

Opinnäytetyö (AMK)

Eetu Mäki

2026

Eetu Mäki

# Lentopallon pelinlukua edistävä sovellus



Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintäteknikka

2026 | 37

Eetu Mäki

## Lentopallon pelinlukua edistävä sovellus

Lentopallossa vastustajajoukkueen passarin suoritusten ennakointi on tärkeä taito, ja se vaatii torjujilta sekä puolustajilta näkövihjeiden nopeaa käsittelyä. Perinteiset menetelmät tämän taidon harjoittamiseksi perustuvat usein manuaaliseen videon tarkasteluun. Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa sovellus, joka automatisoi harjoitustilanteiden luomisen otteluvideoista ja tilastotiedoista. Tavoitteena oli luoda prosessi, jonka avulla valmentajat ja pelaajat voivat muuntaa pelidataa päätöksentekovälineeksi ilman suurta manuaalista videon muokkausta.

Sovellus kehitettiin Python-ohjelmointikielellä sovelluksen logiikan ja graafisen käyttöliittymän osalta, ja se on integroitu R-laskentaympäristöön tilastollista tiedonkäsittelyä varten. Tietokonenäköä käytettiin otteluvideon automaattiseen segmentointiin. Nämä videosegmentit synkronoitiin DataVolley-tiedostojen kanssa, jotka ovat lentopallon analytiikan standardimuoto. Mukana on erillinen manuaalinen kalibrointityökalu, jonka avulla käyttäjät voivat hienosäätää pysäytyskuvan juuri sille hetkelle, kun passari koskettaa palloa.

Tämän projektin päätuloksena on toimiva työpöytäsovellus ”Volleyball IQ Trainer”, joka yhdistää onnistuneesti video- ja tilastotiedot yhtenäiseksi harjoitusympäristöksi. Käyttäjät voivat suodattaa tiettyjä pelitilanteita, kuten täydellisiä vastaanottoja, ja osallistua pelillistettyyn simulaatioon, jossa videon toisto keskeytetään passarin koskettaessa palloa. Tuloksena saatiin urheiluanalytiikkasovellus, jota voidaan käyttää parantamaan kognitiivista harjoittelua lentopallossa.

Asiasanat:

Lentopallo, ohjelmistokehitys, tietokonenäkö, urheiluanalytiikka

Bachelor's | Abstract

Turku University of Applied Sciences

Information and communications technology

2026| 37

Eetu Mäki

Application to develop volleyball players game reading skills

Reading the opposing team's setter is a key skill in high level volleyball. It requires blockers and defenders to process visual cues quickly. Traditional methods for practicing this skill rely on manual video review, which can be time consuming and does not provide immediate interactive feedback. The purpose of this thesis was to implement an application that automates the creation of interactive training clips from match videos and statistical files. The goal was to create a process that allows coaches and players to convert game data into decision making tool without much manual video editing.

The application was developed using Python programming language for its logic and graphical user interface, and it has been integrated to R environment for statistical data processing. Computer vision was used to automatically segment match videos into individual rallies by detecting changes in screen brightness. These video segments were synchronized with DataVolley files, the standard format for volleyball analytics, to obtain accurate data on the quality of receptions and distribution of sets. A separate manual calibration tool is included, allowing users to fine tune the pause frame to the exact moment the setter touches the ball.

The outcome of this project is a desktop application called "Volleyball IQ Trainer", which successfully combines video and statistical data into a singular training environment. Users can filter specific game situations, such as perfect receptions, and participate in a gamified simulation where the video pauses when the setter touches the ball. This app demonstrates that sports analytics software can be used to improve cognitive training in volleyball.

Keywords:

Volleyball, software development, computer vision

# Sisältö

<b>Sanasto</b>	<b>6</b>
<b>1 Johdanto</b>	<b>8</b>
1.1 Tausta ja aiheen esittely	8
1.2 Motivaatio ja merkityksellisyys	8
1.3 Tavoitteet ja rakenne	9
<b>2 Teoreettinen tausta</b>	<b>10</b>
2.1 Havainto-kognitiiviset taidot lentopallossa	10
2.2 Lentopalloanalytiikka ja DataVolley	12
2.3 Automatisoitu videonkäsittely	15
2.4 Pythonin ja R:n integrointi urheiluanalytiikkaan	16
<b>3 Sovelluksen suunnittelu ja toteutus</b>	<b>19</b>
3.1 Arkkitehtuuri	19
3.2 Automatisoitu videonkäsittely	22
3.3 Tietojen jäsentäminen ja integrointi	23
3.4 Tietojen synkronointi	24
<b>4 Käyttöliittymä ja vuorovaikutus</b>	<b>27</b>
4.1 Hallintapaneeli	27
4.2 Kalibrointityökalu	28
4.3 Koulutusmoduuli	30
<b>5 Sovelluksen arviointi ja tulevaisuus</b>	<b>32</b>
<b>Lähteet</b>	<b>33</b>
<b>Kuvat</b>	
Kuva 1. Vastaanoton laatu. (Reynaud, 2015)	14

Kuva 2. Volleyball IQ Trainer -sovelluksen arkkitehtuuri ja tietojenkäsittelyputki.	20
Kuva 3. Volleyball IQ Trainer-sovelluksen hallintapaneeli.	27
Kuva 4. Manuaalinen kalibrointityökalu (manual_adjuster.py).	29
Kuva 5. Koulutusmoduuli (trainer.py).	30
Kuva 6. Koulutusmoduuli esimerkki virheellisestä valinnasta.	31

## **Koodit**

Koodi 1. Esimerkki synkronointimoduulin luomasta final_data.json-tietokannasta.	25
---	----

## **Taulukot**

Taulukko 1. Lentopallon kosketukset.	11
Taulukko 2. Vastaanoton laatu.	14

## Sanasto

DataVolley (.dvw)	Data Projectin kehittämä oma ohjelmistopaketti ja tiedostostandardi, jota käytetään laajalti ammattilaislentopallossa tilastolliseen analyysiin ja pelaajien etsimiseen. .dvw-tiedostomuoto tallentaa pelitapahtumat standardoidulla aakkosnumeerisella syntaksilla. (Data Project S.r.l, 2026)
Graafinen käyttöliittymä (GUI)	Visuaalinen käyttöliittymä, jossa käyttäjä käyttää järjestelmää graafisten elementtien esim. painikkeet, ikkunat) kautta komentorivin sijaan. (Rogers ym., 2015)
Havainnointi-kognitiiviset taidot	Urheilijan kyky paikantaa, tunnistaa ja käsitellä ympäristön vihjeitä ja yhdistää ne olemassa olevaan tietoon ja motorisiin kykyihin sopivan toiminnon valitsemiseksi ja toteuttamiseksi. (Marteniuk, 1984)
Okluusio	Urheilutieteiden tutkimus- ja harjoittelumalli, jossa visuaalinen informaatio poistetaan (peitetään) valikoivasti tietyissä ajankohdissa – tyypillisesti pallon kosketushetkellä tai sitä ennen – urheilijan ennakointikykyjen arvioimiseksi ja harjoittelemiseksi. (Farrow & Abernethy, 2002)
OpenCV	Avoimen lähdekoodin kirjasto kuvankäsittelyyn ja tietokonenäköön, jota käytetään esimerkiksi videoiden analysointiin. (OpenCV, 2026)
PyQt6	Python-ohjelmointikielelle tarkoitettu QT-kirjastoon pohjautuva graafisten käyttöliittymien kehys. (Riverbank Computing, 2026)
QThread	PyQt6-kehysen luokka, jota käytetään säikeiden hallintaan ja rinnakkaisen suorituksen toteuttamiseen.

re	regular expression on pythonin standardikirjaston moduuli, jota käytetään tekstin käsittelyyn säännöllisten lausekkeiden avulla. Sen avulla voidaan etsiä, tunnistaa ja muokata merkkijonoja määriteltyjen hakumallien perusteella.
Tietokonenäkö	Monitieteinen tieteenala, joka käsittelee sitä, miten tietokoneet voivat saavuttaa korkeatasoisen ymmärryksen digitaalisista kuvista tai videoista. Tämän opinnäytetyön kontekstissa se viittaa videokehysten automaattiseen analysointiin kohtausten siirtymien havaitsemiseksi. (Szeliski, 2022)
Python	Ohjelmointikieli, jota käytetään laajasti sovelluskehityksessä, automaatioissa, data-analyysissä, tekoälyssä ja web-sovelluksissa. Se tunnetaan selkeästä syntaksistaan ja laajasta kirjastoalikoimasta. (Python Software Foundation, 2026)
R	Tilastolliseen laskentaan ja datan visualisointiin suunniteltu ohjelmointikieli. Sitä käytetään erityisesti tutkimuksissa, data-analyysissä ja tilastotieteessä suurten aineistojen käsittelyyn ja analysointiin. (The R Foundation, 2026)
JSON	JavaScript Object Notation on kevyt tekstimuotoinen tiedoston esitystapa (dataformaatti), jota käytetään tiedon siirtämiseen järjestelmien välillä. Se on ihmiselle helposti luettavaa ja koneelle helposti käsiteltävää, minkä vuoksi sitä käytetään yleisesti web-sovelluksissa ja API rajapinnoissa. (Crockford, 2026)

# 1 Johdanto

## 1.1 Tausta ja aiheen esittely

Korkeatasoisessa lentopalloissa pelin nopeus on kasvanut, mikä asettaa valtavan paineen joukkueiden puolustustaktiikoille. Puolustuksen kriittinen osa on torjujien kyky ennakoida vastustajan passaria eli käytännössä ennustaa kelle vastustajan hyökkääjistä pallo menee seuraavaksi ennen kuin se lähtee passarin käsistä. Tämä havaintokognitiivinen taito, jota usein kutsutaan ”passarin lukemiseksi”, erottaa huippupelaajat keskiverroista.

Tämä opinnäytetyö keskittyy Volleyball IQ Trainer -ohjelmiston kehittämiseen, joka on suunniteltu automatisoimaan näiden päätöksentekotilanteiden harjoittelu. Ohjelma muuttaa passiivisen ottelun katsomisen aktiiviseksi harjoitteluksi yhdistämällä raaka kuvamateriaali ja otteludatan interaktiiviseksi harjoittelukokemukseksi. Sovellus segmentoi automaattisesti otteluvideon pallorallien mukaan, jäsentää DataVolley (.dvw)-tiedostot pelitilanteiden tunnistamiseksi ja yhdistää nämä tietolähteet yhtenäiseksi harjoittelumoduuliksi.

## 1.2 Motivaatio ja merkityksellisyys

Tein sovelluksen, jotta joukkueellani olisi mahdollisuus kehittyä entistä syvällisemmin pelin luvussa. Yhdistämällä tilastollisen analyysin ja otteluvideot suunnittelin pelillisen harjoitusympäristön. Tavoitteena oli mennä perinteistä analyysia pidemmälle ja luoda interaktiivinen kokemus, joka terävöittää päätöksentekoa kentällä. Videon keskeyttäminen kriittisellä hetkellä päätöksen tekemisen pakottamiseksi on todistetusti tehokas menetelmä reaktioajan parantamiseksi (Farrow & Abernethy, 2002). Näiden skenaarioiden luominen manuaalisesti vaatii kuitenkin valmentajilta satoja videoleikkeiden leikkaamista ja manuaalista merkitsemistä.

### 1.3 Tavoitteet ja rakenne

Tämän opinnäytetyön ensisijaisena tavoitteena on suunnitella ja toteuttaa puoliautomaattinen ohjelmistoputki, joka muuntaa raakamuotoiset lentopallopelitiedot interaktiiviseksi päätöksentekokoulutustyökaluksi.

1. Automaatio: Kehitetään Smart Slicer -työkalu, joka käyttää tietokoneen näkötekniikoita (kirkkauden kynnsarvo) segmentoidakseen jatkuvan pelivideon automaattisesti yksittäisiksi palloralleiksi.
2. Integraatio: Luodaan analysointimoduuli, joka lukee .dvw-tiedostoja ja suodattaa tiettyjä taktisia skenaarioita (esim. kotijoukkue, täydellinen vastaanotto) käyttämällä R-kieltä ja kartoittaa nämä tilastotiedot vastaaviin videoleikkeisiin.
3. Vuorovaikutus: Rakennetaan käyttöliittymä, joka mahdollistaa pysäytyskuvan manuaalisen tarkentamisen, sekä pelillistetty harjoitusmoduuli, joka testaa käyttäjän kykyä ennustaa pallon suunta (vasen, keskelle, oikea).

Tutkimuskysymykset ovat seuraavat:

- Kuinka tietokoneen näkötekniikoita voidaan soveltaa lentopallopelien videomateriaalin segmentointiin?
- Kuinka standarditilastotiedostot (.dvw) voidaan jäsentää ja synkronoida videon kanssa taktisten tulosten automaattista merkitsemistä varten?
- Mikä on tehokkain ohjelmistoarkkitehtuuri näiden elementtien yhdistämiseksi käyttäjäystävälliseksi työpöytäsovellukseksi?

Luvussa 2 esitetään lentopalloanalyysin ja videonkäsittelyn teoreettinen tausta. Luvussa 3 kuvataan sovellusarkkitehtuurin toteutusta, mukaan lukien Python-R-integraatio. Luvussa 4 kuvataan hallintapaneelin sekä kalibrointi- ja koulutusmoduulien käyttöliittymien toiminnallisuus. Lopuksi luvussa 5 käsitellään ohjelmiston tuloksia, rajoituksia ja tulevaisuuden potentiaalia. Opinnäytetyössä on käytetty tekoälyä ideoinnissa, tiedonhankinnassa sekä kielenhuollossa.

## 2 Teoreettinen tausta

### 2.1 Havainto-kognitiiviset taidot lentopallossa

Lentopallo on erillisiin palloralleihin perustuva peli, jossa joukkueilla on kolme kosketusta aikaa palauttaa pallo vastustajan puolelle. Kosketusten rakenteellinen ja funktionaalinen jaottelu on esitetty taulukossa 1. Palloa ei voi pitää hallussa, minkä vuoksi tilanteet kehittyvät nopeasti. Lentopalloa luonnehtivat suuret tilannenopeudet sekä pelaajiin kohdistetut tiukat aikapaineet. Huipputasolla pelattavissa otteluissa pallon lähtönopeus ylittää usein 100 km/h, joten puolustajalla on vastustajan liikkeiden havaitsemiseen, käsittelyyn ja reagoimiseen käytettävissä usein vähemmän aikaa kuin yksinkertaisen reaktion suorittamiseen tarvitaan. Näin ollen huipputason suorituskykyä eivät määritä pelkästään fyysiset ominaisuudet, kuten pituus tai hyppyvoima, vaan myös havaintokognitiiviset taidot. Erityisesti kyky ennakoida tulevia tapahtumia ennen niiden toteutumista. Tämä kyky lukea peliä antaa taitaville pelaajille mahdollisuuden asettautua oikealle paikalle ennen kuin pallon lentorata on täysin selvillä (Maldonado ym., 2014).

Taulukko 1. Lentopallon kosketukset.

Suoritustapa	Tyypillinen kosketusjärjestys	Ensisijainen käyttö	Taktinen tavoite
<b>Hihalyönti</b>	1	Vastaanotto ja puolustus	Hyökkäyksen hallittu vaimentaminen ja pallon suunnan kontrollointi
<b>Sormilyönti</b>	2	Passipeli	Tarkka ajoitus ja pallon optimaalinen sijoittelu hyökkäyksen valmisteluun
<b>Iskulyönti</b>	3	Hyökkäys	Korkea nopeus ja suuntaus tai tarkoituksenmukainen sijoittelu
<b>Torjunta</b>	0	Puolustus verkolla	Pallon radan peittäminen tai hidastaminen

Huom. Torjuntaa ei lasketa joukkueen kolmen kosketuksen kiintiöön, minkä vuoksi se on merkitty numerolla 0.

Puolustustilanteessa vastustajan passari on ensisijainen lähde etukäteisille näköhavainnoille. Passari päättää hyökkäyksen jakautumisen valitsemalla antaako hän pallon laita- vai keskihyökkääjälle. Mitä myöhemmin passari aloittaa liikesuunnan muutoksen ennen passisuorituksen toteutumista, sitä vaikeampaa torjujien ja puolustajien on ennakoida tulevaa. Tämä vähentää huomattavasti torjunnan ja puolustuksen onnistumisen todennäköisyyttä. Tutkimukset ovat osoittaneet, että ammattilaispelaajat erottuvat aloittelijoista kyvyllään hyödyntää varhaisia näkövihjeitä, kuten kehon asentoa, ennakoidakseen passin suunnan ennen kuin pallon lentorata on tiedossa (Piras

ym., 2014). Tämä ennakointikyky vähentää pallon lentoradan aikana tehtävien päätösten epävarmuutta ja helpottaa tehokkaampaa motorista suorittamista.

Tällaisten ennakoivien taitojen kehittäminen on perinteisesti perustunut kentällä kertyneeseen kokemukseen. Tämä prosessi on kuitenkin aikaa vievää ja fyysisesti vaativaa. Havaintokognitiivisen kehityksen nopeuttamiseksi tutkijat ovat ottaneet käyttöön videopohjaisia koulutusparadigmoja, jotka on suunniteltu eristämään ja tehostamaan päätöksentekoprosesseja. Yksi yleisimmin käytetyistä lähestymistavoista on ajallinen okklusioparadigma (Farrow & Abernethy, 2002). Tässä menetelmässä urheilulajikohtaisten liikkeiden videoklippejä pysäytetään järjestelmällisesti kriittisissä kohdissa, tyypillisesti juuri ennen pallokosketusta tai pallon kosketuksen hetkellä, jolloin urheilijan on ennustettava lopputulos pelkästään ennen kosketusta saatujen tietojen perusteella.

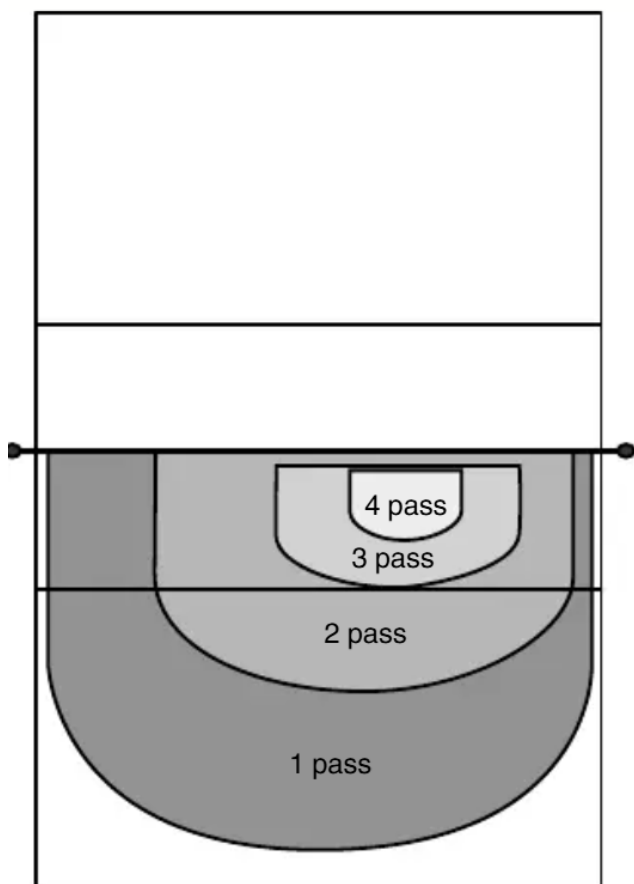
Ajallisen okklusioharjoittelun tehokkuus perustuu sen kykyyn erottaa havainnollinen käsittely ilman fyysisiä suorituksia. Pakottamalla urheilijat luottamaan ennakkoon saataviin kinemaattisiin tietoihin pallon liikeradan sijaan, vahvistavat havainnoin herkkyyttä ja asiantuntemukseen liittyviä toistuvien liikkeiden tunnistamisprosesseja. Systemaattinen katsaus ja meta-analyysi, jossa tutkittiin päätöksentekokoulutusta lentopallossa, osoitti merkittävää parannusta päätöksentekokyvyssä havaintokognitiivisten koulutusohjelmien jälkeen (Suárez ym., 2020). Tämä teoreettinen viitekehys on perustana tässä opinnäytetyössä kehitetyn sovelluksen suunnittelulle, joka automatisoi okklusiopohjaisten tilanteiden luomisen, jotta voidaan toteuttaa havainnointikoulutusta ilman fyysistä väsymystä.

## 2.2 Lentopalloanalytiikka ja DataVolley

Nykyaikaisen lentopallon taktinen monimutkaisuus edellyttää suoritusten tarkkaa tilastollista analyysia sekä oman joukkueen, että vastustajan pelin arvioimiseksi. Viimeisen kahden vuosikymmenen aikana lentopallon tilastointi on standardoitunut italialaisen yrityksen DataProjectin kehittämän DataVolley-

ohjelmistopakettien yleisen käyttöönoton ansiosta. DataVolley toimii ammattilaislentopalloliigojen ja maajoukkueiden globaalina standardina. Ohjelmiston avulla tilastojat, voivat tallentaa jokaisen pallokosketuksen ottelun aikana käyttämällä sovellukseen kehitettyä omaa aakkosnumeerista syntaksia. Tilastointiprosessin tuloksena syntyy tekstitiedosto, jonka tiedostotunniste on .dvw. Tämä tiedosto toimii koko ottelun kronologisena tietokantana. Jokainen rivi vastaa tiettyä tapahtumaa tai kosketusta, kuten syöttöä, vastaanottoa, passia tai hyökkäystä. DataVolleyin syntaksi on suunniteltu tallentamaan laajat yksityiskohdat kompaktissa muodossa. Vakiokoodijono eli yksi rivi määrittelee joukkueen, pelaajan numeron, tehdyn suorituksen ja suorituksen laadun (Data Project S.r.l, 2026).

Syntaksin olennaisin osuus on suorituksen laadun määritelmä. Esimerkiksi syötön vastaanotossa arvosana määrää, mitä taktisia vaihtoehtoja passarilla on käytettävissä. Vastaanoton laatuluokitus on esitetty kuvassa 1. sekä taulukossa 2. Täydellinen vastaanotto, joka on merkitty symbolilla (#) tarkoittaa, että passari voi käyttää kaikkia hyökkäysvaihtoehtoja. Positiivinen vastaanotto (+) tai OK vastaanotto (!) rajoittaa hyökkäystä vaihtelevassa määrin. Yleensä nämä laadut annetaan, kun keskihyökkäys ei ole enää mahdollinen. Negatiivinen vastaanotto (-) vaikeuttaa hyökkäyksen organisointia merkittävästi, ja virhe (=) merkitsee pelin katkeamista tai välitöntä pisteen menetystä. (Reynaud, 2015)



Kuva 1. Vastaanoton laatu. (Reynaud, 2015)

Taulukko 2. Vastaanoton laatu.

Laatu	Merkki	Selitys
Täydellinen	#	Kaikki hyökkäysvaihtoehdot
Positiivinen	+	Kaikki hyökkäysvaihtoehdot, mutta ei optimaalinen paikka
OK	!	Keskihyökkäys ei enää pelattavissa
Negatiivinen	-	Vaikea tilanne
Virhe	=	Suora virhe

Pelkän tekstidatan kirjaamisen lisäksi DataVolleyn-työnkulku sisältää näiden tilastokoodien synkronoinnin digitaalisen videon kanssa. Live ottelun aikana tilastoija painaa pikanäppäintä juuri, kun syöttö tapahtuu, ja luo aikaleiman, joka linkittää tilastotapahtuman vastaavaan videoklippiin. Tämä synkronointi muuttaa .dvw-tiedoston yksinkertaisesta taulukosta helposti haettavaksi multimediatiedostoksi, jonka avulla valmentajat voivat hakea tiettyjä tilanteita, kuten kaikki tietyn pelaajan hyökkäykset.

Vaikka sovellus on erittäin tehokas ottelun jälkeiseen analysointiin ja taktiseen valmistautumiseen, se on edelleen pääasiassa analyysityökalu. Näiden synkronoituneiden tapahtumien vieminen interaktiivisiin muotoihin vaatii huomattavaa manuaalista työtä. Tässä opinnäytetyössä kehitetty sovellus hyödyntää .dvw-tiedoston standardoitua rakennetta. Poimimalla tilastoijan kirjaamat tietyt koodit (kuten # ja + vastaanotot) sekä passijakaumaa kuvaavia ”setters call”-merkintöjä, sovellus osaa lajitella videoleikkeet omiin kategorioihin. Näin analyysiprosessia voidaan tehostaa hyödyntämällä olemassa olevaa dataa ilman, että itse pelivideon sisältöä tarvitsee analysoida erillisillä menetelmillä.

### 2.3 Automatisoitu videonkäsittely

Digitaalinen video koostuu peräkkäisistä kuvista, jotka on otettu vakionopeudella. Jokainen kuva voidaan esittää kaksiulotteisena pikselien intensiteettiarvojen taulukkona, joka ilmaisee luminanssi- ja väritietoja, mikä mahdollistaa visuaalisen sisällön laskennallisen analyysin (Szeliski, 2022). Pitkissä urheilutallenteissa, kuten lentopallon kuvamateriaalissa, tämä tarkoittaa kymmeniä tuhansia kuvia, jotka on käsiteltävä taktisesti merkityksellisten osien erottamiseksi.

Yleisessä videon analysoinnissa jatkuva kuvamateriaalin jakaminen pienempiin osiin tapahtuu videosegmentointitekniikoilla. Alan keskeinen tehtävä on kuvan rajapisteiden tunnistus (Shot Boundary Detection, SBD), jonka tarkoituksena on tunnistaa peräkkäisten kamerakuvien siirtymät (Szeliski, 2022). Klassisissa

menetelmissä havaitaan äkilliset tai vaihteittaiset siirtymät mittaamalla vierekkäisten kuvien välisiä visuaalisia katkoja, esimerkiksi histogrammien vertailujen tai intensiteetin muutosten avulla (Zabih ym., 1995).

Kiinteän kuvakulman urheiluvideoissa segmentointi ei kuitenkaan yleensä perustu televisio-ohjelmien tyyppisiin kuvakulmanvaihtoihin, koska kuvamateriaali voi koostua yhdestä jatkuvasta kameranauhoituksesta. Sen sijaan segmentointi voi perustua tallenteeseen sisältyviin näkövihjeisiin. Tässä työssä käytetty lentopallo-ottelun ottelukohtaisessa videomateriaalissa näyttö muuttuu mustaksi pallojen välissä, mikä tarjoaa luonnollisen rajan pelisegmenttien välillä. Visuaalinen muutos voidaan havaita laskennallisesti analysoimalla kunkin kuvan keskimääräinen pikselien intensiteetti. Koska mustat kuvat sisältävät hyvin alhaiset luminanssiarvot, kynnyksarvoon perustuvaa menetelmää voidaan käyttää aktiivisen pelin ja taukojen välisten siirtymäkohtien automaattiseen tunnistamiseen.

Tällaiset intensiteettiin perustuvat lähestymistavat ovat laskennallisesti tehokkaita ja helppoja toteuttaa. Kynnyksarvoihin perustuvat tekniikat voivat kuitenkin olla herkkiä valaistuksen vaihteluille tai osittaisille näkymän peitteille. Näistä rajoituksista huolimatta ne tarjoavat käytännöllisen ja teoreettisesti perustellun menetelmän automatisoituun segmentointiin urheilutallenteissa.

## 2.4 Pythonin ja R:n integrointi urheiluanalytiikkaan

Urheiluanalytiikan monimutkaisuus on johtanut siirtymiseen perinteisistä taulukkolaskentaohjelmista kohti kehittyneitä ohjelmistoympäristöjä, jotka pystyvät käsittelemään suuria tietojoukkoja, tilastollista mallintamista ja multimediasäveltelyä. Näistä ympäristöistä Python ja R ovat nousseet kahdeksi eniten käytetyksi ohjelmointikieleksi datapohjaisessa urheiluanalyysissä. Vaikka niitä pidetään usein kilpailevina työkaluina laajemmalla datatieteen alalla, niiden tekniset vahvuudet eroavat toisistaan, joka tekee niiden yhdistelmäkäytöstä erityisen hyödyllisen analyttisissä sovelluksissa. Urheiluanalytiikan käytännön

sovelluksissa on kehitetty työkulkuja, joissa yhdistyvät molemmat kielet tietojen käsittelyyn, mallintamiseen ja raportointiin (Eager & Erickson, 2023).

R on ohjelmointikieli, joka on kehitetty erityisesti tilastolliseen laskentaan ja data-analyysiin. Sen suurin vahvuus on laaja ekosysteemi, joka koostuu erilaisista paketeista, jotka on suunniteltu datan käsittelyyn, tilastolliseen mallintamiseen ja aluekohtaisiin analyttisiin tehtäviin. Urheiluanalytiikassa R:ää käytetään usein tilastotietojen jäsentämiseen, edistyneiden suorituskykymittareiden laskemiseen ja monimutkaisten tietojoukkojen suodattamiseen. Koska kieli on suunniteltu tilastotieteilijöiden toimesta, se tarjoaa natiivin tuen tilastollisille tietorakenteille, kuten datakehysille, sekä erittäin optimoidut työkalut tilastolliseen päättelyyn ja tutkimukselliseen data-analyysiin. Lukuisat urheiluanalytiikan työkalut ja tutkimusprosessit hyödyntävät R-paketteja urheilutietojen toistettavan tilastollisen analyysin helpottamiseksi (Nguyen ym., 2021). Näistä vahvuuksista huolimatta R ei yleensä sovellu kovin hyvin itsenäisten työpöytäsovellusten, graafisten käyttöliittymien (GUI) kehittämiseen tai matalan tason toimintojen, kuten laajamittaisen multimediakäsittelyn suorittamiseen.

Python on yleiskäyttöinen ohjelmointikieli, joka tunnetaan monipuolisuudestaan, luettavuudestaan ja laajasta kirjastojen valikoimasta. Python tukee myös laajaa valikoimaa tietojenkäsittelytieteen sovelluksia, mutta sen tärkein etu hybridianalyttisissä sovelluksissa on sen joustavuus sovelluskehitysalustana. Python on erityisen tehokas automaatioissa, tietokonenäön tehtävissä, kuten frame-by-frame-videon analysoinnissa, sekä yksinkertaisten interaktiivisten graafisten käyttöliittymien kehittämisessä. Urheilutieteellisessä tutkimuksessa on kehitetty Python-pohjaisia kehysrakenteita tukemaan suorituskyvyn analysoinnissa yleisesti käytettyjen ajan ja paikan seurantatietojen, tapahtumatietojen ja muiden monimutkaisten tietojoukkojen analysointia (Raabe ym., 2022).

Näiden kahden kielen toisiaan täydentävien vahvuuksien hyödyntämiseksi urheiluanalytiikkasovellukset voivat käyttää monikielistä arkkitehtuuria, jossa samassa sovelluksessa käytetään useita ohjelmointikieliä eri

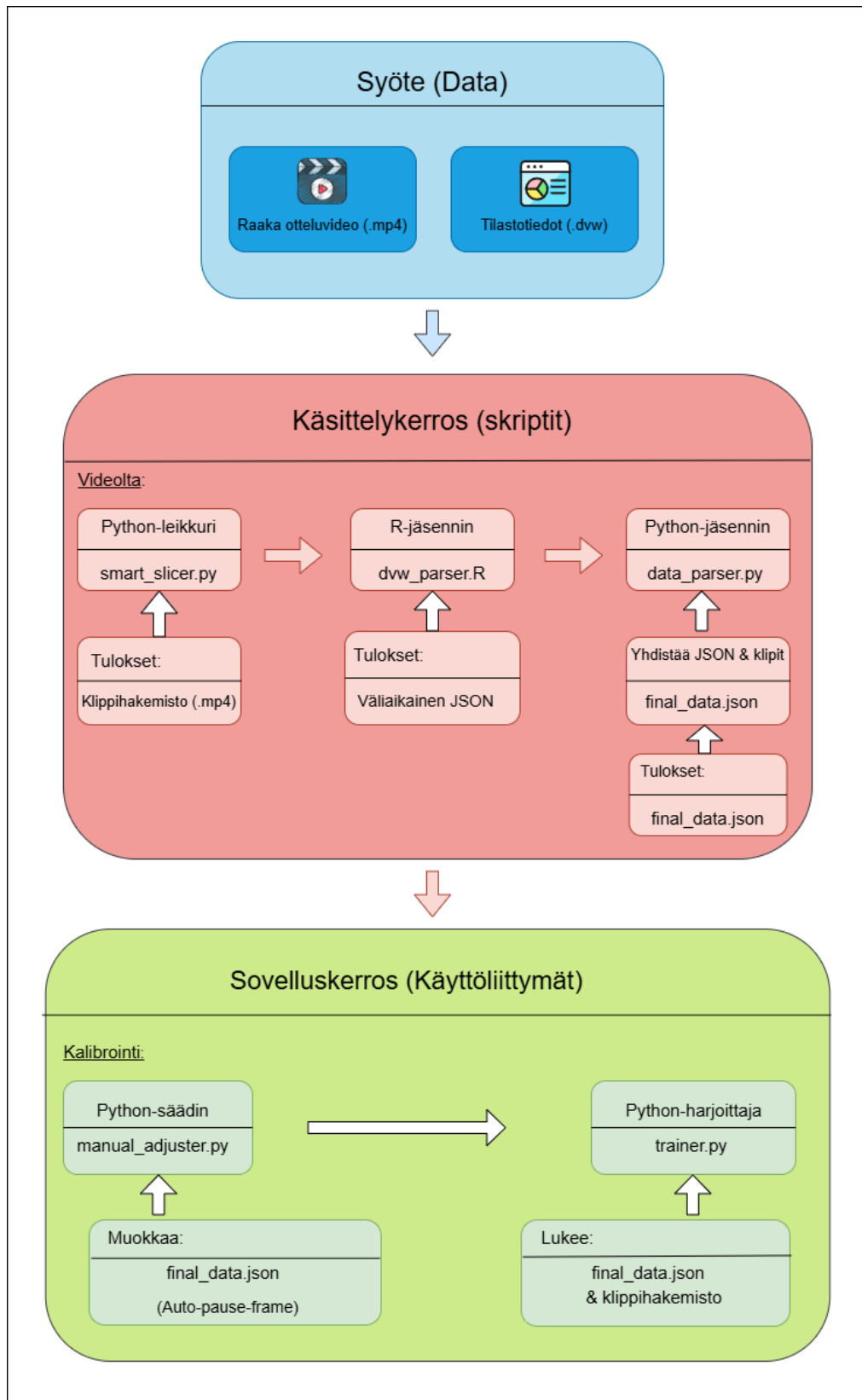
laskentavaatimusten täyttämiseksi. Tällaisissa sovelluksissa Python toimii usein ohjauskerroksena tai sovelluksen käyttöliittymänä, joka hallinnoi käyttäjän vuorovaikutusta, videonkäsittelyn työkaluja ja järjestelmätason tehtäviä. Laskennallisesti intensiiviset tilastolliset analyysit voidaan puolestaan delegoida taustalla toimiville R-skripteille. Tämän vastuunjaon ansiosta analyttiset sovellukset voivat yhdistää tilastollisen mallinnuksen kyvyt interaktiiviseen sovelluskehitykseen.

Koska Python ja R toimivat erillisissä suoritusympäristöissä ja muistialueissa, ne eivät voi suoraan jakaa muistissa olevia muuttujia. Sen sijaan tiedot siirretään kieliriippumattomien dataformaattien, kuten JSON (JavaScript Object Notation) tai CSV (Comma-Separated Values) kautta. Tyypillisessä työnkulussa Python-sovellus voi käynnistää R-skriptin tietojoukon käsittelyä, tietojoukon suodatusta tai laskelmien suorittamista varten ja viedä tulokset strukturoituna tiedostona. Python voi sitten lukea tämän tuloksen ja integroida tilastolliset tulokset muihin sovelluskomponentteihin, kuten synkronoituihin videotiedostoihin. Tällaisia työkaluja on hyödynnetty nykyaikaisissa urheiluanalyysisovelluksissa, joissa videopohjainen tapahtumien tunnistus ja tilastollinen analyysi on integroitava yhteen analyttiseen prosessiin (Wu ym., 2022). Tämä yhteistyöhön perustuva lähestymistapa mahdollistaa analyttisten työkalujen kehittämisen, joissa yhdistyvät tilastollinen tarkkuus sekä interaktiivisuus ja multimediamahtoisuudet.

## 3 Sovelluksen suunnittelu ja toteutus

### 3.1 Arkkitehtuuri

Volleyball IQ Trainer on suunniteltu nelivaiheiseksi tietojenkäsittelyputkeksi, jota hallitaan keskitetyn graafisen käyttöliittymän (GUI) avulla. Sovellus on kehitetty pääasiassa Python-ohjelmointikielellä PyQt6 kehystä hyödyntäen. PyQt6 on QT-kirjastoon perustuva käyttöliittymäkehys, jonka avulla voidaan toteuttaa graafisia työpöytäsovelluksia Pythonilla. Sovellus on suunniteltu ohjaamaan käyttäjää järjestelmällisesti raakadatan keräämisestä interaktiiviseen kognitiiviseen harjoitteluun. Kuvassa 2 esitetään sovelluksen arkkitehtuuri sekä datan kulku eri käsittelyvaiheiden läpi.



Kuva 2. Volleyball IQ Trainer -sovelluksen arkkitehtuuri ja tietojenkäsittelyputki.

Sovelluksen ytimenä toimii koordinointisovellus (main.py), joka toimii käyttäjän hallintapaneelina. Tämä hallintapaneeli valvoo prosessin etenemistä ja hallinnoi siirtymiä neljän erillisen toimintamoduulin välillä: videon pilkkominen, tietojen jäsentäminen, manuaalinen kalibrointi sekä lopullinen koulutusmoduuli. Jotta sovellus pysyy reagoivana resurssinkulutusta vaativien toimintojen, kuten videonkäsittelyn aikana, sovelluksessa käytetään asynkronista monisäikeistä lähestymistapaa. Tarkemmin sanottuna sovellus hyödyntää PyQt6-kehiksen QThread-luokkaa ja siirtää laskennallisesti raskaat OpenCV-kirjastoon perustuvat videonkäsittelytehtävät erilliselle taustatyöskentelysäikeelle. OpenCV on avoimen lähdekoodin tietokonenäköön ja kuvankäsittelyyn tarkoitettu kirjasto. Tämä erottelu estää pääsovelluksen jumittumisen, jolloin graafinen käyttöliittymä pysyy interaktiivisena videon käsittelyn tapahtuessa taustalla.

Sovelluksen keskeinen piirre on sen monikielinen rakenne, erityisesti Pythonin ja R:n integrointi eri laskentakohteiden käsittelyä varten. Python hoitaa yksinomaan multimediakäsittelyn ja käyttöliittymät, kun taas DataVolley-tiedostojen jäsentäminen siirretään erilliselle R-skriptille.

Tiedot kulkevat sovelluksen läpi sekvensointiprosessin kautta. Kun käyttäjä käynnistää analysointivaiheen, Pythoniin perustuva koordinaattori suorittaa R-skriptin aliprosessina ja välittää käyttäjän määrittämät parametrit kuten kohdejoukkueen ja toivotun vastaanoton laadun komentoriviparametreina. R-ympäristö (dvw\_parser.R) käsittelee DataVolley tiedoston, käyttää suodattimia ja vie tarvittavat tilastotiedot väliaikaiseen JSON-tiedostoon.

Python taustaprosessi (data\_parser.py) lukee tämän väliaikaisen JSON-tiedoston suorittaakseen tarvittavan jälkikäsittelyn. Tähän sisältyy tiettyjen hyökkäyskoodien muuntaminen standardoiduiksi kentän suuntamääreiksi ("vasen", "keski" tai "oikea") sekä kentän puolen vaihtamisen logiikan määrittäminen lentopalloerän numeron perusteella. Lopuksi sovellus yhdistää nämä jalostetut tilastotiedot luotuun videoleikekansioon ja kokoaa yhtenäisen final\_data.json tietokannan. Tämä yksittäinen JSON-tiedosto toimii pysyvänä

tietorakenteena, joka linkittää tietyt videotiedostot niiden tuloksiin ja ohjaa sekä kalibrointi- että harjoittelumoduuleja.

### 3.2 Automatisoitu videonkäsittely

Automatisoidun videon segmentoinnin toteutuksesta huolehtii `smart_slicer.py` -moduuli. Skripti on vastuussa raakamuotoisen ottelumateriaalin syöttämisestä sovellukselle ja hakemiston tuottamisesta, joka sisältää erillisiä videoklippejä. Toiminto perustuu Python-kieliseen OpenCV-kirjastoon, joka käsittelee videotiedostoa kuva kerrallaan.

Koska korkean resoluution kuvien suora käsittely on laskennallisesti raskasta, etenkin kun arvioidaan pikselikohtaista kirkkautta jokaisessa kuvassa, ennen kuvan visuaalisten ominaisuuksien analysointia skripti pienentää kuvan resoluution huomattavasti alhaisemmaksi 64x64 pikseliksi. Tämä pienennys vähentää käsittelykuormitusta säilyttäen samalla siirtymien havaitsemiseen tarvittavat kirkkaustiedot. Sovellus käyttää tämän jälkeen NumPy-kirjastoa laskeakseen tämän pienemmän kuvan keskimääräisen pikselien intensiteetin, jolloin saadaan yksi numeerinen arvo, joka edustaa nykyisen kuvan kokonaiskirkkautta.

Kuvan rajapisteiden tunnistusalgoritmi toimii kynnyсарvologiikan mukaan, joka perustuu aiemmin laskettuun keskiarvoon. Sovellus vertaa kuvan kirkkautta ennalta määriteltyyn vakionuotoiseen arvoon, `BLACK_THRESHOLD`. Jos kuvan kirkkaus ylittää tämän kynnyсарvon, se luokitellaan aktiiviseksi pelitoiminnaksi ja lisätään väliaikaiseen muistiin. Mustaksi haalistuva siirtymä pallorallin lopussa aiheuttaa keskimääräisen kuvan kirkkauden laskun alle kynnyсарvon, mikä merkitsee aktiivisen pelitoiminnan loppua.

Tuloksen luotettavuuden varmistamiseksi ja virheellisten lyhyiden videopätkien syntymisen estämiseksi tallennuslogiikkaan on lisätty aikarajoitus. Näitä virheellisiä videoita voivat aiheuttaa esimerkiksi lyhyet valaistuksen vaihtelut, varjot tai kameran liikkeet. Skripti tarkistaa nykyisen videopuskuriin tallennetun materiaalin pituuden suhteessa `MIN_CLIP_DURATION` -vakioon, joka on

määritelty arvoksi 3,0 s. Jos puskurissa on tarpeeksi kuvia tämän keston ylittämiseksi, sekvenssi kootaan ja tallennetaan paikalliselle levyllä itsenäisenä .mp4-tiedostona, minkä jälkeen puskuri tyhjennetään. Tämä logiikka toistuu, kunnes koko lähdevideo on käsitelty, jolloin tuloksena on siististi segmentoitu hakemisto yksittäisistä palloralleista.

### 3.3 Tietojen jäsentäminen ja integrointi

Taktisten tilastotietojen poiminta ja suodatus tavallisista DataVolley-tiedostoista toteutetaan Pythonin (Python Software Foundation, 2026) ja R:n integroinnilla (The R Foundation, 2026). `data_parser.py`-moduuli toimii tämän prosessin koordinaattorina ja käyttää Pythonin `subprocess`-moduulia suorittaakseen taustalla erillisen R-skriptin (`dvw_parser.R`). Python-moduuli välittää käyttäjän määrittämät parametrit kuten kohdejoukkueen (kotijoukkue tai vierasjoukkue) ja halutun vastaanoton laatuluokituksen esim. täydellinen tai positiivinen komentoriviparametreina R-ympäristöön.

Kun R-skripti käynnistetään, se hyödyntää `datavolley`-kirjastoa `.dvw`-tiedoston lukemiseen ja käsittelyyn. Jotta tilastotiedot voidaan kohdistaa tarkasti niihin liittyviin videoleikkeisiin, skriptin on ensin määriteltävä erilliset pallorallit. Tämä tapahtuu tunnistamalla syöttötapahtumat, määrittelemällä jokaiselle pisteelle sarjanumero `rally_id` ja luomalla vastaava klipin nimi (esim. `rally_001.mp4`). Sitten skripti erottelee passitapahtumat tallentaakseen passarin käyttämän "setters call" hyökkäyskoodin ja linkittää tämän koodin edeltävään vastaanotto tapahtumaan. Käsiteltyään käyttäjän antamat joukkue- ja laatusuodattimet R-skripti jäsentää jäljellä olevat tapahtumat datakehikseksi ja tallentaa sen väliaikaisena JSON-tiedostona.

Kun R-skripti on suoritettu loppuun, Python-moduuli lukee tämän väliaikaisen JSON-tiedoston suorittaakseen tarvittavan jälkikäsitteilyn. DataVolley-syntaksi sisältää laajennettuja aakkosnumeerisia merkkijonoja, joissa tietty hyökkäysyhdistelmä on upotettu muihin tapahtumatietoihin. Näiden tietojen luotettavaksi poimimiseksi Python-skripti käyttää `re`-moduulia. Algoritmi etsii

tapahtumamerkkijonosta merkkiä "K", jota seuraa välittömästi kaksi merkkiä (esim. poimitaan "K1C..." jonosta "1C").

Kun tämä kahden merkin hyökkäyskoodi on eristetty, sovellus vertaa sitä ennalta määritettyyn sanastoon (ATTACK\_COMBOS) ja muuntaa monimutkaisen merkintätavan helposti luettavaksi suunnaksi: "Left", "Middle" tai "Right". Varmistaakseen luotettavuuden algoritmi käyttää varasuunnitelmana tilannetta, jossa harvoin käytetty koodi puuttuu sanastosta. Se arvioi erotetun merkkijonon viimeisen merkin ja merkkäa "F":n vasemmalle, "B":n oikealle ja "C":n tai "P":n keskelle. Tämä kaksikerroksinen erottelu takaa, että jokainen suoritus luokitellaan tarkasti.

Tavallisessa lentopallopelissä joukkueet vaihtavat kentän puolta jokaisen erän jälkeen. Näin ollen, kun kameran puolella oleva joukkue passaa "vasemmalle" alueelle, se näkyy videossa vasemmalla puolella. Jos joukkue on kuitenkin toisella puolella verkkoa, sama "vasen" alue näkyy videossa menevän oikealle. Jotta merkinnät kuvastavat tarkasti kameran näkökulmaa, Python-skripti käyttää kääntöalgoritmia. Hakemalla eränumeron ja sen, onko vastaanottava joukkue koti- vai vierasjoukkue. Skripti määrittää kummalla puolen verkkoa joukkue on ja tarvittaessa kääntää "vasemman" oikeaksi ja päinvastoin. Keskikentän hyökkäykset jäävät ennalleen sillä niihin ei kohdistu horisontaalista muutosta puolten vaihtojen yhteydessä.

### 3.4 Tietojen synkronointi

Tietojenkäsittelyprosessin viimeinen vaihe on yksittäisten videoleikkeiden synkronointi niitä vastaavien tilastollisten tietojen kanssa. Koska videon segmentointi ja tilastollinen jäsentäminen tapahtuvat rinnakkain mutta erillisinä prosesseina, sovellus tarvitsee mekanismin, jolla rakenteettomat multimediatiedostot voidaan linkittää tilastotietoihin. Tämän toteuttamiseksi käytetään yhtenäistä nimeämiskäytäntöä, joka toimii tietokannan ensisijaisena avaimena. R-jäsennysvaiheen aikana jokainen kelvollinen palloralli indeksoidaan peräkkäin ja leike nimetään videotiedostonimeksi (esim.

rally\_002.mp4). Tämä tarkoittaa, että JSON-tiedoston avain on aina saman niminen kuin sitä vastaavan videotiedoston nimi (Koodi 1).

```
{
  "rally_002.mp4": {
    "clip_name": "rally_002.mp4",
    "rally_id": 2,
    "reception_grade": "Positive",
    "receiving_team": "home",
    "set_number": 1,
    "full_code": "*14RM+~~~16",
    "related_set_code": "*18ET+K1B",
    "set_type": "Right",
    "auto_pause_frame": 189
  },
  "rally_006.mp4": {
    "clip_name": "rally_006.mp4",
    "rally_id": 6,
    "reception_grade": "Positive",
    "receiving_team": "home",
    "set_number": 1,
    "full_code": "*04RM+~~~56",
    "related_set_code": "*18EQ+K1C",
    "set_type": "Middle",
    "auto_pause_frame": 194
  },
}
```

Koodi 1. Esimerkki synkronointimoduulin luomasta final\_data.json-tietokannasta.

Synkronointilogiikka suoritetaan Python-pohjaisen sovelluksen sisällä. Saatuaan suodatetun luettelon pelitilanteista data\_parser.py -moduulilta pääsovellus luo hajautustaulun (HashMap), jossa kukin tilastotieto on indeksoitu sille määritetyn videoleikkeen nimen mukaan. Tämän jälkeen sovellus hakee paikalliselta käyttöjärjestelmältä lajitellun luettelon kaikista .mp4-tiedostoista, jotka sijaitsevat tällä hetkellä automaattisen videoleikkurin luomassa kansiossa.

Käymällä läpi nämä fyysiset videotiedostot ja vertaamalla niiden tiedostonimiä tilastolliseen hajautustaulukkoon algoritmi suodattaa videotiedostot ja tilastotiedot. Jos videotiedostolle löytyy vastaava merkintä taulukosta, se tarkoittaa, että kyseinen palloralli täyttää käyttäjän ennalta määrittämät suodattimet (esim. kotijoukkueen täydellinen vastaanotto). Videot, joilla ei ole vastaavaa tilastollista osumaa kuten virheeseen johtaneet syötöt tai vierasjoukkueen vastaanotot jätetään huomiotta.

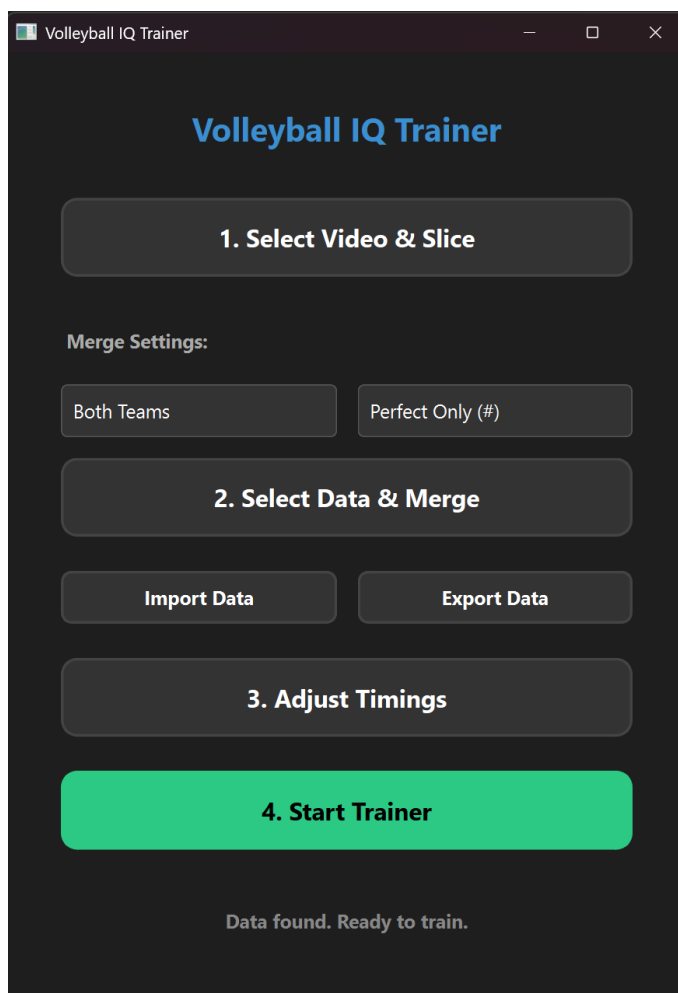
Yhdistämisvaiheen aikana sovellus lisää myös harjoitusmoduulissa tarvittavat parametrit. Tarkemmin sanottuna se liittää jokaiseen kelvolliseen merkintään `auto_pause_frame` muuttujan, jonka oletusarvona on 180 kuvaa. Tämä arvo toimii väliaikaisena aikaleimana pysäytyskuvaa varten ja määrittää pisteen, jonka käyttäjä voi myöhemmin kalibroida.

Lopuksi onnistuneesti synkronoidut tietoparit, jotka sisältävät videotiedoston nimen, vastaanottavan joukkueen, eränumeron, hyökkäyssuunnan ja pysäytysajankohdan, kootaan yhteen paikalliseksi tiedostoksi nimeltä `final_data.json`. Tuloksena oleva tiedosto noudattaa JSON-rakennetta, jossa jokainen videotiedostonimi toimii avaimena, joka viittaa siihen liittyvään tietoaalueeseen, kuten koodissa 1 on esitetty. Tämä yhtenäinen JSON-rakenne toimii sovelluksen paikallisena tietokantana. Suorittamalla kaikki laskennallisesti raskaat käsittelyt ja synkronoinnit etukäteen sovellus varmistaa, että graafiset kalibroinnit- ja koulutusmoduulit voidaan ladata välittömästi ja että ne toimivat ilman viiveitä.

## 4 Käyttöliittymä ja vuorovaikutus

### 4.1 Hallintapaneeli

Käynnistäessään Volleyball IQ Trainer-sovelluksen käyttäjälle avautuu hallintapaneeli ikkuna. Koska sovellus edellyttää toimintojen suorittamista kronologisessa järjestyksessä, hallintapaneeli on suunniteltu johdonmukaiseksi vaiheittaiseksi käyttöliittymäksi. Tämä suunnittelu vähentää käyttäjävirheitä ohjaamalla käyttäjää tarvittavien prosessivaiheiden läpi. Prosessin vaiheet järjestyksessä ovat: videon leikkaaminen, tietojen jäsentäminen, manuaalinen ajoituksen säätö sekä lopuksi harjoitusmoduulin käynnistäminen (Kuva 3).



Kuva 3. Volleyball IQ Trainer-sovelluksen hallintapaneeli.

Suurten multimediatiedostojen käsittelyssä merkittävä käyttökokemukseen (UX) liittyvä haaste on visuaalisen palautteen puute, mikä voi saada käyttäjän luulemaan, että sovellus on kaatunut. Kun tavallinen graafinen käyttöliittymä yrittää käsitellä videoita peräkkäin, sovelluksen pääsäie ylikuormittuu, jonka seurauksena ikkuna jumittuu.

Tämän ongelman ratkaisemiseksi hallintapaneeli hyödyntää reagoivaa järjestelmää, joka perustuu taustalla toimivaan WorkerThread prosessiin. Kun käyttäjä käynnistää "Select Video & Slice" -toiminnon, hallintapaneeli käynnistää välittömästi `lock_interface()` funktion. Tämä toiminto estää väliaikaisesti kaikkien interaktiivisten painikkeiden käytön, mikä estää käyttäjää käynnistämästä vahingossa samanaikaisia prosesseja.

Samalla sovellus muodostaa viestintäkanavan taustakäsittelysäikeen ja hallintapaneelin välille. Kun `smart_slicer.py` moduuli analysoi videota taustalla, se lähettää jatkuvasti väliaikatietoja. Hallintapaneeli vastaanottaa nämä tiedot ja päivittää näytöllä näkyvää etenemispalkkia. Tämä tarjoaa käyttäjälle reaaliaikaista palautetta prosessin etenemisestä. Kun prosessi on valmis, taustasäie lähettää lopetussignaalin, joka avaa käyttöliittymän, antaa onnistumisilmoituksen ja antaa käyttäjän siirtyä seuraavaan vaiheeseen. Tämä varmistaa sujuvan käyttökokemuksen kuormituksesta riippumatta.

## 4.2 Kalibrointityökalu

Vaikka automatisoitu videonkäsittelyprosessi erottelee tehokkaasti yksittäiset pallorallit, havaintotaitojen harjoittelun tehokkuus riippuu millisekun tarkkuudesta. Ennakoivien taitojen tarkkaan harjoitteluun video on pysäytettävä juuri sillä hetkellä, kun passari koskee palloon, joka pakottaa käyttäjän tulkitsemaan näkövihjeitä pallon lentoradan sijaan. Koska lentopallon syötön ja vastaanoton kesto vaihtelevat, kiinteästi koodattu pysäytysaika (esim. 3 sekuntia) ei toimi. Tämän vaihtelevuuden ratkaisemiseksi sovelluksessa on oma manuaalinen kalibrointityökalu `manual_adjuster.py` (Kuva 4).



Kuva 4. Manuaalinen kalibrointityökalu (manual\_adjuster.py).

Kalibrointityökalu toimii tietokantaeditorina. Käynnistyessään moduuli lukee final\_data.json-tiedoston, jotta se osaa yhdistää videoklipit niitä vastaaviin tietoihin. Graafinen käyttöliittymä on toteutettu käyttämällä OpenCV:tä kuvien poimimiseen ja PyQt6:n QLabel- ja QPixmap-luokkia kuvien renderöimiseen sovellusikkunassa.

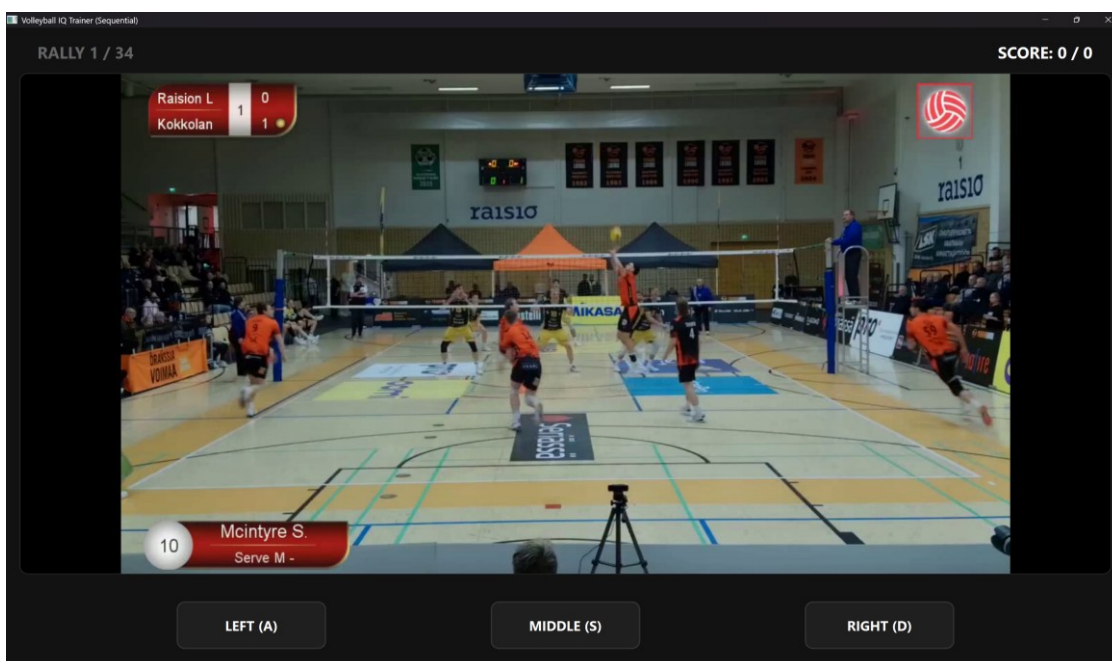
Tarvittavan tarkkuuden saavuttamiseksi käyttöliittymä poikkeaa tavallisesta videoistotoiminnosta. Jatkuvan toiston sijaan videota selataan kuva kerrallaan. Nykyisen kuvan numero näkyy ruudulla reaaliaikaisesti. Käyttäjän tavoitteena on siirtyä siihen kuvaan, jossa passari koskettaa palloa.

Kun oikea kuva on löydetty, käyttäjä painaa "Enter" -näppäintä, joka tallentaa kuvan ja siirtyy seuraavaan. Tämä tallennustoiminto päivittää kyseisen videoleikkeen auto\_pause\_frame-muuttujan sovelluksen muistissa. Kun haluttujen videoleikkeiden kalibrointi on valmis, sovellus korvaa final\_data.json -

tiedoston päivitettyillä auto\_pause\_frame-muuttujilla. Manuaalinen kalibrointi varmistaa, että harjoitusmoduuli toimii halutulla tarkkuudella.

#### 4.3 Koulutusmoduuli

trainer.py -skriptissä toteutettu harjoitusmoduuli on Volleyball IQ Trainer sovelluksen ydin. Se on käyttöliittymä, joka on suunniteltu helpottamaan aktiivisesti havaintokognitiivista harjoittelua. Kun raakavideo on käsitelty ja tiedot synkronoitu, tämä moduuli muuntaa jäsenneilyn tietokannan pelillistetyksi oppimisympäristöksi (Kuva 5).



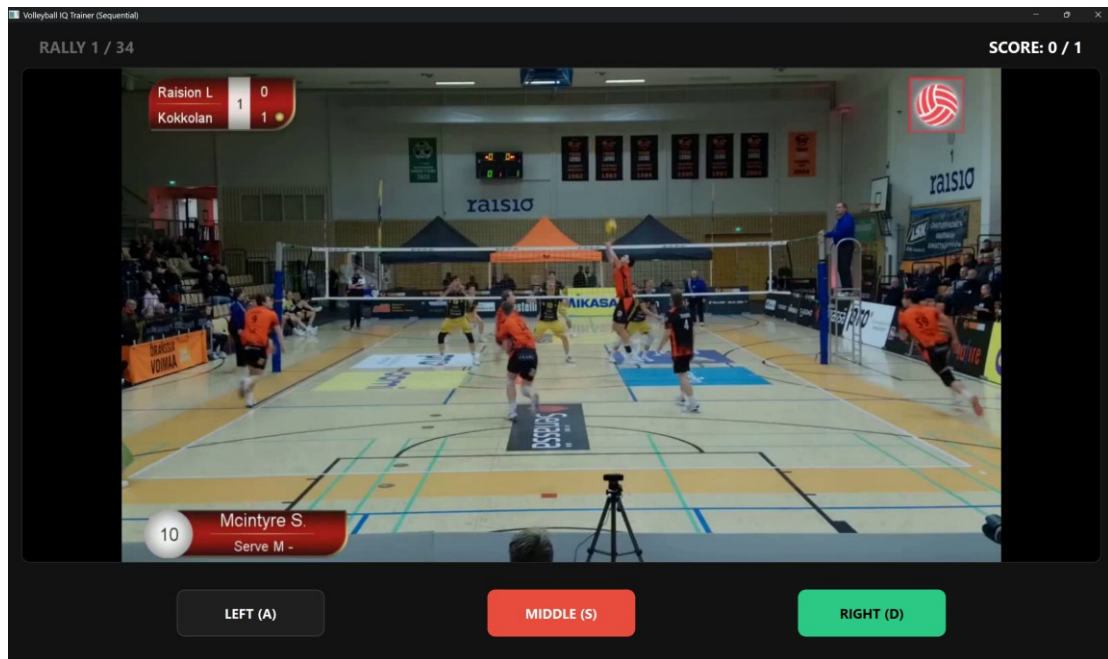
Kuva 5. Koulutusmoduuli (trainer.py).

Käynnistyessään moduuli lataa final\_data.json tiedoston muistiin.

Käyttöliittymän asettelu koostuu videoistoalueesta, vasemman yläkulman pallorallien laskurista, oikean yläkulman tulostaulusta sekä kolmesta painikkeesta, jotka vastaavat hyökkäyssuuntia "Left", "Middle" ja "Right".

Kun harjoitusmoduuli käynnistetään, sovellus käyttää OpenCV:tä videon näyttämiseen käyttöliittymässä. Videon toiston aikana sovellus vertaa jatkuvasti

nykyistä kuvan indeksiä `auto_pause_frame` muuttuun. Kun kuvan indeksi saavuttaa tämän arvon, video pysäytetään ja sovellus jää odottaa käyttäjän arvausta. Sovellus vertaa tätä valintaa `final_data.json` tiedoston `set_type` muuttuun. Opetuksellisen tehokkuuden maksimoimiseksi käyttöliittymä antaa välittömän palautteen. Oikealla valinnalla käyttäjän valitsema suunta käyttöliittymässä pysy vihreänä ja pistetaulu päivittyy. Virheellisessä valinnassa väärin valittu suunta näkyy käyttöliittymässä punaisena ja oikea suunta näytetään vihreällä (Kuva 6). Käyttäjän syöttämän tiedon ja sovelluksen arvioinnin jälkeen videoklippi jatkuu, jolloin käyttäjä voi seurata tilanteen loppuun asti.



Kuva 6. Koulutusmoduuli esimerkki virheellisestä valinnasta.

## 5 Sovelluksen arviointi ja tulevaisuus

Videopohjaisten havaintokognitiivisten harjoitusmateriaalien luominen on erittäin aikaa vievää, sillä se joudutaan usein luomaan manuaalisesti. Vaikka tämän tyyppisen harjoittelun edut on dokumentoitu hyvin urheilutieteessä, sen käytännön soveltamista on usein rajoittanut valmennushenkilöstölle aiheutuva taakka. Luomalla sovelluksen, joka yhdistää Pythonin ja R:n vahvuudet, tässä työssä onnistuttiin automatisoimaan raakakuvamateriaalin ja DataVolley (.dvw) tiedostojen muuntaminen harjoitteluympäristöksi.

Sovellus tarjoaa välitöntä lisäarvoa lentopallojoukkueille. Valmentajat voivat nopeasti luoda räätälöityjä harjoitusmoduuleja tiettyjä vastustajia varten, jolloin urheilijat saavat suuren määrän päätöksentekotilanteita ilman oikean pelitilanteen fyysistä väsymystä.

Nykyisen toteutuksen onnistumisesta huolimatta sovellusta voi kehittää laadukkaammaksi. Tällä hetkellä sovellus käyttää manuaalista kalibroitimoduulia tunnistukseen tarkasti sen hetken, kun passari koskee palloon. Tulevissa versioissa voitaisiin integroida edistyneitä, syväoppimiseen perustuvia asennon arviointimalleja (kuten MediaPipePose tai YOLO) passarin liikkeiden automaattiseen seuraamiseen ja pallokosketukseen. Tämä poistaisi manuaalisen vaiheen kokonaan ja loisi aidosti automaattisen käsittelyprosessin.

Lisäksi nykyinen sovellus on rakennettu PyQt6-kehyksellä, mikä tarkoittaa, että sovellus täytyy ladata jokaiselle tietokoneelle erikseen. Harjoittelun helpottamiseksi seuraavissa versioissa kannattaisi keskittyä siirtymiseen pilvipohjaiseksi verkkosovellukseksi. Siirtyminen antaisi urheilijoille mahdollisuuden käyttää harjoitusmoduuleja suoraan mobiililaitteilla tai tableteilla, mikä helpottaisi harjoittelun lisäämistä päivittäisiin rutiineihin.

## Lähteet

Crockford, D. (2026). Introducing JSON. Haettu 31. 3. 2026 osoitteesta <https://www.json.org/json-en.html>

Data Project S.r.l. (2026). DataVolley. Bologna, Italia. Haettu 16. 2. 2026 osoitteesta <https://www.dataproject.com/>

Eager, E. A.;& Erickson, R. A. (8 2023). *Football Analytics with Python & R*. O'Reilly. Haettu 8. 3. 2026 osoitteesta <https://www.oreilly.com/library/view/football-analytics-with/9781492099611/>

Farrow, D.;& Abernethy, B. (2002). Can anticipatory skills be learned through implicit video-based perceptual training? *Can anticipatory skills be learned through implicit video-based perceptual training?* Journal of sports sciences. <https://doi.org/10.1080/02640410252925143>

Maldonado, S. V.;Abellán, J.;Sáez, N.;López, L. M.;& Contreas, O. (10. 2014). Decision-making and visual perception skills in youth volleyball players and non-players. Haettu 16. 2. 2026 osoitteesta [https://www.researchgate.net/publication/266686596\\_Decision-making\\_and\\_visual\\_perception\\_skills\\_in\\_youth\\_volleyball\\_players\\_and\\_non-players](https://www.researchgate.net/publication/266686596_Decision-making_and_visual_perception_skills_in_youth_volleyball_players_and_non-players)

Marcelino, R.;Mesquita, I.;& Afonso, J. (2008). International Journal of Performance Analysis in Sport. *The weight of terminal actions in Volleyball. Contributions of the spike, serve and block for the teams' rankings in the World League 2005.* <https://doi.org/10.1080/24748668.2008.11868430>

Marteniuk, R. G. (31. 10. 1984). Information Processes in Movement Learning: Capacity and Structural Interference Effects. Haettu 16. 2. 2026 osoitteesta <https://doi.org/10.1080/00222895.1986.10735370>

Nguyen, R. N.;Day, J. T.;Warton, D. I.;& Lane, O. (14. 1. 2021). fitzRoy - An R Package to Encourage Reproducible Sports Analysis. *The R Journal*. <https://doi.org/10.32614/RJ-2021-005>

OpenCV. (2026). OpenCV. Haettu 18. 3. 2026 osoitteesta <https://opencv.org/>

Piras, A.;Lobiatti, R.;& Squatrito, S. (30. 4. 2014). Response Time, Visual Search Strategy, and Anticipatory Skills in Volleyball Players.

<https://doi.org/10.1155/2014/189268>

Python Software Foundation. (2026). Python. Haettu 31. 3. 2026 osoitteesta

<https://www.python.org/>

Python Software Foundation. (2026). re — Regular expression operations.

Haettu 19. 3. 2025 osoitteesta <https://docs.python.org/3/library/re.html>

Raabe, D.;Biermann, H.;Bassek, M.;Wohlan, M.;Komitova, R.;Rein, R.;Kuppens, G.;& Memmert, D. (26. 8. 2022). floodlight - A high-level, data-driven sports analytics framework. *Journal of Open Source Software*.

<https://doi.org/10.21105/joss.04588>

Reynaud, C. (1. 6. 2015). The Volleyball Coaching Bible, Vol. II. (C. Reynaud, Toim.) Haettu 27. 2 2026

Riverbank Computing. (2026). What is PyQt? Haettu 18. 3. 2026 osoitteesta

<https://www.riverbankcomputing.com/software/pyqt/>

Rogers, Y.;Sharp, H.;& Preece, J. (26. 5. 2015). Interaction Design: Beyond Human-Computer Interaction. Haettu 18. 3. 2026 osoitteesta

[https://books.google.fi/books?id=n0h9CAAQBAJ&printsec=frontcover&hl=fi&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.fi/books?id=n0h9CAAQBAJ&printsec=frontcover&hl=fi&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

Suárez, M. C.;Serenini, A. L.;Fernández-Echeverría, C.;Collado-Mateo, D.;& Arroyo, M. M. (21. 5. 2020). The Effect of Decision Training, from a Cognitive Perspective, on Decision-Making in Volleyball: A Systematic Review and Meta-Analysis. <https://doi.org/10.3390/ijerph17103628>

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2 p.). Sveitsi: Springer. <https://doi.org/10.1007/978-3-030-34372-9>

The R Foundation. (2026). The R Project for Statistical Computing. Haettu 31.

3. 2026 osoitteesta <https://www.r-project.org/>

Wu, F.;Wang, Q.;Bian, J.;Xiong, H.;Ding, N.;Lu, F.;Cheng, J.;& Dou, D. (2. 6.

2022). A Survey on Video Action Recognition in Sports: Datasets, Methods and Applications. <https://doi.org/10.48550/arXiv.2206.01038>

Zabih, R.; Miller, J.; & Mai, K. (1995). A feature-based algorithm for detecting and classifying scene breaks. *Haettu* 5. 3. 2026 osoitteesta  
<https://www.semanticscholar.org/paper/A-feature-based-algorithm-for-detecting-and-scene-Zabih-Miller/e37e7533b1689127f5eb151694f4add687767f96>