

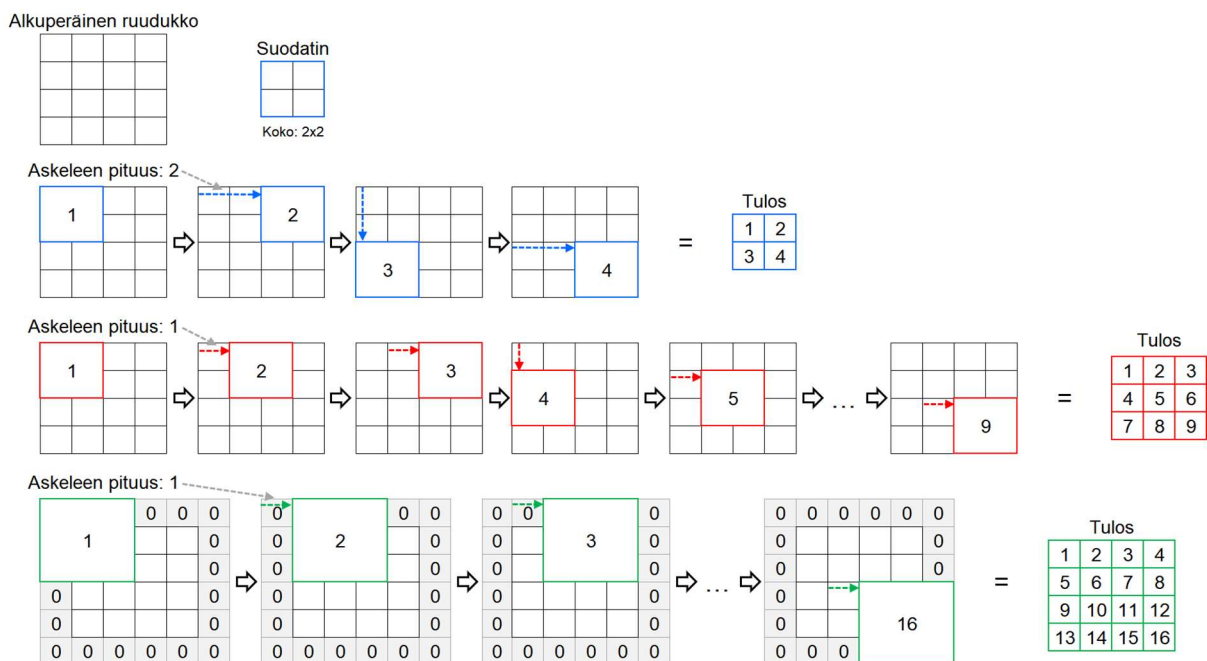
Ominaisuuksia poimivan/erottelevan (kuva 11) osion ensimmäiset kerrokset eristävät kuvasta paikallisia (pienestä osasta kuvan pikseleistä) matalan tason piirteitä, kuten tietyssä kulmassa

olevia suoria ja kaarevia muotoja. Syvemmissä kerroksissa aiempien tasojen paikalliset piirteet yhdistetään korkean tason piirteiksi, esimerkiksi kaarevat muodot voivat yhdessä muodostaa soikion tai ympyrän. Vielä syvemmissä kerroksissa aiempien tasojen muodot voivat muodostaa esimerkiksi ihmisen kehon. [45]

Konvoluutioneuroverkon luokitteluosion muodostaa neuroverkko, jonka kerrokset ovat täysin kytkettyjä. Ominaisuuksien poiminnasta/erottamisesta osion viimeisen yhdistetyn kerroksen matriisien arvot litistetään vektoriksi, jonka arvot välitetään neuroverkon syötekerrokseen. [50] Seuraavissa kappaleissa käydään läpi konvoluutioneuroverkkoon liittyviä laskentoja ja matriisien arvojen litistäminen vektoriksi.

### 3.6.1 Suodattimen liu'uttaminen ja nollien lisääminen

Konvoluutioneuroverkon ominaisuuksien poiminta-/erottamisvaiheessa suoritettavissa laskennoissa (kuvan yli tapahtuva konvoluutio ja yhdistäminen) alkuperäisen kuvan pikseleiden tai piirrekarttojen yli liu'utetaan suodatin (kernel). Kuvassa 12 esitetään ruudukon avulla visuaalisesti, miten liu'uttaminen tapahtuu ja miten siirtymän/askeleen pituus, sekä nollien lisääminen kuvan pikseleiden ympärille vaikuttavat tuloksena muodostettavan ruudukon kokoon. Liu'uttamisen aikana suodattimen ruutujen alle jäävien ruutujen arvoilla tehdään laskutoimenpiteitä tai valitaan arvoista esimerkiksi maksiarvo, tulos tai valittu arvo tallennetaan ruudukkoon (kuvassa 12 oikean reunan värilliset ruudukot).



Kuva 12. Suodattimen liu'uttaminen ruudukon (kuvan) yli eri askel pituuksilla (alimmissa: ruudukon ympärille on lisätty nolliä). Kuva: Kalle Lehtonen

Suodattimen liu'uttaminen aloitetaan kuvan (kuvassa 12 ruudukon) yläkulmasta, siirtymän/askeleen pituus (stride) määrää suodattimen etenemisen/liikkumisen edelliseen vaiheeseen verrattuna. Kuvassa 12 ruudukon läpikäynti aloitetaan ruudukon vasemmasta yläkulmasta ja edetään ruudukon oikeaan reunaan. Tämän jälkeen tutkitaan ruudukon vasemman yläkulman alla olevat ruudut ja edetään ruudukon oikeaan reunaan. Tätä toistetaan, niin monta kertaa, kunnes ruudukon kaikki ruudut on läpikäyty. Tuloksena syntyy ruudukko, jonka arvot edustavat alkuperäisen ruudukon paikallisia arvoja.

Kuvan 12 ylimmässä esimerkissä  $4 \times 4$  kokoisen ruudukon yli liu'utetaan  $2 \times 2$  kokoinen suodatin, askeleen pituudella kaksi, tällöin tallennettavan ruudukon koko on  $2 \times 2$  ruutua. Kuvan 12 keskimmaisessä esimerkissä ruudukon yli liu'utetaan  $2 \times 2$  kokoinen suodatin, askeleen pituudella yksi, tällöin tallennettavan ruudukon koko on  $3 \times 3$  ruutua. Suodattimen koko ja siirtymän pituus vaikuttavat lopputuloksena syntyvän ruudukon kokoon. Molemmissa edellisissä esimerkeissä tuloksena syntyneen ruudukon koko on alkuperäistä ruudukkoa pienempi. Alkuperäisen ruudukon koko saadaan säilytettyä, lisäämällä ruudukon ympärille nollia (padding, zero-padding). Kuvan 12 alimmassa esimerkissä ruudukon ympärille on lisätty nollia, tämän jälkeen ruudukon yli liu'utetaan  $3 \times 3$  kokoinen suodatin, askeleen pituudella yksi, tällöin tallennettavan ruudukon koko on  $4 \times 4$  ruutua.

### 3.6.2 Konvoluutiovaihe

Konvoluutiovaiheessa (convolution phase) kuvan yli tapahtuvassa konvoluutiossa suoritettava laskenta kohdistuu kerrallaan alkuperäisen kuvan pieneen paikalliseen osaan, tämä toteutetaan liu'uttamalla kuvan yli konvoluutiosuodatin (convolution kernel). [45] Konvoluutiosuodattimia on olemassa monenlaisia, joilla jokaisella on eri tarkoitus esimerkiksi konvoluutioneuroverkon ominaisuuksien poimimisen/erottamisen ensimmäisessä vaiheessa etsiä kuvasta tietyssä kulmassa olevia suoria tai kaarevia muotoja.

Alkuperäinen ruudukko (2d)

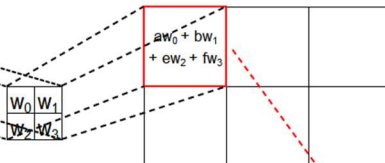
a	b	c	d
e	f	g	h
i	j	k	l

Konvoluutiosuodatin (2d)

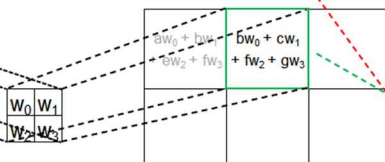
$w_0$	$w_1$
$w_2$	$w_3$

Askeleen pituus: 1

a	b	c	d
e	f	g	h
i	j	k	l

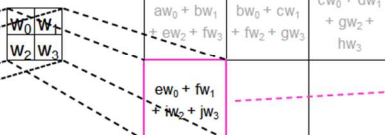


a	b	c	d
e	f	g	h
i	j	k	l



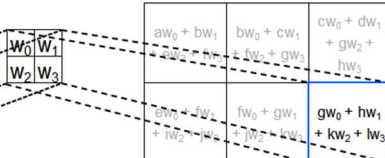
⋮

a	b	c	d
e	f	g	h
i	j	k	l



⋮

a	b	c	d
e	f	g	h
i	j	k	l

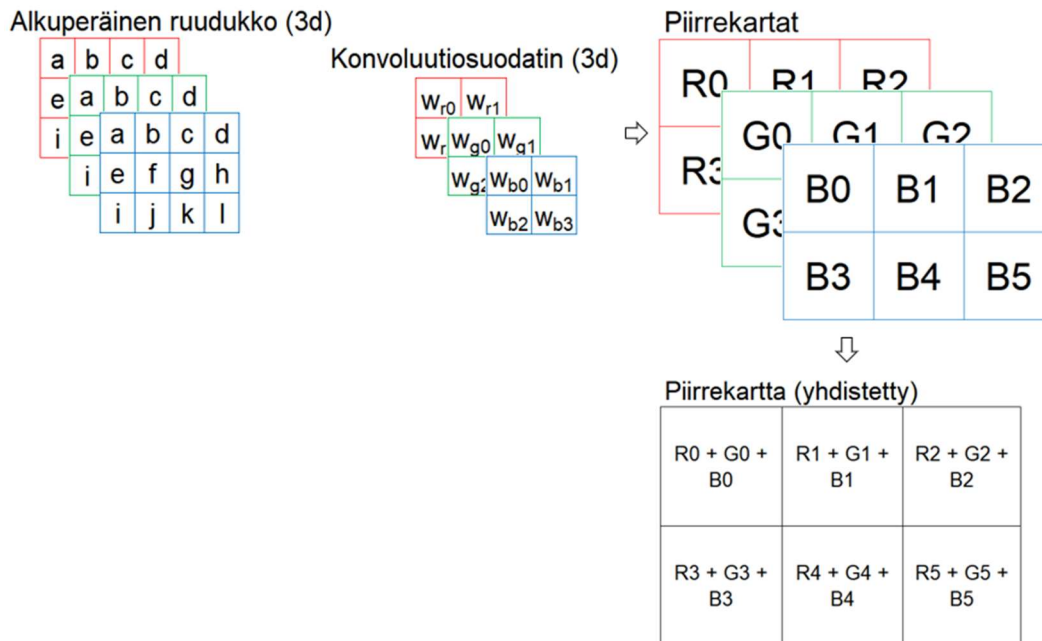


Piiirrekartta

$aw_0 + bw_1$ $+ ew_2 + fw_3$	$bw_0 + cw_1$ $+ fw_2 + gw_3$	$cw_0 + dw_1$ $+ gw_2 + hw_3$
$ew_0 + fw_1$ $+ iw_2 + jw_3$	$fw_0 + gw_1$ $+ jw_2 + kw_3$	$gw_0 + hw_1$ $+ kw_2 + lw_3$

Kuva 13. Konvoluutiokerroksen piiirrekartan arvojen laskeminen ja muodostaminen ruudukon/kuvan yli tapahtuvalla konvoluutiolla. Kuva: Kalle Lehtonen

Kuvan yli tapahtuvan konvoluution arvot lasketaan painotettua summaa käyttäen, yksittäisen pikselin arvo kerrotaan konvoluutiosuodattimen samassa kohdassa olevalla arvolla ja lopuksi lasketut arvot lasketaan yhteen. Kuvassa 13 esitetään ruudukkojen avulla visuaalisesti, miten ruudukon yli tapahtuvan konvoluution laskenta tapahtuu. Kuvan yli tapahtuvan konvoluution tuloksena syntyvää ruudukkoa kutsutaan konvoluutiokerroksen piiirrekartaksi. Epälineaarisuuden lisäämiseksi konvoluutiokerroksen piiirrekarttaan käytetään/sovelletaan usein saranafunktiota (Rectified Linear Unit, ReLU), jonka jälkeen muodostunut piiirrekartalle suoritetaan uudestaan konvoluutiiovaihe tai siirretään yhdistämisvaiheen käsittelyyn [50].



Kuva 14. Konvoluutiokerroksen piirrekartan arvojen laskeminen ja muodostaminen kolmiulotteisen ruudukon/kuvan yli tapahtuvalla konvoluutiolla. Kuva: Kalle Lehtonen

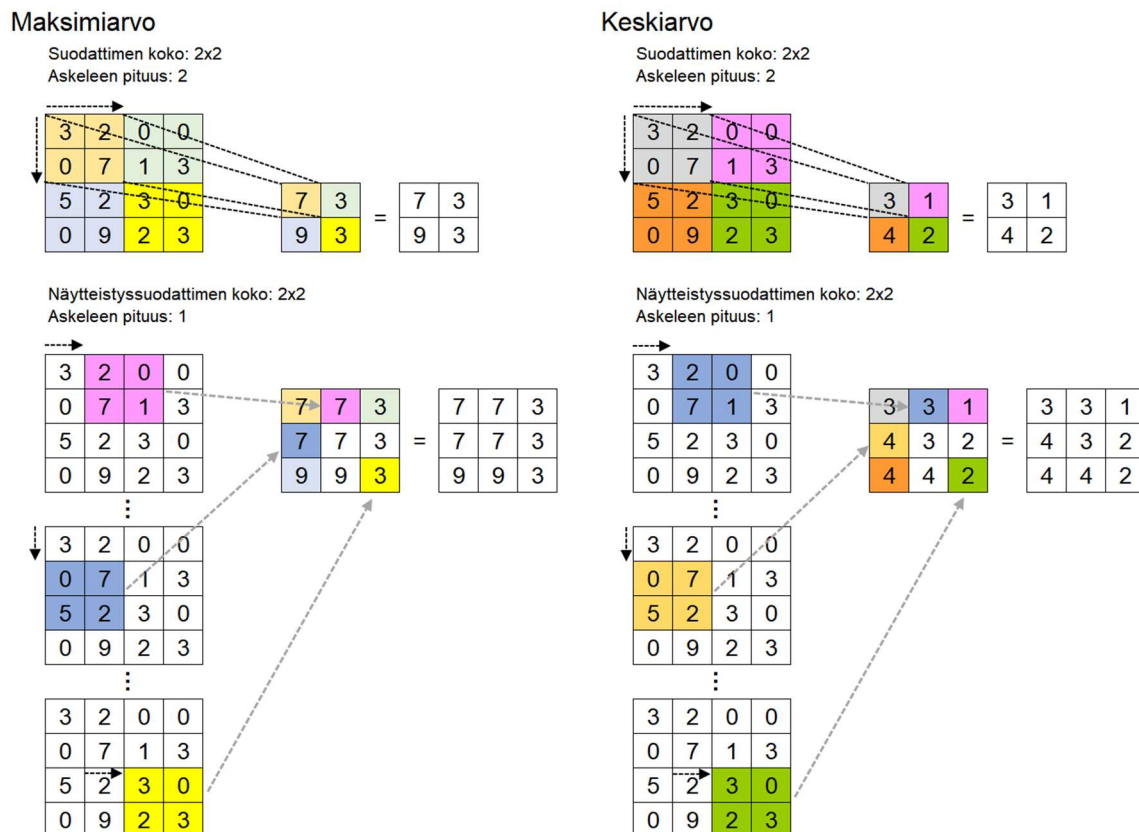
Konvoluutiossa käytettävän konvoluutiosuodattimen syvyys on sama kuin alkuperäisen kuvan syvyys. Mustavalkoisen kuvan syvyys on yksi (kuva 13), niin kuvan ylitapahtuvassa konvoluutiossa käytettävän konvoluutiosuodattimen syvyys on myös yksi. Värikuvassa kerroksia on kolme (punainen, vihreä ja sininen), joten konvoluutiosuodattimessa on sama määrä kerroksia. [56] Kuvassa 14 esitetään kolmiulotteisten ruudukkojen avulla visuaalisesti, miten ruudukon yli tapahtuvan konvoluution laskenta tapahtuu. Jokaiselle kerrokselle (punainen, vihreä ja sininen) suoritetaan ruudukon yli tapahtuva konvoluutio vastaavan konvoluutiosuodattimen kerroksen kanssa, tuloksena jokaiselle kerrokselle syntyy oma piirrekartta. Jokaisen piirrekartan laskenta suoritetaan Kuvassa 13 esitetyllä tavalla. Piirrekarttojen laskennan suorittamisen jälkeen muodostetaan yhdistetty piirrekartta laskemalla piirrekarttojen kerroksien vastaavat ruudut yhteen.

Konvoluutioneuroverkon ominaisuuksia poimivan/erottelevan osion rakennetta voidaan pitää samanlaisena, kuin työssä aiemmin läpi käydyn neuroverkon rakennetta. Konvoluutioneuroverkon yksittäiset konvoluutiokerrokset vastaavat neuroverkon piilokerroksia. Konvoluutioneuroverkon konvoluutiovaiheessa on oltava vähintään yksi konvoluutiosuodin, yleensä vaiheessa käytetään kuitenkin kymmeniä erilaisia konvoluutiosuodattimia. Konvoluutiovaiheessa käytettävän konvoluutiosuodattimen voidaan ajatella vastaavan neuroverkon piilokerroksen neuronina. Mikäli konvoluutiovaiheessa on käytössä esimerkiksi 2x2 kokoinen konvoluutiosuodin, niin kerroksessa on neljä piilotettua yksikköä tässä vaiheessa. Mikäli lisätään toinen samankokoinen suodin, niin piilotettujen

yksikköjen määrä on neljä (ja niin edelleen). Suodattimien määrän lisääminen lisää piilotettujen yksiköiden määrää, tämä monimutkaistaa konvoluutioneuroverkon rakennetta ja mahdollistaa monimutkaisempien kuvioiden havaitsemisen. Vastaavasti neuronien lisääminen neuroverkkoon, parantaa neuroverkon kykyä mallintaa monimutkaisia suhteita. [50].

### 3.6.3 Yhdistämisvaihe

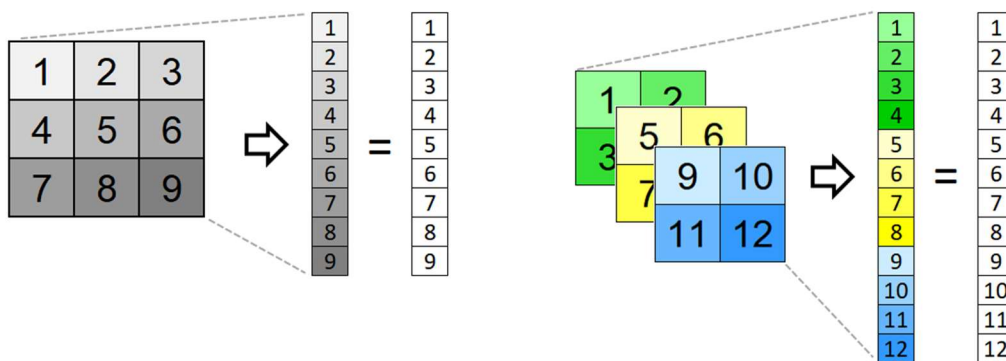
Yleensä konvoluutioneuroverkoissa on kymmeniä/satoja konvoluutiokerroksia. Mikäli käytettäisiin pelkästään konvoluutiokerroksia, niin parametrien määrä olisi valtava. Yhdistämisvaiheen (pooling phase) tarkoituksena on vähentää parametrien määrää, jolloin laskentaan käytettävä aika ja laskentatehon tarve vähenevät. Konvoluutioneuroverkoissa on yleinen käytäntö, että yhden tai kahden konvoluutiokerroksen jälkeen lisätään yhdistämisvaihe/-kerros. Yleensä konvoluutioneuroverkoissa käytetään joko maksimiarvoon tai keskiarvoon perustuvaa tietojen yhdistämistä. [50]. Yhdistämisvaiheen syötteenä on konvoluutiokerroksen piirrekartta, jonka pikselien arvoihin on sovellettu epälineaarisuus. Kuvassa 15 esitetään ruudukkojen avulla visuaalisesti, miten ruudukon yli tapahtuvan maksimiarvoon tai keskiarvoon perustuva yhdistämisvaihe tapahtuu. Ruudukon yli liu'utettavaa ruudukkoa kutsutaan yhdistämissootimeksi (pooling kernel).



Kuva 15. Maksimiarvoon tai keskiarvoon perustuvaa tietojen yhdistäminen. Kuva: Kalle Lehtonen

### 3.6.4 Piirrekarttojen muuttaminen vektoriksi

Konvoluutioneuroverkon ominaisuuksia poimivan/erottelevan osion viimeisen yhdistämisvaiheen tuottamia piirrekarttoja ei voi syöttää sellaisenaan luokitteluvaiheessa olevalle neuroverkolle. Konvoluutioneuroverkon luokitteluvaiheen alussa (kaksiulotteiset matriisit) piirrekartat muutetaan tai litistetään yksiulotteiseksi vektoriksi. Kuvassa 16 esitetään ruudukkojen avulla visuaalisesti, yhden ja kolmen kaksiulotteisen ruudukon muuntaminen vektoriksi.



Kuva 16. Kaksi- ja kolmiulotteisen ruudukon muuttaminen vektoriksi. Kuva: Kalle Lehtonen

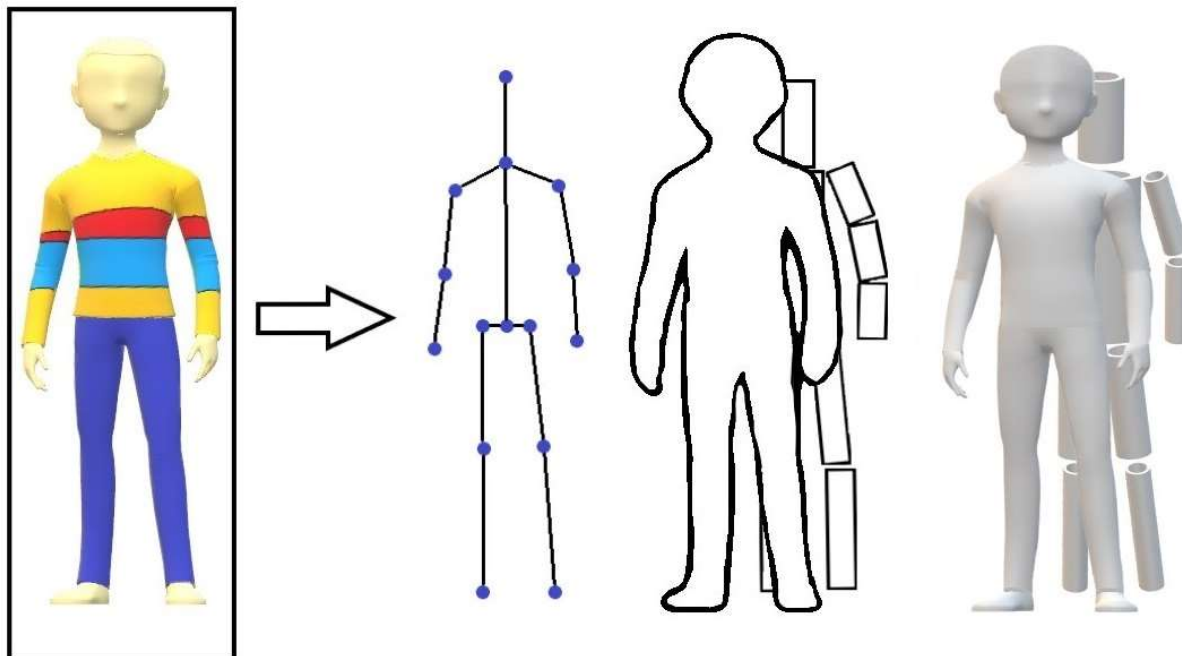
## 3.7 Ihmisen asennon arviointi

Ihmisen asennon arviointi (Human pose estimation, HPE) on tietokonenäköön (computer vision) liittyvä tehtävä, jossa pyritään valokuvasta tai videolta paikantamaan ihmiskehon avainpisteitä: niveliä (esimerkiksi vasemman jalan polvi, oikean jalan kantapäätä) ja kehon osia (esimerkiksi oikean jalan kantapäätä tai vasemman jalan isovarvas). Paikantamisen lisäksi asennon arvioinnissa pyritään yhdistämään kehon osat ja nivelet toisiinsa esimerkiksi tikkuukko- tai kolmiulotteiseksi kehon malliksi. Asennon arvioinnin aikana luotava malli esitetään kaksi- tai kolmiulotteisesti (2D tai 3D). Tässä työssä pituushyppääjän asennon arviointi toteutettiin kaksiulotteisesti.

Ihmisen asennon arviointia käytetään erilaisissa sovelluksissa ihmiskehon toimintojen tunnistamiseen, ennustamiseen, havaitsemiseen ja seurantaan. Asennon arviointia käytetään muun muassa videovalvonnassa, ihmisen ja tietokoneen vuorovaikutukseen liittyvissä sovelluksissa (esimerkiksi palvelurobotti), viihdekäyttöön liittyvillä aloilla (esimerkiksi tietokonepeleissä, animaatioissa ja elokuvissa), lääketieteessä (esimerkiksi potilaiden kuntoutuksessa), lisätyn todellisuuden ja virtuaalitodellisuuteen liittyvissä ratkaisuisissa

(esimerkiksi näyttelijä voidaan korvata elokuvassa animaatiohahmolla) sekä urheilussa (esimerkiksi urheilijan asennon ja toiminnan seuraaminen sekä analysointi). [57, 58]

### 3.7.1 Ihmiskehon mallit



Kuva 17. Ihmiskehon mallintaminen: kinemaattinen malli, tasomallit ja tilavuusmallit. Kuva: Kalle Lehtonen

Asennon arvioinnin tuloksena ihmiskehosta luodaan malli, usein käytettyjä malleja on kolme (kuva 17): kinemaattinen malli, tasomalli ja tilavuusmalli. Kinemaattisessa (luuranko/tikkukko) mallissa yhdistetään avainpisteet toisiinsa viivoilla, mallia käytetään kaksi- ja kolmiulotteisissa asennon arvioinneissa. Avainpisteiden määrä kinemaattisessa mallissa vaihtelee asennon arviointimenetelmän mukaan noin 10...30 kappaleen välillä. Tasomallissa ihmiskeho mallinnetaan esimerkiksi kaksiulotteisena kuvana tai käyttämällä suorakulmaisia neliöitä, tasomallia käytetään kaksiulotteisessa asennon arvioinnissa. Tilavuusmallissa ihmiskehosta pyritään muodostamaan realistinen kolmiulotteinen tilavuusmalli, joka voi olla kolmiulotteinen verkkomalli tai kehon osien muodostamiseen voidaan käyttää esimerkiksi sylintereitä ja kartioita. [57, 59, 60] Tässä työssä tehdyssä sovelluksessa käytettiin ihmiskehon asennon arviointimenetelmää, joka perustuu kinemaattiseen malliin.

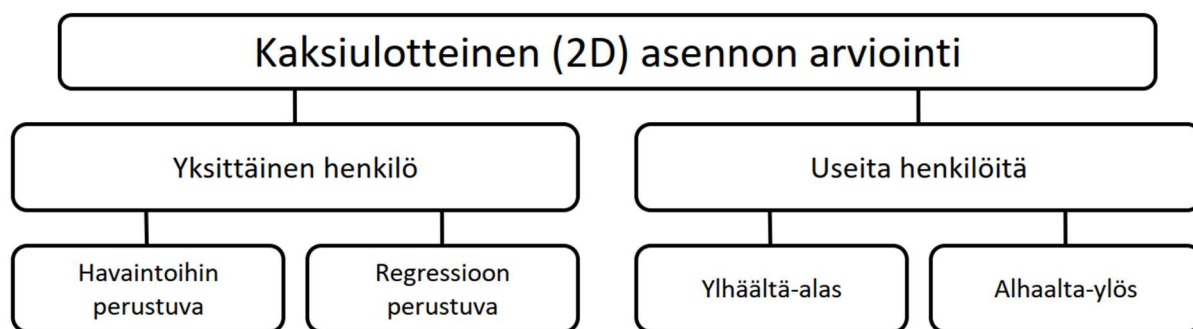
### 3.7.2 Ihmisen asennon arvioinnin haasteet

Ihmiskehon avainpisteiden tunnistamisessa on monia haasteita, kuten okklusio, lähekkäin tai toisiinsa kontaktissa olevat henkilöt, kehon monimutkainen tai epätavallinen asento, vaatetus, vaihtelevat kuvauskulmat, henkilö on vain osittain näkyvässä kuvassa tai videolla ja kuvan tai

videon alhainen resoluutio. Okklusio tarkoittaa sitä, että kehon kaikki avainpisteet eivät ole näkyvillä kuvassa tai videolla. Kehon avainpiste voi jäädä esimerkiksi henkilön oman raajan, toisen henkilön tai muun esineen taakse piiloon. Okklusion lisäksi lähekkäin tai kontaktissa olevien usean henkilön asennon arvioinnissa avainpisteiden yhdistäminen oikeaan henkilöön vaikeutuu. Kehon monimutkainen tai epätavanomainen asento (esimerkiksi joogassa) ja vaihtelevat kuvauskulmat vaikeuttavat kehon avainpisteiden tunnistamista. Avainpisteiden tunnistamiseen vaikuttaa myös henkilön vaatetus, joka voi urheilijalla olla esimerkiksi t-paita ja verryttelyhousut tai sitten ihonmyötäinen urheiluasu. Mikäli on kuvassa tai videolla ei ole näkyvissä esimerkiksi jalkoja, niin asennon arviointi vaikeutuu. [57, 60] Mikäli kuva tai video on kuvattu kaukaa kohteena olevasta henkilöstä, niin kaikkien avainpisteiden tunnistaminen ja niiden yhdistäminen toisiinsa ei välttämättä onnistu. Lisäksi kuvassa tai videolla oleva tausta voi puolestaan aiheuttaa ongelmia, mikäli asennon arviointimenetelmä löytää sieltä kehoon liittyviä avainpisteitä [16].

Moni yllä olevista haasteita johtuu siitä, että käytettävissä ei ole opetusaineistoa, jotka sisältäisivät edellä mainittuja tapauksia. Esimerkiksi opetuksessa käytettävät aineistot on yleensä kuvattu korkealla resoluutiolla, mikäli asennon arvioinnissa käytetty kuvassa tai videossa on matala resoluutio, niin asennon arviointi välttämättä onnistu. Opetusaineistot sisältävät paljon aineistoa niin sanottua normaalia asennon arviointia varten (esimerkiksi kävely ja juoksu), mutta niissä ei ole mukana aineistoa, jossa keho olisi epätavallisessa asennossa. [57]

Erikoistyyssä [1] kuvattiin hyppyvideoita kahden pituushyppääjän pituushyppysuorituksista. Hyppääjillä oli päällään erilainen vaatetus (kuva 21). Hyppyvideoihin liittyviä asioita käsitellään neljännen luvun ensimmäisessä kappaleessa.



Kuva 18. Kaksiulotteisen (2D) asennon osa-alueet. Kuva: Kalle Lehtonen

Riippumatta siitä arvioidaanko ihmisen asentoa kaksi- tai kolmiulotteisesti, niin asennon arviointimenetelmät voidaan luokitella kahteen osa-alueeseen (kuva 18): yksittäisen henkilön

ja usean henkilön asennon arvioinniksi. Tässä työssä pituushyppysuorituksien analysointiin käytettiin kaksiulotteisia menetelmiä. Erikoistyössä [1] kuvattujen pituushyppyvideoiden kuvaruuduissa on vain hyppysuorituksen tekevä pituushyppääjä, tämän vuoksi asennon arvioinnissa käytettävän menetelmän tarvitsee tunnistaa yksittäinen henkilö.

Ennen syväoppimisen ja neuroverkkojen yleistymistä ihmisen asennon arvioinnissa käytettiin muun muassa erilaisia käsintehtyjen (käyttäjän muodostamia) ominaisuuksien etsintää käyttäviä menetelmiä ja kuvarakenteiksi (pictorial structures) kutsuttua menetelmää. Käsintehtyjen ominaisuuksien etsinnässä kuvasta voidaan muun muassa etsiä seuraavia ominaisuuksia: reunoja, ääriviivoja, suuntautuneita gradienttihistogrammeja ja värihistogrammeja. Kuvarakenteita käytävissä menetelmissä kehon osien sijaintiin liittyvät korrelaatiot esitetään puurakenteisena graafisena mallina. Kuvarakenteita käyttävät menetelmät toimivat hyvin, kun kaikki kehonosat olivat näkyvissä. Molemmat yllä mainituista menetelmistä perustuvat käsintehtyihin ominaisuuksiin, jotka rajoittavat mallin yleistämistä ennen näkemättömille syötteille ja avainpisteiden sijainnin tarkkuutta. [61, 62]

Syväoppimisessa käytetyissä neuroverkoissa käsintehtyjen ominaisuuksien etsimiselle/muodostamiselle ei ole tarvetta, vaan neuroverkko muodostaa ominaisuudet automaattisesti. [50]. Konvoluutioneuroverkoja (convolutional neural networks) käyttämällä on saatu parempia tuloksia tietokonenäköön liittyvissä tehtävissä, näin on käynyt myös ihmisen asennon arviointiin liittyvissä tehtävissä [63].

Yksittäisen ihmisen asennon arviointiin käytettävät menetelmät jaetaan tyypillisesti kahteen osa-alueeseen (kuva 18): havaintoihin ja regressioon perustuviin menetelmiin. Havaintoihin perustuvissa menetelmissä käytetään yleensä lämpökarttoja (esimerkiksi Gaussin lämpökartta), jonka yksittäisen pikselin arvo kuvaa todennäköisyyttä sille, että kehon avainpiste löytyy kyseisestä pikselistä. Havaintoihin perustuvat menetelmät ovat kaksivaiheisia, ensimmäisessä vaiheessa luodaan lämpökartat ja toisessa vaiheessa avainpisteiden sijainnit yhdistetään toisiinsa. Ensimmäisessä vaiheessa kehon jokaiselle avainpisteelle muodostetaan erillinen lämpökartta. Toisessa vaiheessa lämpökartoista etsitään avainpisteiden sijainnit ja avainpisteet (kehonosat) yhdistetään toisiinsa. Lämpökarttamenetelmillä on saavutettu erinomaisia tuloksia kaksiulotteisessa asennon arvioinnissa. Menetelmän hyvinä puolina voidaan pitää sitä, että lämpökarttoja voidaan visualisoida käyttäjälle ja menetelmää voidaan soveltaa monimutkaisiin tapauksiin. Useiden lämpökarttojen käyttäminen samanaikaisesti vaatii kuitenkin laskentatehoa

ja tallennustilaa. Lisäksi menetelmä on vaikeaa/työlästä laajentaa soveltumaan kolmiulotteiseen asennon arviointiin. [57, 61, 64, 65]

Regressioon perustuvissa menetelmissä ihmisen asennon arviointi toteutetaan yhdessä vaiheessa, eli syötekuvasta saadaan kartoitettua kehon avainpisteet ilman välivaiheita. Regressioon perustuvien menetelmien tehokkuus perustuu siihen, että ne pystyvät oppimaan epälineaaristen piirteiden esityksiä. Regressiota käyttävät menetelmät ovat nopeita ja niitä voidaan käyttää esimerkiksi reaaliaikaisessa asennon arvioinnissa, menetelmää voidaan soveltaa pieniä muutoksia tekemällä kolmiulotteiseen asennon arviointiin. Regressiota käyttävät menetelmät eivät sovellu usean henkilön samanaikaisesti tapahtuvan asennon arviointiin. [61, 63] Regressiota käyttävillä menetelmillä ei päästä kehon avainpisteiden arvioinnissa samaan tarkkuuteen kuin havaintoihin perustuvilla menetelmillä. [66]

Usean ihmisen samanaikaisesti suoritettavan asennon arviointimenetelmät jaetaan kahteen osaluokkaan (kuva 18): ylhäältä-alas (top-down) ja alhaalta-ylös (bottom-up) menetelmiksi. Ylhäältä-alas-menetelmässä pyritään ensin kuvasta tunnistamaan/rajaamaan alue, jossa ihminen sijaitsee ja tämän jälkeen alueen sisältä pyritään paikantamaan kehon avainpisteet. Alhaalta-ylös menetelmässä puolestaan pyritään ensin paikantamaan kehon avainpisteet, jonka jälkeen muodostetaan arvio ihmisen sijainnista kuvassa. Mikäli kuvassa olevien ihmisten määrä kasvaa, niin ylhäältä-alas menetelmää käytettäessä laskentatehon tarve nousee merkittävästi, kun taas ylhäältä-alas -menetelmää käytettäessä laskentatehon tarve pysyy puolestaan suhteellisen samana. Mikäli kuvassa olevat ihmiset ovat sijoittuneet kuvaan siten, että osa ihmisistä jää taka-alalle, niin alhaalta-ylös menetelmillä voi olla haasteellista ryhmitellä kuvasta löytyneet kehonosat ja nivelet samaan/oikeaan kehoon kuuluvaksi. [60]

### 3.7.3 Asennon arviointimenetelmät

Syväoppimista käyttäviä ihmisen asennon arviointimenetelmiä on monia erilaisia, tässä luvussa käydään läpi eri menetelmien ominaisuuksia. Taulukkoon 1 on koottu joidenkin asennon arviointimenetelmien ominaisuuksia. DeepPose oli ensimmäisiä syvää oppimista ja konvoluutioneuroverkkoja soveltava asennonarviointimalli, jonka erinomaisen suorituskyvyn ansiosta asennon arvioinnissa alettiin tutkimaan ja käyttämään konvoluutioneuroverkkoja enemmän [57]. DeepCut tunnistaa ihmiskehon ylävartalon nivelien avainpisteet ranteeseen asti ja alavartalossa vastaavasti nilkkaan asti, kasvoista/päästä tunnistetaan yksi avainpiste [67]. OpenPose on yksi suosituimmista asennon arviointimenetelmistä, jonka avulla pystyy tunnistamaan ihmiskehosta 135 avainpistettä, määrässä on mukana myös käsien (kuten

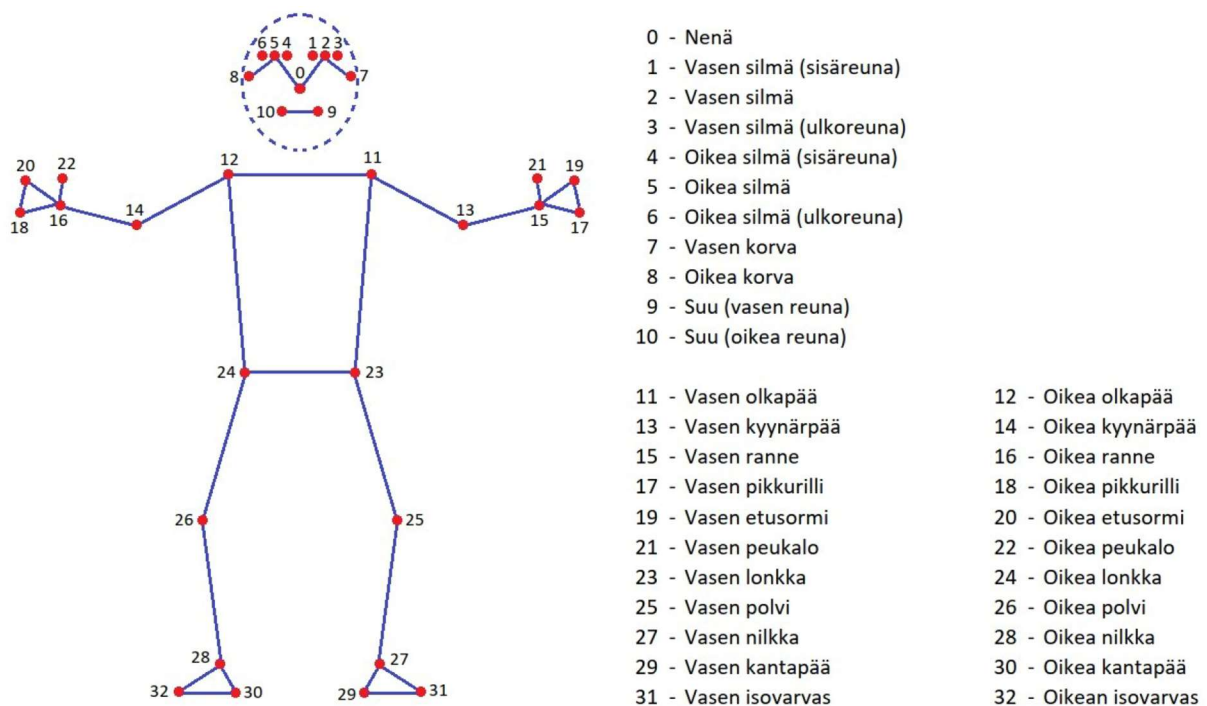
sormien), kasvojen ja jalkaterien avainpisteet. [68, 69] OpenPose tunnistaa avainpisteiden sijainnit tarkasti kuvista, joissa on korkea resoluutio. Menetelmän suorituskyky ei kuitenkaan ole yhtä hyvä, kun kuvissa on matala resoluutio tai kun kuvissa esiintyy okklusiota [57]. PoseNet on Google Creative Lab:n kehittäämä asennon arviointimenetelmä, jota voidaan käyttää esimerkiksi kannettavan tietokoneen kameralla tai älypuhelimien verkkoselaimella [70]. High-Resolution Net (HRNet) tunnistaa ihmisen kasvoista viisi avainpistettä, ylävartalon avainpisteiden tunnistaminen kattaa nivelet ranteeseen asti ja alavartalon vastaavasti nilkkaan asti [71]. MoveNetin kehittämisessä on ollut mukana Google ja siitä on olemassa kaksi versiota: Lightning versiossa on panostettu suorituskykyyn/nopeuteen (voidaan käyttää esimerkiksi älypuhelimissa) ja Thunder versiossa on panostettu puolestaan tarkkuuteen [72]. MoveNet tunnistaa ylävartalon nivelien avainpisteet ranteeseen asti ja alavartalossa vastaavasti nilkkaan asti, kasvoista/päästä tunnistetaan viisi avainpistettä [72].

Taulukko 1. Asennon arviointimenetelmien ominaisuuksia [57, 68, 69, 73, 74]. Taulukko: Kalle Lehtonen

Asennon arviointimenetelmä	Julkaisuvuosi	Avainpisteiden määrä	Avainpisteiden sijainti	Samanaikainen tunnistaminen	Tunnistusmenetelmä	Tulos kaksi-/kolmiulotteinen
DeepPose	2014	Ei tiedossa	Ei tiedossa	Yksittäinen henkilö	Regressio	2D
DeepCut	2016	14	Pää, ylä- ja alavartalo	Yksittäinen ja useita henkilöitä	Alhaalta-ylös	2D
OpenPose	2017	135	Naama, kädet, pää, ylä- ja alavartalo	Yksittäinen ja useita henkilöitä	Alhaalta-ylös	2D useita henk. 3D yksi henk.
PoseNet	2017	17	Pää, ylä- ja alavartalo	Yksittäinen ja useita henkilöitä	Ylhäältä-alas	2D/3D
High-Resolution Net (HRNet)	2019	17	Pää, ylä- ja alavartalo	Yksittäinen ja useita henkilöitä	Ylhäältä-alas	2D
MediaPipe Pose	2020	33	Pää, ylä- ja alavartalo	Yksittäinen henkilö	Ylhäältä-alas	2D/3D
MoveNet	2021	14	Pää, ylä- ja alavartalo	Yksittäinen ja useita henkilöitä	Alhaalta-ylös	2D/3D

MediaPipe on Googlen kehittämä tuoteperhe, jonka kirjastoja ja työkaluja voidaan hyödyntää tekoälyä-/koneoppimista käyttävän sovelluksen tekemisessä. Tuoteperhe sisältää tällä hetkellä ratkaisuja tietokonenäköön, tekstinkäsittelyyn ja audioon liittyvien sovellusten tekoon. Tietokonenäköön liittyviä ratkaisuja löytyy muun muassa: kohteiden havaitsemiseen kuvista/videolta, kuvien luokitteluun/segmentointiin, käsieleiden tunnistaminen (esimerkiksi viittomakielen viittomien tunnistaminen) ja ihmisen kasvojen tunnistaminen kuvasta tai videosta. Ihmiskehon avainpisteiden tunnistamiseen on tarjolla kolme eri ratkaisua: kasvot (Face landmark detection, 478 avainpistettä), käsi/kämmen (Hand landmark detection, 21 avainpistettä/käsi) ja keho (Pose landmark detection, 33 avainpistettä). Näiden kolmen lisäksi Google kehittää ratkaisua (Holistic landmark detection), joka yhdistää edellä mainitut ratkaisut. [75]

Tässä työssä käytetty Googlen kehittämä MediaPipe Pose-asennontunnistusalgoritmi, tunnistaa ihmisen kehosta 33 avainpistettä (kuva 19). Menetelmä tunnistaa ihmisen päästä/kasvoista 11 ja kehosta 22 avainpistettä [76]. Kämmenistä tunnistetaan kolme sormea ja jalkateristä kantapää ja isovarvas. Erikoistyön kalibrointivideoiden kuvaamisen aikana törmäsin YouTubessa MediaPipe Posen käyttöönottoon liittyvään videoon, jota seuraamalla sain varsin helposti asennon arvioinnin toimimaan. Kaikkien asennon arviointimenetelmien käyttäminen pituushyppyjen analysointiin ei olisi mahdollista, koska niissä jalkateristä ei tunnisteta kantapään ja isovarpaan avainpisteitä.



Kuva 19. MediaPipe Pose-asennontunnistusalgoritmin avainpisteet. [77] Kuva: Kalle Lehtonen

MediaPipe Pose-asennontunnistusalgoritmin toimintaa voidaan muuttaa hyperparametrien arvoja muuttamalla. Videota, jolta etsitään kehon avainpisteitä, voidaan käsitellä joko kuvaruutuja joko yksittäisinä kuvina tai toisiinsa liittyvinä kuvina eli videovirtana (työssä on käytetty tätä). Videovirtana käsiteltäessä ensimmäisistä kuvaruuduissa pyritään havaitsemaan kuvaruudussa parhaiten näkyvimmän henkilön avainpisteet. Seuraavissa kuvaruuduissa pyritään seuraamaan edellisissä kuvaruuduissa havaittuja avainpisteitä. Avainpisteiden tunnistamisessa käytetyn mallin monimutkaisuuden voi valita kolmesta mallista, työssä on käytetty monimutkaisinta mallia. Työssä on käytetty ominaisuutta, jolla voidaan vähentää kuvaruutujen välistä värinää. Ihmiskehon vähimmäisluottamusarvolla valitaan taso, jolla ihmiskehon havaitseminen katsotaan onnistuneen (oletusarvo: 0.5, työssä on käytetty arvoa 0.65). Avainpisteiden vähimmäisluottamusarvolla valitaan taso, jolla avainpisteiden

seuraamisen katsotaan onnistuneen (oletusarvo: 0.5, työssä on käytetty arvoa 0.65). Mikäli seuraaminen ei ole onnistunut, niin seuraavassa kuvaruudussa aloitetaan kehon avainpisteiden tunnistaminen uudestaan. [78]

## 4 MATERIAALIT

Tässä työssä käytetyt pituushyppyvideot kuvattiin ja esikäsiteltiin ennen tämän pro gradu - tutkimuksen aloittamista tehdyssä erikoistyössä: “Pituushyppyanalytiikka hyppyvideoiden avulla - videomateriaalin kerääminen ja esikäsittely” [1]. Tämän luvun kappaleissa käydään lyhyesti läpi erikoistyön pääkohtia, pituushyppyvideoiden kuvaamista ja niiden esikäsittelyä.

### 4.1 Hyppyvideot

Kuvaustapahtumaan valmistautumisesta ja tehtävistä toimenpiteistä tehtiin erikoistyössä [1] suunnitelma, jossa käytiin läpi kuvauksissa tarvittava välineistö, missä järjestyksessä valmistelevia asioita tehdään ennen varsinaisia hyppyjen kuvaamista sekä hyppytapahtuman etenemistä suoritusten aikana ja miten mittauspöytäkirjoja täytetään. Hyppysuoritusten kuvaaminen järjestettiin syyskuussa 2022 Turussa Paavo Nurmen stadionin viereisellä harjoituskentällä. Hyppyjen kuvaustapahtumaan osallistui kokenut kansallisen tason ja hyppyranssa alussa oleva (16...17-vuotias) pituushyppääjä.

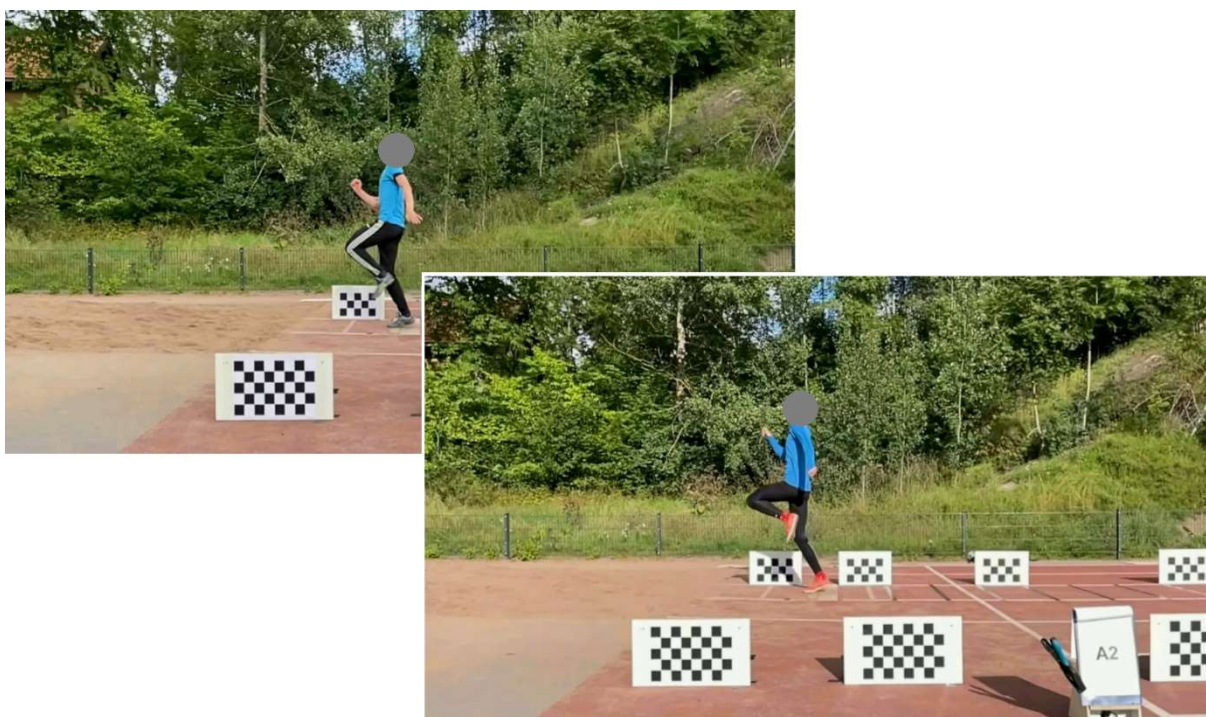


Kuva 20. Puhelimen kuvaruutu näkymä kuvausten aikana. Kuva: Kalle Lehtonen

Pituushyppääjien hyppysuoritukset kuvattiin älypuhelimien kameralla, näkymä puhelimen kuvaruudusta on esitetty kuvassa 20. Älypuhelin oli sijoitettu noin 6,5 m päähän

pituushyppypaikan vauhdinottoradan reunasta. Näin saatiin kuvattua 1.0x suurennoksella mieshyppääjän ponnistus ja sitä edeltävä askel, sekä 0.5x suurennoksella useampi askel.

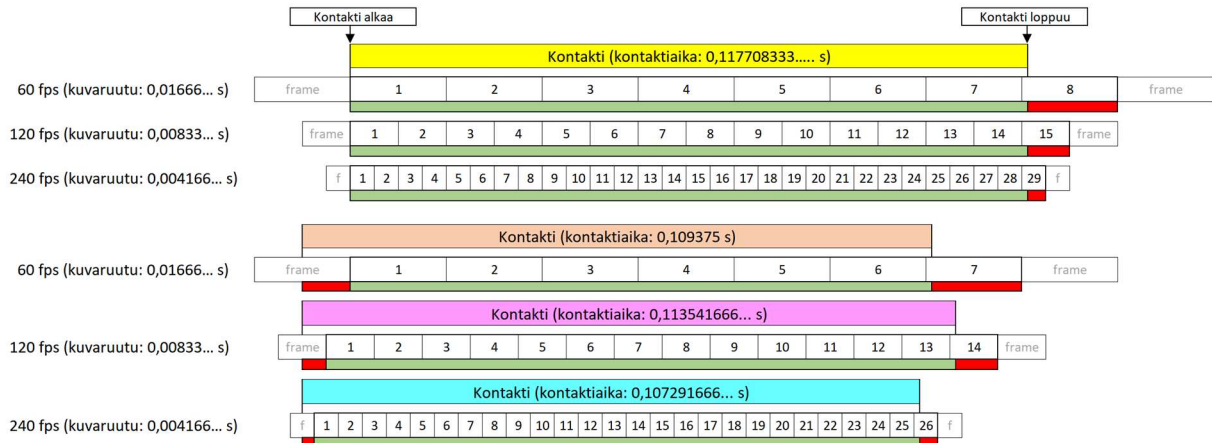
Vauhdinottoradan etu- ja takapuolelle oli aseteltu shakkiruudukoita, joita käytettiin apuna erikoistyössä hyppyvideon tärinän poistossa. Shakkiruudukkoja käytettiin yhden pikselin muuntokertoimen (pikselin pituus vs. todellinen pituus) laskennassa, tätä tietoa puolestaan käytettiin esimerkiksi hypyn askelpituuden ja nopeuksien laskennoissa. Vauhdinottoradan pintaan oli liimattu ilmastointiteippiä, joiden avulla voitiin arvioida askelpituuden määrittämiseen tehdyn sovelluksen tarkkuutta. Havainnekuva välineistön sijoittelusta kentälle löytyy Liitteestä 1.



Kuva 21. Hyppääjien vaatetus Kuva: Kalle Lehtonen

Pituushyppääjien hyppyvideot kuvattiin erikoistyössä niin, että hyppääjä tulee kuvaan kuvaruudun oikeasta reunasta. Videoiden kuvaruudussa on näkyvillä vain hyppysuorituksen tekevä pituushyppääjä. Hyppyvideoita kuvattiin kahdesta hyppääjästä, 16 ... 17-vuotiaalla hyppääjällä oli päällään t-paita ja verryttelyhousut (kuvan 21 ylempi kuva), kokeneella kansallisen tason hyppääjällä oli puolestaan päällään pitkähihainen tekninen paita ja pitkät ihon myötäiset harjoitteluhousut (kuvan 21 alempi kuva). Hyppypaikalle oli aseteltu shakkiruudukkoja vauhdinottoradan taakse, sekä kameran ja vauhdinottoradan väliin, hyppypaikan taustalla oli puita ja matala verkkoaita. Hyppypaikan asetellut shakkiruudukot ja hyppypaikan tausta eivät merkittävästi vaikuttaneet avainpisteiden tunnistamiseen, ja vain yksittäisissä kuvaruuduissa avainpisteiden tunnistaminen oli epäonnistunut.

Hyppysuoritusten kuvaamiseen valittiin älypuhelin, jolla on mahdollista kuvata 240 fps kuvataajuudella. Useamman kuvaruudun tallentaminen parantaa analysointivaiheessa tehtävien laskentojen tarkkuutta, koska tällöin saadaan määritettyä tarkemmin hetki, jolloin esimerkiksi hyppääjän jalka osuu radan pintaan tai irtoaa radan pinnasta.



Kuva 22. Virheiden määrittäminen eri kuvataajuuksilla. Kuva: Kalle Lehtonen

Ponnistusta edeltävän vauhdinottoaskeleen kontakti aika on noin 0,10...0,15 s (100...150 ms), niin kontakti näkyisi 60 fps kuvataajuudella kuvatulla videolla noin 6...9 kuvaruudun ajan. Vastaavasti 120 fps kuvataajuudella kuvatulla videolla kontakti näkyisi 12...18 kuvaruudun ajan ja 240 fps kuvataajuudella 24...36 kuvaruudun ajan. Kuvassa 22 on hahmoteltu eri kontakti aikojen avulla eri kuvataajuuksilla syntyvää virhettä. Kuvassa 22 kuvaruutujen leveydet vastaavat yhden kuvaruudun aikaa, vihreissä kohdissa kontakti on tunnistettu oikein ja punaisissa kohdissa kontaktia ei ole tunnistettu oikein. Kuvan 22 ylemmässä osassa (keltainen) kontakti alkaa samaan aikaan kuvaruudun vaihtumisen kanssa, eli kontaktin oikean alkamiskohdan tunnistaminen onnistuisi täydellisesti. Kontaktin loppuessa jokaisella kuvataajuudella syntyy virhettä (kuvassa punaiset kohdat), jolloin arvioidusta kontakti ajan kesto on liian pitkä todellisuuteen verrattuna. Kontaktin loppuosan aiheuttama virhe voi olla maksimissaan yhden kuvaruudun keston verran, kuvataajuudella 60 fps virhe on <math><16,7\text{ ms}</math>, vastaavasti 120 fps <math><8,3\text{ ms}</math> ja 240 fps <math><4,2\text{ ms}</math>. Kontakti aikoihin (100...150 ms) suhteutettuna, 60 fps kuvataajuudella virhe on 16,7...11,1 %, 120 fps kuvataajuudella virhe on 8,3...5,5 % ja 240 fps kuvataajuudella virhe on 4,2...2,8 %.

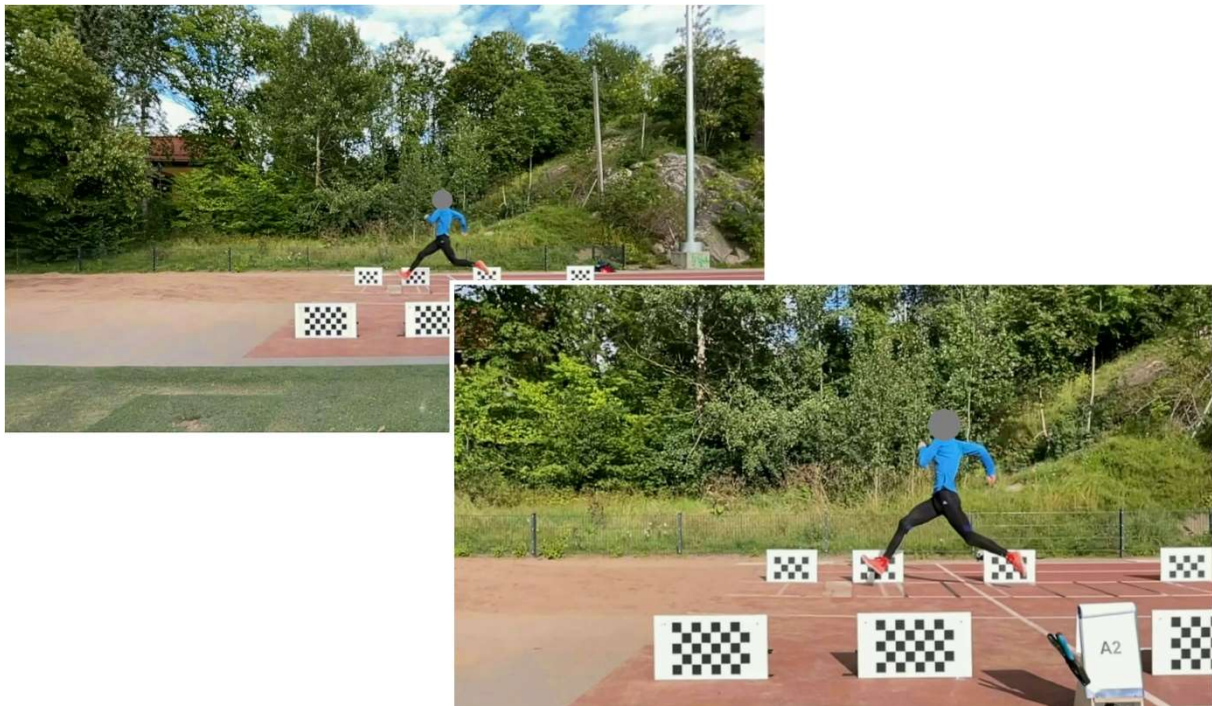
Kuvan 22 alemmassa osassa kontakti aika alkaa kuvaruudun puolesta välistä, jolloin kontaktikohta on mahdollista tunnistaa vasta seuraavassa kuvaruudussa (vihreällä merkitty osuus). Tällöin kontaktin alussa punaisella merkityjä osia ei huomioida kontakti ajan laskennassa, vaan ajan laskenta alkaa vasta seuraavan kuvaruudun alusta. Mikäli kontaktin

loppuminen ei osu kuvaruutujen vaihtumiskohtaan, niin arvioidusta kontaktiajan pituus on jälleen liian pitkä todellisuuteen verrattuna. Kuvan 22 alemmassa osassa kontaktin alkuosan puuttuva aika ja loppuosan lisäaika kumoavat toistensa vaikutusta kontaktiajan laskennassa, jolloin laskennan lopputulos paranee.

Mikäli kontaktin alkamiskohta tunnistettaisiin hieman kuvaruudun alun jälkeen ja kontaktin loppumiskohta osuisi täsmälleen samaan kohtaan kuin kuvaruudun loppumiskohta, niin arvioidusta kontaktiajan kesto olisi liian lyhyt todellisuuteen verrattuna. Maksimissaan kontaktiajan kesto olisi yhden kuvaruudun keston verran liian lyhyt.

Kontaktiajanvirhe voi olla siis yhden kuvaruudun verran liian lyhyt tai se voi olla yhden kuvaruudun verran liian pitkä, eli virhe voi vaihdella eli virhe on  $\pm$  kuvaruudun keston verran. Kuvataajuudella 60 fps virhe on välillä -16,7...16,7 ms, vastaavasti 120 fps -8,3...8,3 ms ja 240 fps  $< -4,2...4,2$  ms.

Pituushyppääjä liikkuu hypyn aikana noin 8...10 m/s nopeudella, jolloin hyppääjä liikkuisi yhden kuvaruudun aikana 60 fps kuvataajuudella kuvatulla videolla noin 13,3...16,7 cm. Vastaavasti 120 fps kuvataajuudella kuvatulla videolla hyppääjä liikkuisi noin 6,8...8,3 cm ja 240 fps kuvataajuudella noin 3,3...4,2 cm. Kuvataajuudella 240 fps päästään hyppääjän sijainnin määrittämisessä suhteellisen tarkkaan tulokseen.



Kuva 23. Kuvaruudut eri suurennoksella kuvatuista hyppyvideoista. Kuva: Kalle Lehtonen

Hyppyjen analysoinnissa käytettiin yhdeksää hyppyvideota, jotka oli kuvattu 1.0x suurennoksella (kuvassa 23 alempi kuva). Hyppyvideoista neljä oli kuvattu kansallisen tason hyppääjän hypyistä ja viisi nuoremman hyppääjän hypyistä. Yhdessä nuoremman hyppääjän hyppyvideolla oli ponnistuksen lisäksi kaksi edellistä kontaktia (yhden kontaktin sijasta), tämän vuoksi työn aikana ohjelmoitu sovellus ei pystynyt analysoimaan kyseistä videota. Hyppyvideoita kuvattiin myös 0.5x suurennoksella (kuvassa 23 ylempi kuva) ja niiden käyttämistä testattiin, mutta useilta analysoinnin kannalta tärkeistä kuvaruuduista ei löytynyt kehon avainpisteitä, joten näiltä hyppyvideoilta ei pystytty määrittämään suureita.

## **4.2 Hyppyvideoiden esikäsittely**

Erikoistyössä hyppyvideot nimettiin uudelleen, leikattiin lyhyemmiksi, kalibroitiin ja stabiloitiin. Hyppyvideoiden tiedostojen nimiin lisättiin seuraavat tiedot: hypyn numero, hyppääjän identifiointitunnus, käytetyn suurennoksen arvo ja tieto siitä, hyväksyttiinkö hyppy analysoitavaksi hyppytapahtumassa. Alkuperäiset hyppyvideot kestivät 12...19 s, hypyn analysoinnin kannalta oleelliset asiat tapahtuvat noin 1...2 s aikana. Videoiden alusta poistettiin kuvaruutuja, joissa hyppääjä ei vielä näy kuvaruudussa. Lisäksi videoilta poistettiin ponnistuksen jälkeisiä kuvaruutuja, koska ne olivat tämän työn tekemisen kannalta tarpeettomia. Puhelimen linssin aiheuttaman linssivääristymän poistamiseksi videot kalibroitiin, tämä mahdollistaa videoilta laskettavien arvojen tarkemman laskemisen. Hyppääjän viimeiset vauhdinottoaskeleiden ja ponnistuksen aiheuttamat tärähdykset välittyivät maata pitkin kamerajalkaan kiinnitettyyn puhelimeen ja aiheuttivat videossa kuvaruudun heilumista. Heilumisen poistamiseksi hyppyvideot stabiloitiin, toimenpide pienensi hieman kuvaruudun korkeutta ja leveyttä.

## 5 TOTEUTUS

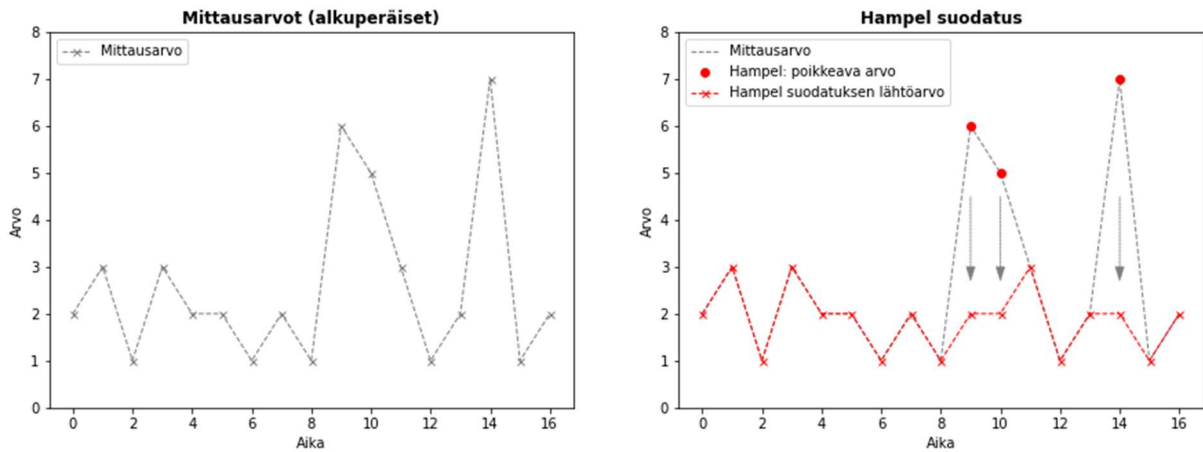
Erilaisten suureiden määrittämiseksi hyppyvideoilta työn tekemisen aikana ohjelmoitiin neljä toisistaan erillistä sovellusta, jotka suoritetaan vaiheittain. Ensimmäisessä vaiheessa etsitään hyppyvideoilta hyppääjän kehon avainpisteet Googlen MediaPipe Pose-asennontunnistusalgoritmin avulla. Toisessa vaiheessa avainpisteiden sijainnit käydään läpi ja niistä etsitään poikkeavia arvoja, tämän tiedon perusteella tehdään arvoihin korjauksia. Lisäksi tässä vaiheessa sijaintitiedoille tehdään suodatus ja määritetään kohdat, joissa hyppääjän jalka on kontaktissa radan pintaan. Kolmannessa vaiheessa edeltävissä vaiheissa kerätyistä tiedoista lasketaan hypyn teknistä toteutusta kuvaavat suureet kuten jalkojen asentojen kulmia ja viimeisen askeleen pituus. Viimeisessä vaiheessa laskennassa saadut tulokset tallennetaan tulokset sisältävän taulukkoon, lisäksi hyppysuorituksesta tallennetaan kuvakaappauksia. Jokaisen vaiheen suorittamisen jälkeen tulokset tallennetaan taulukkoon, näin kaikkien aiemmin suoritettujen vaiheiden tiedot ovat käytettävissä seuraavan vaiheen lähtötietona. Toteutukseen liittyvän luvun jälkeen, seuraavissa luvuissa käydään läpi eri sovelluksissa tehtäviä toimia ja laskentoja.

### 5.1 Toteutuksen menetelmät

Tämän luvun kappaleissa käydään läpi teoriaan liittyviä asioita, joita on käytetty sovelluksissa, joilla muokataan ja analysoidaan pituushyppyvideoista saatuja tietoja. Hampel-suodatinta käytettiin tunnistamaan avainpisteiden sijaintitiedoista poikkeavia arvoja, jotka johtuivat virheellisistä avainpisteiden tunnistamista. Perspektiivin muuttamisen avulla sivulta kuvattu hyppyvideolta valittu alue muutettiin siten, että alue olisi kuvattu ylhäältä alaspäin, jotta voitiin arvioida kontaktikohdan sijainti. Hyppyihin liittyvien kulmien laskennassa käytettiin tangenttifunktiota. Hyppyvideoiden kuvaruudussa yksi pikseli vastaa jotain todellista mitta-alueen luonnossa (cm/pikseli), jotta todellinen mitta saatiin selvitettyä, niin suoritettiin skaalaaminen hyppypaikalle sijoitettujen shakkiruudukkojen avulla.

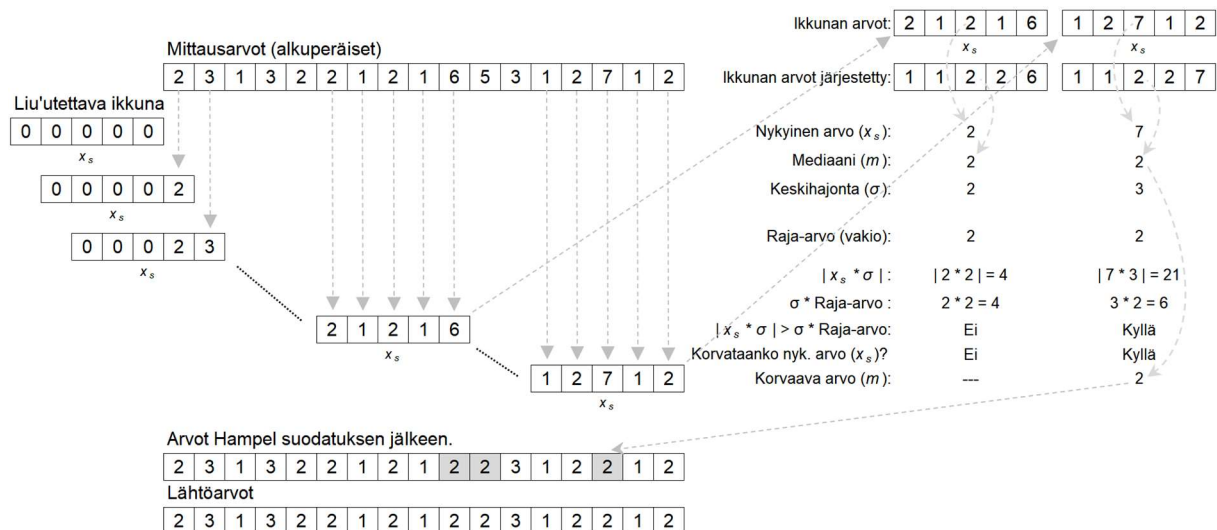
#### 5.1.1 Hampel-suodatin

Hampel-suodattimen avulla voidaan havaita mittaustiedoista poikkeavia arvoja ja laskea niille korvaavat arvot. Kuvassa 24 on esitetty Hampel-suodattimen löytämät poikkeavat arvot ja niille lasketut korvaavat arvot. Kuvassa 25 on esitetty Hampel-suodattimen toiminta, molemmissa kuvissa 24 ja 25 mittausarvot ovat samoja.



Kuva 24. Hampel-suodatuksen vaikutus mittausarvoihin. Kuva: Kalle Lehtonen

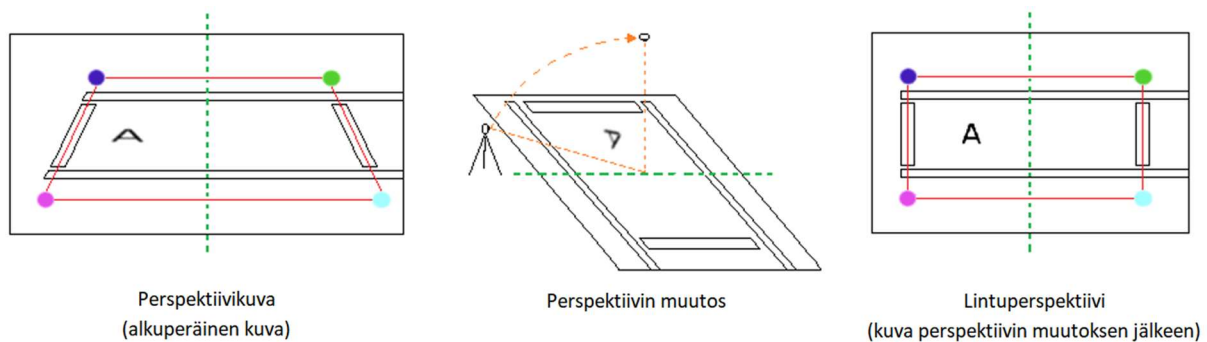
Hampel-suodattimessa poikkeavien arvojen havaitsemiseen käytetään absoluuttista mediaanipoikkeaman (median absolute deviation) laskentaa ja mittaustietojen yli liu'utettavaa ikkunaa. Poikkeavien arvojen tunnistamisessa alkuperäisten mittausarvojen yli liu'utetaan (yleensä parittoman pituista) ikkunaa (kuvassa 25 ikkunan pituus on viisi). Laskennassa arvioinnin kohteena on ikkunan keskimäinen arvo, eli nykyinen arvo ( $x_s$ ). Ensin ikkunan alle jäävistä arvoista etsitään mediaani ( $m$ ) ja lasketaan keskihajonta ( $\sigma$ ). Mikäli nykyisen arvon ( $x_s$ ) ja keskihajonnan ( $\sigma$ ) tulo itseisarvo on isompi kuin, keskihajonnan raja-arvon (vakio) tulo, niin alkuperäinen mittausarvo eli nykyinen arvo ( $x_s$ ) korvataan mediaaniarvolla ( $m$ ). [79] Kuvan 25 toiseksi alimpaan taulukkoon on merkitty harmaalla kohdat, joissa alkuperäinen arvo on korvattu.



Kuva 25. Hampel-suodattimen toiminta ja laskennan vaiheet. Kuva: Kalle Lehtonen

### 5.1.2 Perspektiivin muuttaminen

Pituushyppysuorituksen ponnistusta edeltävän askeleen pituuden määrittämiseksi alkuperäisen kuvan perspektiivi muutettiin lintuperspektiiviksi. Kuvassa 26 hahmotetaan perspektiivin muuttamista kameraa siirtämällä. Vasemmanpuoleinen kuva on alkuperäinen kuva, joka on kuvattu esimerkiksi kamerajalustan päältä. Mikäli olisi mahdollista, niin kamera siirrettäisiin sivusta kohteen yläpuolelle (kuvan 26 keskimmäinen kuva) perspektiivin muuttamiseksi, jolloin kuvattaisiin lintuperspektiivistä (ylhäältä alaspäin) ja tuloksena syntyisi kuvan 26 oikeanpuoleinen kuva [80]. Kameran siirtäminen tai useamman kamerasiirron käyttäminen ei ole aina mahdollista, tällöin perspektiivin muuttaminen voidaan tehdä laskennallisesti. Erikoistyössä [1] [K. Lehtonen. 2023. Pituushyppyanalytiikka hyppyvideoiden avulla – videomateriaalin kerääminen ja esikäsittely. Erikoistyökurssin työseloste. Turun yliopisto.] pituushyppysuoritukset oli kuvattu vauhdinottoradan sivusta, joten perspektiivin muuttaminen tehtiin sovelluksessa laskennallisesti.

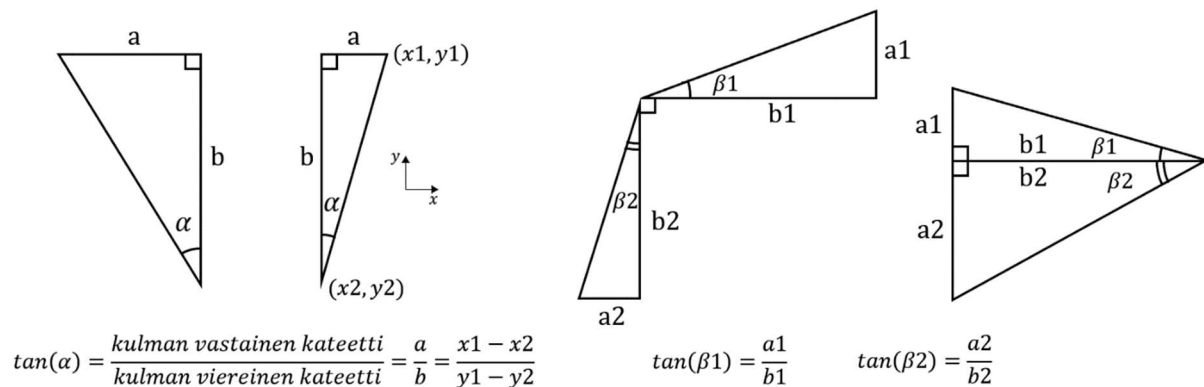


Kuva 26. Perspektiivin muuttaminen Kuva: Kalle Lehtonen

Tässä työssä (sovellukset on ohjelmoitu Python-ohjelmointikielellä) perspektiivin muuttaminen tehtiin OpenCV:n (Open Source Computer Vision Library) kuvankäsittelykirjaston metodeja käyttämällä. Perspektiivin muutos lasketaan OpenCV:n `getPerspectiveTransform` -metodia käyttäen, syötearvoina metodille annetaan alkuperäisestä kuvasta neljän pisteen koordinaatit (pisteet kuvan 26 vasemmanpuoleisessa kuvassa) ja neljän pisteen sijainnit uudessa kuvassa (pisteet kuvan 26 oikeanpuoleisessa kuvassa), metodi palauttaa  $3 \times 3$  kokoisen muutosmatriisin. OpenCV:n `warpPerspective` -metodi laskee perspektiivin muutoksen ja palauttaa kuvan (kuvan 26 oikeanpuoleinen kuva), metodille annetaan syötearvoina alkuperäinen kuva, edellä laskettu muutosmuutosmatriisi ja luotavan/palautettavan kuvan koko. [81]

### 5.1.3 Kulmien laskeminen

Työssä laskettiin ponnistusvaiheeseen liittyviä kulmia, suurimmat kulmat olivat noin 160 ... 170° (ponnistavan jalan polvikulma viimeisen kontaktin alussa) ja pienimmät noin -5 ... 5° (esimerkiksi kehon kaltevuuskulma viimeisen kontaktin alussa ja lopussa). Kulmien laskennassa käytettiin tangenttifunktiota, koska kahdesta xy-koordinaatista on helpoin laskea kulman vastaisen ja viereisen kateettien pituudet. Mikäli kulma oli yli 90° kulma tai muodostuneessa kolmiossa ei ollut 90° kulmaa, niin laskettiin kulma kahdessa osassa (kuvan 27 kaksi oikeanpuoleista kuvaa).

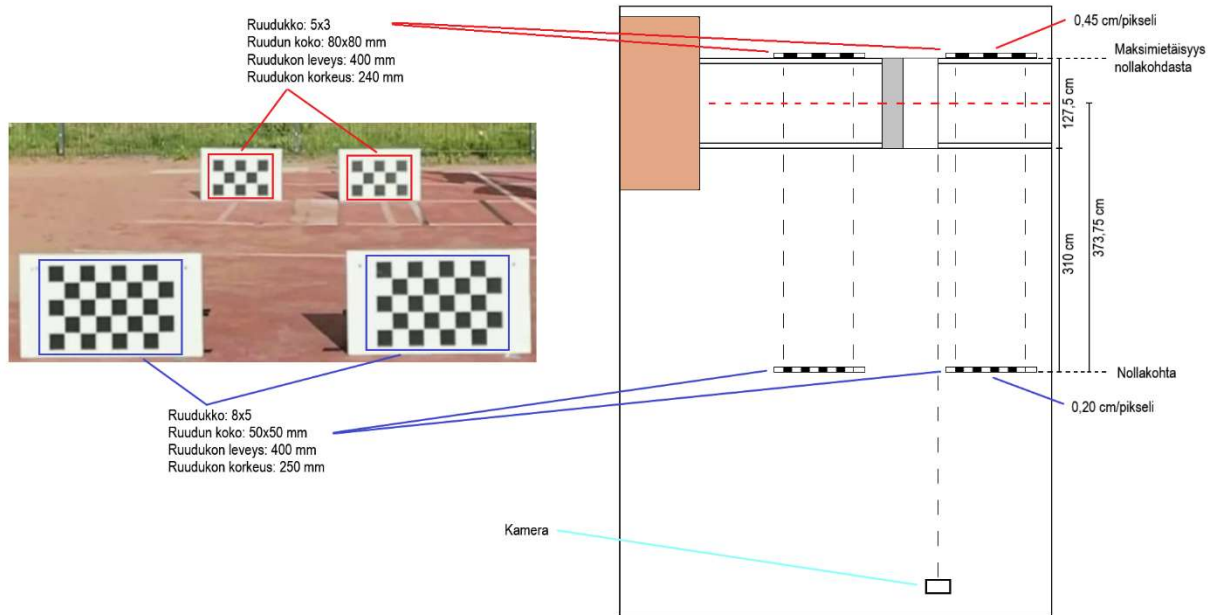


Kuva 27. Tangentin laskeminen. Kuva: Kalle Lehtonen

### 5.1.4 Skaalaaminen

Hypypysuoritukseen liittyvien ponnistusvaiheen nopeuksien ja lonkan aleneman laskemiseksi, pitää määrittää kuljettu matka. Hypypysuoritukseen nopeuksiin liittyvät ajat saadaan laskettua hyppyvideon kuvaa/sekunti ja kuvaruutujen määrä tietojen avulla. Kuljetun matkan selvittäminen aloitetaan laskemalla molempien vauhdinottoradan takana olevien shakkiruudukkojen (kuvassa 28 5x3 kokoiset ruudukot) avulla, kuinka monta senttimetriä yksi pikseli vastaa luonnossa (arvot lasketaan erikseen pysty- ja vaakasuuntaan). Lisäksi lasketaan kuvan etualalla olevien shakkiruudukkojen (kuvassa 28 8x3 kokoiset ruudukot) vastaavat arvot. Kun takana oleville shakkiruuduille (2 kappaletta) on laskettu luonnollinen koko (cm/pikseli arvot), niin lasketaan näistä arvoista vaaka- ja pystysuunnille keskiarvot. Samanlainen keskiarvon laskenta suoritetaan etualalla oleville shakkiruudukoille. Keskiarvojen laskennan jälkeen voidaan laskea todellisuudessa kuljettu matka, jos esimerkiksi liikutaan vaakasuuntaan tietty pikselimäärä vauhdinottoradan takana olevien shakkiruutujen samassa linjassa (vauhdinottoradan kauimman viivan päällä). Matkan laskeminen senttimetreissä onnistuu myös, mikäli liikkuminen tapahtuu sivusuunnassa samassa linjassa kameran ja

vauhdinottoradan välissä olevien shakkiruudukkojen kanssa. Pituushyppääjä ei kuitenkaan juokse vauhdinottoradan takareunan (tai edessä olevien shakkiruutujen) kanssa samassa linjassa, vaan suurin piirtein keskellä rataa (merkitty kuvan 28 oikeanpuoleiseen kuvaan punaisella katkoviivalla).



Kuva 28. Kameralla kuvattu näkymä hyppylankun kohdalta (vasen) ja vastaava lintusperspektiivistä (oikea). Kuva: Kalle Lehtonen

Vauhdinottoradan suuntaisen matkan laskemiseksi keskelle vauhdinottorataa pitää suorittaa skaalaaminen. Kuvassa 29 on esitetty kaava 2, jolla voidaan laskea yhden pikselin leveys (cm/pikseli) arvo tietyn matkan päässä nollakohdasta, vastaavalla tavalla lasketaan pikselille korkeus (cm/pikseli) arvo. Kuvassa 29 on laskettu esimerkki, jossa vaakasuuntainen siirtymä taaimmaisten shakkiruutujen kohdalla on 0,45 cm/pikseli ja etummaisten shakkiruutujen kohdalla 0,20 cm/pikseli. Yhden pikselin leveys lasketaan vauhdinottoradan keskilinjalle (merkitty kuvan 28 oikeanpuoleiseen kuvaan punaisella katkoviivalla).

$$pikselin\ leveys\ (cm) = \frac{et_x}{et_{100}} (pl_{100} - pl_0) + pl_0 \quad (2)$$

, jossa

- $et_x$  - etäisyys nollakohdasta, johon halutaan määrittää pikselin leveys
- $et_{100}$  - maksimietäisyys nollakohdasta
- $pl_{100}$  - pikselin leveys maksimietäisyydellä nollakohdasta
- $pl_0$  - pikselin leveys nollakohdassa

**Esimerkki (arvoilla laskettuna):**

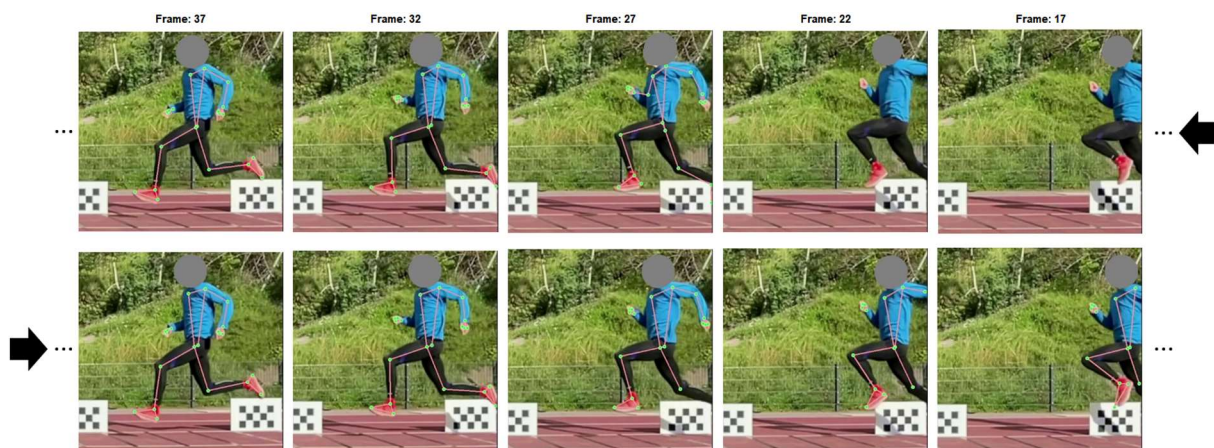
$$pikselin\ leveys\ (cm) = \frac{(310 + 127,5/2)\ cm}{(310 + 127,5)\ cm} (0,45 - 0,20)\ cm + 0,20\ cm \approx 0,4135\ cm$$

Kuva 29. Pikselin leveyden laskeminen tietyssä kohtaa kuvaa, laskettu esimerkki arvoilla keskelle vauhdinottorataa. Kuva: Kalle Lehtonen

## 5.2 Vaihe 1 - avainpisteiden etsintä

Ensimmäisessä vaiheessa erikoistyyssä [1] kuvatuista ja käsitellyistä hyppyvideon kuvaruuduista etsitään kehon nivelten ja osien sijainnit. Sovellus käy läpi videon jokaisen kuvaruudun ja etsii niistä MediaPipe Pose-asennontunnistusalgoritmia käyttämällä 33 kehon avainpistettä (kuva 19). Sovellus tallentaa taulukkoon kuvaruudun numeron ja kehon avainpisteiden x- ja y-koordinaattien sijainnit kuvaruudulla.

Avainpisteiden etsintä aloitetaan videon viimeisestä kuvaruudusta (video toistetaan lopusta alkuun), tällöin hyppääjä on kokonaisuudessaan kuvaruudussa ensimmäisessä kuvaruudussa ja kehon avainpisteet tunnustetaan heti. Mikäli avainpisteiden etsintä aloitettaisiin videon ensimmäisestä kuvaruudusta, niin ensimmäisissä kuvaruuduissa hyppääjä ei vielä näy kuvaruudulla. Seuraavissa kuvaruuduissa tulee näkyville osittain hyppääjän jalka tai käsi, tämän seurauksena kehon avainpisteitä ei vielä tunnisteta.



Kuva 30. Avainpisteiden tunnistaminen alusta loppuun (ylempi kuvasarja) ja lopusta alkuun (alempi kuvasarja) toistetuilta hyppyvideoilta. Kuva: Kalle Lehtonen

Kuvan 30 kuvasarjoissa nuolet osoittavat ensimmäisen kuvaruudun, josta kehon avainpisteitä aletaan etsimään. Kuvan 30 ylemmässä kuvasarjassa hyppyvideo toistetaan normaalisti (alusta loppuun), tällöin MediaPipe Pose-asennontunnistusalgoritmi havaitsee suurimman osan kehon avainpisteistä (vasta) kuvaruudussa 27. Kuvan 30 alemmassa kuvasarjassa hyppyvideo toistetaan takaperin (lopusta alkuun), tällöin asennontunnistusalgoritmi havaitsee suurimman osan avainpisteissä (jo) kuvaruudussa 17. Tätä aiemmissa kuvaruuduissa osa avainpisteistä tunnustetaan, mutta osa avainpisteistä tunnustetaan väärin, esimerkiksi molempien jalkojen avainpisteet on tunnustettu samaan jalkaan (liitteen 5 ensimmäinen kuva kuvaruutu: 16). Suorittamalla avainpisteiden etsintä takaperin toistetulta videolta, tällöin saadaan tallennettua avainpisteiden sijainteja useammasta videon alkuosan kuvaruuduista.

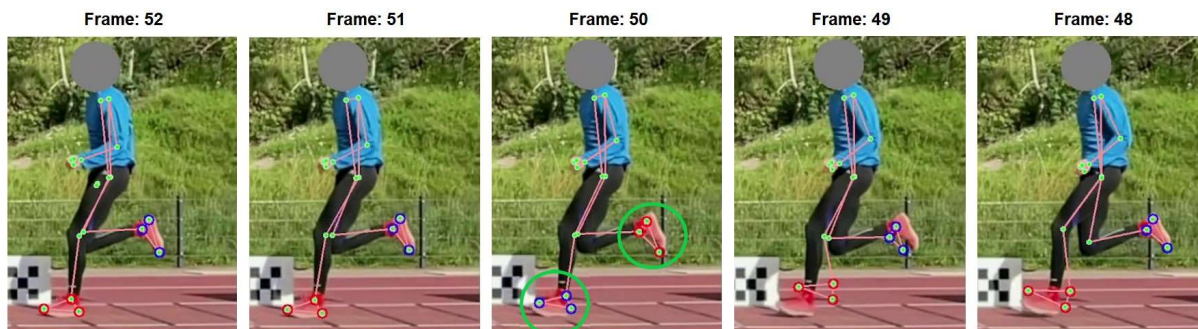
### 5.3 Vaihe 2 – mittausarvojen käsittely ja kontaktikohtien määrittäminen

Toisessa vaiheessa suoritetaan kaksi toimenpidettä: avainpisteiden xy-koordinaattien tarkastelu/muokkaaminen ja kontaktikohtien etsiminen. Osassa pituushyppyvideoista MediaPipe Pose-asennontunnistusalgoritmi oli joissain kuvaruuduissa sekoittanut kehon vasemman ja oikean puolen avainpisteet keskenään. Tämän korjaamiseksi lonkan ja jalkojen avainpisteiden x- ja y-koordinaateista etsitään poikkeavia arvoja, mikäli samassa kuvaruudussa kehon vasemman- ja oikeanpuolen molemmat avainpisteet (esimerkiksi nilkan) ovat poikkeavia arvoja, niin arvot vaihdetaan keskenään.

Kontaktikohtien määrittämisessä etsitään kahden viimeisen kontaktikohdan alkamis- ja päättymiskohdat. Kontaktikohdilla tarkoitetaan kohtia, joissa hyppääjän jalka koskee urheilukentän pintaa. Erikoistyössä [1] kuvatuista hyppyvideoista löytyy kaksi kontaktikohtaa: toiseksi viimeinen kontakti (ponnistusta edeltävä kontakti) ja viimeinen kontakti (varsinainen ponnistus).

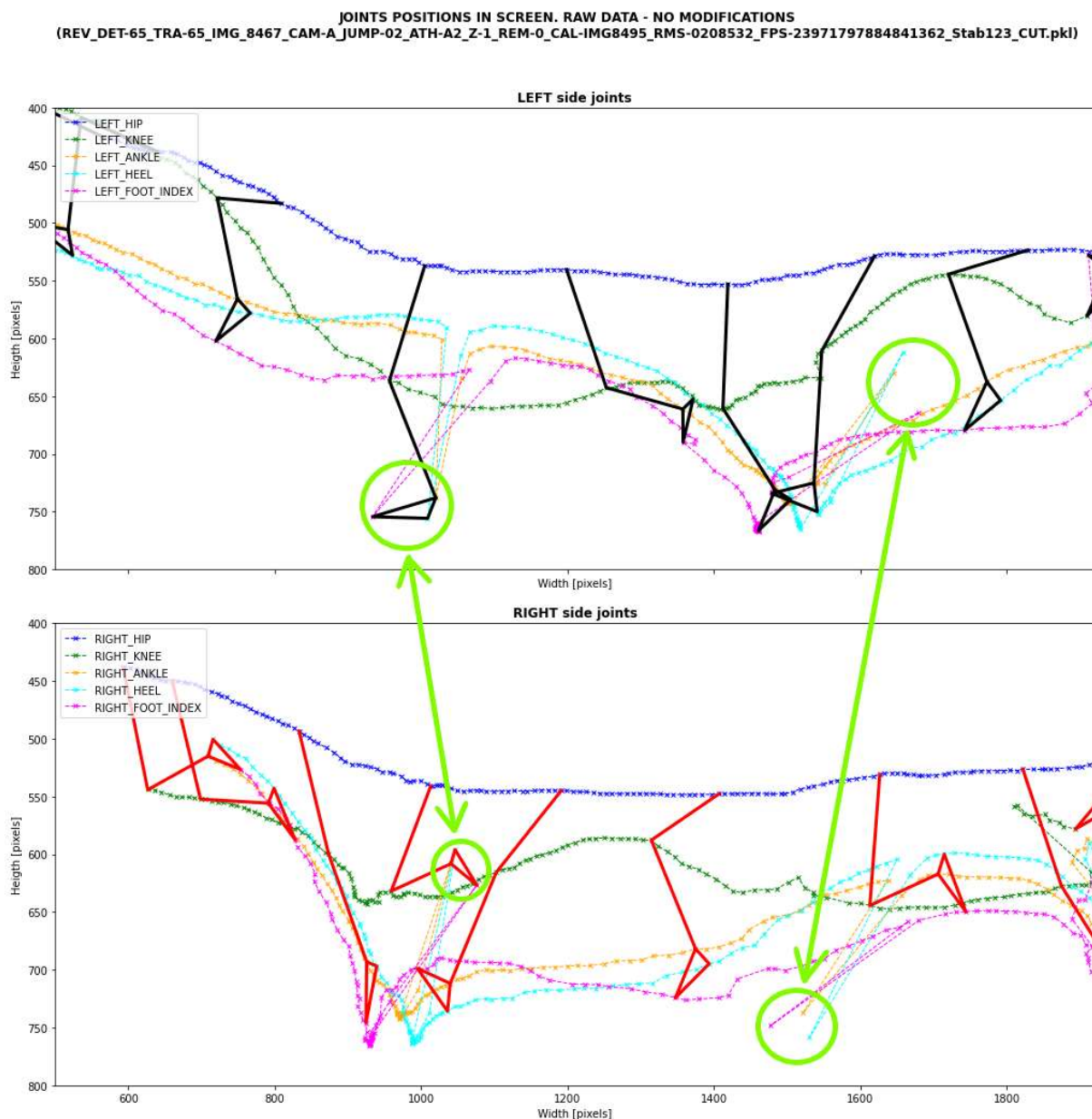
#### 5.3.1 Mittausarvojen käsittely

MediaPipe Pose-asennontunnistusalgoritmi oli sekoittanut joissain kuvaruuduissa kehon vasemman ja oikean puolen avainpisteet keskenään (esimerkiksi vasemman puolen nilkan koordinaatit oli tallennettu oikean puolen nilkan tietoihin ja toisin päin). Avainpisteiden koordinaattien korjaamiseksi alkuperäiset lonkan ja jalkojen avainpisteiden xy-koordinaattien (x- ja y-koordinaatit kuvaruudulla) sijainnit tutkitaan ja tarpeen vaatiessa xy-koordinaattien arvot vaihdetaan kehon toiselle puolelle (vasemmalta oikealle tai oikealta vasemmalle). Aluksi lonkan ja jalkojen avainpisteiden x- ja y-koordinaatit ajetaan Hampel-suodattimen läpi ja niistä etsitään poikkeavia arvoja. Mikäli samassa kuvaruudussa kehon molempien puolien y-koordinaattien arvot ovat poikkeavia arvoja, niin kyseisten indeksien xy-koordinaatit vaihdetaan keskenään.



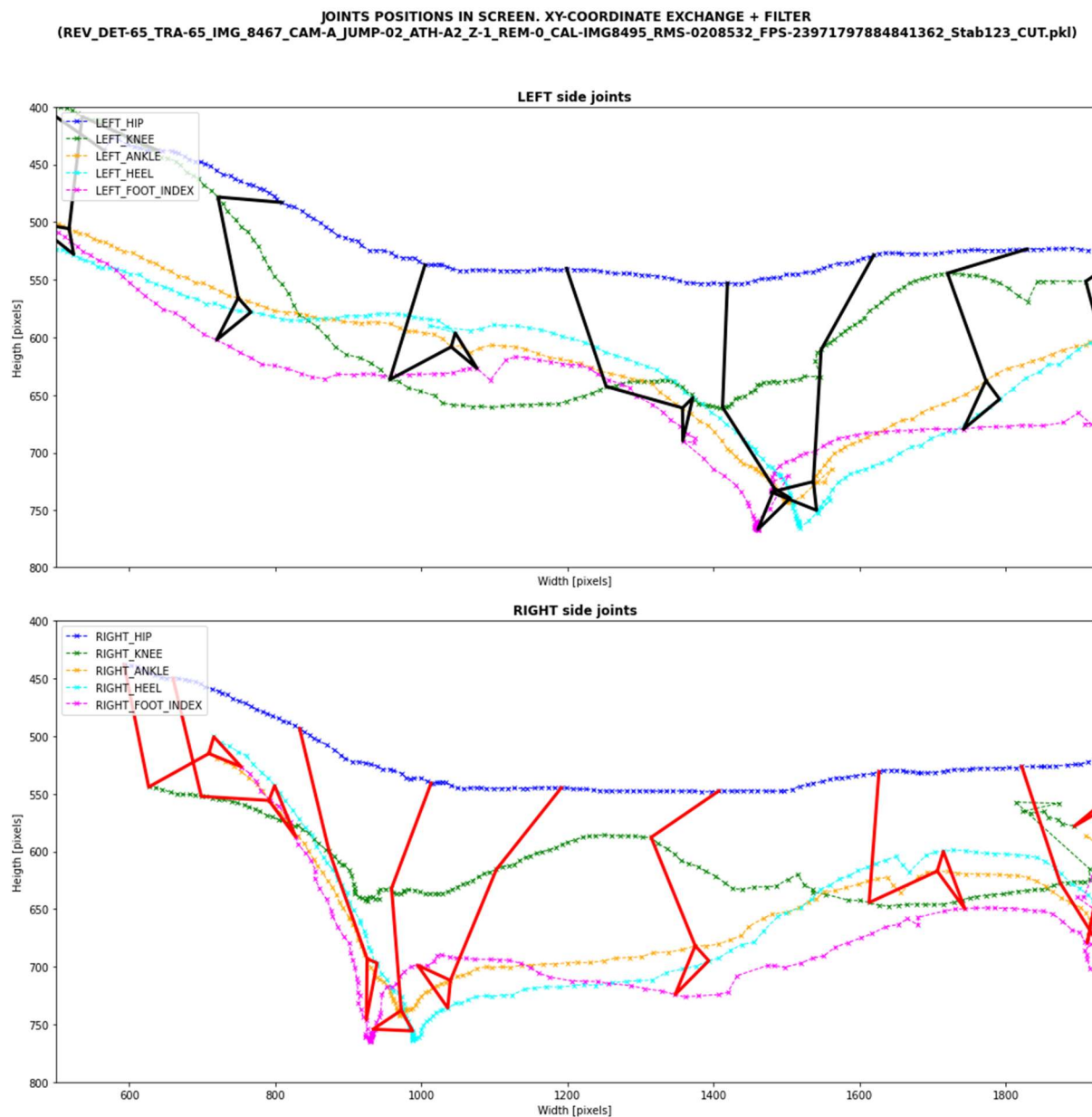
Kuva 31. Avainpisteiden sijainnin vaihtuminen. Kuva: Kalle Lehtonen

Kuvassa 31 kaikkiin kuvaruutuihin on lisätty (punaiset/siniset) ympyrät nilkan, kantapään ja isovarpaan avainpisteiden ympärille. Punaiset ympyrät on lisätty vasemman puolen ja siniset ympyrät on lisätty oikean puolen avainpisteiden kohdalle. MediaPipe Pose-asennontunnistusalgoritmi tunnistaa oikein avainpisteet (kehon vasemmalle ja oikealle puolelle) kuvan 31 kuvaruuduissa 48, 49, 51 ja 52. Kuvaruudussa 50 avainpisteiden XY-koordinaatit ovat vaihtuneet puolelta toiselle, tämän seurauksena sovellus vaihtaa avainpisteiden xy-koordinaattien paikat kehon toiselle puolelle.



Kuva 32. Alkuperäiset mittausarvot lonkan ja jalkojen avainpisteiden sijainnista kuvaruudulla. Kuva: Kalle Lehtonen

Kuvan 32 ylimmässä kuvassa on kehon vasemman puolen lonkan ja jalan avainpisteiden sijainnit kuvaruudulla, kuvan alemmassa kuvassa ovat oikean puolen vastaavat avainpisteet. Lisäksi kuviin on lisätty avainpisteitä yhdistävät viivat, joista näkee jalkojen liikkumisen suorituksen aikana. Kuvan 32 molempiin kuviin on merkitty vihreillä ympyröillä kohdat, joissa xy-koordinaatit ovat vaihtuneet keskenään. Kuvan 31 kuvaruudussa 50 tapahtuva avainpisteiden sijainnin vaihtuminen löytyy kuvasta 32 (oikeanpuoleinen vihreä merkintä).



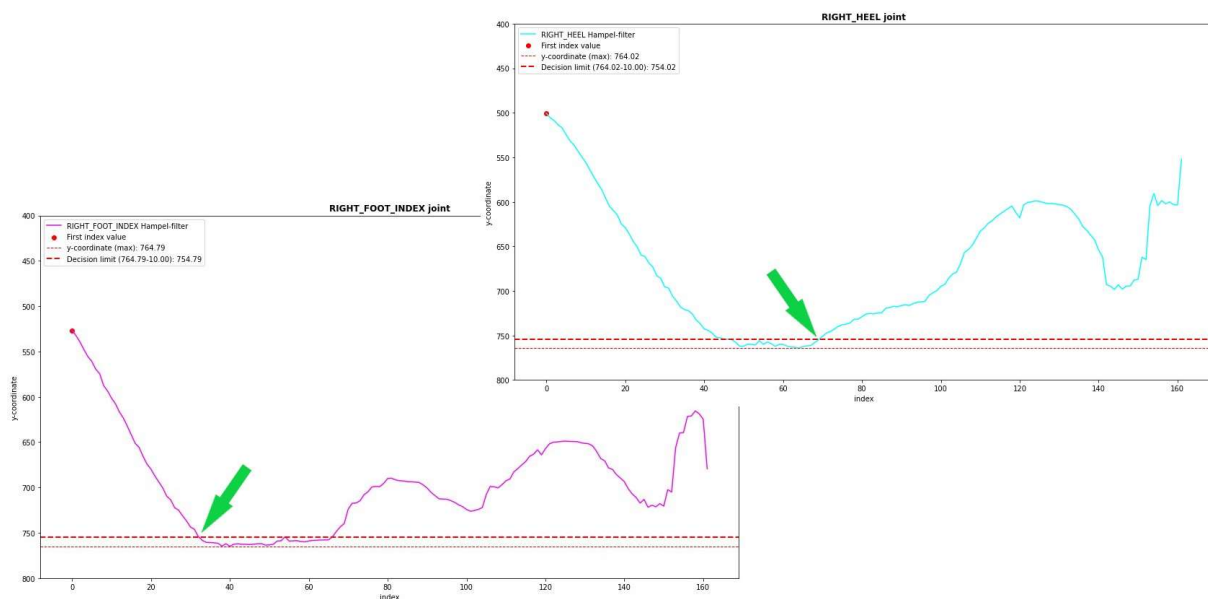
Kuva 33. Lonkan ja jalkojen avainpisteiden sijainnista kuvaruudulla, kun xy-koordinaattien vaihto ja Hampel-suodatus on tehty. Kuva: Kalle Lehtonen

Edellä suoritetun xy-koordinaattien vaihdon jälkeen lonkan ja jalkojen avainpisteiden koordinaateille tehtiin vielä suodatus Hampel-suodatinta käyttäen, näitä avainpisteiden xy-koordinaatteja käytettiin hyppysuorituksista tehdyissä tarkasteluissa. Hampel-suodatuksesta saatavat tiedot tallennetaan mittaustiedot sisältävään taulukkoon. Kuvassa 33 on kuvattu xy-koordinaattien arvot, koordinaattien vaihdon ja Hampel-suodatuksen jälkeen.

### 5.3.2 Kontaktikohtien määrittäminen

Kontaktikohtien määrittämisessä etsitään ponnistusta edeltävän kontaktin ja ponnistuksen alkamis- ja päättymiskohdat. Etsittäviä kontaktikohtia on neljä:

- Toiseksi viimeisen askeleen kontakti (ei ponnistava jalka):
  - ensimmäinen kontaktikohta: kantapää
  - viimeinen kontaktikohta: isovarvas.
- Viimeisen askeleen (ponnistus) kontakti (ponnistava jalka):
  - ensimmäinen kontaktikohta: kantapää
  - viimeinen kontaktikohta: isovarvas.

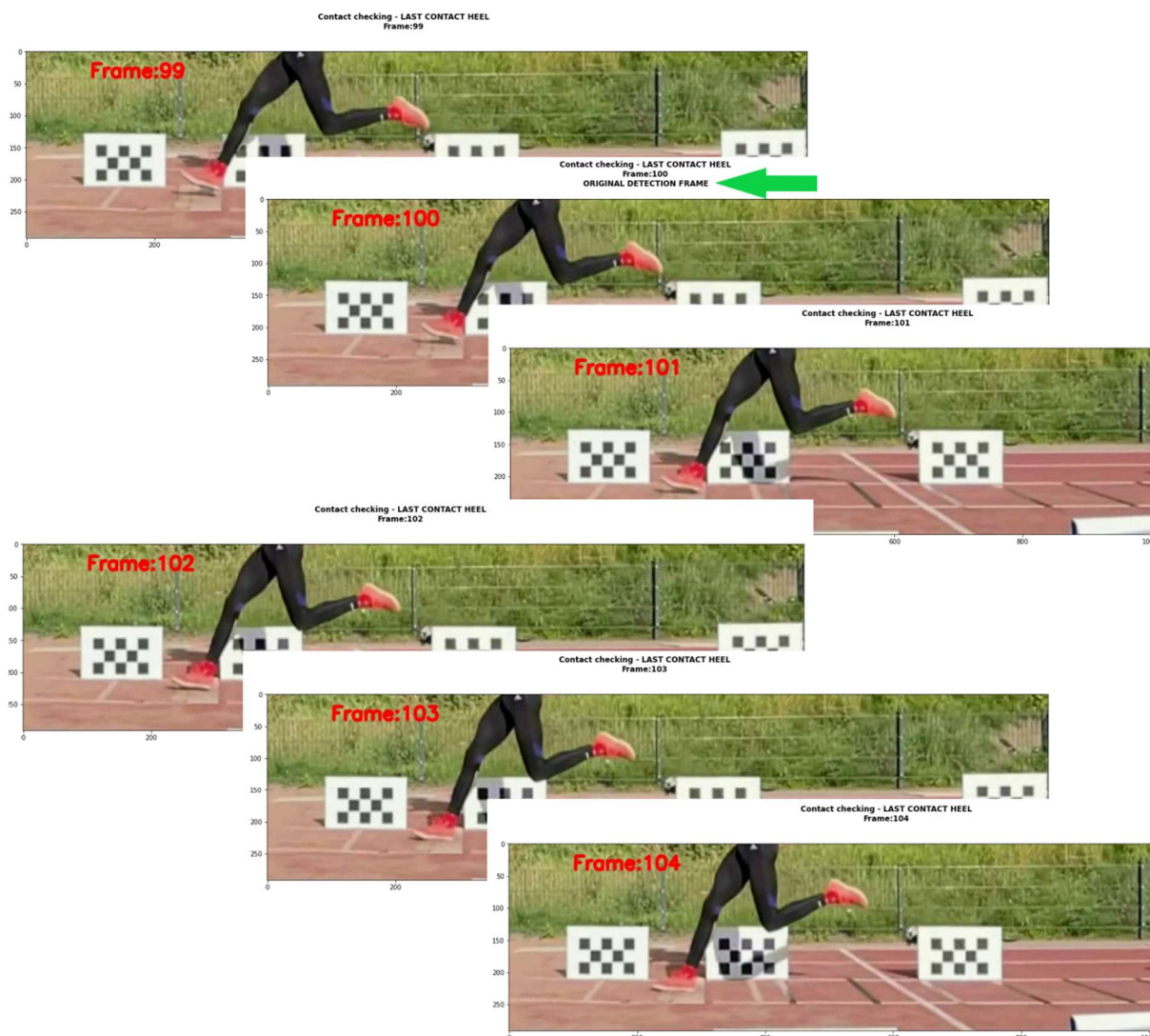


Kuva 34. Viimeisen kontaktin (ponnistusaskel) isovarpaan (vasemmalla) ja kantapään (oikealla) y-koordinaattien arvot kuvaruuduittain. Kuva: Kalle Lehtonen

Kontaktikohtien määrittämisessä etsitään kontaktikohtien alku- ja loppukohtia hyppyvideolta. Hyppyvideot käsiteltiin erikoistyössä [1] niin, että niistä löytyy kaksi viimeistä kontaktia. Kontaktien alkukohtien (kantapää osuu kentän pintaan) määrittämiseksi etsitään vasemman ja oikean jalan kantapään avainpisteiden sijainneista maksimikohdat. Kontaktin loppumiskohtien määrittämisessä etsitään vasemman ja oikean jalan isovarpaan avainpisteiden sijainneista maksimikohdat. Kuvan 34 kuvissa xy-koordinaatistojen nollakohdat ovat kuvaruudun vasemmassa yläkulmassa. Kuvan 34 molempiin kuviin on piirretty kaksi vaakasuuntaista

katkoviivaa, näistä alempi on piirretty maksimiarvon tasolle, ylempi viiva on päätösraja. Kuviin vihreällä nuolella merkityistä kohdista (indekseistä) aletaan etsimään kontaktikohtia.

Kuvan 34 kuviin on piirretty ponnistavan (oikeanpuoleisen) jalan kantapään (oikeanpuoleinen kuva) ja isovarpaan (vasemmanpuoleinen kuva) avainpisteiden sijainnit kuvaruuduissa (x-akselilla on kuvaruudun numero). Molempiin kuviin on merkitty avainpisteen y-koordinaatin maksiarvo (alempi punainen katkoviiva) ja päätösraja (ylempi punainen katkoviiva), joka on kymmenen pikseliä pienempi kuin maksimiarvo. Sovellus tulostaa näytölle 21 kappaletta kuvia, näistä keskimäinen kuva on tulostettu kohdasta, jossa avainpisteen y-koordinaatin arvo ylittää päätösrajan (merkitty kuvan 34 kuviin vihreällä nuolella). Mikäli sovellus on löytänyt oikean maksimikohdan, niin esimerkiksi kontaktin alkamiskohdan etsinnässä kuvasarjan keskimäinen kuva on kohta, jossa kontakti alkaa (kantapää osuu kentän pintaan). Kuvasarjan keskikohtaa edeltävät olisivat tällöin kuvia kohdista, joissa kontakti ei ole vielä alkanut (kantapää on vielä ilmassa). Keskikohdan jälkeiset kuvat olisivat tällöin kohtia, joissa kontakti on alkanut (kantapää on kentän pinnassa). Sovelluksen tulostamista kuvista käyttäjä valitsee kuvaruudun, jossa kontakti käyttäjä mielestä todellisuudessa alkaa.



Kuva 35. Kuusi kuvaa sovelluksen käyttäjälle tulostamasta kuvasarjasta, joista käyttäjä valitsee varsinaisen kontaktin alkamiskohdan. Kuva: Kalle Lehtonen

Kuvaan 35 on koottu kuusi käyttäjälle tulostettua kuvaruutua viimeisen askeleen ensimmäisen kontaktikohdan alkamiskohdan määrittämiseen liittyen. Sovelluksen avulla määritetyn keskimmäisen kuvaruudun on numero 100 (merkitty kuvaan 35 vihreällä nuolella), kantapää ei kuitenkaan ole vielä tässä kuvassa kontaktissa kentän pintaan. Kuvasarjan perusteella kontakti alkaa kuvaruudussa 103 tai 104, kun kantapään osuessa kentän pintaan. Kun kaikkien toiseksi viimeisen ja viimeisen kontaktikontaktikohtien alkamis- ja loppumiskohdat on käyty läpi (sovellus tulostaa neljä erillistä kuvasarjaa), niin mittauksiedot sisältävään taulukkoon tallennetaan tieto siitä, onko jalka kontaktissa kentän pintaan vai ei.

## 5.4 Vaihe 3 – hyppyjen analysointi

Hyppysuorituksista määritetyt suureet on koottu taulukkoon 2, vastaavia suureita on määritetty vuoden 2018 hallimaailmanmestaruuskisojen raportissa [2]. Tämän luvun kappaleissa käydään läpi yksityiskohtaisemmin, miten suureet on laskettu.

Taulukko 2. Vauhdinotto- ja ponnistusvaiheissa määritettävät suureet. Taulukko: Kalle Lehtonen

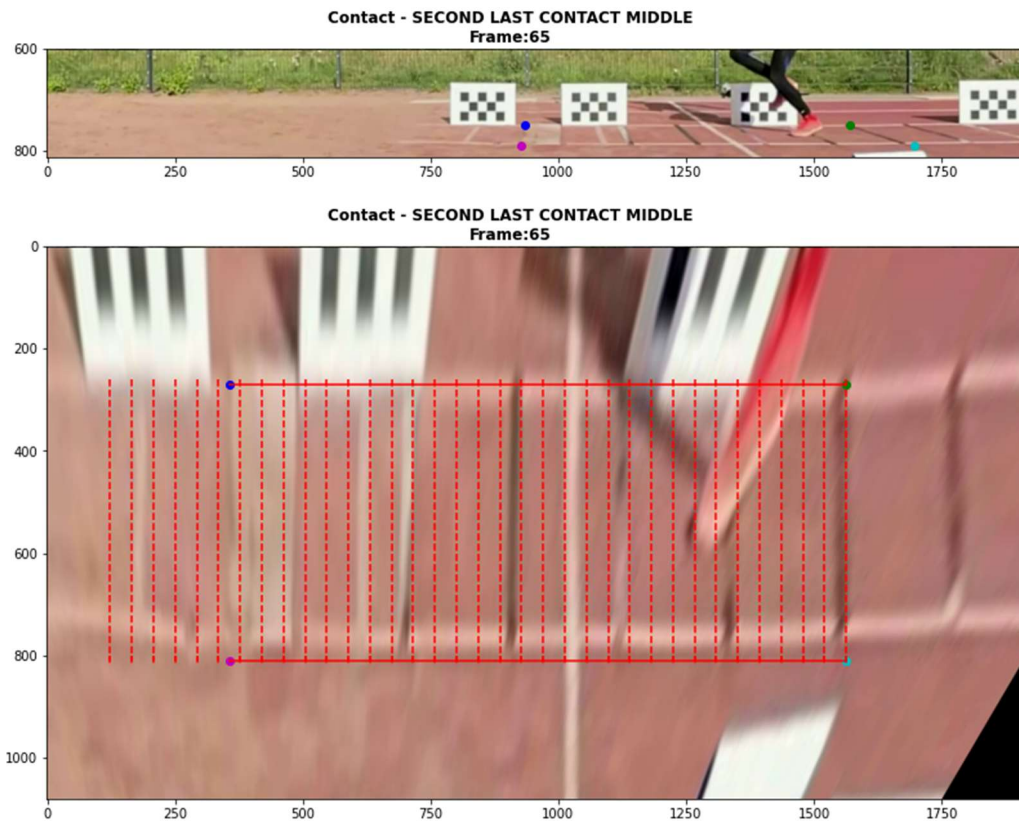
Vauhdinottovaihe	Ponnistusvaihe
Toiseksi viimeisen ja viimeisen kontaktin kontaktiaika	
Viimeisen askeleen lentoaika	
Viimeisen askeleen pituus	
Lonkan pystysuuntainen sijainnin muutos	
Keskimääräinen nopeus viimeisen askeleen aikana	
	Ponnistusvaiheen nopeudet: vaaka-, pysty- ja resultanttinopeus, vaakasuuntaisen nopeuden muutos
	Hyppykulma
	Ponnistavan jalan polven kulmat: viimeisen kontaktin alkaessa, pienin polvikulma, polven kulman muutos
	Ponnistavan jalan polvikulman muutosnopeus
	Lonkan kaltevuuskulmat (kantapää - lonkka)
	Kehon kaltevuuskulmat (lonkka - olkapää)
	Heilahtavan jalan kulma (lonkka - polvi)
	Heilahtavan jalan keskimääräinen kulmanopeus

### 5.4.1 Kontaktiajat ja viimeisen askeleen lentoaika

Kontaktiaikojen laskemiseksi mittaustiedot sisältävästä taulukosta lasketaan, kuinka monessa kuvaruudussa jalka on ollut kontaktissa hyppyalustaan ja tämä määrä kerrotaan yhden kuvaruudunkuvaruudun kestolla (laskettu hyppyvideon kuvaa/sekunti tiedon avulla). Lentoajan laskeminen suoritetaan samalla tavalla, mutta kuvaruutujen määrä on askelten väliin jäävä määrä.

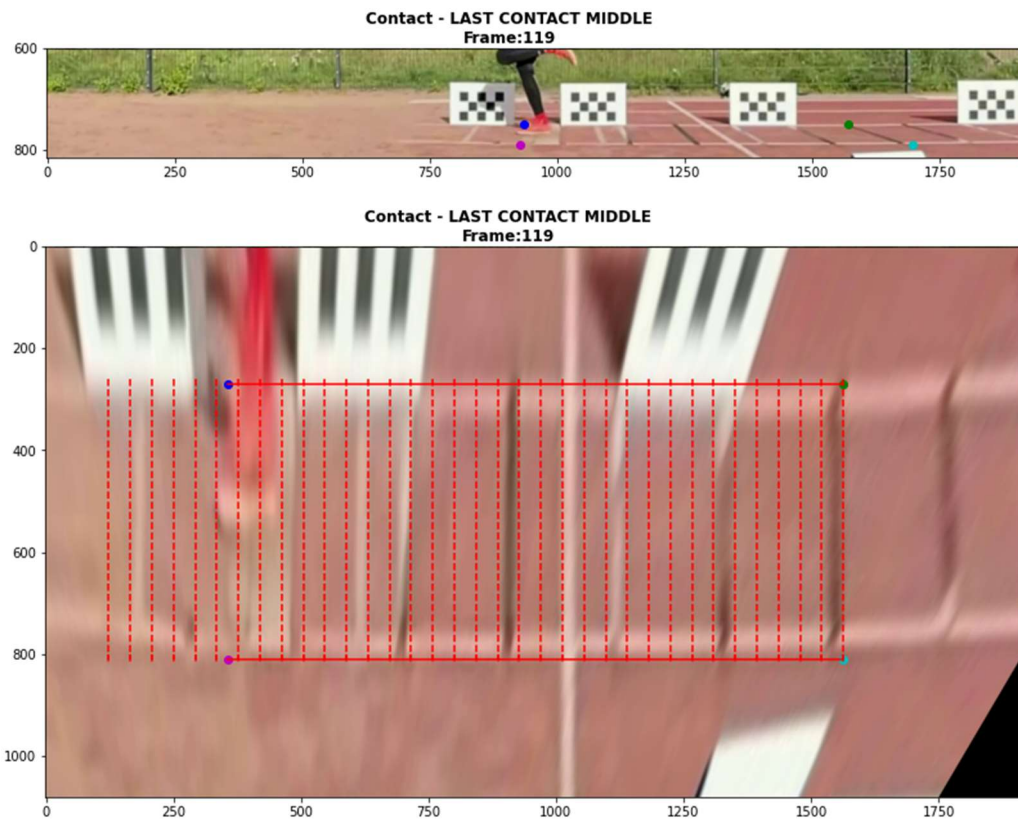
### 5.4.2 Viimeisen askeleen pituus

Viimeisen askeleen pituuden määrittämiseksi alkuperäiselle kuvalle tehdään perspektiivin muuttaminen lintuperspektiiviin. Kuvan 36 kuvissa esitellään toiseksi viimeisen kontaktikohdan alkuperäisen (ylempi kuva) kuvan perspektiivin muuttamista lintuperspektiiviin (alempi kuva). Vastaavat kuvat löytyvät kuvan 37 kuvista, joissa esitellään viimeisen kontaktikohdan alkuperäisen kuvan muuttamista lintuperspektiiviin.



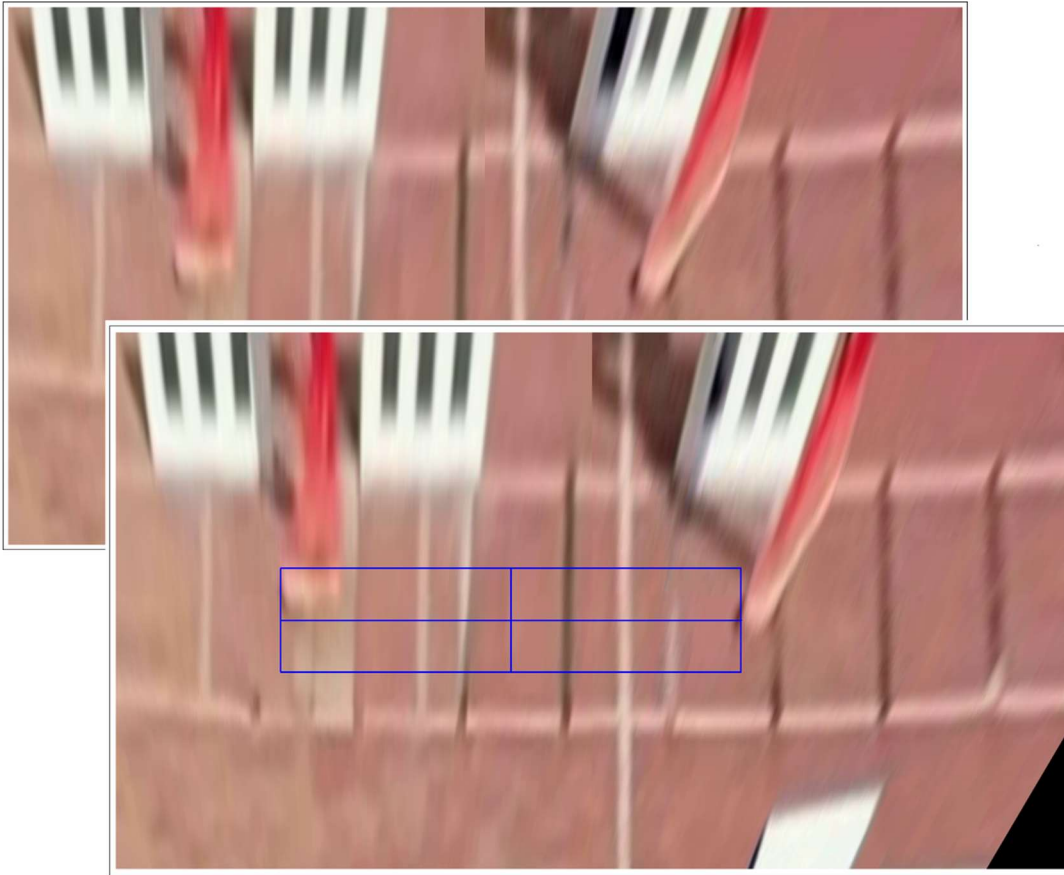
Kuva 36. Ylhäällä alkuperäinen kuva toiseksi viimeisestä kontaktista ja alhaalla kuva, jossa perspektiivin muutos on suoritettu. Kuva: Kalle Lehtonen

Kuvan 36 (ja 37) ylempään kuvaan on merkitty puolisuunnikkaan kulmapisteet eri väreillä. Kulmapisteet ovat laskennallisen perspektiivin muutoksen kulmapisteitä, tähän alueeseen perspektiivin muutos kohdistetaan/toteutetaan. Kuvan 36 (ja 37) alemmassa kuvassa on perspektiivin muutoksen jälkeinen kuva lintuperspektiivistä, johon on merkitty kulmapisteet vastaavilla väreillä kuin ylempässä kuvassa.



Kuva 37. Ylhäällä alkuperäinen kuva viimeisestä kontaktista ja alhaalla kuva, jossa perspektiivin muutos on suoritettu. Kuva: Kalle Lehtonen

Ennen pituushyppysuoritusten kuvaamista erikoistyössä [1] vauhdinottoradan pintaa liimattiin 5 cm leveää mustaa ilmastointiteippiä 50 cm välein (etureunojen etäisyys), lisäksi hyppylankun lyhyemmän sivunpituus oli 30 cm. Näiden tietojen ja lintuperspektiivi kuvien avulla on mahdollista laskea viimeisen askeleen pituus. Luonnossa kuvien 36 ja 37 kuvissa hyppylankun lankun etureunan sininen kulmapisteen ja vihreän kulmapisteen etäisyys toisistaan on 285 cm (purppuran ja vaalean sinisen kulmapisteiden etäisyys on sama). Kulmapisteiden etäisyyden avulla kuvien 36 ja 37 alemmista kuvista voitiin laskea yhden pikselin leveys luonnollisessa koossa.

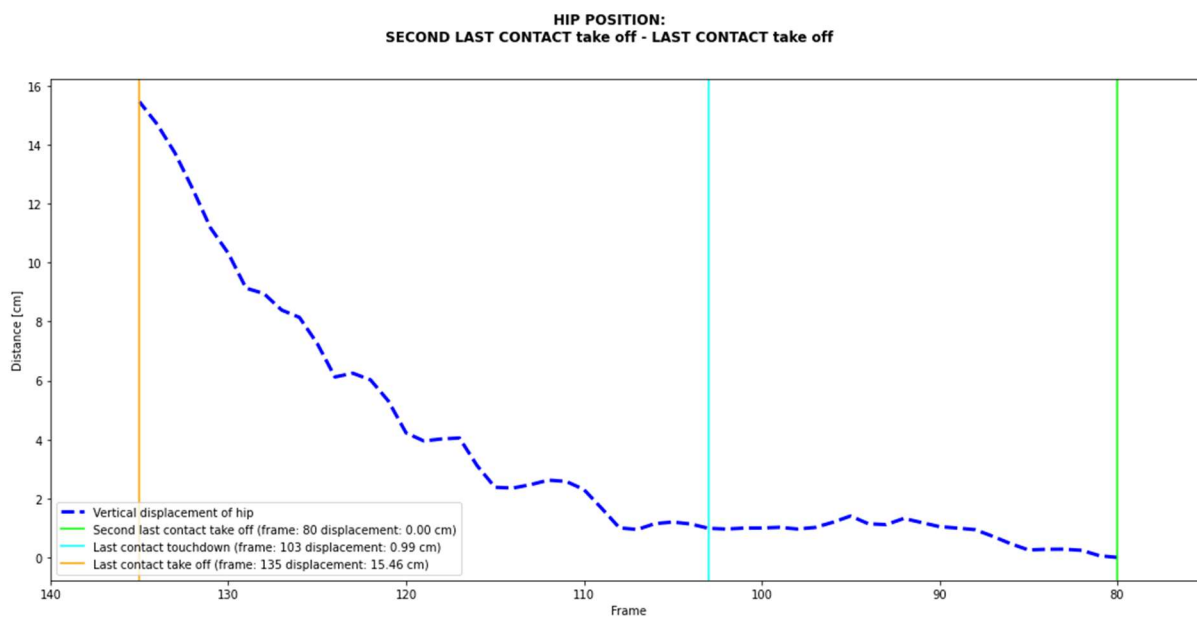


Kuva 38. Toiseksi viimeisen ja viimeisen kontaktikohtat yhdistettynä samaan kuvaan, alemmassa kuvassa pikselien valinta kuvasta (sininen ruudukko). Kuva: Kalle Lehtonen

Toiseksi viimeisen kontaktin varpaan ja viimeisen kontaktin kantapään etäisyyden eli viimeisen askeleen pituuden laskemiseksi kuvien 36 ja 37 alemmat kuvat yhdistettiin toisiinsa kuvaksi 38, josta valittiin varpaasta varpaaseen (sininen laatikko kuvan 38 alakuvassa). Sinisen laatikon leveys vastaa pikselien määrää kontaktikohtien välissä, tämän ja yhden pikselin leveyden avulla laskettiin viimeisen askeleen pituus.

### 5.4.3 Lonkan pystysuuntainen sijainnin muutos

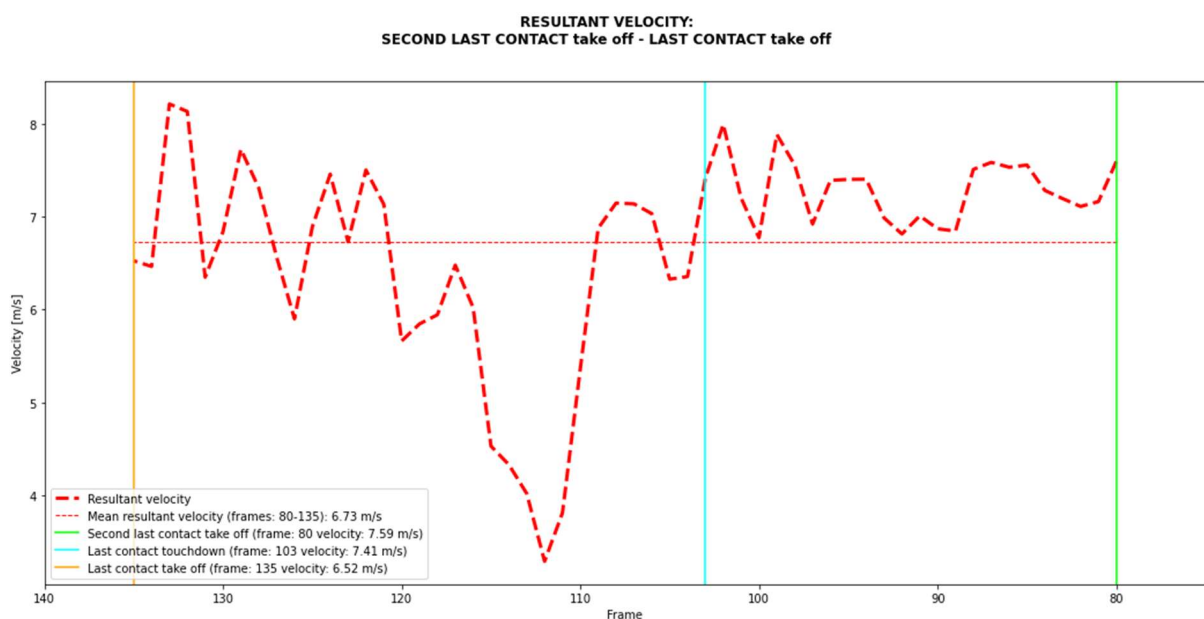
Lonkan pystysuuntainen sijainnin muutos lasketaan käyttäen vasemman ja oikean lonkan avainpisteiden y-koordinaattien keskimääräistä sijaintia. Laskennassa käytettiin skaalaamalla selvitettyjä pikselien pystysuuntaisia luonnollisia (cm/pikseli) kokoja. Sijainnin muutoksen nollakohta on toiseksi viimeisen kontaktin irtoamiskohta (kuvassa 39 vihreä pystyviiva).



Kuva 39. Lonkan pystysuuntainen sijainnin muutos toiseksi viimeisen kontaktin irtoamisen ja viimeisen kontaktin irtoamisen välillä. Kuva: Kalle Lehtonen

#### 5.4.4 Keskimääräinen nopeus viimeisen askeleen aikana

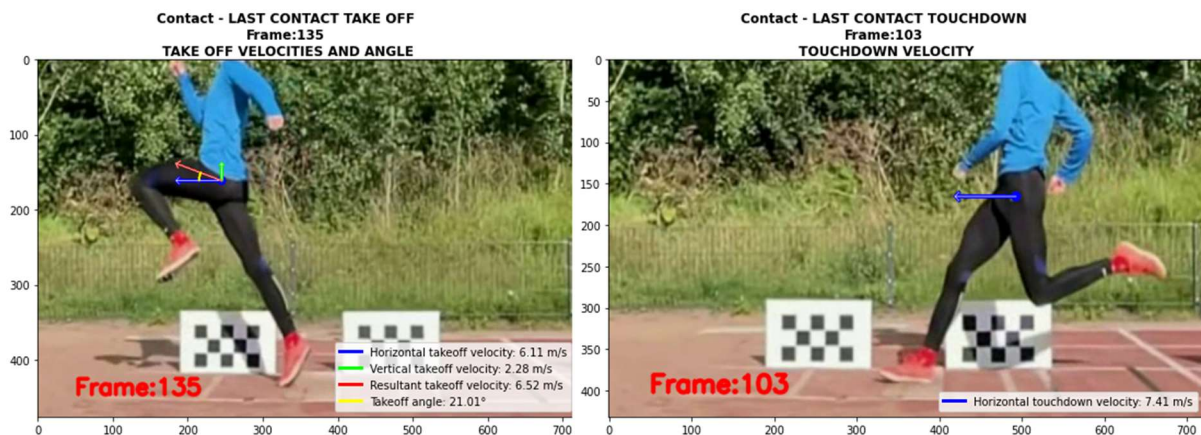
Keskimääräinen nopeus viimeisen askeleen aikana lasketaan toiseksi viimeisen kontaktin irtoamisen ja viimeisen kontaktin irtoamisen välille, laskennassa käytetään vasemman ja oikean lonkan avainpisteiden xy-koordinaattien keskimääräistä sijaintia. Vaaka- ja pystykoordinaattien sijainnin/matkan muutos laskettiin edelliseen kuvaruutuun verrattuna. Nopeuden laskennassa tarvittavan matkan laskennassa käytettiin skaalaamalla selvitettyjä pikselien vaaka- pystysuuntaisia luonnollisia (cm/pikseli) kokoja. Yhden kuvaruudun kesto laskettiin hyppyvideon kuvaa/sekunti tiedon avulla. Keskimääräisen nopeudet viimeisen askeleen aikana on esitelty kuvassa 40.



Kuva 40. Resultanttinopeus toiseksi viimeisen kontaktin irtoamisen ja viimeisen kontaktin irtoamisen välillä. Kuva: Kalle Lehtonen

### 5.4.5 Ponnistusvaiheen nopeudet

Ponnistusvaiheen vaaka- ja pystysuuntaiset nopeudet lasketaan käyttäen vasemman ja oikean lonkan avainpisteiden keskimääräistä sijaintia. Nopeuksien laskennassa tarvittavan matkan laskennassa käytettiin skaalaamalla selvitettyjä pikselien vaaka- ja pystysuuntaisia luonnollisia (cm/pikseli) kokoja. Vaaka- ja pystykoordinaattien sijainnin/matkan muutos laskettiin edelliseen kuvaruutuun verrattuna. Yhden kuvaruudun kesto laskettiin hyppyvideon kuvaa/sekunti tiedon avulla. Nopeuksien laskennan tulokset on esitelty kuvassa 41. Resultanttinopeus laskettiin vaaka- ja pystysuuntaisten nopeuksien avulla käyttämällä Pythagoraan lausetta. Ponnistuksen aikainen vaakasuuntaisen nopeuden muutos lasketaan vähentämällä ponnistusaskelen kontaktin loppumiskohdan vaakasuuntaisesta nopeudesta kontaktin alkamiskohdan vaakasuuntaisen nopeus.



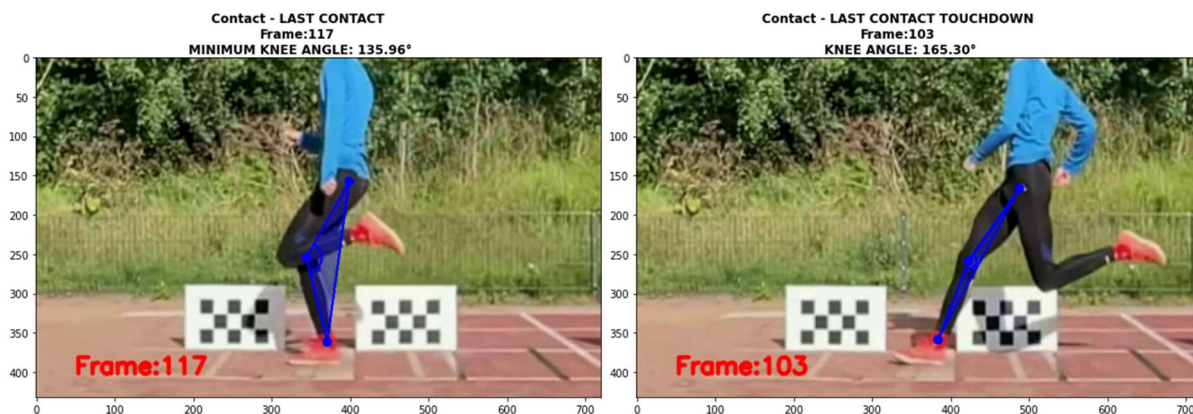
Kuva 41. Viimeisen kontaktikohdan nopeudet ja ponnistuskulma. Kuva: Kalle Lehtonen

### 5.4.6 Hyppykulma

Hyppykulma lasketaan käyttäen vasemman ja oikean lonkan avainpisteiden keskimääräistä sijaintia. Vaaka- ja pystykoordinaattien sijainnin muutos laskettiin viimeisen kontaktin irtoamiskohdan kuvaruudun ja edelliseen kuvaruudun perusteella. Hyppykulman laskennan tulos on merkitty kuvan 41 vasemmanpuoleiseen kuvaan.

### 5.4.7 Ponnistavan jalan polven kulmat

Ponnistusvaiheen kontaktista kentän pintaan lasketaan polvikulma ponnistavan jalan osuessa kentän pintaan (kuvan 42 oikeanpuoleinen kuva), sekä viimeisen kontaktin aikainen pienin polvikulma (kuvan 42 vasemmanpuoleinen kuva). Kulmat laskettiin tangenttifunktiota käyttämällä, ponnistavan jalan kantapäähän, polven ja lonkan xy-pisteistä. Polven kulman muutos laskettiin vähentämällä ponnistusvaiheen kontaktikohdan kulmasta pienin polvikulma.



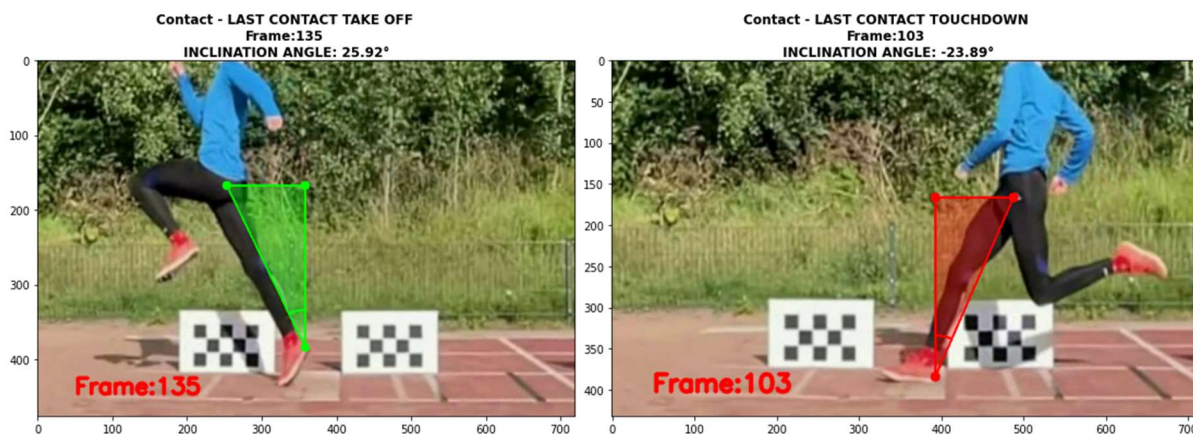
Kuva 42. Viimeisen kontaktikohdan polven minimikulma ja ensimmäisen kontaktikohdan polven kulma. Kuva: Kalle Lehtonen

#### 5.4.8 Ponnistavan jalan polvikulman muutosnopeus

Ponnistavan jalan polvikulman muutosnopeus lasketaan ponnistavan jalan polven kulman muutoksen arvon (laskettu edellisessä kappaleessa, kuva 42) ja muutokseen kuluneen ajan avulla. Ajan määrittämiseksi lasketaan, kuinka monen kuvaruudun aikana kulman muutos on tapahtunut, jonka jälkeen kuvaruutujen määrä kerrotaan yhden kuvaruudunkuvaruudun kestolla (laskettu hyppyvideon kuvaa/sekunti tiedon avulla).

#### 5.4.9 Lonkan kaltevuuskulmat (kantapää - lonkan)

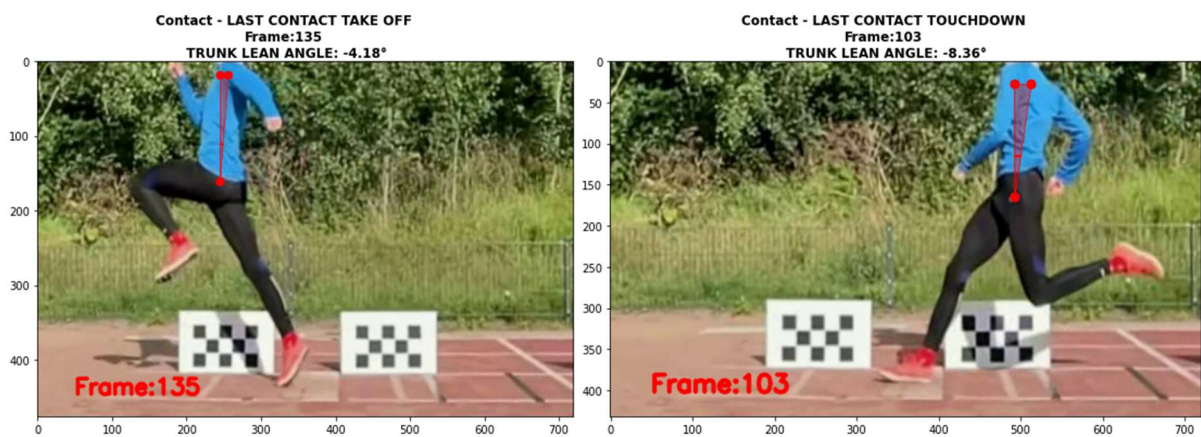
Lonkan kaltevuuskulma lasketaan ponnistusvaiheessa ponnistavan jalan osuessa kentän pintaan (kuvan 43 oikeanpuoleinen kuva), sekä ponnistavan jalan irrotessa kentän pinnasta (kuvan 43 vasemmanpuoleinen kuva). Kaltevuuskulmat laskettiin tangenttifunktiota käyttämällä, ponnistavan jalan kantapään ja lonkan avainpisteiden xy-koordinaattien sijaintien avulla. Mikäli kulma on negatiivinen, niin lonkan xy-koordinaatti on kontaktikohdan takana. Mikäli kulma on positiivinen, niin lonkan xy-koordinaatti on kontaktikohdan edessä.



Kuva 43. Viimeisen kontaktikohdan kantapään ja lonkan muodostamat kaltevuuskulmat. Kuva: Kalle Lehtonen

#### 5.4.10 Kehon kaltevuuskulmat (lonkka - olkapää)

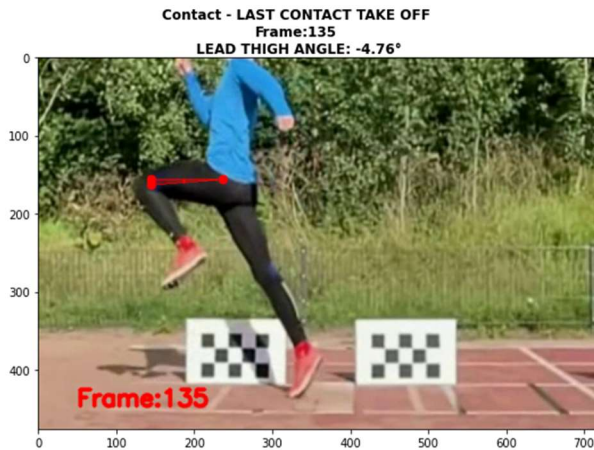
Kehon kaltevuuskulma lasketaan ponnistusvaiheessa ponnistavan jalan osuessa kentän pintaan (kuvan 44 oikeanpuoleinen kuva), sekä ponnistavan jalan irrotessa kentän pinnasta (kuvan 44 vasemmanpuoleinen kuva). Kehon kaltevuuskulmien laskemiseksi vasemman ja oikean puolen lonkan xy-koordinaateista laskettiin keskiarvo, myös hartioiden xy-koordinaateista laskettiin keskiarvo. Keskiarvoa käyttämällä laskettiin tangenttifunktiolla kaltevuuskulma. Mikäli kulma on negatiivinen, niin hartioiden xy-koordinaatti on lonkan xy-koordinaatin takana. Mikäli kulma on positiivinen, niin hartioiden xy-koordinaatti on lonkan xy-koordinaatin edessä.



Kuva 44. Viimeisen kontaktikohdan kehon kaltevuuskulmat. Kuva: Kalle Lehtonen

#### 5.4.11 Heilahtavan jalan kulma (lonkka - polvi)

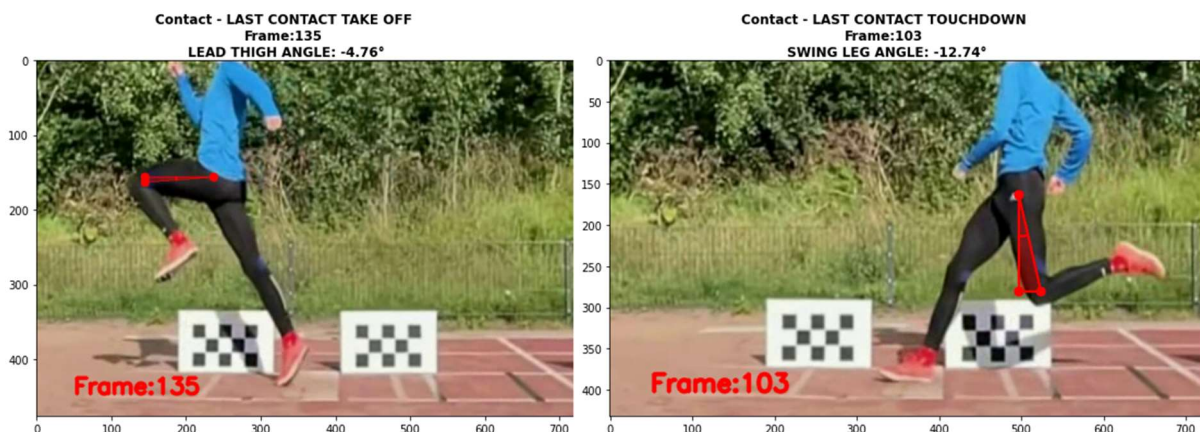
Heilahtavan jalan kulma lasketaan ponnistusvaiheessa ponnistavan jalan irrotessa kentän pinnasta (kuva 45). Heilahtavan jalan kulma laskettiin tangenttifunktiota käyttämällä heilahtavan jalan polven ja lonkan xy-pisteistä. Mikäli kulma on negatiivinen, niin polvi horisontaalisen linjan alapuolella. Mikäli kulma on positiivinen, niin polvi on horisontaalisen linjan yläpuolella.



Kuva 45. Heilahtavan kulma. Kuva: Kalle Lehtonen

#### 5.4.12 Heilahtavan jalan keskimääräinen kulmanopeus (astetta/sekunti)

Heilahtavan jalan keskimääräinen kulmanopeus laskentaa varten lasketaan ponnistusvaiheessa (ponnistavan jalan osuessa kentän pintaan) heilahtavan jalan polvenkulma (kuvan 46 oikeanpuoleinen kuva), lisäksi laskennassa käytetään edellisessä kappaleessa laskettua heilahtavan jalan kulmaa (kuva 45 ja kuvan 46 vasemmanpuoleinen kuva). Keskimääräisen kulmanopeuden laskemiseksi lasketaan myös, kuinka monessa kuvaruudussa jalka on ollut kontaktissa hyppyalustaan ja tämä määrä kerrotaan yhden kuvaruudun kestolla (laskettu hyppyvideon kuvaa/sekunti tiedon avulla). Heilahtavan jalan keskimääräisen kulmanopeuden laskemiseksi heilahtavan jalan kulman muutos (viimeisen askeleen kontakti alkaa ja loppuu) jaetaan kontaktiajalla.



Kuva 46. Heilahtavan jalan kulmat. Kuva: Kalle Lehtonen

### 5.5 Vaihe 4 – tulosten kokoaminen

Hyppyjen analysointivaiheessa saadut tulokset ja tiedot kootaan yhteen tässä vaiheessa ja tallennetaan. Jokaisesta hypystä lasketut tiedot tallennetaan taulukkoon ja kontaktikohdista

tallennetaan hyppyvideolta kuvakaappaukset. Liitteestä 2 löytyy esimerkki yhdestä hypystä tallennettavasta taulukosta, yhden hyppääjän kaikista hypyistä tallennetut tiedot on puolestaan koottu liitteisiin 3 ja 4. Lisäksi kansioon tallennetaan esimerkiksi kuvia avainpisteiden sijainneista hypyn kontaktikohdissa ja lonkan pystysuuntaisesta sijainnista hypyn aikana.

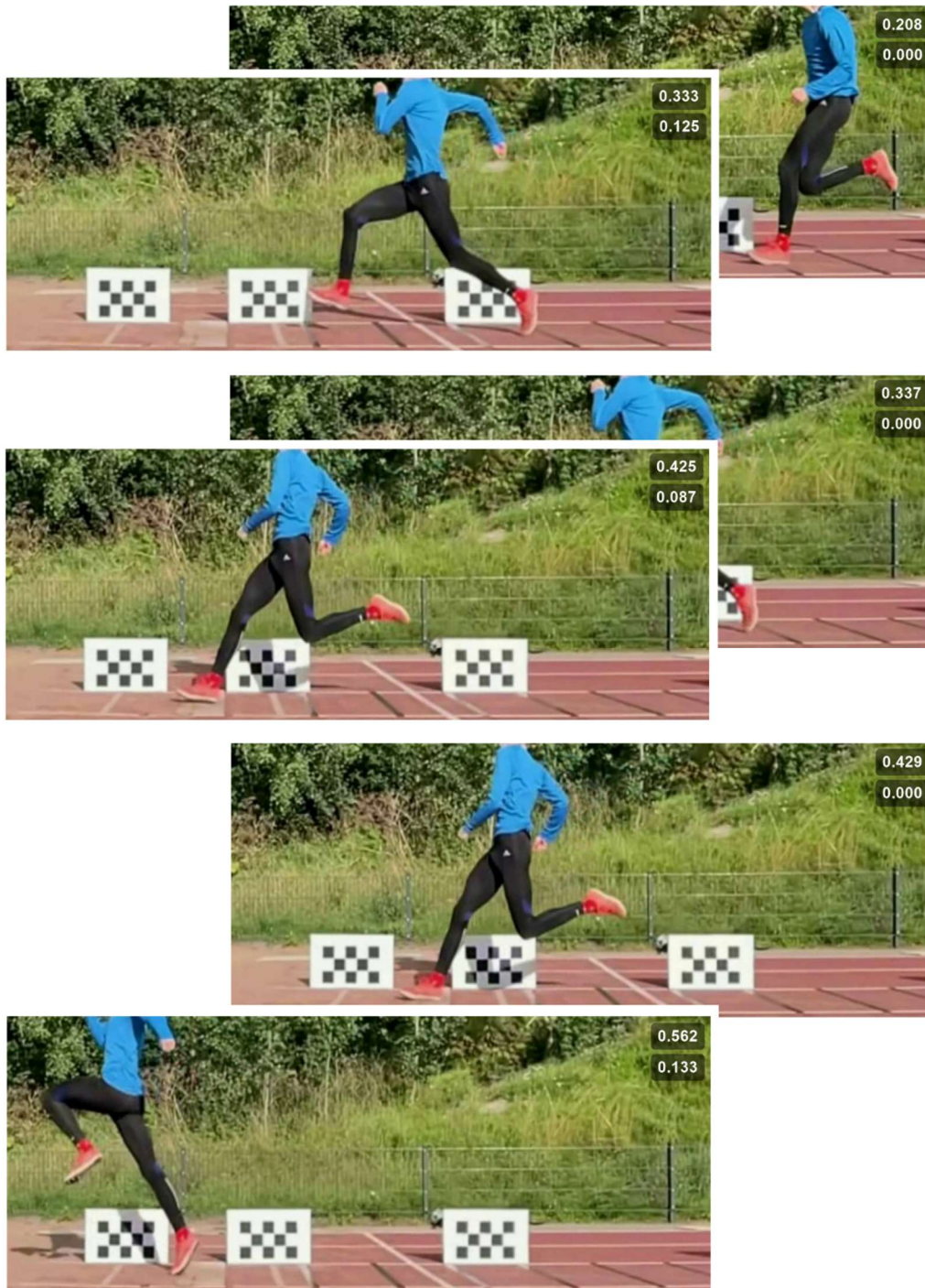
## 6 TULOKSET JA NIIDEN TARKASTELU

Kansainväliselle yleisurheiluliitolle tehdyssä vuoden 2018 hallimaailmanmestaruuskisoissa tehdyssä pituushypyn biomekaniikkaraportissa [2] analysoitiin kisaan osallistuneiden 15 hyppääjän parasta suoritusta. Kansainväliselle yleisurheiluliitolle tehtyä raporttia käytettiin tässä työssä määritettyjen arvojen luotettavuuden arviointiin ja vertailemiseen, koska vastaavaa raporttia ei ole julkaistu kansallisen tason tai tätä alemman tason hyppääjistä. Vertailtaessa tässä työssä määritettyjä arvoja kansainvälisen yleisurheiluliiton raportin arvoihin, on otettava huomioon hyppääjien taitotaso (kansainvälinen kärki, kansallinen taso ja 16...17-vuotias hyppääjä).

Edeltävän toteutusta käsittelevän luvun analysointivaiheen (vaihe 3) kappaleessa luodut kuvat, ovat kaikki samalta hyppyvideolta. Tästä hyppyvideosta analysointivaiheissa lasketut arvot on koottu taulukkoon, joka löytyy liitteestä 2 (Hypystä tallennetut tiedot). Molempien hyppääjien hyppykohtaiset analysointivaiheissa lasketut arvot on koottu erillisiin liitteisiin 3 (Hypyistä tallennetut tiedot (hyppääjä noin 16...17-vuotias)) ja 4 (Hypyistä tallennetut tiedot (kansallisen tason top-10)). Liitteisiin 3 ja 4 on lisätty raportista mainitut minimi-, maksimi- ja keskiarvot.

Tulosten tarkastelussa käytettiin apuna urheilusuoritusten analysointiin suunniteltua Kinovea-sovellusta, jonka avulla voidaan videon kuvaruuduista esimerkiksi määrittää aikoja, kulmia ja etäisyyksiä. [83] Kulmien määrittäminen Kinovea-sovelluksella on varsin tarkkaa puuhaa ja pienillä muutoksilla kulma saadaan muuttumaan useita asteita. Kinovea-sovelluksella tarkastellaan osaa analysointivaiheessa (vaihe 3) lasketuista arvoista.

## 6.1 Kontakti- ja lentoajat



Kuva 47. Kinovea-sovelluksen avulla lasketut kontakti- ja lentoajat, ylimmässä kuvaparissa on toiseksi viimeinen kontakti, keskimmäisessä on ilmalento ja alimmassa on viimeinen kontakti. Kuva: Kalle Lehtonen

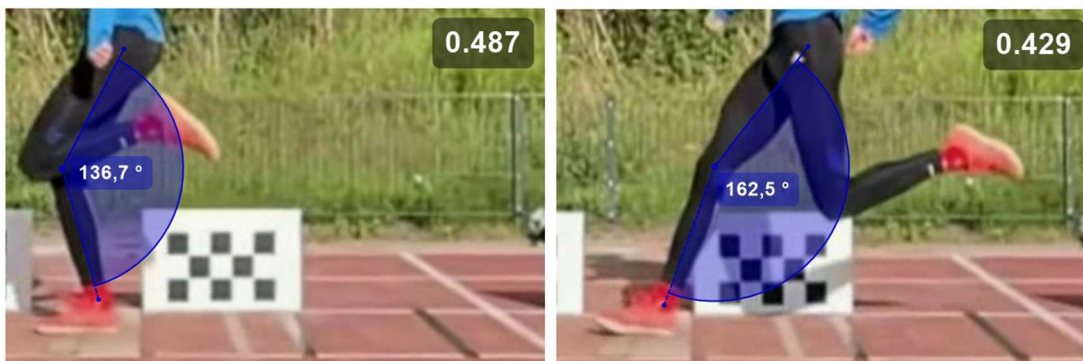
Kontakti- ja lentoaikojen analysoimiseksi etsittiin Kinovea-sovelluksella hyppyvideolta kuvaruudut, joissa toiseksi viimeinen ja viimeinen kontaktit alkavat ja loppuvat (kuva 47). Kuvan 47 kuvaparien yläkulmassa on kaksi kelloa, joista ylempi on videon alusta kulunut aika ja alempi on kontakti- tai lentoaikaan liittyvä ajastin. Tässä työssä lasketut kontakti- ja lentoajat

(kontaktiaika: 0,1251 s, lentoaika: 0,0876 s ja kontaktiaika:0,1377 s) vastaavat hyvin Kinovea-sovelluksella määritettyjä (0,125 s, 0,087 s ja 0,133 s) aikoja.

IAAF:n raportissa [2] ponnistusta edeltävän (toiseksi viimeisen) kontaktin kontaktiaikojen keskiarvo oli 0,121 s (vaihteluväli: 0,084...0,128 s), tähän verrattuna nuoremman hyppääjän (Liite 3) 0,146...0,150 s ja kansallisen tason hyppääjän (Liite 4) 0,125...0,133 s kontaktiajat olivat samassa suuruusluokassa (nuoremmalla hyppääjällä kontaktiajat olivat hieman pidempiä).

IAAF:n raportissa viimeisen vauhdinottoaskeleen lentoaikojen keskiarvo oli 0,074 s (vaihteluväli: 0,060...0,084 s), tähän verrattuna nuoremman hyppääjän (Liite 3) 0,075...0,167 s ja kansallisen tason hyppääjän (Liite 4) 0,083...0,100 s kontaktiajat olivat samassa suuruusluokassa (nuoremmalla hyppääjällä kontaktiajat olivat hieman pidempiä). Ponnistuksen (viimeisen kontaktin) kontaktiaikaa ei ole ilmoitettu IAAF:n raportissa, joten arvoja ei voida vertailla tämän työn tuloksiin.

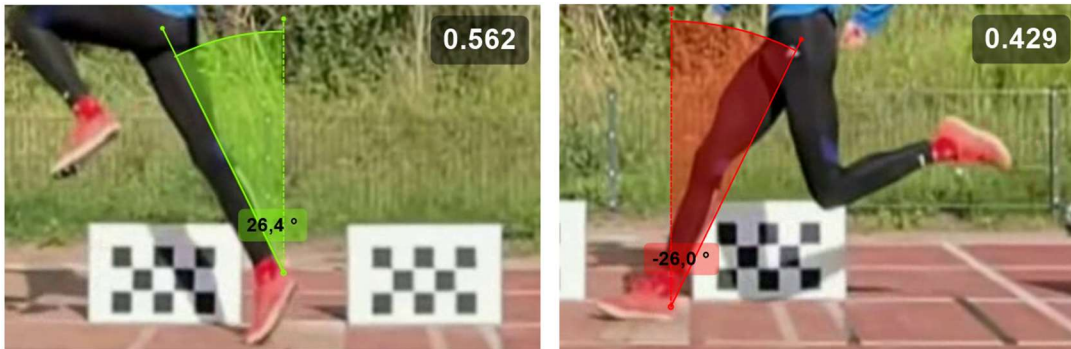
## 6.2 Kulmat



Kuva 48. Kinovea-sovelluksen avulla arvioidut ponnistavan jalan kulmat, vasemmanpuoleisessa kuvassa viimeisen kontaktin minimi polvikulma ja oikeanpuoleisessa kuvassa viimeisen kontaktin alkamiskohdan polvikulma. Kuvien yläkulmissa on videon alusta kulunut aika. Kuva: Kalle Lehtonen

Ponnistavan jalan polven kulmien määrittämiseksi Kinovea-sovelluksella etsittiin hyppyvideolta vastaavat kuvaruudut, joista oli määritetty vastaavat kulmat. Kuvan 48 molempiin kuviin on asetettu käsin lonkan, polven ja nilkan sijainnit. Kinovea-sovelluksella määritetyt viimeisen kontaktin alkamiskohdan polvikulma (kuvan 48 oikeanpuoleinen kuva) 162,5° ja viimeisen kontaktin minimi polvikulma (kuvan 48 vasemmanpuoleinen kuva) 136,7°, vastaavat hyvin tässä työssä määritettyjä kulmia 165,30° ja 135,96° (vastaavat kuvaruudut kuvassa 42).

IAAF:n raportissa [2] viimeisen kontaktin alkamiskohdan polvikulmien keskiarvo oli  $165,6^\circ$  (vaihteluväli:  $160,1\dots172,2^\circ$ ), tähän verrattuna nuoremman hyppääjän (Liite 3)  $162,9\dots173,5^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $165,3\dots169,1^\circ$  polvikulmat olivat samassa suuruusluokassa. IAAF:n raportissa viimeisen kontaktin minimi polvikulmien keskiarvo oli  $137,4^\circ$  (vaihteluväli:  $120,5\dots150,8^\circ$ ), tähän verrattuna nuoremman hyppääjän (Liite 3)  $123,1\dots143,7^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $136,0\dots142,2^\circ$  minimi polvikulmat olivat samassa suuruusluokassa.



Kuva 49. Kinovea-sovelluksen avulla arvioidut kehon kaltevuuskulmat, viimeisen kontaktin loppumiskohta vasemmalla ja alkamiskohta oikealla. Kuva: Kalle Lehtonen

Lonkan kaltevuuskulmien määrittämiseksi Kinovea-sovelluksella etsittiin hyppyvideolta vastaavat kuvaruudut, joista oli määritetty vastaavat kulmat. Kuvan 49 molempiin kuviin on asetettu käsin lonkan, kantapään ja kolmannen pisteen (tarvitaan kulmaan määrittämisessä) sijainnit. Kinovea-sovelluksella määritetyt kehon kaltevuuskulmat viimeisen kontaktin alkamiskohdassa (kuvan 49 oikeanpuoleinen kuva)  $-26,0^\circ$  ja viimeisen kontaktin loppumiskohdassa (kuvan 49 vasemmanpuoleinen kuva)  $26,4^\circ$ , vastaavat hyvin tässä työssä määritettyjä kulmia  $-23,89^\circ$  ja  $25,92^\circ$  (vastaavat kuvaruudut kuvassa 43).

IAAF:n raportissa viimeisen kontaktin alkamiskohdan lonkan kaltevuuskulmien keskiarvo oli  $-34,7^\circ$  (vaihteluväli:  $-32,2\dots-39,3^\circ$ ). Kulmat on määritetty negatiivisiksi, koska lonkka on tukipisteen takana. IAAF:n raportin tuloksiin verrattuna nuoremman hyppääjän (Liite 3)  $-20,2\dots-30,3^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $-23,8\dots-27,2^\circ$  lonkan kaltevuuskulmat olivat suurempia. IAAF:n raportissa viimeisen kontaktin loppumiskohdan lonkan kaltevuuskulmien keskiarvo oli  $18,0^\circ$  (vaihteluväli:  $13,5\dots22,7^\circ$ ), tähän verrattuna nuoremman

hyppääjän (Liite 3)  $24,1...27,2^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $22,6...25,9^\circ$  kaltevuuskulmat olivat suurempia.



Kuva 50. Kinovea-sovelluksen avulla arvioitu heilahtavan jalan kulma viimeisen kontaktin alkamiskohdassa. Kuva: Kalle Lehtonen

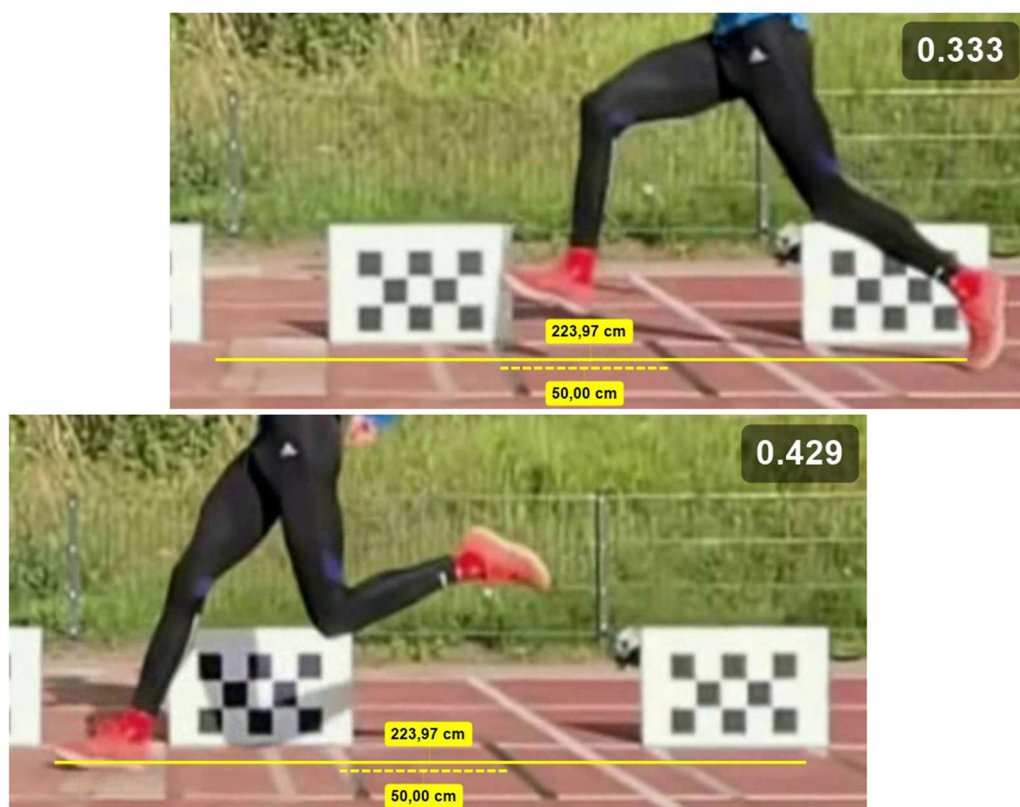
Heilahtavan jalan kulman määrittämiseksi viimeisen kontaktin alkamiskohdassa etsittiin Kinovea-sovelluksella hyppyvideolta vastaavat kuvaruudut, joista oli määritetty vastaava kulma. Kuvaan 50 on asetettu käsin lonkan, polven ja kolmannen pisteen (tarvitaan kulmaan määrittämisessä). Kinovea-sovelluksella määritetty heilahtavan jalan kulma viimeisen kontaktin alkamiskohdassa (kuva 50)  $-10,1^\circ$ , vastaavat hyvin tässä työssä laskettua kulmaa  $-12,74^\circ$  (vastaava kuvaruutu kuvan 46 oikeanpuoleisessa kuvassa). Tätä pienempiä kulmia ei Kinovea-sovelluksella pyritty tässä työssä määrittämään (esimerkiksi kehon kaltevuuskulmat ja heilahtavan jalan kulma). Heilahtavan jalan kulmia viimeisen kontaktin alkamiskohdassa ei ole ilmoitettu IAAF:n raportissa, joten arvoja ei voida vertailla tämän työn tuloksiin. IAAF:n raportissa viimeisen kontaktin loppumiskohdassa heilahtavan jalan kulmien keskiarvo oli  $-16,2^\circ$  (vaihteluväli:  $-33,3...-7,5^\circ$ ). IAAF:n raportin tuloksiin verrattuna nuoremman hyppääjän (Liite 3)  $-38,9...-13,3^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $-7,1...-0,7^\circ$  heilahtavan jalan kulmat viimeisen kontaktin loppumiskohdassa olivat samassa suuruusluokassa.

IAAF:n raportissa viimeisen kontaktin alkamiskohdan kehon kaltevuuskulmien keskiarvo oli  $-5,6^\circ$  (vaihteluväli:  $-16,3...-0,7^\circ$ ). Kulmat on määritetty negatiivisiksi, koska hyppääjän yläkeho on kallistunut taaksepäin. IAAF:n raportin tuloksiin verrattuna nuoremman hyppääjän (Liite 3)  $-15,8...-7,2^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $-8,4...-5,6^\circ$  kehon kaltevuuskulmat olivat samassa suuruusluokassa. IAAF:n raportissa viimeisen kontaktin loppumiskohdan kehon kaltevuuskulmien keskiarvo oli  $0,8^\circ$  (vaihteluväli:  $-9,4...10,4^\circ$ ), tähän verrattuna nuoremman

hyppääjän (Liite 3)  $-9,5\dots-5,6^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $-6,1\dots-3,8^\circ$  kehon kaltevuuskulmat olivat samassa suuruusluokassa.

Hyppykulman laskenta tehtiin lonkan avainpisteiden keskimääräistä sijaintia käyttäen. Vaaka- ja pystykoordinaattien sijainnin muutos laskettiin viimeisen kontaktin irtoamiskohdan kuvaruudun ja edelliseen kuvaruudun perusteella. IAAF:n raportissa hyppykulmien keskiarvo oli  $22,7^\circ$  (vaihteluväli:  $19,4\dots26,0^\circ$ ), tähän verrattuna nuoremman hyppääjän (Liite 3)  $12,9\dots22,4^\circ$  ja kansallisen tason hyppääjän (Liite 4)  $16,4\dots21,9^\circ$  hyppykulmat olivat samaa luokkaa.

### 6.3 Viimeisen askeleen pituus



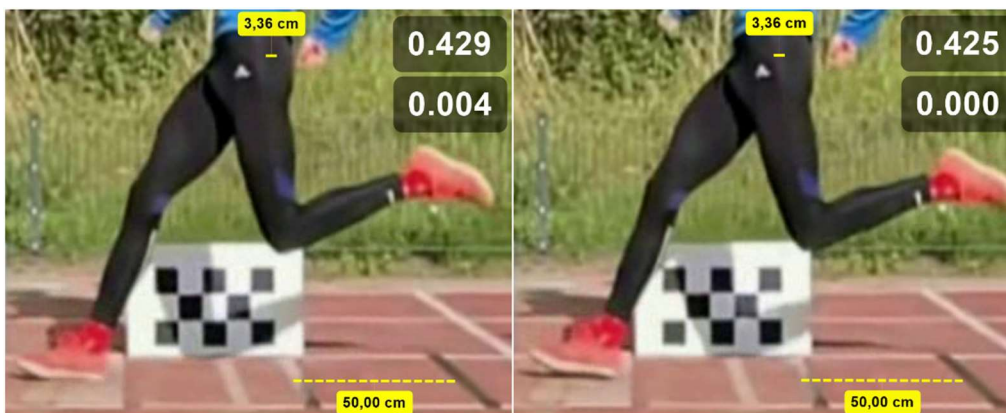
Kuva 51. Kinovea-sovelluksen avulla arvioitu askeleen pituus. Kuva: Kalle Lehtonen

Viimeisen askeleen pituuden määrittämiseksi Kinovea-sovelluksella etsittiin hyppyvideolta vastaavat kuvaruudut, joista oli määritetty viimeisen askeleen pituus. IAAF:n raportissa [2] askeleen pituus oli määritetty kontaktien välille varpaasta varpaaseen. Kuvan 51 ylemmässä kuvassa on toiseksi viimeisen kontaktin irtoamiskohta ja alemmassa kuvassa on viimeisen kontaktin alkamiskohta. Kuvaruutuihin Kinovea-sovelluksella käsin piirretyt viivat ovat molemmissa kuvissa samoja. Katkonainen viiva on piirretty teippausten väliin (50 cm), tämän viivan pituus määrittää skaalan, jonka perusteella Kinovea-sovellus laskee yhtenäisen viivan

pituuden. Kinovea-sovelluksella määritetty viimeisen askeleen pituus on noin 224 cm, vastaa hyvin tässä työssä laskettua askeleen pituutta 2,19 m. Kuvan 51 alemmasta kuvasta näkee, että hyppääjän varvas ei aivan vielä kosketa radanpintaa, tämän seurauksena työssä tehdyllä sovelluksella laskettu askeleen pituus jää hieman liian lyhyeksi.

IAAF:n raportissa viimeisen askeleen keskiarvo oli 2,18 m (vaihteluväli: 2,03...2,36 m), tähän verrattuna nuoremman hyppääjän (Liite 3) 2,06...2,61 m ja kansallisen tason hyppääjän (Liite 4) 2,19...2,33 m askeleen pituudet olivat samaa luokkaa.

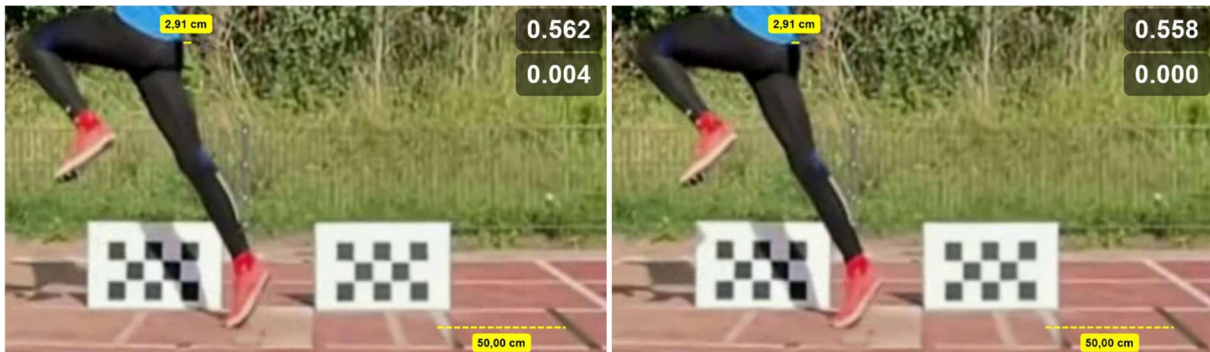
#### 6.4 Nopeudet



Kuva 52. Kinovea-sovelluksen avulla peräkkäisistä kuvaruuduista määritetty kuljettu matka ja aika.  
Kuva: Kalle Lehtonen

Työssä tehdyssä sovelluksessa nopeuksien laskennassa käytettiin kahden peräkkäisen kuvaruudun lonkan avainpisteiden sijainnin keskiarvoa. Kinovea-sovellusta käyttämällä voi määrittää nopeudelle arvon käyttämällä kuljettua matkaa ja aikaa. Kuvan 52 vasemmanpuoleisessa kuvassa viimeisen kontaktin alkamiskohta ja oikeanpuoleisessa kuvassa on tätä edeltävä kuvaruutu. Kuvaruutuihin Kinovea-sovelluksella käsin piirretyt viivat ovat molemmissa kuvissa samoja. Katkonainen viiva on piirretty teippausten väliin (50 cm), viivan pituus määrittää skaalan, jonka perusteella Kinovea-sovellus laskee yhtenäisen viivan pituuden (3,36 cm). Oikeanpuoleisessa kuvassa yhtenäisen viivan oikeanpuoleinen pää on asetettu suurin piirtein lonkan kohdalle, jossa hyppääjän housuissa on vaalean ja tumman osan raja. Vasemmanpuoleisessa kuvassa yhtenäisen viivan vasemmanpuoleinen pää on asetettu hyppääjän housuissa olevan vaalean ja tumman osan rajaan. Vaalean ja tumman osan raja liikkuu kuvaruutujen välillä 3,36 cm ja aikaa tähän kuluu 0,004 s. Edellisten perusteella laskettu vaakasuuntainen nopeus on 8,4 m/s, työssä laskettu vastaava nopeus oli 7,41 m/s, joten nopeuksien välillä on varsin suuri ero.

IAAF:n raportissa [2] viimeisen kontaktin alkamiskohdan vaakasuuntaisten nopeuksien keskiarvo oli 10,08 m/s (vaihteluväli: 9,68...10,43 m/s). IAAF:n raportissa ei ole ilmoitettu viimeisen kontaktin alkamiskohdan vaakasuuntaista nopeutta, joten ne on laskettu viimeisen kontaktin irtoamiskohdan nopeuden ja kontaktin aikaisen nopeuden muutoksen arvojen avulla. IAAF raporttiin verrattuna nuoremman hyppääjän (Liite 3) 5,92...7,16 m/s ja kansallisen tason hyppääjän (Liite 4) 6,58...7,52 m/s viimeisen kontaktin alkamiskohdan vaakasuuntaisten nopeuksien arvot olivat pienempiä.



Kuva 53. Kinovea-sovelluksen avulla peräkkäisistä kuvaruuduista määritetty kuljettu matka ja aika.  
Kuva: Kalle Lehtonen

Kuvan 53 vasemmanpuoleisessa kuvassa on viimeisen kontaktin päättymiskohta ja oikeanpuoleisessa kuvassa on tätä edeltävä kuvaruutu. Kuvista ei pysty määrittämään lonkan sijaintia niin, että se voitaisiin valita molemmista kuvista. Tämän vuoksi yhtenäisen viivan päät on piirretty Kinovea-sovelluksella lähelle lonkkaa (takapuoleen), josta voidaan määrittää kuvaruutujen välillä kuljettu matka. Kuvaruutujen välillä tapahtuvan liikkeen pituus on 2,91 cm ja aikaa tähän kuluu 0,004 s. Edellisten perusteella laskettu vaakasuuntainen nopeus on 7,3 m/s, työssä laskettu vastaava nopeus oli 6,52 m/s, nopeuksien suuruuksien välillä on suuri ero.

IAAF:n raportissa viimeisen kontaktin loppumiskohdan vaakasuuntaisten nopeuksien keskiarvo oli 8,67 m/s (vaihteluväli: 8,10...9,12 m/s). IAAF:n raporttiin verrattuna nuoremman hyppääjän (Liite 3) 5,91...6,43 m/s ja kansallisen tason hyppääjän (Liite 4) 4,50...6,87 m/s viimeisen kontaktin loppumiskohdan vaakasuuntaisten nopeuksien suuruudet olivat pienempiä. IAAF:n raportissa keskimääräinen nopeus viimeisen askeleen aikana oli 9,54 m/s (vaihteluväli: 9,23...9,97 m/s). IAAF:n raporttiin verrattuna nuoremman hyppääjän (Liite 3) 5,89...6,14 m/s ja kansallisen tason hyppääjän (Liite 4) 6,65...6,99 m/s viimeisen askeleen keskimääräisten nopeuksien arvot olivat pienempiä. IAAF:n raportissa viimeisen kontaktin alkamis- ja loppumiskohdan vaakasuuntaisten nopeuksien eron keskiarvo oli -1,41 m/s (vaihteluväli: -1,62...-0,96 m/s). IAAF:n raporttiin verrattuna nuoremman hyppääjän (Liite 3) -1,25...0,18

m/s ja kansallisen tason hyppääjän (Liite 4)  $-2,08\dots-0,62$  m/s viimeisen kontaktin vaakasuuntaisten nopeuksien suuruudet olivat samaa luokkaa.

## 6.5 Muut suureet

Kontakti- ja lentoaikojen, erilaisten kulmien ja nopeuksien, sekä viimeisen askeleen pituuden lisäksi, IAAF:n raportissa [2] määritettiin hyppääjän massakeskipisteen alenema ja kulman muutosnopeuksia. Massakeskipisteen aleneman sijasta, tässä työssä pyrittiin määrittämään vastaava tulos käyttämällä hyppääjän lonkan avainpisteiden sijaintien keskiarvoa. IAAF:n raportissa massakeskipisteen aleneman keskiarvo oli  $-3,0$  cm (vaihteluväli:  $-1,0\dots-8$  cm). IAAF:n raportissa nollakohdaksi on asetettu toiseksi viimeisen kontaktin irtoamiskohta, arvot ovat negatiivisia, koska hyppääjän kehon massakeskipiste on laskenut nollakohdan alle (ennen viimeisen kontaktin päättymistä). Hyppääjän lonkan avainpisteiden sijaintien keskiarvoa käyttämällä, lonkan alenemat nuoremmalla hyppääjällä (Liite 3) vaihtelivat välillä  $0,02\dots2,67$  cm ja kansallisen tason hyppääjällä (Liite 4)  $0,17\dots2,22$  cm. IAAF:n raporttiin verrattuna kummankaan hyppääjän lantion alenema ei ollut negatiivinen.

Viimeisen kontaktin kulman muutosnopeudet laskettiin ponnistavalle ja heilahtavalle jalalle. Näissä laskennoissa käytettiin muiden suureiden laskennoissa käytettyjä kulmia, muutokseen kulunut aika puolestaan laskettiin kuvaruutujen määrän ja keston perusteella. IAAF:n raportissa ponnistavan jalan polvikulman muutosnopeuden keskiarvo oli  $-478$  °/s (vaihteluväli:  $-669\dots-412$  °/s), tähän verrattuna nuoremman hyppääjän (Liite 3)  $-510\dots-448$  °/s ja kansallisen tason hyppääjän (Liite 4)  $-502\dots-436$  °/s polvikulman muutosnopeuksien suuruudet olivat samaa luokkaa. IAAF:n raportissa heilahtavan jalan polvikulman muutosnopeuden keskiarvo oli  $530$  °/s (vaihteluväli:  $363\dots704$  °/s), tähän verrattuna nuoremman hyppääjän (Liite 3)  $271\dots461$  °/s ja kansallisen tason hyppääjän (Liite 4)  $501\dots594$  °/s polvikulman muutosnopeuksien suuruudet olivat samaa luokkaa.

## 7 MENETELMIEN ARVIOINTI

Tämän luvun kappaleissa arvioidaan työssä käytettyjä menetelmiä, joita käytettiin työssä tehdyissä sovelluksissa. Kinovea-sovellusta käytettiin työssä määritettyjen suureiden arviointiin (sovelluksella ei ollut mahdollista arvioida kaikkia suureita), samansuuruisiin tuloksiin päästiin kontakti- ja lentoaikojen, sekä ponnistusvaiheen kulmien ja viimeisen askeleen pituuden suureiden määrittämisessä. Kinovea-sovelluksen avulla arvioidut vaakasuuntaisten nopeuksien arvot, olivat isompia, kuin tässä työssä tehdyllä sovelluksella määritetyt nopeudet.

Työssä tehdyllä järjestelmällä määritettyjen suureiden arvot olivat vastaavan suuruisia IAAF:n raportin [2] kanssa seuraavilla suureilla: kontakti- ja lentoajat, kulmat (ponnistavan/heilahtavan jalan kulmat, kehon kaltevuuskulmat ja hyppykulma), viimeisen askeleen pituus, ponnistavan jalan polvikulman muutosnopeus ja heilahtavan jalan keskimääräinen kulmanopeus. Työssä tehdyllä sovelluksella määritetyt lonkan kaltevuuskulmat ja -nopeudet, olivat suuruudeltaan pienempiä IAAF:n raporttiin verrattuna. IAAF:n raportissa käytettiin kehon massakeskipistettä pystysuuntaisen aleneman laskennassa. Työssä tehdyssä sovelluksessa pystysuuntaisen aleneman laskennassa käytettiin lonkan avainpisteitä, tällä menetelmällä ei saatu IAAF:n raporttiin verrattuna vastaavia tuloksia. Hypyistä määritetyt nopeudet jäivät IAAF:n raportin tuloksia pienemmäksi, tässä on kuitenkin huomioitava hyppääjien taso. Työssä määritettyjä nopeuksia ei pystytä vertaamaan todelliseen arvoon, esimerkiksi askelpituuden määrittämisen tuloksia pystyttiin vertaamaan radan pinnan teippauksiin.

### 7.1 Avainpisteiden tunnistaminen

Hyppyjen analysoinnin yhteydessä hyppyvideoilta ei tuntunut löytyvän kohtia, joissa olisi tunnistettu virheellisesti avainpiste hyppyvideon taustalta, mutta tarkemman tutkimisen jälkeen tällaisia kuvaruutuja myös löytyi. Liitteeseen 5 on koottu esimerkkikuvia MediaPipe Pose-asennontunnistusalgoritmin virheellisesti tunnistamista avainpisteistä, liitteessä olevien kuvien avainpisteiden xy-koordinaattien arvoille ei ole tehty muokkauksia.

Liitteen 5 ensimmäinen (kuva 1) kuvapari on hyppyvideon alkuosasta. Hyppyvideo toistetaan takaperin (lopusta alkuun), kuvaparin vasemmanpuoleisessa kuvassa (kuvaruutu 30) toisen jalan kantapäähän ja isovarpaan avainpisteet ovat hävinneet kuvaruudusta, asennon arviointialgoritmi kuitenkin löytää virheellisesti näiden avainpisteiden sijainnit aivan kuvaruudun reunasta. Kun koko jalka on hävinnyt näkyvistä (kuvaruutu 16), niin asennon

arviointialgoritmi tunnistaa molempien jalkojen avainpisteet samaan jalkaan. Jostain syystä asennon arviointialgoritmi sekoittaa näissä kohdissa avainpisteiden tunnistamisen, kohdat ovat kuitenkin niin alussa hyppyvideota, että niillä ei ole merkitystä analysoinnin kannalta.

Liitteen 5 toinen (liitteen kuva 2) kuvasarja on kohdasta, jossa toiseksi viimeinen kontakti tapahtuu, takaperin toistetulla hyppyvideolla ensimmäisessä kuvaruudussa (kuvaruutu 51) molempien jalkojen avainpisteet on tunnistettu oikein kuvaruudusta. Kuvaruuduissa 50...47 toisen/molempien jalkojen nilkan, kantapään ja isovarpaan avainpisteiden sijainnit vaeltavat ja eivät ole oikeissa kohdissa. Kuvaruuduissa 46...45 tilanne palautuu jälleen normaaliksi. Toiseksi viimeisen kontaktikohdan alussa ei lasketa kulmia avainpisteiden sijainnin perusteella, joten avainpisteiden sijainneilla ei ole merkitystä suureiden laskentaan. Kontaktikohdan alkukohdan alkamisen määritetään käyttäjän toimesta, joten esimerkiksi kontaktiajan määrittäminen ei perustu avainpisteiden sijaintiin.

Liitteen 5 kolmannessa (liitteen kuva 3) kuvasarjassa ilmassa olevan jalan isovarpaan sijainti alkaa vaeltamaan ja sen sijaintia ei tunnisteta oikein useissa peräkkäisissä kuvaruuduissa. Takaperin toistetulla hyppyvideolla jo ensimmäisessä kuvaruudussa (kuvaruutu 99) isovarpaan avainpisteen sijainti ei ole silmämääräisesti tarkasteltuna oikeassa kohdassa. Näiden avainpisteiden sijaintien perusteella ei lasketa suureita.

Liitteen 5 neljännessä (liitteen kuva 4) kuvasarjassa viimeisen kontaktin aikana nilkkojen, kantapäiden ja isovarpaiden avainpisteiden sijainnit eivät ole oikeissa kohdissa. Takaperin toistetulla hyppyvideolla ensimmäisessä kuvaruudussa (kuvaruutu 117) avainpisteet ovat silmämääräisesti tarkasteltuna oikeissa kohdissa. Tämän jälkeen heilahtavan jalan isovarpaan avainpiste karkaa väärään kohtaan (kuvaruutu 116) ja tämän jälkeen kahdessa seuraavassa kuvaruudussa (kuvaruudut 115 ja 114) nilkkojen, kantapäiden ja isovarpaiden avainpisteet tunnistaan kuvaruudulta väärästä kohtaa.

Liitteen 5 viidennessä (liitteen kuva 5) kuvasarjassa ilmalennon aikana molempien jalkojen polvien, nilkkojen, kantapäiden ja isovarpaiden avainpisteiden sijainnit eivät ole oikeissa kohdissa. Takaperin toistetulla hyppyvideolla ensimmäisessä kuvaruudussa (kuvaruutu 91) avainpisteet ovat silmämääräisesti tarkasteltuna oikeissa kohdissa. Seuraavassa kuvaruudussa (kuvaruutu 90) lonkan alapuolella olevien avainpisteiden sijainnit ovat täysin väärissä kohdissa ja avainpisteitä ei tunnisteta ollenkaan seuraavasta kuvaruudusta (kuvaruutu 89). Mielestäni tämän kaltainen avainpisteiden sijainnin häviäminen oli harvinainen tapahtuma.

Liitteen 5 kuudennessa (liitteen kuva 6) kuvasarjassa viimeisen kontaktin aikana molempien jalkojen nilkkojen, kantapäiden ja isovarpaiden avainpisteiden sijainnit eivät ole oikeissa kohdissa. Takaperin toistetulla hyppyvideolla ensimmäisessä kuvaruudussa (kuvaruutu 120) avainpisteet ovat silmämääräisesti tarkasteltuna oikeissa kohdissa. Seuraavassa kuvaruudussa (kuvaruutu 119) nilkkojen, kantapään ja isovarpaan avainpisteiden sijainnit ovat täysin väärissä kohdissa, seuraavissa kuvaruuduissa avainpisteet alkavat vaeltaa/liikkua kohti oikeaa sijaintia. Lopulta, avainpisteet ovat silmämääräisesti tarkasteltuna oikeissa kohdissa kuvaruudussa 113. Kinovea-sovelluksella arvioituna kuvaruudussa (kuvaruutu 119) avainpisteiden virheellisten xy-koordinaattien perusteella polvikulma on noin  $97^\circ$ .

Liitteen 5 seitsemänteen (liitteen kuva 7) kuvaan on piirretty alkuperäisten (vasemmanpuoleinen kuva) ja Hampel-suodatettujen (oikeanpuoleinen kuva) avainpisteiden xy-koordinaattien sijainnit, suodatus muuttaa ponnistavan (oikeanpuoleisen) jalan virheellisen avainpisteen xy-koordinaattia (ympyröity kuvista vihreällä). Tämän seurauksena työssä tehdyllä sovelluksella viimeisen kontaktin pienin polvikulma on kyseiselle hypylle  $139,8^\circ$ . Suodatuksella pystytään havaitsemaan ja korjaamaan yksittäisten avainpisteiden xy-koordinaattien arvoja, mutta mikäli useita peräkkäisiä avainpisteiden xy-koordinaatteja on tunnistettu virheellisesti (ympyröity liitteen 5 kuvaan 7 oranssilla), niin niiden havaitseminen ja korjaaminen ei onnistu työssä tehdyllä sovelluksen Hampel-suotimella. Mikäli näiden avainpisteiden xy-koordinaatteja tarvittaisiin laskennoissa, niin sovelluksella lasketut tulokset olisivat virheellisiä.

## **7.2 Avainpisteiden tunnistaminen videon toistaminen**

Tässä työssä tehdyssä sovelluksessa avainpisteiden etsiminen aloitettiin hyppyvideon lopusta ja edettiin kohti videon alkua eli avainpisteiden etsiminen suoritettiin toistamalla video takaperin. Näin saatiin tunnistettua avainpistettä mahdollisimman monesta hyppyvideon alkuosan kuvaruudusta. Jälkikäteen arvioituna, takaperin suoritettulla avainpisteiden etsimisellä ei mielestäni saavutettu merkittävää etua, koska videon alkuosan kuvaruuduista ei suoritettu arvojen laskentaa. Arvojen laskenta alkaa vasta, kun hyppääjä on ensimmäisen kerran kontaktissa kentän pintaan.

## **7.3 Avainpisteiden puolten vaihtuminen**

Työssä käytetty MediaPipe Pose-asennontunnistusalgoritmi sekoitti joissain kuvaruuduissa kehon vasemman ja oikean puolen avainpisteet keskenään. Avainpisteet sekoittuivat tilanteissa,

joissa molemmat jalat olivat näkyvillä, joten avainpisteiden vaihtuminen ei johtunut okklusiosta. Työssä tehdyllä sovelluksella avainpisteiden vaihtumisen tunnistaminen ja pisteiden sijaintien vaihtaminen toimivat hyvin.

#### **7.4 Kyynärpään avainpisteet**

Työssä käytetty MediaPipe Pose-asennontunnistusalgoritmi ei tunnistanut hyppyvideon kuvaruuduista (suurimmasta osasta), hyppääjän kehon taakse jäävän kyynärpään avainpisteen xy-koordinaattien sijainteja. Kyynärpäätä ei tunnistettu algoritmilla, vaikka kyynärpää oli selkeästi näkyvillä kuvaruudulla. Kyynärpään sijainnilla ei ole tämän työn kannalta merkitystä, mutta mikäli esimerkiksi käsien asentoja olisi haluttu tarkastella tarkemmin hypyn aikana, niin se ei olisi ollut mahdollista.

#### **7.5 Hyppyvideoiden suurennos**

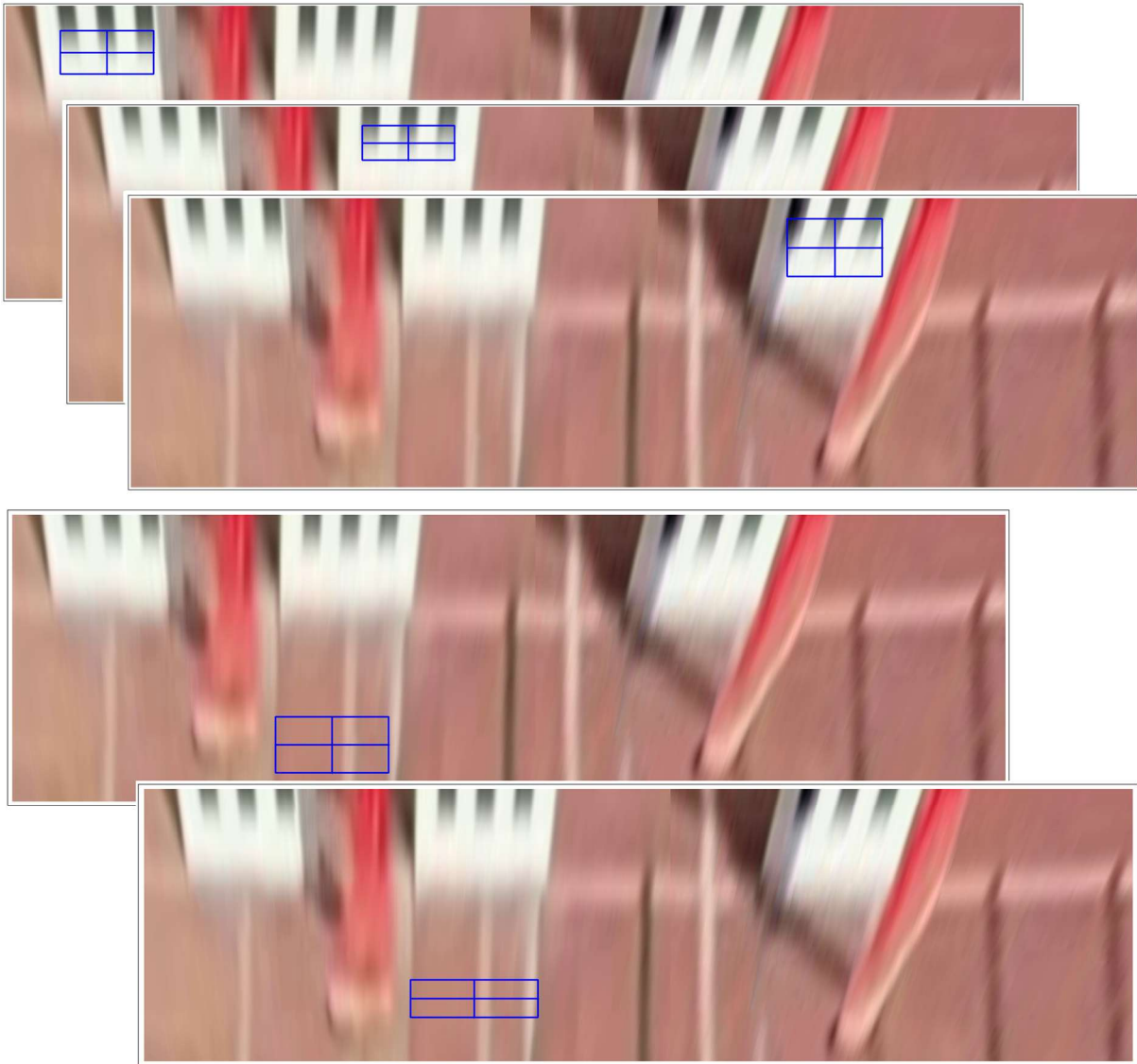
IAAF:n raportissa [2] hyppyistä arvioitiin kolmen viimeisen vauhdinottoaskeleen kontakti- ja lentoaikoja. Tätä varten erikoistyössä [1] kuvattiin hyppyvideoita 0.5x suurennoksella, jolloin hyppyistä saatiin tallennettua useampia vauhdinottoaskeleita. (Suurennoksella 1.0x kuvatuilla hyppyvideoilla näkyy vain viimeinen vauhdinottoaskel.) MediaPipe Pose-asennontunnistusalgoritmi ei kuitenkaan löytänyt avainpisteiden koordinaatteja useista (0.5x suurennoksella kuvattujen videoiden) kuvaruuduista, tämän vuoksi 0.5x suurennoksella kuvatuilta hyppyvideoita ei pystytty määrittämään hyppyyn liittyviä arvoja. Työn tekemisen aikana testattiin MediaPipe Pose-asennontunnistusalgoritmin hyperparametrien arvojen muuttamista, mutta 0.5x suurennokselle soveltuvia parametreja ei löytynyt. Hyperparametrejä muuttamalla olisi luultavasti mahdollista löytää 0.5x suurennoksella kuvatuilta hyppyvideoilta avainpisteitä useammista kuvaruuduista.

Tässä työssä hyppyjen analysoinnissa käytettiin lopulta vain suurennoksella 1.0x kuvattuja pituushyppyvideoita. Suurennoksella 1.0x kuvatuilla videoilla hyppääjän koko oli kuvaruudussa isompi, tämä helpotti asennon arviointialgoritmin toimintaa ja tarvittavat avainpisteet tunnistettiin kuvaruuduista. Suurennoksella 1.0x saatiin luultavasti myös määritettyä tarkemmin esimerkiksi hyppyyn liittyviä kulmia (verrattuna 0.5x suurennos). Erikoistyössä 0.5x ja 1.0x suurennoksella kuvatut hyppyvideot, kuvattiin samalta etäisyydeltä. Jälkikäteen mietittynä etäisyyttä olisi voinut pienentää (0.5x suurennos), jolloin hyppääjän koko kuvaruudussa olisi kasvanut isommaksi ja useampia vauhdinottoaskeleita olisi saatu tallennettua.

## 7.6 Skaalaus ja hyppyyn liittyvät nopeudet

Hyppyihin liittyvien nopeuksien arvot olivat pienimpiä, kun niitä verrataan IAAF:n raportin [2] arvoihin. Tässä on kuitenkin huomioitava, että IAAF:n raportissa hyppääjät olivat maailmankärkeä ja heillä hyppyyn liittyvät nopeudet ovat korkeampia. Tässä työssä määritetyt resultanttinopeudet ponnistusvaiheen lopussa olivat 6,5...7,2 m/s. Määritettyjä nopeuksien arvoja ei kuitenkaan voida verrata todelliseen arvoon, tämä olisi vaatinut esimerkiksi nopeustutkan käyttämistä hyppäjien nopeuksien arvioinnissa.

Nopeuden laskeminen edellyttää kuljetun matkan ja ajan määrittämistä videon kuvaruuduilta. Kuljetun matkan laskemiseksi työssä määritettiin kuvaruudun pikselin luonnollinen/todellinen koko käyttämällä skaalaamista. Skaalaamisessa käytettiin apuna vauhdinottoradan etu- ja takapuolelle asetettuja shakkiruudukkoja (kuva 28), joiden avulla kuvaruudun pikselin luonnollinen/todellinen määritettiin vauhdinottoradan keskikohtaan. Yhden pikselin avulla suoritettu nopeuden määrittämisessä tarkastellaan muutosta pienellä alueella, tämän seurauksena virhe kasvaa. Nopeuden määrittämisen olisi voinut toteuttaa siten, että laskennassa olisi otettu mukaan useampia kuvaruutuja. Lonkan avainpisteiden siirtymä olisi laskettu ensimmäisen ja viimeisen kuvaruutujen avulla, kulunut aika olisi laskettu valittujen kuvaruutujen määrän ja keston perusteella.



Kuva 54. Shakkiruudukkojen leveyden arviointi. Kuva: Kalle Lehtonen

Työn aikana tehdyllä sovelluksella pystyi määrittämään askeleen pituuden valitsemalla kuvaruudusta halutun levyisen alueen, jonka perusteella sovellus laski viimeisen askeleen pituuden. Kuvan 54 kuvissa on kuvaruudulta valittu tiedossa olevia leveyksiä. Kuvan 54 ylimmässä kolmen kuvan kuvasarjassa pikselit on valittu vasemmanpuoleisesta, keskimmaisesta ja oikeanpuoleisesta shakkiruudukosta, kaikkien ruudukkojen leveys on 40 cm. Leveyden määrittämiseksi työssä tehtyä sovellusta suoritettiin useamman kerran ja kyseiset alueet valittiin manuaalisesti kuvaruudusta. Valintojen perusteella sovellus laski kuvan 54 vasemmanpuoleisen ja keskimmäisen ruudukkojen leveyksien olevan 39,90...40,85 cm välillä. Oikeanpuoleisen ruudukon leveys oli puolestaan 41,05...41,79 cm välillä.

Sovelluksessa suoritettu skaalaaminen (tarvittiin pikselin leveyden/korkeuden todellisen koon määrittämisessä) tehtiin vasemmanpuoleista ja keskimmäistä ruudukkoa käyttäen, koska

pääosa hyppyyn liittyvien suureiden määrittämisestä tapahtuu tällä alueella. Tästä syystä sovelluksen näille ruudukoille lasketut leveydet ovat varsin lähellä ruudukon todellista leveyttä. Oikeanpuoleinen ruudukon on puolestaan kauempana skaalauskohtasta, sovelluksen laskema ruudukko on noin 1...2 cm leveämpi kuin ruudukon todellinen leveys.

Kuvan 54 alemmassa kahden kuvan kuvasarjassa pikselit on valittu 50 cm ja 55 cm teippausten kohdalta, sovellus laski näille arvot välille 48,88...49,58 cm ja 55,01...55,49 cm. Vastaavat arvot laskettiin sovelluksella myös 100, 105, 250 ja 255 cm teippauksiin, joista saatiin arvot 97,28...99,40 cm, 102,0...104,60 cm, 248,63...250,28 cm ja 255,0...256,65 cm.

Hyppääjän nopeuksiin liittyvien suureiden laskennassa käytettiin lonkan avainpisteiden sijainnin keskiarvoa. Pystysuunnassa lonkan avainpisteiden sijainnit ovat kuvaruudulla korkeammalla, kuin skaalauksessa käytetyt shakkiruudukot. Tämä aiheuttaa jonkin verran virhettä skaalauksen avulla laskettuun kuljettuun matkaan (lonkan avainpisteiden sijainnin keskiarvo).

Kuvaan 40 on piirretty hypyn resultanttinopeus toiseksi viimeisen kontaktin irtoamisen ja viimeisen kontaktin irtoamisen välillä, hieman ennen viimeisen kontaktin irtoamista (kuvassa oranssin pystyviiva) nopeus on yli 8 m/s. Vastaavanlainen resultanttinopeuden muutos esiintyy myös useissa muissa hyppysuorituksissa, eli resultanttinopeudet ovat korkeampia ennen viimeisen kontaktin irtoamiskohtaa. Viimeisen kontaktin aikaisten nopeuksien määrittäminen tapahtuu kuvaruudulla vaakasuunnassa suurin piirtein samassa kohtaa, jossa yhden pikselin koko luonnollisessa (cm/pikseli) koossa on määritetty skaalaamalla.

## **7.7 Massakeskipisteen/lonkan alenema, lonkan/kehon kaltevuuskulmat**

IAAF:n raportissa [2] laskettiin hyppääjän massakeskipisteen sijainnin alenema. Kehon asennon arviointimenetelmillä ei pystytä määrittämään ihmiskehon massakeskipisteen sijaintia. Massakeskipiste olisi pitänyt määrittää avainpisteiden sijainnin perusteella laskennallisesti, tätä ei kuitenkaan tämän työn puitteissa toteutettu. Vastaavanlaisen (massakeskipisteen aleneman) arvon laskemiseksi, tässä työssä aleneman laskennassa käytettiin lonkan avainpisteiden keskiarvoa. Avainpisteistä laskettujen alenemien arvot eivät vastaa IAAF:n raportin tuloksia.

IAAF:n raportissa lonkan ja kehon kaltevuuskulmien laskennoissa oli käytetty kehon massakeskipistettä. Suorana seisovan ihmisen massakeskipiste on lonkan avainpisteitä korkeammalla. Massakeskipiste voi olla myös ihmiskehon ulkopuolella, riippuen siitä missä asennossa ihmisen keho ja raajat ovat. Työssä lasketut lonkan kaltevuuskulmat ja kehon

kaltevuuskulmat laskettiin käyttämällä lonkan avainpisteiden sijaintien keskiarvoa, molempien kulmien arvot olivat suurempia kuin IAAF:n raportissa.

## 8 YHTEENVETO

Työn tarkoituksena oli tutkia, onko mahdollista ohjelmoida sovellus tai sovelluksia, joilla voidaan edullisella laitteistolla kuvatuilta pituushyppyvideoilta määrittää hyppyyn liittyviä suureita, käyttämällä tekoälyyn perustuvaa esikoulutettua ihmiskehon asennontunnistusalgoritmia kehon avainpisteiden xy-koordinaattien sijaintien määrittämiseen pituushyppyvideon kuvaruuduilta. Avainpisteiden sijainnit pystyttiin määrittämään MediaPipe Pose-asennontunnistusalgoritmin avulla tämän työn kannalta riittävällä tarkkuudella ja sijaintien avulla pystyttiin määrittämään hyppysuoritukseen liittyviä suureita vauhdinottovaiheen loppuosasta ja ponnistusvaiheesta: kontakti-/lentoajat, hyppyyn liittyviä kulmia, viimeisen askeleen pituus ja muutos-/kulmanopeuksia. Hyppyvideolta MediaPipe Pose-asennontunnistusalgoritmilta avainpisteiden sijaintien tunnistamiseen kului aikaa noin 45...60 s. Hyppyyn liittyvien suureiden laskentaan kuluu aikaa vaihteittain suoritettavilla sovelluksilla arviolta noin 5...15 minuuttia. Käyttäjältä vaaditaan toimenpiteitä esimerkiksi kontaktikohtien ja askelpituuden määrittämisessä. Tämän takia menetelmä ei sovellu reaaliaikaiseen tarkasteluun harjoituksen aikana.

Työssä määritettyjen suureiden osalta, kontakti-/lentoaikojen, hyppyyn liittyvät kulmien ja viimeisen askeleen pituus olivat samassa suuruusluokassa, kuin kansainvälisen yleisurheiluliiton pituushyppyraportissa [2]. Hyppystä määritetyt nopeudet (huomioitava myös hyppääjien taso: kansainvälinen, kansallinen ja 16...17-vuotias) ja lonkan/massakeskipisteen alenema eivät puolestaan olleet raportin kanssa samassa suuruusluokassa, näiden suureiden osalta vaadittaisiin lisää tutkimusta ja sovelluksien muokkaamista/kehittämistä. Mikäli työssä tehdyllä sovelluksella määritettyjä suureita haluttaisiin käyttää esimerkiksi pituushyppääjän kehittymisen seuraamiseen kauden aikana, niin pitäisi saada mitattua riippumattomat mittaustulokset (esimerkiksi nopeuden mittaaminen nopeustukalla), joita voitaisiin käyttää työssä tehdyillä sovelluksella määritettyjen suureiden arvojen validointiin.

MediaPipe Pose-asennontunnistusalgoritmi tunnistoi joissain kuvaruuduissa vasemman ja oikean puolen koordinaattien sijainnit väärin ja tallensi koordinaatin väärälle puolelle, vaihtuneiden koordinaattien korjaaminen onnistui työssä tehdyllä sovelluksella. Useita peräkkäisiä outliereita esiintyi hyppyvideoilla kohdissa, joilla ei ollut vaikutusta suureiden määrittämiseen.

Työssä käytettyjä menetelmiä voisi pituushypyn lisäksi soveltaa myös muissa yleisurheilulajeissa (pika-/aitajuoksu, kävely, seiväshyppy ja keihäänheitto), esimerkiksi kontaktiaikojen ja kehon kulmien määrittämisessä. Myös askelpituuksien ja nopeuksien määrittäminen onnistuisi muissa lajeissa (kentälle pitää asetella shakkiruutuja). Työssä käytettyjen menetelmien avulla voisi myös määrittää, kuinka kaukana ponnistuslankkaa tai yliastumisviivaa viimeinen kontakti kentän pintaan tapahtuu. Työn tekemisen aikana havaittiin, että hyppääjän käsien avainpisteiden xy-koordinaatteja ei tunnistettu monista kuvaruuduista, joten tämä voi rajoittaa esimerkiksi keihäänheitossa käsien liikkeiden määrittämistä. Työssä käytettyjä menetelmiä voisi soveltaa myös muihin urheilulajeihin esimerkiksi hiihtoon (kehonkulmat, liu'un pituus ja nopeus), kori-/lentopallo (kehon kulmat, ponnistusta edeltävän askeleen pituus, ponnistusvaiheen kesto/nopeudet, askeleiden kontaktiajat, ilmalennon aika, ponnistuksen korkeus/pituus) ja painon-/voimannosto (kehon kulmien määrittäminen, suorituksen vaiheiden kesto). Työssä käytettyjen menetelmiä voitaisiin soveltaa joukkuelajeihin esimerkiksi harjoituksissa, jossa vain yksittäinen urheilija tekee suorituksen (useamman ihmiskehon avainpisteet voivat sekoittua keskenään).

Sovelluksien jatkokehittämisessä, voisi tässä työssä tehdyt vaiheittain suoritettavat sovellukset koota yhdeksi sovellukseksi, tämä voisi lyhentää hyppyyn liittyvien suureiden määrittämisaikaa. Lisäksi shakkiruutujen sijainnin määrittäminen pitäisi suorittaa kokonaan erillisessä vaiheessa (suoritettaisiin kerran ennen suureiden laskentaa). Useamman vauhdinottoaskeleen suureiden määrittämiseksi voisi jatkokehittämisessä kuvata hyppyjä erilaisilta etäisyyksiltä ja tutkia löytääkö MediaPipe Pose-asennontunnistusalgoritmi kehonavainpisteet, niin että suureiden määrittäminen on mahdollista. Jatkotutkimusta voisi tehdä myös MediaPipe Pose-asennontunnistusalgoritmin avainpisteiden xy-koordinaattien tarkkuudesta, tämän toteuttamiseksi hyppyvideon kuvaruuduilta pitäisi määrittää avainpisteiden xy-koordinaatit manuaalisesti ja verrata näitä asennontunnistusalgoritmin vastaaviin arvoihin. Jatkotutkimusta voisi suorittaa myös työssä määritettyjen suureiden arvojen vaihtelusta. Todellisen tarkkuuden selvittämiseksi olisi selvitettävä, onko vaihtelu todellista vai suorituksesta riippuvaa tai käytettyihin menetelmiin liittyvää variaatiota. Yhtenä jatkotutkimuksen kohteena voisi olla, asennontunnistusalgoritmin opettaminen pituushyppyyn liittyvillä kuvilla. Avainpisteiden tunnistaminen ja sijaintien tarkkuus voisi parantua, mikäli jatkotutkimuksessa asennontunnistusalgoritmia opetettaisiin pituushyppysuorituksiin liittyvillä kuvilla.

## Lähteet

- [1] K. Lehtonen. 2023. Pituushyppyanalytiikka hyppyvideoiden avulla – videomateriaalin kerääminen ja esikäsittely. Erikoistyökurssin työseloste. Turun yliopisto.
- [2] C.B. Tucker, A. Bissas, S. Merlino. 2019. Biomechanical Report for the IAAF World Indoor Championships 2018: Long Jump Men. Birmingham, UK: International Association of Athletics Federations.
- [3] O.P. Mykkänen. 2014. Pituushypyn lajiansalyysi ja valmennuksen ohjelmointi. Valmentajaseminaari Liikuntabiologian laitos, Jyväskylän yliopisto. Lehdet: 3 ja 7. <https://jyx.jyu.fi/handle/123456789/43955>. Luettu 1.9.2023.
- [4] O. Koskinen. 2016. Pituushypyn lajiansalyysi ja valmennuksen ohjelmointi. Valmentajaseminaarityö. Liikuntabiologian laitos. Jyväskylän yliopisto. Lehti: 2. <https://jyx.jyu.fi/handle/123456789/50244>. Luettu 29.8.2023.
- [5] A.M.J. Mero. 2011. Reaktivoimatuotto, nopeus ja tekniikka pituushypyn ponnistusvaiheessa. Valmennus- ja testausoppi. Kandidaatintutkielma. Liikuntabiologian laitos, Jyväskylän yliopisto. Lehdet: 6, 7, 9, 11, 12 ja 17. <https://jyx.jyu.fi/handle/123456789/26805>. Luettu 29.8.2023.
- [6] R. MacPherson.2022. Understanding biomechanics and body movement - Biomechanics is the study of how living things move. <https://www.verywellfit.com/understanding-biomechanics-3498389>. Luettu: 23.11.2023.
- [7] M. Ihakki. 2018. Liikkeentunnistusjärjestelmän toteutus Kinect v2 -sensoria käyttäen. Insinöörityö, Metropolian ammattikorkeakoulu. Lehdet: 5-7.
- [8] S. Delp. 2022. Biomechanics of movement - Lecture 13.3: Democratizing access to human movement analysis, Stanford University. <https://youtu.be/3ElbNvh2r7U?t=52>. Katsottu: 30.11.2023.
- [9] K. Ludwig, J. Lorenz, R. Schön, R. Lienhart. 2023. All keypoints you need: Detecting arbitrary keypoints on the body of triple, high, and long jump athletes. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

- [10] A. Cioppa, S. Giancola, A. Deliege, L. Kang, X. Zhou, Z. Cheng, B. Ghanem, M. Van Droogenbroeck. 2022. SoccerNet-Tracking: Multiple object tracking dataset and benchmark in soccer videos. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Lehdet: 3491–3502. <https://doi.org/10.48550/arXiv.2204.06918>. Luettu 3.11.2023.
- [11] K. M. Kulkarni, S. Shenoy. 2021. Table tennis stroke recognition using two-dimensional human pose estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Lehdet: 4576–4584.
- [12] P. Liu, J.-H. Wang. 2022. Monotrack: Shuttle trajectory reconstruction from monocular badminton video. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Lehdet: 3513–3522. <https://doi.org/10.48550/arXiv.2204.01899>. Luettu 3.11.2023.
- [13] D. Stepec, D. Skocaj. 2022. Video-based ski jump style scoring from pose trajectory. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops. Lehdet: 682–690. <https://ieeexplore.ieee.org/document/9707593>. Luettu 3.11.2023.
- [14] V. Hudovernik, D. Skočaj. 2022. Video-Based Detection of Combat Positions and Automatic Scoring in Jiu-jitsu. Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports (MMSports '22).
- [15] S. Garg, A. Saxena, R. Gupta. 2022. Yoga pose classification: a CNN and MediaPipe inspired deep learning approach for real-world application. J Ambient Intell Human Comput (2022).
- [16] P. Jafarzadeh, P. Virjonen, P. Nevalainen, F. Farahnakian, J. Heikkonen. 2021. Pose Estimation of Hurdles Athletes using OpenPose. 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering.
- [17] Final reports on scientific projects in the areas of biomechanics, coaching and development. <https://worldathletics.org/about-iaaf/documents/research-centre>. Luettu: 24.11.2023.
- [18] A. Jääskeläinen. 2019. Mitä tapahtuu huomenna kun, tekoäly poistaa järjettömyydet? Helsinki: WSOY. Lehdet: 11, 12 ja 14.

- [19] IBM. Artificial intelligence: Types of artificial intelligence - weak AI vs. strong AI. [https://www.ibm.com/topics/artificial-intelligence?mhsrc=ibmsearch\\_a&mhq=artificial%20intelligence](https://www.ibm.com/topics/artificial-intelligence?mhsrc=ibmsearch_a&mhq=artificial%20intelligence). Luettu 8.5.2023.
- [20] IBM. What is strong AI. <https://www.ibm.com/topics/strong-ai>. Luettu 5.2.2024.
- [21] F. Chollet. 2018. Deep Learning with Python. New York: Manning Publications Co. Lehdet: 5 ja 20-23.
- [22] A. Hautala 2020. Koneoppiminen ja matematiikka – johdanto aiheeseen ja kertausmateriaali. Pro gradu -tutkielma, Helsingin yliopisto, Suomi. Lehdet: 21 ja 22. [https://helda.helsinki.fi/bitstream/handle/10138/322282/Hautala\\_Anni\\_progradu\\_2020.pdf?sequence=2&isAllowed=y](https://helda.helsinki.fi/bitstream/handle/10138/322282/Hautala_Anni_progradu_2020.pdf?sequence=2&isAllowed=y). Luettu: 10.5.2023.
- [23] Helsingin Yliopisto. 2022. Tekoälyn perusteet kurssi materiaali. <https://course.elementsofai.com/fi/4/1>. Luettu 25.04.2023
- [24] T. Roos. 2016. Johdatus tekoälyyn - kurssin luentokalvot. <https://www.cs.helsinki.fi/u/ttonteri/ai/2016/luennot/luento7.pdf>. Luettu 10.5.2023.
- [25] IBM. What is machine learning? <https://www.ibm.com/topics/machine-learning>. Luettu 22.6.2023.
- [26] M. Yao, M. Jia, A. Zhou. 2018. Applied artificial intelligence: A Handbook for business leaders. Topbots. Lehti: 23.
- [27] P. Neittaanmäki, H. Tuominen. 2019. Tekoälyn perusteita ja sovelluksia. Jyväskylän yliopisto, Suomi. Lehdet: 13 ja 52. <https://jyx.jyu.fi/handle/123456789/64975>. Luettu 11.5.2023.
- [28] E. Alpaydin. 2021. Koneoppiminen. Helsinki: Libris media. Lehdet: 108 ja 141. (Englanninkielinen alkuperäisteos: Machine learning. Revised and updated revision. 2021.)
- [29] IBM. What is computer vision? <https://www.ibm.com/topics/computer-vision>. Luettu 10.10.2023.
- [30] J. Sawtell-Rickson. 2022. What is computer vision? <https://builtin.com/machine-learning/computer-vision>. Luettu 11.10.2023.

- [31] V. Saarinen. 2020. Kuusi hyödyllistä näkökulmaa koneoppimisesta. <https://reflector.fi/blog/kuusi-hyodyllista-nakokulmaa-koneoppimisesta/>. Luettu 17.11.2023.
- [32] N. Van Otten. 2023. Data quality in machine learning - Explained, issues, how to fix them & python tools. <https://spotintelligence.com/2023/04/07/data-quality-machine-learning/>. Luettu 17.11.2023.]
- [33] M.R. Berthold, C Borgelt, F. Höppner, F. Klawonn. 2010. Guide to intelligent data analysis - How to intelligently make sense of real data. New York: Springer. Lehti: 103.
- [34] A. Bonnet. 2023. Top 15 free datasets for human pose estimation in computer vision. <https://encord.com/blog/15-best-free-pose-estimation-datasets/>. Luettu: 27.11.2023.
- [35] D. C. Luvizon, D. Picard, H. Tabia. 2018. 2D/3D pose estimation and action recognition using multitask deep learning. <https://arxiv.org/abs/1802.09232>. Luettu: 27.11.2023.
- [36] V. Meel. 2023. What is the COCO Dataset? What you need to know in 2024. <https://viso.ai/computer-vision/coco-dataset/>. Luettu: 27.11.2023.
- [37] D. Nikolaiev. 2021. Overfitting and Underfitting Principles. <https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c>. Luettu 14.7.2023.
- [38] D. Andina, A. Vega-Corona, J. I. Seijas, J. Torres-García. 2007. Neural networks historical review. Computational Intelligence. Lehdet: 39-65.
- [39] T. Knuutila. 2021. Tekoälyn perusteet kurssin johdanto luentomateriaali. Turun yliopisto. Tietotekniikan laitos.
- [40] H. Chen, R. Feng, S. Wu, H. Xu, F. Zhou, Z. Liu. 2022. 2D Human Pose Estimation: A Survey. Multimedia systems.
- [41] IBM. What are neural networks? <https://www.ibm.com/topics/neural-networks>. Luettu 30.11.2023.
- [42] G. Boesch. Deep Neural Network: The 3 Popular Types (MLP, CNN and RNN). <https://viso.ai/deep-learning/deep-neural-network-three-popular-types/>. Luettu 22.6.2023.

- [43] S. Heininen. 2020. Syväoppivat neuroverkot ja niiden sovellukset. Pro gradu -tutkielma, Turun yliopisto. Lehdet: 17 ja 37-38.  
[https://www.utupub.fi/bitstream/handle/10024/150834/Heininen\\_Sara\\_opinnayte.pdf?sequence=1](https://www.utupub.fi/bitstream/handle/10024/150834/Heininen_Sara_opinnayte.pdf?sequence=1). Luettu 14.7.2023.
- [44] P. Louridas. 2020. Algoritmit. Helsinki: Libris media. Lehdet: 178-182, 189-192, 194 ja 197-199. (Englanninkielinen alkuperäisteos: Algorithms. 2020.)
- [45] J. D. Kelleher. 2019. Syväoppiminen, kuinka tekoäly toimii. Helsinki: Libris media. Lehdet: 48, 66, 74-76, 83, 84, 86, 116, 145 ja 147. (Englanninkielinen alkuperäisteos: Deep learning. 2019.)
- [46] C. Bento. 2021. Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis. <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>. Luettu 12.7.2023.
- [47] Baledung. 2023. Multi-Layer Perceptron vs. Deep Neural Network. <https://www.baeldung.com/cs/mlp-vs-dnn>. Luettu 12.7.2023.
- [48] Activation functions. 2023. <https://paperswithcode.com/methods/category/activation-functions>. Luettu 4.7.2023.
- [49] A. Keerthi. 2022. Fantastic activation functions and when to use them. <https://towardsdatascience.com/fantastic-activation-functions-and-when-to-use-them-481fe2bb2bde>. Luettu 4.7.2023.
- [50] M. Elgendy. 2019. Deep Learning for Vision Systems (MEAP edition, Version 6). New York: Manning Publications Co. Lehdet: 30, 31, 55-61, 107, 108, 113-117, 122, 123 ja 195-196.
- [51] N. Schlüter. 2019. Don't Overfit! - How to prevent Overfitting in your Deep Learning Models. <https://towardsdatascience.com/dont-overfit-how-to-prevent-overfitting-in-your-deep-learning-models-63274e552323>. Luettu 25.7.2023.
- [52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, Volume 15, 1929-1958.

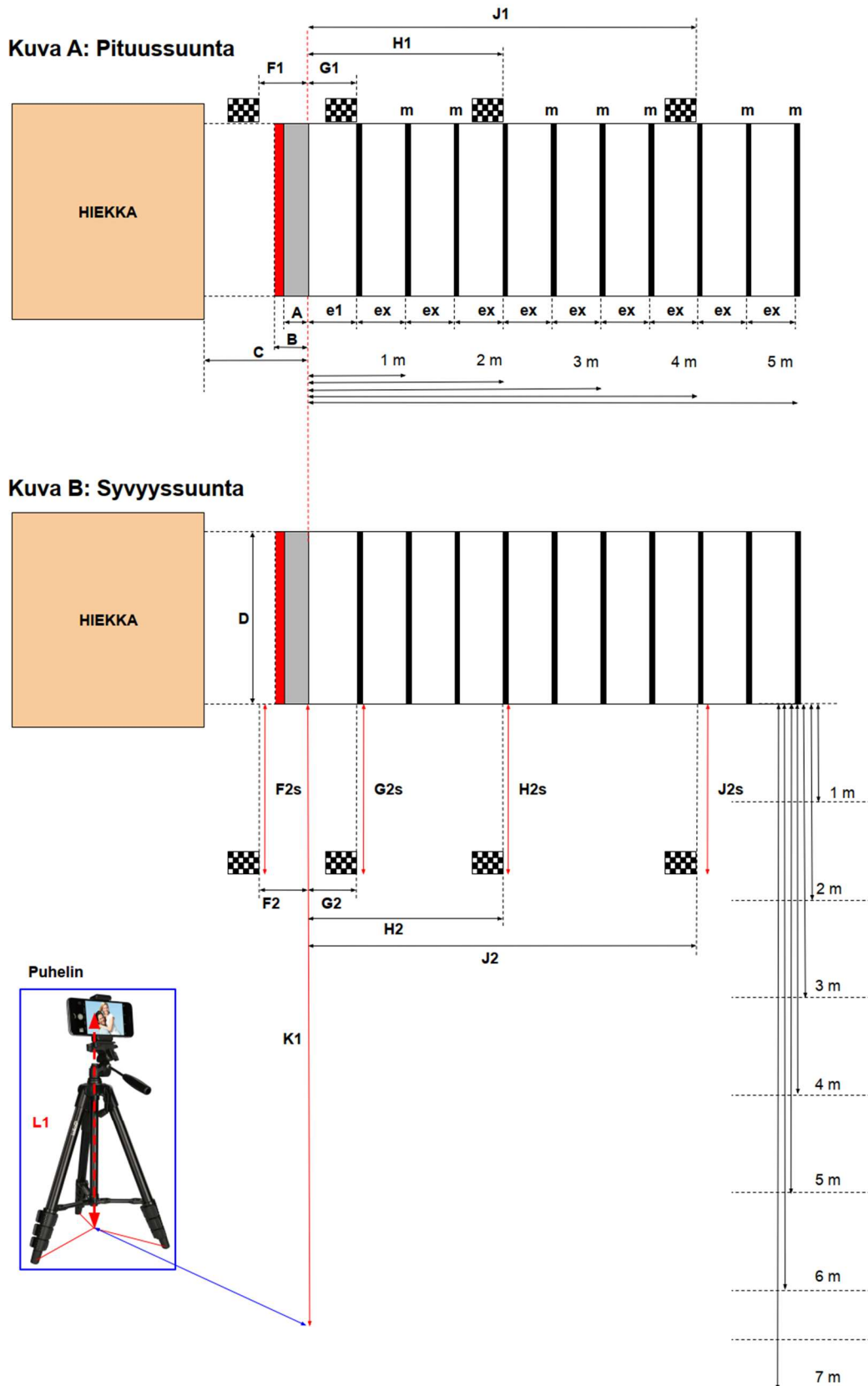
- [53] S. Norris. 2018. Classic Regularization Techniques in Neural Networks. <https://odsc.medium.com/classic-regularization-techniques-in-neural-networks-68bccee03764>. Luettu 25.7.2023.
- [54] S. Jain. 2023. An Overview of Regularization Techniques in Deep Learning. <https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/>. Luettu 25.7.2023.
- [55] K. Kamali. 2023. Deep Learning (Part 3) - Convolutional neural networks (CNN). <https://training.galaxyproject.org/training-material/topics/statistics/tutorials/CNN/tutorial.html>. Luettu: 27.11.2023.
- [56] S. Balachandran. 2020. Machine Learning - Convolution with color images. <https://dev.to/sandeepbalachandran/machine-learning-convolution-with-color-images-2p41>. Luettu 6.9.2023.
- [57] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, M. Shah. 2018/2022. Deep Learning-Based Human Pose Estimation: A Survey. Association for Computing Machinery, 2018/8, Volume 37, Number 4, Article 111. 1-35.
- [58] Z. Liu, J. Zhu, J. Bu, C. Chen. 2015. A survey of human pose estimation: The body parts parsing based methods. Journal of Visual Communication and Image Representation, 2015/10, Numero: 32, Lehdet: 10-19.
- [59] W. Gong, X. Zhang, J. González, A. Sobral, T. Bouwmans, C Tu, E, Zahzah. 2016. Human pose estimation from monocular images: A comprehensive survey. MDPI – Sensors, 2016/11, Julkaisu: 16, Numero: 12, Artikkelin: 1966. Lehdet: 1-39.
- [60] Y. Chen, Y. Tian, M. He. 2020. Monocular human pose estimation: A survey of deep learning-based methods. Computer Vision and Image Understanding, 2020/3, Julkaisu: 192, Lehdet: 1-20.
- [61] Q. Dang, J. Yin, B. Wang, W. Zheng. 2019. Deep learning based 2D human pose estimation: A survey. IEEE Xplore - Tsinghua Science and Technology, 2019/12, Julkaisu: 6, Numero: 6, 663-676.

- [62] T. Munea, Y. Jembre, H. Weldegebriel, L. Chen, C. Huang, C. Yang. 2020. The progress of human pose estimation: A survey and taxonomy of models Applied in 2D human pose estimation. *IEEE Access*, 2020, Julkaisu: 8, Lehdet: 133330-133348.
- [63] M. Gamra, M. Akhloufi. 2021. A review of deep learning techniques for 2D and 3D human pose estimation. *Image and Vision Computing*, Numero: 114, Artikkele: 104282, Lehdet: 1-23.
- [64] J. Li, S. Bian, A. Zeng, C. Wang, B. Pang, W. Liu, C. Lu. 2021. Human pose regression with residual log-likelihood estimation. *International Conference on Computer Vision*.
- [65] S. Dubey, M. Dixit. 2022. A comprehensive survey on human pose estimation approaches. *Multimedia Systems*, Numero: 29, Lehdet: 167–195.
- [66] S. Liang, X. Sun, Y. Wei. 2018. Compositional Human Pose Regression. *Computer Vision and Image Understanding*, Numero: 176-177, Lehdet: 1-8.
- [67] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, B. Schiele. 2016. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation.
- [68] E. Odemakinde. 2023. Human Pose Estimation with Deep Learning - Ultimate Overview in 2023. <https://viso.ai/deep-learning/pose-estimation-ultimate-overview/>. Luettu 12.6.2023.
- [69] J.-M. Chung, L.-Y. Ong, M.-C. Leow. 2022. Comparative Analysis of Skeleton-Based Human Pose Estimation. *Future internet*, Volume 14, Issue 12.
- [70] D. Oved. 2018 Real-time Human Pose Estimation in the Browser with TensorFlow.js. <https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5>. Luettu 15.6.2023.
- [71] K. Wason. 2020. Human Pose Estimation: OpenPose vs. HRNet. <https://kartikwason.medium.com/human-pose-estimation-openpose-vs-hrnet-e8fa37768929>. Luettu 13.6.2023.
- [72] K. LeViet. 2021. Pose estimation and classification on edge devices with MoveNet and TensorFlow Lite. <https://blog.tensorflow.org/2021/08/pose-estimation-and-classification-on-edge-devices-with-MoveNet-and-TensorFlow-Lite.html>. Luettu 15.6.2023.

- [73] Folio3 AI Editorial Team. 2022. OpenPose guide 2022 - How to use/install, features and more. <https://www.folio3.ai/blog/openpose-ai-software/>. Luettu 15.6.2023.
- [74] K. Sun, B. Xiao, D. Liu, J. Wang. 2019. Deep high-resolution representation learning for human pose estimation. Conference on Computer Vision and Pattern Recognition.
- [75] Google. MediaPipe Solutions guide. <https://developers.google.com/mediapipe/solutions/guide>. Luettu 19.6.2023.
- [76] Google. Pose landmark detection guide. [https://developers.google.com/mediapipe/solutions/vision/pose\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/pose_landmarker). Luettu 19.6.2023.
- [77] Google. MediaPipe, Pose landmark detection guide. [https://developers.google.com/mediapipe/solutions/vision/pose\\_landmarker#get\\_started](https://developers.google.com/mediapipe/solutions/vision/pose_landmarker#get_started). Luettu 21.6.2023.
- [78] MediaPipe Pose, Cross-platform configuration options. <https://github.com/google/mediapipe/blob/master/docs/solutions/pose.md#cross-platform-configuration-options>. Luettu 25.10.2023
- [79] MathWorks. 1994-2023. Hampel filter documentation. [https://se.mathworks.com/help/dsp/ref/dsp.hampelfilter-system-object.html#bvj3ldg-1\\_sep\\_bvkftv4-6](https://se.mathworks.com/help/dsp/ref/dsp.hampelfilter-system-object.html#bvj3ldg-1_sep_bvkftv4-6). Luettu 12.9.2023.
- [80] L.B. Luo, I.S. Koh, K.Y. Min, J. Wang, J.W. Chong. 2010. Low-cost implementation of bird's-eye view system for camera-on-vehicle. 2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE).
- [81] OpenCV. Geometric transformation of images goals. [https://opencv24-python-tutorials.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_geometric\\_transformations/py\\_geometric\\_transformations.html](https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_geometric_transformations/py_geometric_transformations.html). Luettu 10.11.2023.
- [82] OpenCV. Geometric image transformations. [https://docs.opencv.org/4.x/da/d54/group\\_imgproc\\_transform.html#gaf73673a7e8e18ec6963e3774e6a94b87](https://docs.opencv.org/4.x/da/d54/group_imgproc_transform.html#gaf73673a7e8e18ec6963e3774e6a94b87). Luettu 10.11.2023.
- [83] Kinovea features. <https://www.kinovea.org/features.html>. Luettu 25.10.2023.

# Liitteet

## Liite 1. Havainnekuva välineistön sijoittelusta kentälle



## Liite 2. Hypystä tallennetut tiedot

### GENERAL INFORMATION:

Video file path:  
 Pickle file path:  
 Video: IMG\_8467  
 Camera: CAM-A  
 Jump number: 02  
 Event: A  
 Athlete: 2  
 Zoom: 1 (1: 100% or 05: 50%)  
 Remove video: REM-0 (REM-0: Video is analyzed.)

### LEG:

Second last contact: LEFT Swinging leg  
 Last contact: RIGHT Jumping leg

### CONTACT and FLIGHT TIMES:

Second last contact time: 0.1251 s  
 Last step flight time: 0.0876 s  
 Last contact time: 0.1377 s Jump

### LAST STEP:

Last step length: 2.19 m From toe to toe  
 Last step resultant velocity: 6.73 m/s

### HIP LOWERING:

Hip lowering: 0.95 cm If positive, no lowering. (NOTE: Hip vs Centre of mass)

### SECOND LAST CONTACT:

Horizontal velocity at touchdown: 6.98 m/s  
 Horizontal velocity at take-off: 7.59 m/s

### LAST CONTACT - jump:

Horizontal velocity at touchdown: 7.41 m/s  
 Horizontal velocity at take-off: 6.11 m/s  
 Horizontal velocity change (to-td): -1.30 m/s  
 Vertical velocity at take-off: 2.28 m/s  
 Resultant velocity at take-off: 6.52 m/s

Take-off angle: 21.0°

Knee angle at touchdown: 165.3°  
 Minimum knee angle: 136.0°  
 Knee range of motion: 29.3°  
 Knee angular velocity: -469°/s

Body inclination angle at td: -23.9° (NOTE: Hip vs Centre of mass)  
 Body inclination angle at to: 25.9° (NOTE: Hip vs Centre of mass)

Trunk lean angle at td: -8.4° (NOTE: Hip vs Centre of mass)  
 Trunk lean angle at to: -4.2° (NOTE: Hip vs Centre of mass)

Thigh angle of swing leg (td): -12.7°  
 Thigh angle of swing leg (to): -4.8° If positive over horizontal line.  
 Thigh angular vel of swing leg: 527°/s

### Liite 3. Hyppystä tallennetut tiedot (hyppääjä noin 16...17-vuotias)

#### GENERAL INFORMATION:

Video file path:

Pickle file path:

**Video:**

Camera:

Jump number:

Event:

**Athlete:**

Zoom:

Remove video:

#### LEG:

Second last contact:

Last contact:

#### CONTACT and FLIGHT TIMES:

Second last contact time:

Last step flight time:

Last contact time:

#### LAST STEP:

Last step length:

Last step resultant velocity:

#### HIP LOWERING:

Hip lowering:

#### SECOND LAST CONTACT:

Horizontal velocity at touchdown:

Horizontal velocity at take-off:

#### LAST CONTACT - jump:

Horizontal velocity at touchdown:

Horizontal velocity at take-off:

Horizontal velocity change (to-td):

Vertical velocity at take-off:

Resultant velocity at take-off:

Take-off angle:

Knee angle at touchdown:

Minimum knee angle:

Knee range of motion:

Knee angular velocity:

Body inclination angle at td:

Body inclination angle at to:

Trunk lean angle at td:

Trunk lean angle at to:

Thigh angle of swing leg (td):

Thigh angle of swing leg (to):

Thigh angular vel of swing leg:

IMG_8470	IMG_8473	IMG_8474	IMG_8476
CAM-A	CAM-A	CAM-A	CAM-A
05	08	09	11
A	A	A	A
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
1	1	1	1
REM-0	REM-0	REM-0	REM-0
LEFT	LEFT	LEFT	LEFT
RIGHT	RIGHT	RIGHT	RIGHT
0.1502 s	0.1377 s	0.1460 s	0.1460 s
0.0751 s	0.1210 s	0.1001 s	0.1669 s
0.1711 s	0.1544 s	0.1377 s	0.1419 s
2.06 m	2.34 m	2.11 m	2.61 m
5.89 m/s	6.11 m/s	5.96 m/s	6.14 m/s
2.03 cm	0.02 cm	2.67 cm	0.91 cm
6.81 m/s	5.50 m/s	5.95 m/s	5.94 m/s
6.91 m/s	6.54 m/s	6.55 m/s	6.25 m/s
5.92 m/s	7.16 m/s	6.51 m/s	6.09 m/s
6.10 m/s	5.91 m/s	6.43 m/s	6.11 m/s
0.18 m/s	-1.25 m/s	-0.09 m/s	0.02 m/s
1.43 m/s	1.50 m/s	2.61 m/s	1.74 m/s
6.27 m/s	6.09 m/s	6.94 m/s	6.36 m/s
17.0°	12.9°	21.9°	22.4°
162.9°	164.1°	165.9°	173.5°
123.1°	130.5°	141.0°	143.7°
39.8°	33.6°	25.0°	29.8°
-477°/s	-448°/s	-460°/s	-510°/s
-30.3°	-25.8°	-23.9°	-20.2°
27.2°	24.9°	23.9°	24.1°
-7.2°	-11.5°	-12.3°	-15.8°
-5.6°	-8.1°	-7.8°	-9.5°
-18.6°	-21.8°	-12.9°	-15.2°
-17.0°	-26.3°	-13.6°	-38.9°
318°/s	271°/s	461°/s	253°/s

#### IAAF results:

Min	Mean	Max
0,084	<b>0,121</b>	0,128
0,06	<b>0,074</b>	0,084
---	---	---
2,03	<b>2,18</b>	2,36
9,23	<b>9,54</b>	9,97
-1,00	<b>-3,00</b>	-8,00
---	---	---
---	---	---
8,1	<b>8,67</b>	9,12
-1,62	<b>-1,41</b>	-0,96
3,22	<b>3,63</b>	4,2
8,93	<b>9,40</b>	9,81
19,4	<b>22,7</b>	26
160,1	<b>165,9</b>	172,2
120,5	<b>137,4</b>	150,8
18,7	<b>28,6</b>	41,5
-669	<b>-478</b>	-412
-32,2	<b>-34,7</b>	-39,3
13,5	<b>18,0</b>	22,7
-16,3	<b>-5,6</b>	0,7
-9,4	<b>0,8</b>	10,4
---	---	---
-33,1	<b>-16,2</b>	-7,5
363	<b>530</b>	704

## Liite 4. Hypyistä tallennetut tiedot (kansallisen tason top-10)

### GENERAL INFORMATION:

Video file path:

Pickle file path:

**Video:**

Camera:

Jump number:

Event:

**Athlete:**

Zoom:

Remove video:

### LEG:

Second last contact:

Last contact:

### CONTACT and FLIGHT TIMES:

Second last contact time:

Last step flight time:

Last contact time:

### LAST STEP:

Last step lenght:

Last step resultant velocity:

### HIP LOWERING:

Hip lowering:

### SECOND LAST CONTACT:

Horizontal velocity at touchdown:

Horizontal velocity at take-off:

### LAST CONTACT - jump:

Horizontal velocity at touchdown:

Horizontal velocity at take-off:

Horizontal velocity change (to-td):

Vertical velocity at take-off:

Resultant velocity at take-off:

Take-off angle:

Knee angle at touchdown:

Minimum knee angle:

Knee range of motion:

Knee angular velocity:

Body inclination angle at td:

Body inclination angle at to:

Trunk lean angle at td:

Trunk lean angle at to:

Thigh angle of swing leg (td):

Thigh angle of swing leg (to):

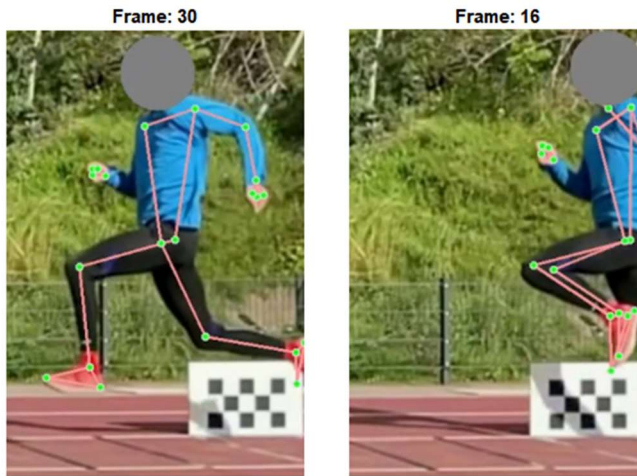
Thigh angular vel of swing leg:

IMG_8467	IMG_8469	IMG_8471	IMG_8475
CAM-A	CAM-A	CAM-A	CAM-A
02	04	06	10
A	A	A	A
<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>
1	1	1	1
REM-0	REM-0	REM-0	REM-0
LEFT	LEFT	LEFT	LEFT
RIGHT	RIGHT	RIGHT	RIGHT
0.1251 s	0.1252 s	0.1293 s	0.1335 s
0.0876 s	0.1001 s	0.0834 s	0.1001 s
0.1377 s	0.1293 s	0.1335 s	0.1293 s
2.19 m	2.22 m	2.19 m	2.33 m
6.73 m/s	6.98 m/s	6.65 m/s	6.99 m/s
0.95 cm	2.22 cm	0.57 cm	0.17 cm
6.98 m/s	8.14 m/s	7.04 m/s	8.12 m/s
7.59 m/s	6.97 m/s	7.25 m/s	8.83 m/s
7.41 m/s	7.52 m/s	6.58 m/s	7.46 m/s
6.11 m/s	6.87 m/s	4.50 m/s	6.84 m/s
-1.30 m/s	-0.65 m/s	-2.08 m/s	-0.62 m/s
2.28 m/s	2.11 m/s	1.62 m/s	2.26 m/s
6.52 m/s	7.18 m/s	4.78 m/s	7.21 m/s
21.0°	21.9°	16.4°	21.6°
165.3°	165.3°	169.1°	165.5°
136.0°	139.8°	142.2°	136.2°
29.3°	25.5°	26.9°	29.3°
-469°/s	-436°/s	-460°/s	-502°/s
-23.9°	-23.8°	-27.2°	-25.6°
25.9°	25.1°	22.6°	25.9°
-8.4°	-7.0°	-8.4°	-5.6°
-4.2°	-6.1°	-4.3°	-3.8°
-12.7°	-8.5°	-15.9°	-13.9°
-4.8°	-5.2°	-7.1°	0.7°
527°/s	590°/s	501°/s	594°/s

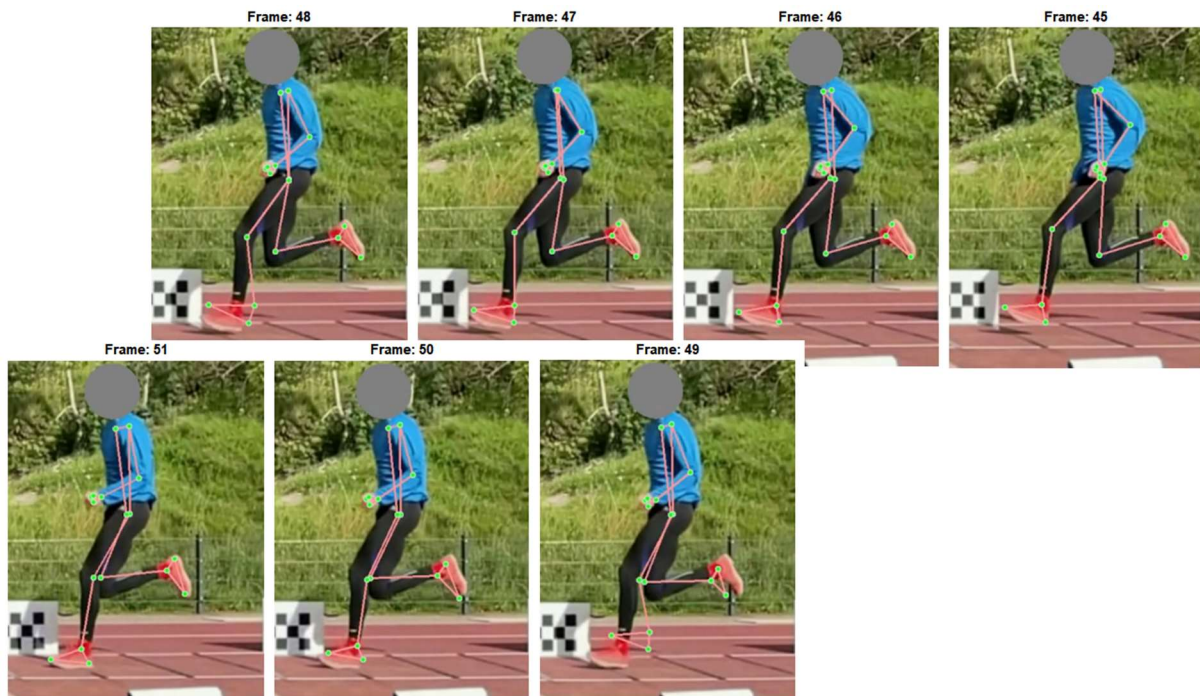
### IAAF results:

Min	Mean	Max
0,084	<b>0,121</b>	0,128
0,06	<b>0,074</b>	0,084
---	---	---
2,03	<b>2,18</b>	2,36
9,23	<b>9,54</b>	9,97
-1,00	<b>-3,00</b>	-8,00
---	---	---
---	---	---
---	---	---
8,1	<b>8,67</b>	9,12
-1,62	<b>-1,41</b>	-0,96
3,22	<b>3,63</b>	4,2
8,93	<b>9,40</b>	9,81
19,4	<b>22,7</b>	26
160,1	<b>165,9</b>	172,2
120,5	<b>137,4</b>	150,8
18,7	<b>28,6</b>	41,5
-669	<b>-478</b>	-412
-32,2	<b>-34,7</b>	-39,3
13,5	<b>18,0</b>	22,7
-16,3	<b>-5,6</b>	0,7
-9,4	<b>0,8</b>	10,4
---	---	---
-33,1	<b>-16,2</b>	-7,5
363	<b>530</b>	704

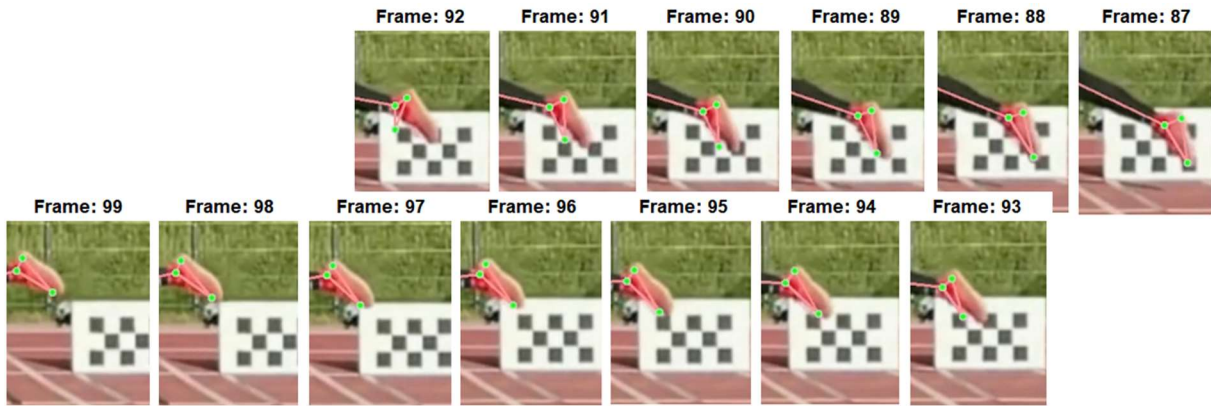
## Liite 5. Avainpisteiden tunnistaminen (mittaustietoja ei muokattu)



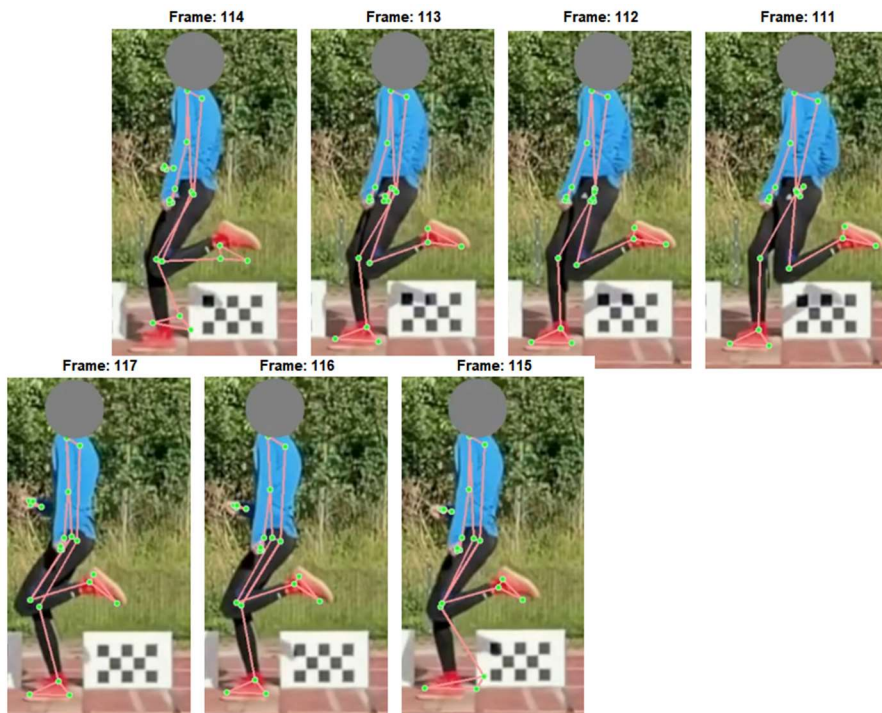
Liite 5 - Kuva 1. Hyppääjä ilmestyy kuvaruudun reunasta kuvaan. Kuva: Kalle Lehtonen



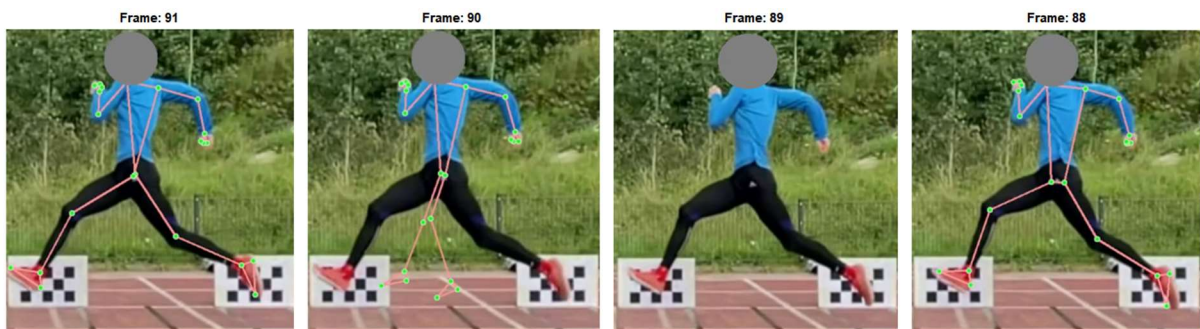
Liite 5 - Kuva 2. Kantapäiden, nilkkojen ja iso varpaiden avainpisteiden sijainnit eivät oikeilla paikoilla (kuvaruudut: 47 ... 50) ennen toiseksi viimeisen kontaktin alkua (kuvaruutu: 51). Kuva: Kalle Lehtonen



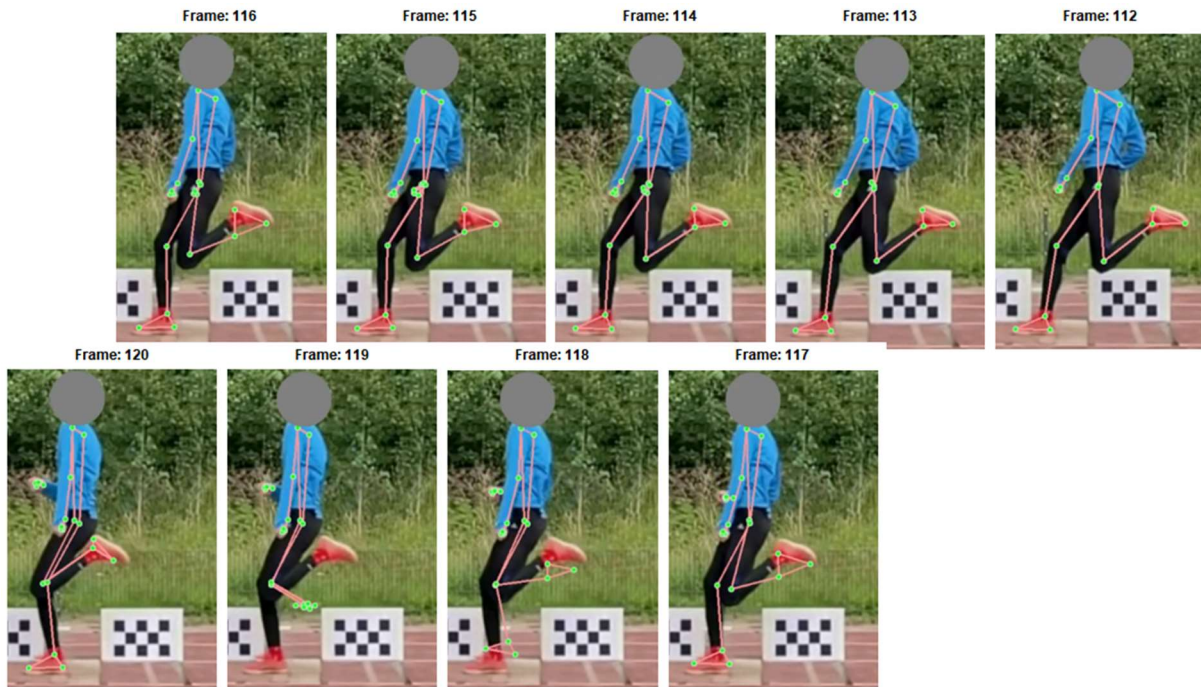
Liite 5 - Kuva 3. Isovarpaan avainpisteen sijainti vaeltaa. Kuva: Kalle Lehtonen



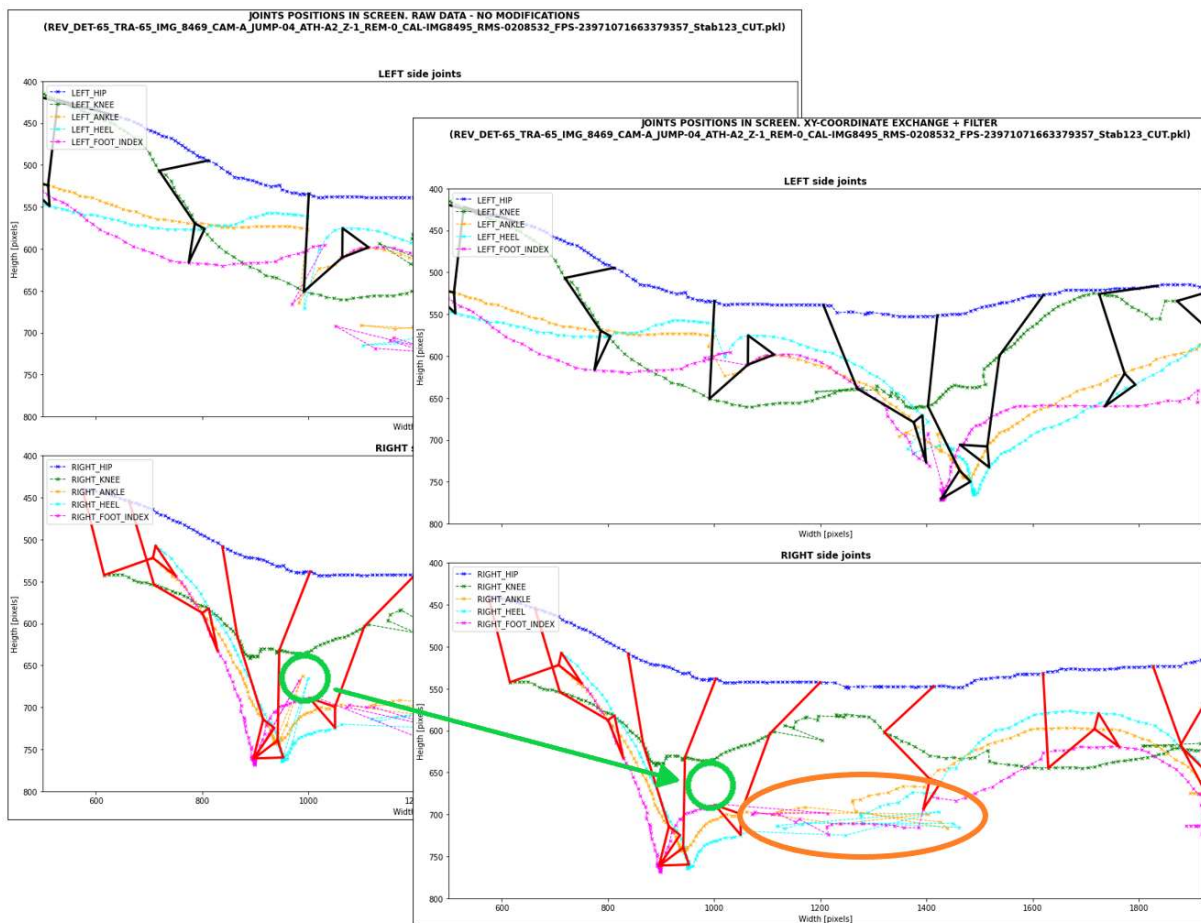
Liite 5 - Kuva 4. Kantapäiden, nilkkojen ja iso varpaiden avainpisteiden sijainnit eivät oikeilla paikoilla viimeisen kontaktin aikana. Kuva: Kalle Lehtonen



Liite 5 - Kuva 5. Kaikkien avainpisteiden koordinaatit katoavat kuvaruudussa 89, jonka jälkeen kuvaruudussa 90 osa avainpisteistä tunnistetaan taustasta. Kuva: Kalle Lehtonen



Liite 5 - Kuva 6. Vapaan jalan nilkan avainpisteiden koordinaatti alkaa vaeltamaan. Myöhemmissä kuvavuodissa myös molempien jalkojen nilkan/kantapään/isovarpaan avainpisteet tunnistetaan väärästä kohtaa. Kuva: Kalle Lehtonen



Liite 5 - Kuva 7. Avainpisteiden alkuperäiset (vasemmalla) ja suodatetut (oikealla) xy-koordinaatit. Kuva: Kalle Lehtonen