



**UNIVERSITY
OF OULU**

TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA

**Petri Jaakkola
Mikael Rinkinen
Janne Vihriälä**

**IHMISEN PERUSTUNNETILOJEN TULKINTA JA
HYÖDYNTÄMINEN SOSIAALISISSA
ROBOTEISSA**

Kandidaatintyö
Tietotekniikan tutkinto-ohjelma
Kesäkuu 2023

Jaakkola P., Rinkinen M., Vihriälä J. (2023) Ihmisen perustunnetilojen tulkinta ja hyödyntäminen sosiaalisissa roboteissa. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 45 s.

TIIVISTELMÄ

Tietotekniikan kehittyessä tekoälyn ja robottien käyttö on moninkertaistunut monilla eri teollisuuden- ja tekniikan aloilla. Ihmisen tärkeintä aistia, näkökykyä, hyödynnetään robotiikassa keinotekoisena näön, eli konenäön toteuttamisessa. Koska sosiaaliset robotit yleistyvät, on tärkeää, että ihmisen ja robotin vuorovaikutus on mahdollisimman luontevaa. Tällöin varsinaisen näkökyvyn lisäksi kasvojen ilmeet ja niiden tulkinta ovat keskeisessä asemassa, jotta roboti pystyy tulkitsemaan ihmistä oikein. Tässä projektissa toteutettiin kasvojen perustunnetilojen tulkitseminen InMoov-robotin robottipään avulla, jonka 3d-mallit ovat vapaasti saatavilla. Robotti ohjelmoitiin tunnistamaan ihmisen kasvot sen silmässä olevalla web-kameralla ja tulkitsemaan kasvoista perustunnetiloja: neutraali, iloinen, vihainen, hämmästynyt, surullinen ja pelko. Tunteen tunnistamisen jälkeen roboti reagoi eri tunteisiin eri liikkeellä. Järjestelmän toteuttamisessa käytettiin pääasiassa OpenCV-ohjelmistokirjastoa konenäön toteuttamiseen, Tensorflow-koneoppimisalustan korkean tason ohjelmointirajapintaa Kerasta tunteen tunnistamiseen ja Robot Operating Systemiä järjestelmän ytimenä ja itse robotin pään liikkeiden ohjaamiseen. Kuviin perustuvista lähestymistavoista käytettiin neuroverkkoja, jotka syväoppimisen kautta opetettiin tunnistamaan kasvojen tunneilmaisuja kuvasta opetusdatan pohjalta. Tulokseksi projektissa saatiin järjestelmä, joka pystyi tunnistamaan käytetyissä olosuhteissa tunteita varsin hyvin.

Avainsanat: tekoäly, konenäkö, neuroverkot, syväoppiminen, kasvonilmeet, tunteet

Jaakkola P., Rinkinen M., Vihriälä J. (2023) Detection and Use of Human Basic Emotions in Social Robots. University of Oulu, Degree Programme in Computer Science and Engineering, 45 p.

ABSTRACT

As computer science is developing the usage of artificial intelligence and robots has multiplied in several fields of industry and technology. The most important sensor of humans, vision, is utilized in robotics in the implementation of artificial vision, i.e. machine vision. As social robots become more common it is essential for human-robot interaction to be as natural as possible. In this case, in addition to vision facial expressions and their interpretation are vital components for fluent interaction. In this project, the interpretation of basic human facial expressions was implemented using the InMoov-robot head, the 3d-models of which are distributed freely. The system was programmed to recognize the facial expressions of a human in front of a webcam and identify some basic emotions, namely neutral, happy, angry, surprised, sad and fear and react differently to each emotion. The system was created using these main components: OpenCV library for machine vision, Keras, a high-level Application Programming Interface for the Tensorflow machine learning platform, for recognizing emotions and Robot Operating System as a core for running the whole system and controlling the robot head. The project explored neural networks function based on deep learning. Neural networks were trained to recognize emotions from facial expression images. The result of the project was a system capable of recognizing facial expressions quite well considering the challenges of the subject.

Keywords: AI, machine vision, neural networks, deep learning, facial expressions, emotions

ALKULAUSE

Tämä kandidaatintyö on laadittu Oulun yliopiston tieto- ja sähkötekniikan tiedekunnassa kurssia "Sulautettujen ohjelmistojen projekti" varten.

Oulussa 23. kesäkuuta 2023

Petri Jaakkola
Mikael Rinkinen
Janne Vihriälä

SISÄLLYSLUETTELO

TIIVISTELMÄ	
ABSTRACT	
ALKULAUSE	
SISÄLLYSLUETTELO	
LYHENTEIDEN JA MERKKIEN SELITYKSET	
1. JOHDANTO	8
2. IHMISEN NÄÖSTÄ KONENÄKÖÖN	10
2.1. Ihmisen silmän toiminta ja silmän liikkeet	10
2.2. Tietokonenäkö	11
2.3. Konenäkö	11
2.4. Tietokonenäkö kasvojen ilmeiden tunnistuksessa	13
3. KASVOJEN HAVAINNOINTI JA ILMEIDEN TUNNISTAMINEN	14
3.1. Kasvojen havainnointi	14
3.1.1. Ominaisuuksiin perustuvat lähestymistavat	14
3.1.2. Kuviin perustuvat lähestymistavat	15
3.2. Ilmeiden tunnistaminen	17
3.2.1. Tavanomaiset lähestymistavat	18
3.2.2. Syväoppimiseen perustuvat lähestymistavat	18
4. IHMISMÄISEN SOSIAALISEN ROBOTIN KEHITYS JA HAASTEET	19
4.1. Kasvojen mekaniikka realistisessa robottipäässä	19
4.2. Tekoälyn sosiaalinen älykkyys	20
5. TOTEUTUS	22
5.1. Laitteisto	22
5.2. Ohjelmistot	24
5.2.1. OpenCV	24
5.2.2. YuNet	24
5.2.3. YuNetin ja Haar Cascaden väliset erot	24
5.2.4. TensorFlow ja Keras	25
5.2.5. Robot Operating System, ROS	26
5.2.6. FER-2013	26
5.3. Konenäkösovelluksen toteuttaminen	27
5.3.1. Kasvojen tunnistus kuvasta	27
5.3.2. Tunteiden tunnistamisohjelma	27
5.4. ROS:lla yhdistetty kokonaisuus	28
6. TESTAUS	30
6.1. Ohjelmistokokonaisuuden testaus	30
6.1.1. Ohjelmiston suorituskyky	30
6.1.2. Ilmeentunnistusohjelman tarkkuus	31
6.2. Ohjelmiston ajo fyysisellä robotilla	32
6.3. Tulosten analysointi	34
7. POHDINTA	37
8. PROJEKTIN KUVAUS	38
9. YHTEENVETO	39

10. VIITTEET	40
11. LIITE	45

40
45

LYHENTEIDEN JA MERKKIEN SELITYKSET

AI	Artificial Intelligence, tekoäly
API	Application Programming Interface, ohjelmointirajapinta
ASM	Active Shape Model, aktiivisen muodon malli
CA	Constellation Analysis, konstellaatioanalyysi
CNN	Convolutional Neural Network, konvoluutioneuroverkko
CPU	Central processing unit, keskusyksikkö
CVS	comma-separated values, pilkulla erotetut arvot
deg/s	degree per second, astetta per sekunti
DL	Deep Learning, syväoppiminen
FA	Feature Analysis, ominaisuusanalyysi
FBA	Feature-Based Approaches, ominaisuuksiin perustuvat lähestymistavat
FER	Facial Expression Recognition, kasvojen ilmeiden tunnistaminen
FS	Feature Search, ominaisuuksien etsintä
GPT	Generative pre-trained transformer, luova esikoulutettu muuntaja
GPU	Graphics Processing Unit, grafiikkaprosessori
IBA	Image-Based Approaches, kuviin perustuvat lähestymistavat
ICML	International Conference on Machine Learning, kansainvälinen koneoppimisen konferenssi
LBP	Local Binary Pattern, paikallinen binäärimalli
LLA	Low Level Analysis, matalan tason analyysi
LSM	Linear Subspace Methods, lineaarisen aliavaruuden menetelmät
MTCNN	Multi-task Cascaded Convolutional Networks, monitoimiset CSCNN-verkot
RGB	Red-Green-Blue, puna-viher-sini
ReLU	Rectified Linear Unit, oikaistu lineaarinen yksikkö
ROS	Robot Operating System, robotin käyttöjärjestelmä
SA	Statistical Approaches, tilastolliset menetelmät
USB	Universal Serial Bus, universaali sarjaväylä

1. JOHDANTO

Tekoälyn (eng. Artificial Intelligence, AI) merkittävämpi kehitys alkoi 1900-luvulla, jolloin tietotekniikan laitteita alettiin valmistaa helpottamaan tai korvaamaan ihmisen suorittamia tehtäviä [1]. Nykyään tekoälyn sovelluskohteita on kaikkialla, ja esimerkiksi internetsovellukset, pilvipalvelut, erilaiset älylaitteet, jakamispalvelut ja vuorovaikutus ovat tuoneet mukanaan tekoälyn aikakauden [1]. OpenAI:n vuonna 2022 julkaistu matemaattinen, luova kielimalli, ChatGPT [2], on viimeisintä tekoälyn tekniikkaa edustava tekstipohjainen tekoälytyökalu, keskustelubotti, joka on nopeasti saanut suosiota. Se on koneoppimisen kautta koulutettu erityisesti ymmärtämään ja tulkitsemaan lukuisia ihmiskieliä ja mahdollistaa käyttäjille kysymysten esittämisen sekä kykenee antamaan aidontuntuisen vastauksen keskustelunomaisesti [3, 4].

Osa sekä tekoälyä että robottiteknologiaa on myös koneen opettaminen tunnistamaan visuaalista informaatiota. Robotteja käytetään tekoälyn sovelluksien testaamiseen, jotta niitä voidaan sitten soveltaa käytäntöön. Teollisuusrobotit soveltuvat hyvin monotonisiin, tarkkuutta ja nopeutta vaativiin sekä vaarallisiin tehtäviin. Robotteja käytetäänkin monilla eri teollisuuden osa-alueilla korvaamaan ihminen. Elektroniikka- ja metalliteollisuudessa robotteja käytetään kustannus- ja laatusyistä. Ne ovat tehokkaampia ja edullisempia kuin ihmistyövoima. Robotteja voidaan käyttää myös paikoitussovelluksissa, joiden avulla pystytään etsimään kohdekappale ja saamaan tietoja siitä. Mittaussovelluksissa robotteja käytetään muun muassa etäisyyden, pinta-alan, tilavuuden ja kaarevuuden mittaamiseen. Elintarviketeollisuudessa robotit sopivat tuotteiden koodinlukuun ja tuotteiden tarkistukseen tuotannossa. Konenäön avulla parannetaan koneiden tekemiä prosesseja, ja automatisoinnin myötä helponnetaan ihmisen toimintaa. [5, 6, 7, 8]

Tekoälyn ja robotiikan kehityksen ansiosta robotteja voidaan vähitellen käyttää myös yhteistyöhön ihmisten kanssa. Sosiaaliset robotit voivat toimia ihmisten kanssa erilaisissa asiakaspalvelutilanteissa, mutta robotteja suunnitellaan myös esimerkiksi opetukseen ja varsinkin hoitotyöhön, jossa on pula riittävästä henkilökunnasta [9, 10, 11]. Tämä on yksi syy siihen, miksi robotin ja ihmisen vuorovaikutusta pyritään kehittämään luonnollisemmaksi.

Tärkeä ala on myös lääketiede, jossa konenäköä käytetään päivittäisissä rutiinitehtävissä, kuten soluanalyseissa tai ultraäänikuvien tulkinnessa [8]. Sitä voidaan käyttää myös analysoitaessa syöpäsoluja röntgenkuvista ja muissakin sovelluskohteissa, muun muassa radiologiassa ja patologiassa [12]. Nykyään suoritetaan myös leikkauksia ihmisen ohjaaman robotin avulla.

Ihmiselle näkökyky ja näkemisen ymmärtäminen tapahtuu tietokonenäköön perehtyneen Lindköpingin yliopiston professori Michael Felsbergin mukaan luonnostaan [13]. Näkö on ihmisen aisti, jonka avulla kerätään eniten tietoa ympäristöstä, havainnoidaan ja opitaan asioita [14]. Ihminenkin oppii vähitellen, mutta näkökyvylle on olennaista, että nähtyä pystytään tulkitsemaan melkein välittömästi, ja nähdään asiat sellaisena kuin ne ovat [14]. Koneelle näkeminen sen sijaan on monimutkaisempaa, koska koneelle täytyy ensin opettaa tietoja algoritmien avulla näkemisen kohteesta. Robotti voidaan saada näkemään erilaisia asioita, mutta robotti pitää saada myös ymmärtämään näkemäänsä. Robotin näkemä kuva ei sisällä konkreettista tietoa esimerkiksi esineistä tai etäisyyksistä, koska koneilla ei ole ihmisen havainnointi- ja päättelykykyä. Robotin kamera pystyy ymmärtämään valon

voimakkuuden, mutta vasta käytettävän ohjelmiston avulla se pystyy tekemään tulkinnan kuvasta [13].

Helpompi tapa robotin opettamisessa olisi, jos robotti voisi oppia aluksi matkimalla ihmistä ja sitten kokeilemalla. Etuna robotin oppimisessa kuitenkin on, että robotin oppimaa tietoa voidaan siirtää muillekin roboteille tietoverkossa. [11]

Jotta sosiaalisia robotteja voidaan käyttää ihmisten keskuudessa, robotit tarvitsevat ihmisille ominaisen kyvyn kommunikoida. Kommunikoinnissa avuksi voidaan ottaa jossain määrin uuden teknologian keskusteleva tekoäly, ChatGPT, jota voidaan hyödyntää vuorovaikutuksessa [2]. Mutta kuten kaikessa teknologiassa, ChatGPT:n käytössä on omat rajoituksensa ja haasteensa [15], jos on kyse eettisesti arveluttavista tiedoista. Ihmismäisten robottien, eli androidien, ulkonäkö auttaa luontevampaan vuorovaikutukseen, vaikka toisaalta liiallinen ihmismäisyys voi vaikuttaa päinvastoin, sillä tarkasti ihmisen ulkonäköä jäljittelevä robotti voi vaikuttaa oudolta tai pelottavalta [16].

Tässä projektissa kehitettiin konenäkösovellus ja koneoppimisen ja opetusdatan kautta tutkittiin Inmoov-robottipään silmien kykyä nähdä ja tunnistaa ihmisen kasvojen ilmeiden kautta perustunnetiloja: neutraali, iloinen, vihainen, hämmästynyt, surullinen ja pelokas, ja analysoida, kuinka hyvin konenäkö tunteet tunnistaa. Projektissa tutkittiin myös minkälaisia haasteita on otettava huomioon konenäkösovelluksen toteutuksessa sosiaaliselle robotille. Toiminnot toteutettiin opetetun kuvamateriaalin, ja sen jälkeen robottipäähän asennetun web-kameran tuottaman kuvatulkin avulla kameraa vastapäätä olevasta ihmisestä. Lisäksi mukana on robotin reagointi eri ilmeisiin jollakin luonnollisella tavalla, esimerkiksi erilaisilla pään liikkeillä. Projektissa esitetään myös keskustelevan tekoälyn, ChatGPT:n vastausten tuottamista ihmiselle. Projektin on osa kokonaisuutta, jossa toteutetaan robotille erilaisia toimintoja, jotta se pystyy olemaan luonnollisemmalla tavalla vuorovaikutuksessa ihmisten kanssa.

2. IHMISEN NÄÖSTÄ KONENÄKÖÖN

Ihmisellä silmän näkökyky on tärkein aisti, kun hahmotetaan ympäristöä ja otetaan vastaan informaatiota siitä, ja aivan samoin myös robotilla näkö on sen tärkein sensori [11, 14]. Teknologian kehityksen myötä on syntynyt tarve kehittää konenäköä, jonka etuna ihmisen silmään verrattuna on tarkkuus, nopeus ja lisäksi se on väsymätön, ja siten mahdollistaa laajemman ja tehokkaamman hyödyntämisen teknologiateollisuudessa. Batchelorin mukaan [17] konenäkö (eng. machine vision) ja tietokonenäkö (eng. computer vision) tarkoittavat eri asioita, mutta robotiikan, eli robottien kehittämisen yhteydessä [14] on nykyään yleistä käyttää termejä konenäkö ja tietokonenäkö tarkoittaen koko laajaa keinotekoisesti kehitetyn näön käsitettä.

Tietokonenäköön liittyy tieteellinen ja tutkimuksellinen lähestymistapa kuvankäsittelyyn ja kuvan analysointiin. Tässä sisääntulo on tiedostossa oleva kuva. Kuvien käsittelyssä ja tutkittaessa käytetään matemaattisia menetelmiä [17]. Pietikäisen ja Silvenin [8] mukaan tiedeyhteisössä käytetään yleisesti englanninkielistä termiä tietokonenäöstä. Tällöin keskeisessä asemassa matemaattisen teorian lisäksi ovat algoritmit ja yhteys ihmisen näköaistiin.

Konenäkö on sitä vastoin sovelluksiin tähtäävä monipuolinen ja laaja systeemisuunnittelua sisältävä järjestelmätekniinen ala [8]. Siinä tarvitaan laaja-alaisempaa tietoa antureista, valaistuksesta ja optiikasta. Sisäänmenona voi olla jokin tuote liukuhihnalla, ja laitteen tai prosessin ohjaus on ulostulona [17]. Robotiikka jaetaan perinteisesti teollisuusrobotiikkaan ja palvelurobotiikkaan, joista palvelurobotiikan kehitys on vähitellen lisääntymässä [11].

2.1. Ihmisen silmän toiminta ja silmän liikkeet

Ihmisen silmän näköaistimus syntyy, kun jotakin kohdetta katsottaessa siitä heijastuu valoa silmään. Silmän sarveiskalvo yhdessä linssin kanssa taittaa valon verkkokalvolle silmän takaosaan tarkkan näön alueelle, jonne muodostuu ylösalaisin oleva tarkka kuva. Tämän seurauksena verkkokalvolta lähtee sähköisiä impulsseja pitkin näköhermoa aivoissa sijaitsevalle näköalueelle, jossa kuva kääntyy oikeinpäin. Saatu viesti tulkitaan siellä kuvaksi, jolloin näköaistimus syntyy. [18]

Ihmisen molempien silmien avulla pystytään yhdistämään kaksi hieman eri kulmista nähtyä kuvaa. Aivoissa molemmista silmistä tuleva tieto yhdistyy, ja siksi ihminen kokee ympäristön kolmiulotteisena. Tämän johdosta etäisyyksien arviointi onnistuu ihmiselle paremmin. [19]

Ihmisen näköaisti rajoittuu pieneen osaan sähkömagneettisen säteilyn spektristä, aallonpituuksille noin 400-800 nm (näkyvä valo). Eläimillä näkö-alue voi olla kapeampi, tai alue voi olla kokonaan näkyvän valon ulkopuolella. Ne voivat nähdä enemmän värejä, tai olla täysin värisokeita. Kuvia muodostetaan muillakin tavoilla kuin pelkästään sähkömagneettisen säteilyn perusteella, esimerkiksi ääniaaltojen avulla. [20]

Ihmisen silmässä on verkkokalvolla kahdenlaisia soluja, tappi- ja sauvasoluja. Tappisolut ovat keskittyneet tarkkan näkemisen alueelle verkkokalvolla, kun taas sauvasolut ovat jakaantuneet tasaisesti ympäri verkkokalvoa. Tappisoluista 65 % on herkkiä punaiselle valolle, 33 % vihreälle ja 2 % siniselle valolle. Ihmisen havaitsemat

värisävyt syntyvät näiden kolmen värin (RGB) yhdistelminä. Näkö jaetaan kahteen osaan, skotooppiseen eli hämäränäköön ja fotooppiseen eli päivänäköön. [20]

Silmän toimintaan liittyvät myös sakkadiset silmänliikkeet, eli hypähdykset, mikä tarkoittaa silmien yhteisen kiinnkohdan muuttamista. Silmänliikkeelle on ominaista nopea kiihdytys silloin, kun liike alkaa ja vastaavasti pysähtyessä, se voi olla jopa $40\,000\text{ deg/s}^2$. Silmänliikkeiden huippunopeuden vaihtelu on tällöin 400 - 600 deg/s välillä, ja liikkeet kestävät tavallisesti 30 - 120 ms. Sakkadisen liikkeen pituus on 1-40 astetta, sisältäen tavallisesti myös päänliikettä, jos katseen liike kiinnekohtaan on suurempi kuin 30 astetta [21]. Sakkadi liike korjaa nopean liikkeen ja pään asennon vaihtumisen [22].

Silloin, kun ihmiset pyrkivät pitämään katseen tarkasti kohteessa, puhutaan pienistä sakkadeista, mikrosakkadiliikkeistä, joiden suuruus on asteen luokkaa ja nopeudeltaan jopa 30 ms:n välein [21].

2.2. Tietokonenäkö

Noin 60 vuoden ajan on yritetty kehittää koneille tapoja nähdä ja ymmärtää visuaalista tietoa. Tietokonenäkö kuuluu alana tekoälyyn, ja sen avulla tietokoneille ja järjestelmille on mahdollista saada merkityksellistä tietoa digitaalisista kuvista, videoista ja muista visuaalisista kohteista. Saatujen tietojen perusteella tietokoneet toimivat ja antavat suosituksia. Tietokonenäkö toimii hyvin paljon samalla tavoin kuin ihmisenäkö, mutta ihminen pystyy kuitenkin erottamaan esineet toisistaan, arvioimaan välimatkaa ja liikettä. Ihminen kykenee myös päättelemään, näyttääkö kuva oikealta, vai onko se jotenkin virheellinen [23]. Tietokonenäkö pyrkii siis teknisestä näkökulmasta automatisoimaan niitä tehtäviä, jotka muodostavat ihmisen näköaistin [24].

Tietokonenäön avulla, kun käytetään kameroita, tietoa ja algoritmeja, koneet suorittavat toiminnot nopeammin verrattuna ihmisen silmän verkkokalvon, näköhermojen ja visuaalisen aivokuoren toimintaan. Toimiakseen tietokonenäkö tarvitsee paljon tietoa fyysisistä esineistä ja tapahtumista, joita analysoidaan useaan kertaan, jotta lopulta päästäisiin oikeaan ja tarkoitukseen sopivaan tulkintaan. Hyödynnettäessä tietokonenäköä ohjelma tunnistaa asioita reaali maailmasta ymmärtämistä varten. [23, 25, 26]

Tietokoneelle ihmisen älykkyyden mallintaminen reaali maailman kohteisiin on kuitenkin monimutkaisempaa. Juuri mitään tietokonenäön ongelmaa ei ole tyydyttävästi ratkaistu. Yksi syy tähän on, että ihmisen näköaisti on yksinkertaisesti liian hyvä moneen eri tehtävään. Esimerkiksi kasvojen tunnistuksessa, ihminen pystyy tunnistamaan kasvot eri valaistuksissa ja eri näkökulmista. Tietokone ei pysty kilpailemaan ihmisen näennäisesti rajattoman muistin kanssa, jolla pystymme muistamaan eri ihmisten kasvoja pitkälle tulevaisuuteen tai menneisyydestä. [24]

2.3. Konenäkö

Yleisesti konenäön tavoitteena on, että saadaan kone ymmärtämään kameran tai jonkun muun sensorin kuvaama näkymä, josta saatua tietoa hyödynnetään sitten erilaisissa

sovelluksissa. Koneen tulee tunnistaa kohteita ja määrittää niihin liittyviä tietoja muun muassa sijainti, asento, kohteissa tapahtuvat muutokset ja tulkita erilaisten havaintojen merkitys [8]. Konenäkö on tällä hetkellä yksi tärkeimpiä tekoälyyn liittyvistä osa-alueista. Sillä tarkoitetaan tiedon muuntamista kuvasta tai videosta, ja haluttuun tulokseen päätyminen voi vaatia uuden esitystavan tekemisen useampaankin kertaan [27].

Sekä konenäkö että robottinäkö liittyvät vahvasti tietokonenäköön. Mutta konenäköä käytetään, kun viitataan tietokonenäön hyödyntämiseen teollisuudessa, automaattisissa tarkastuksissa ja prosessien ja robottien hallinnassa. Kuitenkaan robottinäkö ja konenäkö eivät ole sama asia, sillä robottinäkö viittaa aina robotin ohjaukseen. Esimerkkinä konenäöstä voidaan mainita tuotantolinjalla tapahtuva tarkastaminen, jossa hyödynnetään kameraa. Tämä ei kuitenkaan liity robotiikkaan. Tekniikan alan lisäksi robotiikka on tiedettä, eli robottien tutkiminen ja kehittäminen ovat olleet matematiikkaa ja fysiikkaa soveltavaa teknistieteellistä tutkimusta. [11, 25]

Konenäköjärjestelmien toiminta tapahtuu siten, että tietokone vastaanottaa numeromatriiseja kameralta [27], mutta esimerkiksi kuvan tarkennus ei tapahdu automaattisesti [8]. Siksi onkin hyvin haastavaa tunnistaa kaksiulotteisesta näkökulmasta kolmiulotteisia kohteita, sillä sama kohde voi näyttää aivan erilaiselta eri kulmasta [8]. Kohteet voivat osittain olla toistensa peittämiä, ja joko kohteet tai kamera, tai molemmat saattavat olla liikkeessä suhteessa toisiinsa nähden, mikä voi aiheuttaa myös ongelmia, mutta mahdollisesti myös helpottaa ymmärtämään nähtävän kohteen sisältöä [8].

Konenäön avulla otettavan kuvaan vaikuttavat käytettävä kamera ja linssi, valaistus sekä tietokoneen ja kamerasen rajapinta. Valaistuksella on tärkeä merkitys siksi, sillä sen avulla saadaan näkyviin kohteena olevan kuvan tulkinnan kannalta tärkeät piirteet. Linssejä käytetään terävän kuvan saamiseksi kameralle ja myös näkökentän kokoa voidaan säätää. Kuvan siirtyminen tietokoneen muistiin tapahtuu, kun kamera muuttaa kuvan ensin videosignaalksi, jonka tietokone tietokoneen ja kamerasen rajapinnassa ottaa vastaan USB-väylän kautta ja muuttaa sen kuvaksi tallentaen sen muistiin. [7]

Monilla aloilla tutkitaan konenäköön liittyviä asioita. Näistä läheisimpiä tieteenalvoja ovat digitaalinen kuvankäsittely ja hahmontunnistus. [8]

Viime aikoina tekoälyn kehityksessä on alettu soveltamaan isoja kielimalleja paitsi tekstien, myös kuvien prosessointiin sekä luomiseen [28]. Viimeisintä teknologiaa on vuonna 2022 julkaistu sanfranciscolaisen David Holzin johtama avoimen beta-vaiheen generatiivinen tekoälyohjelma ja -palvelu, Midjourney. Se toimii Discord-viestintäpalvelun kautta ja sille voidaan antaa luonnollisella kielellä "kehoitteita", joiden perusteella palvelu antaa neljän kuvan sarjan ehdotuksia, joita voidaan sitten vielä skaalata [28, 29]. Midjourney sisältää myös describe-toiminnon, jolla voidaan luoda sanallinen kuvaus tekoälylle syötetystä kuvasta. Toiminto edustaa uudenlaista kuvan tulkintaa, sillä kuvaa ei pyritä sovittamaan ennalta määritellyyn luokkaan, vaan Midjourney:n describe tuottaa luovasti tekstiä, joka kuvailee annetun kuvan sisältöä isojen kielimallien mukaisesti ennalta määrittelemättömällä tavalla. Tällä tavalla tuotettua kuvausta voidaan käyttää myös tekoälyn kehoitteena tuottamaan lisää samankaltaisia kuvia. [30] Tämä saattaa tulevaisuudessa mullistaa konenäön alaa ja parantaa tarkkuutta, sekä luoda uusia mahdollisuuksia. Ison kielimallin mukaiset tekoälymallit ovat kuitenkin valtavan suuria ja vaativat suurta laskentatehoa, mikä rajoittaa niiden käytettävyyttä monessa nopeaa viivettä vaativassa sovelluksessa.

2.4. Tietokonenäkö kasvojen ilmeiden tunnistuksessa

Tärkeä osa ihmisten välistä kommunikaatiota on sanaton viestintä kasvojen ilmeillä. Ja koska se on tärkeä osa ihmisten välistä kommunikaatiota, muodostaa se myös yhden kommunikaatiöväylän ihmisen ja robotin välille. Tästä syystä yksi tekoälyn tehtävä on havaita ja tulkita ihmisen tunnetiloja kasvojen liikkeiden kautta. Tyypillisesti tämä tapahtuu robottipäähän asennetun kameran tuottaman videokuvan tulkinnan avulla.

On havaittu, että tietyt perusilmeet ovat samankaltaisia kulttuurista riippumatta [31]. Ilmeiden tunnistamiseen liittyy kuitenkin lukuisia teknisiä haasteita. Yksi näistä on se, että tekoälyn opettamista varten kerätyt kuvaesimerkit ilmeistä ja niitä vastaavista tunnetiloista eivät välttämättä vastaa todellisuutta. Jokaisella ihmisellä on oma tyyliinsä tuottaa ilmeitä [31], ja omanlaisensa kasvot. Ilmeentunnistuksen tarkkuus heikkenee yleensä merkittävästi, jos opetukseen käytetyn datan henkilö on eri kuin se, jonka ilmeitä tulkitaan [32]. Silloin kun tekoälyn opettamiseen käytetyssä materiaalissa suurin osa on tietyn etnisyyden edustajia, tietokonenäkö tulkitsee todennäköisemmin ilmeen väärin eri etnisyyteen kuuluvan ihmisen kasvojen perusteella [33]. Tosielämässä käytetyt ilmeet ovat myös monesti hienovaraisempia ja siten vaikeampia tulkita kuin yleensä tunnistusalgoritmien kehityksessä käytetyt korostetut tunnetilat, kuten suuri yllättyneisyys tai vihaisuus [34].

Puhtaammin mekaaniset osuudet tunnistusalgoritmista, toisin sanoen numeroiden murskaamisen osuudet, tietokonenäkö pystyy käsittelemään jo varsin hyvin. Esimerkiksi kasvojen erottaminen taustasta tai muista esineistä onnistuu näppärästi, myös silloin, kun osa kasvoista on kameralta piilossa [35]. Tarpeeksi tarkoilla kameroilla pystytään myös havaitsemaan luotettavasti se, mihin suuntaan ihmisen silmät osoittavat milläkin hetkellä [36]. Konenäkö osaa erottaa myös eri ihmiset toisistaan niin suurella tarkkuudella, että kasvojentunnistusta voidaan käyttää käyttäjän varmuuksena tietokoneen tai matkapuhelimen sisäänkirjautumisessa.

Tällä hetkellä perusilmeiden, kuten vihaisuuden, inhon tai yllättyneisyyden tunnistuksen tarkkuus vaihtelee suuresti. Käytetystä algoritmista ja kasvotietokannasta riippuen tunnistuksen tarkkuus heittelee noin 70 %:sta jopa 100 %:n tarkkuuteen [34, 37]. Mutta sellaista menetelmää, jolla eri tosielämän tilanteissa tunnistettaisiin 100 %:n tarkkuudella, ei ole olemassa.

3. KASVOJEN HAVAINNOINTI JA ILMEIDEN TUNNISTAMINEN

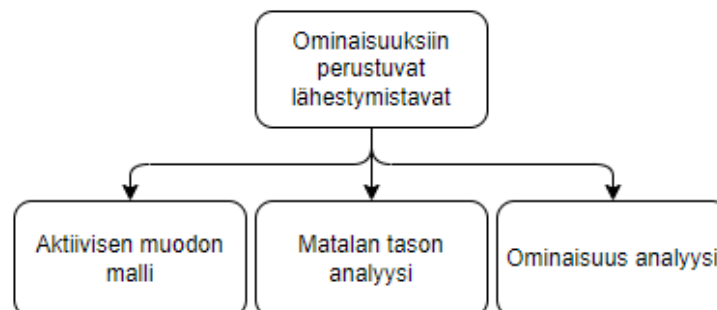
Ihmisten välisessä vuorovaikutuksessa kasvojen ilmeet ja muut näkyvät eleet ovat tärkeitä, koska ne antavat ymmärrystä vastapuolen tunnetiloista [38, 39]. Samalla tavalla myös ihminen-robotti vuorovaikutuksessa, kasvojen tunnistaminen ja ilmeiden tunnistaminen antavat robotille tärkeitä vihjeitä vuorovaikutuksen etenemisestä ja mahdollistavat robotin toiminnan tilanteeseen sopivammin [38]. Jo pelkkä kasvojen havaitseminen mahdollistaa robotin seuraamaan sitä, kenen kanssa se on vuorovaikutuksessa [38]. Kasvojen havaitsemisen, tunnistamisen ja ilmeiden analysoinnin tapahtuessa ihmisillä luonnollisesti, se konenäön avulla on monimutkainen tapahtuma, joka vaatii paljon laskentaa [39, 40]. Tavanomainen kasvojen ilmeiden tunnistaminen (eng. Facial Expression Recognition, FER) sisältää kolme askelta: kasvojen havainnoinnin, ominaisuuksien erottamisen ja ilmeen luokittelun [41, 42].

3.1. Kasvojen havainnointi

Kasvojen havainnoinnin voidaan sanoa olevan yksi tärkeimmistä erityistapauksista objektien havainnoimisessa kuvista [40]. Kasvojen havainnointiin on useita erilaisia digitaalisia kuvankäsittelymenetelmiä, joita varten on luotu useita erilaisia algoritmeja [39, 40]. Kasvojen havainnoinnissa algoritmien päätehtävä on ensin todeta, esiintyykö kuvassa kasvoja vai ei, ja kasvojen esiintyessä palauttaa kasvojen sijainnin kuvasta [40]. Algoritmit voidaan jakaa kahteen eri päälähestymistapaan: ominaisuuksiin perustuviin lähestymistapoihin (eng. Feature-Based Approaches, FBA) ja kuviin perustuviin lähestymistapoihin (eng. Image-Based Approaches, IBA) [39, 40].

3.1.1. Ominaisuuksiin perustuvat lähestymistavat

Ominaisuuksiin perustuvien lähestymistapojen alakategorioiden jakautuminen on esitetty kuvassa 1.



Kuva 1. Ominaisuuksiin perustuvien lähestymistapojen jakautuminen omiin alakategorioiden.

Ominaisuuksiin perustuva lähestymistapa perustuu ominaisuuksien löytämiseen kuvasta [39, 40]. Menetelmien avulla kuvasta etsitään reunoja, kulmia ja muita rakenteita kahdessa ulottuvaisuudessa [39, 40]. Ominaisuuksiin perustuva lähestymistapa voidaan jakaa vielä kolmeen eri alakategoriaan, jotka kuvaavat kyseisiä algoritmeja paremmin [39].

Aktiivisen muodon malli

Aktiivisen muodon malli (eng. Active Shape Model, ASM), joka havainnoi kasvot kuvasta etsimällä kasvoista piirteitä, kuten silmiä, nenää, leukaa ja suuta, ja generoi löytämien havaintojen koordinaattien perusteella kasvojen piirteiden sijainnista kuvassa ehdotuksen kasvojen kartasta [39, 43]. Algoritmit vertaavat löydettyä karttaa referenssikarttaan ja sen perusteella hyväksyvät tai hylkäävät ehdotetun kartan kasvojen sijainnista kuvassa [43].

Matalan tason analyysit

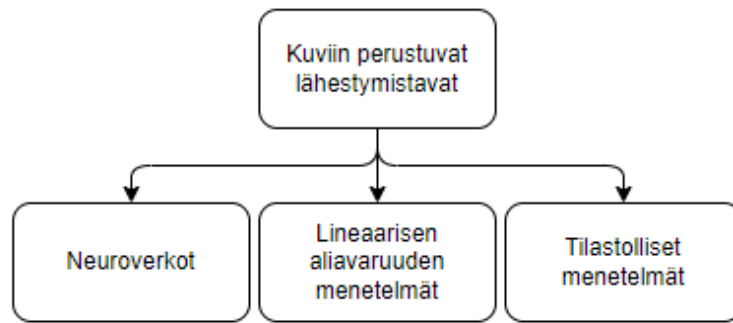
Toinen alakategoria on matalan tason analyysit (eng. Low Level Analysis, LLA) [39]. Matalan tason analyysi kasvojen havaitsemisessa perustuu pikselitietojen analysointiin, jonka avulla kuvasta etsitään yksittäisiä osia [39, 40]. Pikselitiedoista voidaan analysoida esimerkiksi reunoja, harmaaväri vaihtelua ja väritietoja [39, 40].

Ominaisuusanalyysi

Kolmas alakategoria on ominaisuusanalyysi (eng. Feature analysis, FA), johon perustuvat algoritmit on jaoteltavissa vielä kahteen eri alakategoriaan: ominaisuuksia etsiviin algoritmeihin (eng. Feature Search, FS), kuten Viola-Jones algoritmi ja paikalliset binäärimallit (eng. Local Binary Patterns, LBP), ja konstellaatioanalyysi (eng. Constellation Analysis, CA) [39]. Ominaisuuksia etsivät algoritmit etsivät kuvista ominaisuuksia, kuten esimerkiksi silmää tai otsaa, ja alkavat tämän löydetyn ominaisuuden ympäriltä etsimään seuraavia ominaisuuksia, jotka liittyvät löydettyyn ominaisuuteen [39].

3.1.2. Kuviin perustuvat lähestymistavat

Kuviin perustuvien lähestymistapojen alakategorioiden jakautuminen on esitetty kuvassa 2. Kuviin perustuvissa lähestymistavoissa kasvoja havainnoivat algoritmit perustuvat pääsääntöisesti ikkunaan, jota skannataan [39]. Ikkuna skannataan pikseli kerrallaan ja havainnoidaan, onko ikkunassa esiintyvä näkymä kuva kasvoista, vai kuva ei-kasvoista [39]. Voidaan sanoa, että kuviin perustuvat lähestymistavat pyrkivät löytämään parhaan mahdollisen yhtäläisyyden opetuskuviin ja testattavien kuvien väliltä [40]. Myös kuviin perustuva lähestymistapa voidaan tarkemmin jakaa kolmeen eri alakategoriaan [39].



Kuva 2. Kuiin perustuvien lähestymistapojen jakautumisesta omiin alakategorioihin.

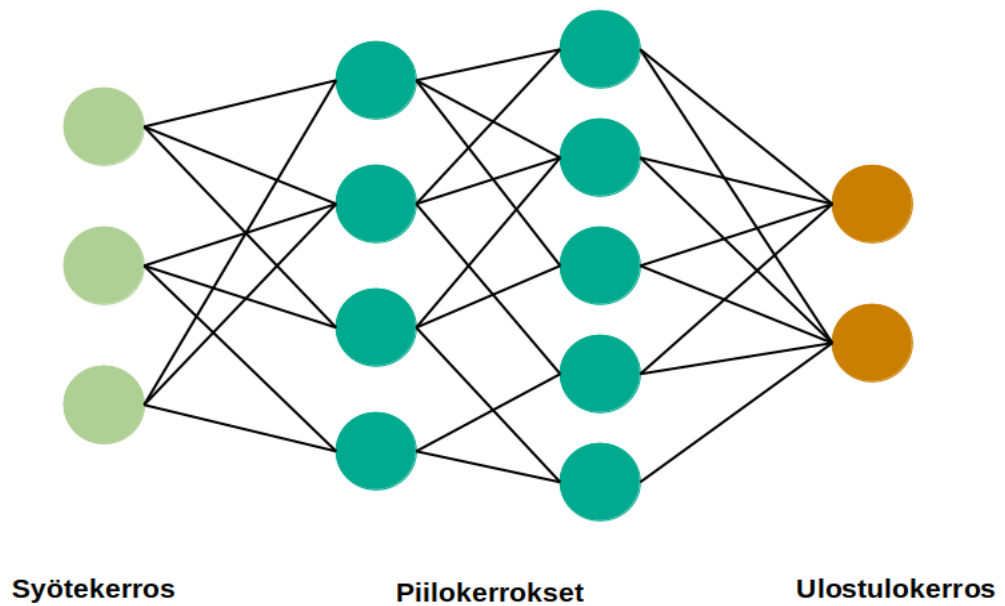
Neuroverkot

Ensimmäinen alakategoria on neuroverkot (eng. Neural Networks), joihin perustuvat lähestymistavat opettavat itselleen, kuinka tunnistaa kasvojen rakenne kuvasta annetun opetusdatan perusteella, ja tunnistustapahtumassa muodostaa ennuste ikkunasta skannatusta kuvasta, onko siinä löydetty havainto kasvojen rakenteesta, vai ei-kasvojen rakenteesta [39].

Tekoälyn taustalla on monenlaisia neuroverkkoja, joista parhaiten kuvantunnistukseen soveltuu algoritmi konvoluutioneuroverkko (eng. CNN). Siinä on alussa erillinen konvoluutiokerros (eng. Convolutional Layer), välissä olevat alinäytteistyskerrokset (eng. Pooling layer) ja lopussa täysin kytketty (eng. Fully-connected layer). Ensimmäisissä kerroksissa etenevä tieto on yksinkertaisempaa, kuten värejä ja reunoja, mutta kun tieto kuvasta etenee pidemmälle konvoluutiokerroksissa, tunnistetaan isompia elementtejä ja muotoja kohteesta, kunnes kohde tunnistetaan. [44]

Eteenpäin syöttävien yksinkertaisten neuroverkkomallien toiminnassa mallit itse määrittelevät tavat käsitellä tietoa. Näissä signaali siirtyy tiettyjen sääntöjen mukaan lähtien syötetasosta ja etenee piilotasojen läpi tulostasolle, jossa se tunnistetaan [45]. Neuroverkot sisältävät useita neuroverkkokerroksia, joiden rakenne on samankaltainen kuin ihmisaivojen rakenne, jossa tiedonsiirto tapahtuu verkottumalla muodostuneiden hermosolujen avulla [46]. Neuroverkoissa ensimmäinen kerros on datan käsittelyn aloittava syötekerros, jota seuraa yksi tai useampi piilokerros ja viimeisenä on ulostulokerros, johon tiedon käsittely päättyy [45]. Kerrokset rakentuvat neuroneista, joihin sisältyy käskylle välitettäviä parametrejä. Syvistä neuroverkoista puhutaan silloin, kun piilokerroksia on useampia, ja niitä voi olla jopa tuhansia. Piilokerroksilla jokaisella on oma tehtävänsä [45].

Kuvassa 3 on esitetty muokattu kuva Tuomisen ja Neittaanmäen syvän neuroverkon kerroksisesta toiminnasta, joka jäljittelee ihmisen aivoja [45].



Kuva 3. Syvän neuroverkon rakenne.

Kerroksissa neuronit suorittavat neuroverkon laskutoimituksia ja muita operaatioita. Ennen kuin neuronin tulos lähetetään seuraavalle neuronille, summa viedään ReLU-aktivointifunktioon (eng. Rectified Linear Unit), joka muuttaa lineaarisen syötteen epälineaariseksi. [45]

Lineaarisen aliavaruuden menetelmät

Toinen alakategoria kuviin perustuvissa lähestymistavoissa on lineaarisen aliavaruuden menetelmät (eng. Linear Subspace Methods, LSM), jossa havainnointi perustuu osajoukkoihin, jotka ovat suurempaa vektoriavaruutta [39]. Lineaarisen aliavaruuden menetelmät voidaan jakaa neljään eri alakategoriaan: Eigenface, todennäköisyyspohjainen Eigenspace, Fisherfaces ja Tensorfaces [39].

Tilastolliset menetelmät

Kolmas alakategoria on tilastolliset menetelmät (eng. Statistical Approaches, SA), joka on yksi tutkituimmista aiheista kasvojen tunnistuksessa [39]. Tilastolliset menetelmät jaetaan viiteen eri alakategoriaan: pääkomponenttianalyysi, tukivektori-kone, diskreetti kosinimuunnos, lokaalisuutta säilyttävä projektiio ja riippumaton komponenttianalyysi [39].

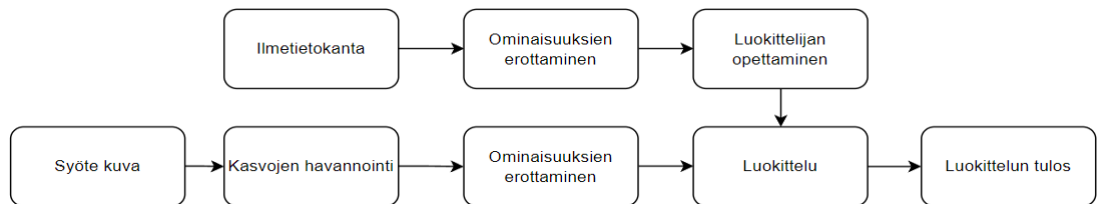
3.2. Ilmeiden tunnistaminen

Kasvojen ilmeen tunnistusjärjestelmien tavoite on ennustaa ja luokitella tunnetiloja [41]. Ensimmäisen vaiheen, eli kasvojen havaitsemisen jälkeen, kuvasta erotetaan

halutut kasvojen ominaisuudet, joiden perusteella pystytään arvioimaan kasvoissa esiintyvää ilmettä ja tunnetilaa [41]. Erotettujen piirteiden avulla kuva luokitellaan luokittelijan avulla, joka on opetusdatan avulla opetettu luokittelemaan ilmeitä [41]. Ilmeiden tunnistaminen voidaan jakaa kahteen päälähestymistapaan: tavanomaisiin lähestymistapoihin ja syväoppimiseen perustuviin lähestymistapoihin [42, 41].

3.2.1. Tavanomaiset lähestymistavat

Tavanomaiset lähestymistavat kasvojen ominaisuuksien erottamisessa, kuten ASM ja LBP, sisältävät yleensä luodut määriykset tai vain pinnallista oppimista [42, 41]. Nämä lähestymistavat vaativat vähemmän laitteistolta, eivätkä tarvitse paljoa dataa, jolloin ne ovat toimivia pienemmissä otoseteissä [42]. Tavanomaisen lähestymistavan omaavan FER-järjestelmän toimintaa on havainnollistettu kuvassa 4.

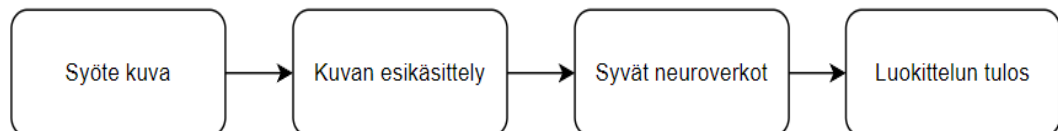


Kuva 4. Tavanomaisten lähestymistapojen toiminta.

3.2.2. Syväoppimiseen perustuvat lähestymistavat

Viime vuosien aikana laskentatehon kasvaessa tekoälyyn kuuluvan osa-alueen syväoppimisen (eng. deep learning, DL) lähestymistavat ovat yleistyneet laajasti [41]. Nämä lähestymistavat perustuvat syötedatasta syväoppimiseen neuroverkkojen avulla, jolloin luokittelu tapahtuu myös verkon sisällä [41]. Neuroverkkomallit pystyvät tutkimaan syötettyä dataa ja arvioimaan parhaimman tuloksen.

Syväoppiminen sopii silloin, kun testattavaa tietoa on paljon ja se on monimutkaista. Syväoppimista käytetään nykyään kaikissa älypuhelimissa, ja muun muassa puheen ja kasvojen tunnistuksessa [46]. Kuvassa 5 on havainnollistettu syväoppimiseen perustuvan FER-järjestelmän toimintaa [42].



Kuva 5. Syväoppimiseen perustuvien lähestymistapojen toiminta.

4. IHMISMÄISEN SOSIAALISEN ROBOTIN KEHITYS JA HAASTEET

Merkittävä osa tämän päivän sosiaalisista roboteista ei pyri mallintamaan tarkasti ihmisen kasvoja. Usein robotin kasvot ovat silmiä lukuun ottamatta liki ilmeettömät ja täysin valkoiset, kuten roboteilla Quori [47] ja Nao [48], tai robotti saattaa mallintaa esimerkiksi eläintä, kuten hyljevauvaa jäljittelevä robotti Paro [49, 50].

Tämänkaltaiset robotit eivät kykene tuottamaan monipuolisesti erilaisia ilmeitä, ja niiden tuottama viestintä jää siten suppeammaksi kuin ihmisen päätä jäljittelevän, ilmeitä tuottavan robottipään viestintä.

Yksi syy, minkä vuoksi osa robotinvalmistajista ei pyri ihmisen ulkonäön ja ilmeiden tarkkaan tuottamiseen on niin kutsuttu outo laakso -ilmiö. Ilmiö on tunnistettu jo 50 vuotta sitten, ja sen mukaan kun robottipää tai esimerkiksi vahanukke lähestyy ihmisen ulkonäkää, pienet eroavaisuudet aidon ihmisen kasvoihin verrattuna muuttuvat häiritseviksi tai jopa uhkaaviksi [51].

4.1. Kasvojen mekaniikka realistisessa robottipäässä

Ihmisenkaltaisia robottipäitä on kuitenkin mahdollista valmistaa jo nykyteknologialla tavalla, joka ei valmistajan mukaan tuota katsojalle häiritsevää outo laakso -vaikutusta [52]. HEFES -nimisen robottipään ilmeenmuodostus tapahtuu 32 pienen pään sisällä sijaitsevan servomoottorin avulla, jotka venyttävät elastista ihoa muodostaakseen erilaisiin tunnetiloihin liittyviä ilmeitä, kuten ärtymykseen kuuluvaa kurttuista otsaa tai yllättyneisyyteen liittyviä suureksi revähtäneitä silmämunia [52]. Nykyiset robottipäät kykenevät myös liikuttamaan niskaansa ja silmiään [53]. Luotujen ilmeiden määrä jää kuitenkin pieneen murto-osaan ihmisen ilmeikkyyteen verrattuna. Esimerkiksi Hanson Robotics -yhtiön vuonna 2014 luoma androidi Phillip K. Dick kykenee luomaan 48 ilmettä, kun taas ihminen kykenee useaan tuhanteen eri ilmeeseen [53]. Kuvassa 6 on vuodelta 2022 androidi Phillip K. Dick, jota on kehitetty muistuttamaan ihmistä mahdollisimman paljon.



Kuva 6. The Philip K. Dick android. / Rasmus Lerdorf, Flickr¹

4.2. Tekoölyn sosiaalinen älykkyys

Tekoölyn uskottavuutta mittaamaan on kehitetty koe, joka tunnetaan Turingin kokeena. Koe onnistuu, jos tekoöly onnistuu vakuuttamaan ihmisen, että hän on ollut vuorovaikutuksessa toisen ihmisen, eikä robotin kanssa. Yksi tapa järjestää koe on laittaa tekoöly vastaamaan ihmisen kirjoittamiin viesteihin tekstin välityksellä. Tällä tavalla voidaan eristää itse vastauksen mielekkyys puheen syntetisoinnin tai ilmeiden tuottamisen haasteista. Tekstipohjaisen kommunikaation välityksellä on onnistuttu jo luomaan tekoölyjä, jotka onnistuvat läpäisemään Turingin kokeen [54], mikä on suuri virstanpylväs ihminen-robotti-vuorovaikutuksessa. Turingin koe ei ole tekoölyn kehityksen päätepiste, mutta tarjoaa silti mielenkiintoisen tavan testata tekoölyä.

OpenAI:n GPT 3.5 -malli osaa tuottaa jo pitkiä tekstinpätkiä niin hyvällä laadulla, että lopputulosta on vaikeaa erottaa ihmisen kirjoittamasta tekstistä. Joissain tapauksissa GPT-mallin tuottama teksti on ollut myös laadukkaampaa kuin ihmisen tuottama teksti, esimerkiksi koetilanteessa ChatGPT [2] on saanut parhaimpaan 10 %:iin kuuluvat pisteet kokeeseen osallistuvien ihmisten joukosta. [55]

Aikaisemmat tekoölymallit ovat perustuneet usein toimintaperiaatteelle, jossa ohjelmoijat ovat luoneet katalogin valmiita vastauksia, joista tekoöly sitten valitsee sen, joka sopii parhaiten tilanteeseen. Tämä on tarkoittanut sitä, että vastaukset ovat olleet laadultaan hyviä, mutta että niitä on ollut rajallinen määrä valittavissa. Nykyiset isojen kielimallien tekoölyt puolestaan luovat vastauksen itse käyttäen luomiseen suurta määrää opetusdataa. Tämä avaa uusia mahdollisuuksia tekstin tyylin ja muodon suhteen, sekä lisää mahdollisten vastausten määrää loputtomasti. GPT 3.5-malli osaa jo ottaa vastapuolen perustunnetiloja huomioon laatiessaan vastausta. ChatGPT:tä

¹Kuva: The Philip K. Dick android: <https://www.mentalfloss.com/posts/philip-k-dick-author-facts/> / Rasmus Lerdorf, Flickr (Lisenssi CC BY 2.0)

pyydettiin kirjoittamaan lyhyt tervehdys ilman mainintaa kuulijan tunteesta, sekä syötteellä, jossa mainittiin kuulijan olevan vihainen.

Syöte: "tervehdi lyhyesti henkilöä"

Vastaus: "Hei sinulle!"

Toivottavasti päiväsi on sujunut hyvin tähän asti. Halusin vain pysähtyä hetkeksi lähettämään sinulle lämpimiä terveisiä. Toivon, että kohtaat päivässäsi iloa, onnistumisia ja paljon positiivisia hetkiä. Muista, että olet arvokas ja ainutlaatuinen juuri sellaisena kuin olet. Pidä huolta itsestäsi ja nauti hetkestä!

Ystävällisin terveisin, ChatGPT"

Syöte: "tervehdi lyhyesti henkilöä, joka vaikuttaa vihaiselta"

Vastaus: "Hei!"

Toivottavasti voit hyvin. Halusin vain pysähtyä hetkeksi ja sanoa, että välitän. Jos haluat puhua tai jakaa ajatuksiasi, olen täällä kuuntelemassa. Toivon sinulle rauhallista ja positiivista päivää. Muista, että sinulla on voimaa käsitellä haasteita, ja toivottavasti löydät pian iloa ja levollisuutta.

Ystävällisin terveisin, ChatGPT"

Syötteellä, jossa mainitaan kuulijan vaikuttavan vihaiselta, GPT tuottaa tervehdyksen, joka on tyyliltään ymmärtävä ja sovitteleva. Jos tunnetilaa ei ole mainittu, tervehdys on tyyliltään positiivinen. Todellisuudessa ihminen ei koe ainoastaan yhtä perustunnetta kerrallaan. Uudella ison kielimallin mukaisella tekoälyllä voidaan mahdollistaa se, että erilaisia tunnetiloja otetaan huomioon joustavasti vastausta laatiessa. ChatGPT:tä pyydettiin vielä tervehtimään henkilöä, joka vaikuttaa sekä vihaiselta että surulliselta.

Syöte: "tervehdi lyhyesti henkilöä, joka vaikuttaa vihaiselta ja vähän surulliselta"

Vastaus: "Hei sinulle,

Haluaisin lähettää sinulle lämpimiä terveisiä tänään. Huomaan, että ilmeesi voi olla vihainen ja surullinen. Haluan muistuttaa sinua siitä, että tunnetilat vaihtelevat, ja on täysin luonnollista kokea erilaisia tunteita. Jos haluat puhua tai jakaa ajatuksiasi, olen täällä kuuntelemassa. Toivon, että löydät rauhaa ja lohtua päivääsi. Muista, että aina on mahdollisuus parempaan ja että olet vahva. Toivon sinulle valoa ja positiivisia hetkiä eteesi.

Lämpimin terveisin, ChatGPT"

Tällä kertaa vastauksen tyyli on ymmärtävä, mutta myös vähän kannustava. Puhtaasti sosiaalisen älykkyyden näkökulmasta ollaan siis jo pisteessä, että tekoäly pystyy kommunikoimaan ihmiselle luontevalla tavalla. Tästä siihen, että robottipää mahdollisesti läpäisisi Turingin kokeen, liittyy kuitenkin vielä paljon ratkaistavia haasteita. Robottipään estetiikan ja puheentuottamisen pitäisi vielä kehittyä ja järjestelmän viive pitäisi saada laskettua tarpeeksi alhaiseksi.

5. TOTEUTUS

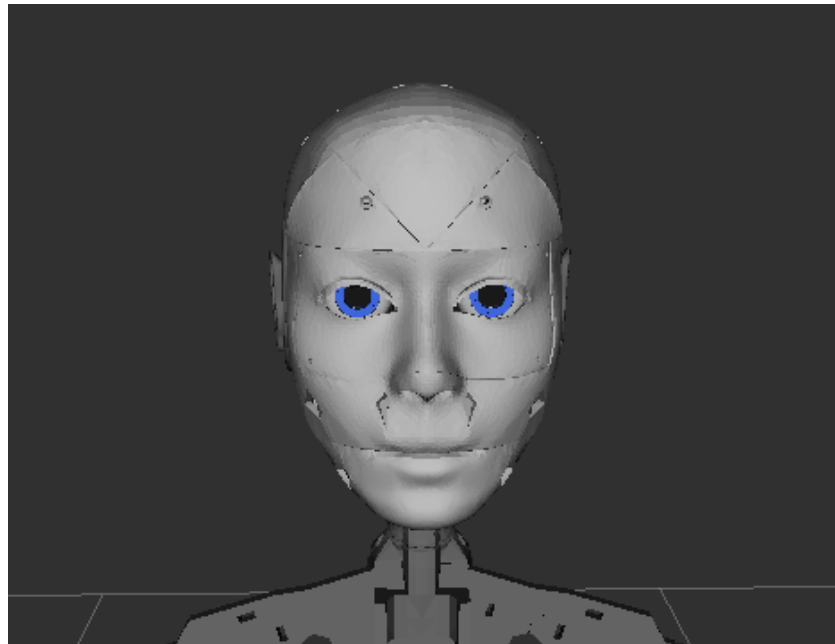
Projektissa luotiin järjestelmä, jossa web-kameralla kuvataan käyttäjän kasvojen ilmeitä, ja ilmeiden perustella ohjataan robottia. Projektissa esitellään robotin kykyä tunnistaa robotin kanssa vuorovaikutuksessa olevan ihmisen perusilmeitä, mutta ihmisen selkeiden perusilmeiden vaikutusta robottiin testattiin vain iloisen ja vihaisen ilmeen osalta, jolloin robotti käänsi katseen sivulle ja takaisin. Käytettiin kuutta perusilmettä: neutraali, iloinen, vihainen, surullinen, yllätynyt ja pelokas. Tärkein ohjelmistojen valintaan vaikuttava kriteeri oli se, että ohjelmistot olivat ohjeineen ja ongelmanratkaisuineen saatavilla vapaasti Internetistä. Olennaisena tekijänä myös ohjelmistojen valintaan vaikutti sellaisten ohjelmistojen valinta, kuten OpenCV ja Keras, jotka on koettu toimivan hyvin vastaavissa ilmeiden tulkintaa testaavissa projekteissa. Huomioon on olennaista myös ottaa tehtävän tietokonenäön haasteellisuus ja ohjelmistolle asetettavat vaatimukset.

Kuvien perusteella tapahtuvan neuroverkkoihin pohjautuvan lähestymistavan ja ilmeiden tunnistamisessa käytettävän syväoppimiseen pohjautuvan koneoppimisen avulla ohjelmalle pystytään opettamaan pohjatiedot ja käyttäjän toiminnan avulla, miten tunnistetaan, luokitellaan ja ennustetaan tulkittavia asioita. Nykyisten tietokoneiden avulla pystytään käsittelemään suurempia tietomääriä paremmin, ja lisäksi neuroverkot jäljittelevät ihmisen aivojen toimintaa. [45]

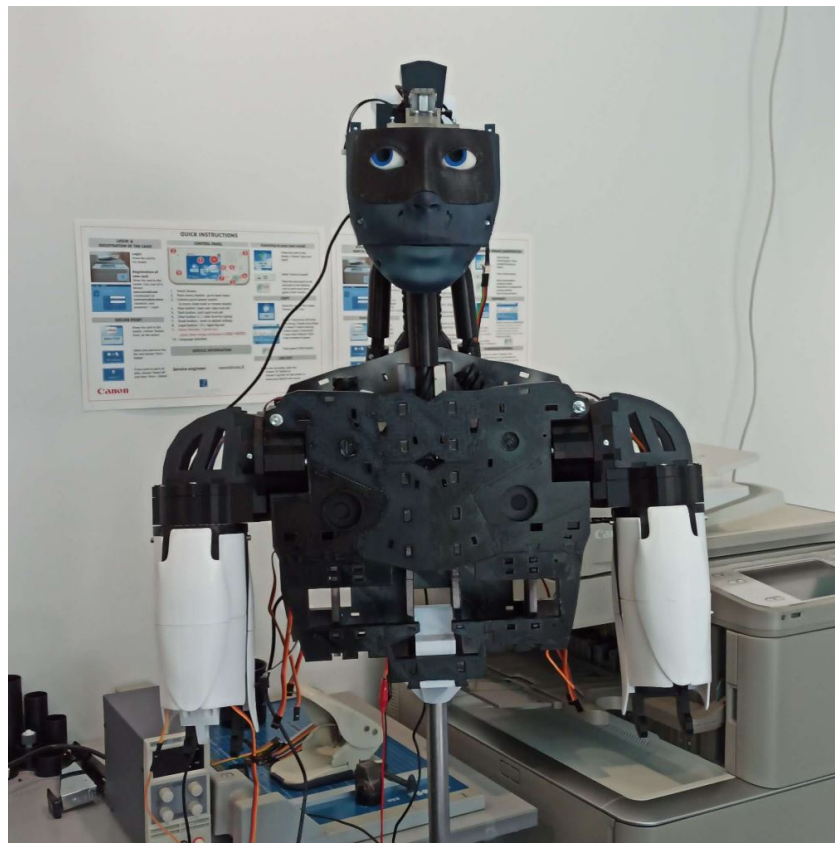
5.1. Laitteisto

Järjestelmän toteuttamisessa käytettiin tietokonetta, web-kameraa ja robotin päätä. InMoov-robotti [56] on ensimmäinen avoimen lähdekoodin 3D-tulostettu ihmisen kokoinen robotti. Sen ohjaamiseen tarvitaan toinen tietokone ja robotti, joka kopioi kasvojen ilmeitä. Tässä työssä käytettiin Oulun yliopistolta käytössä olevaa robottia. Se on itserakennettu avoimen lähdekoodin mukaan. Robotti koostuu servojen avulla liikuteltavasta päästä, sen silmässä olevasta web-kamerasta sekä sillä on mikrofoni ja kaiuttimet, jos tarvitaan äänen tunnistamista ja tuottamista. Tässä työssä käytettiin päästä silmiä ja kykyä kääntää pää sivulle.

Robotin silmien ohjaaminen tapahtuu kahden Dynamixel XL-320 -servon avulla, toinen näistä ohjaa vaakasuuntaisliikettä ja toinen pystysuuntaista. Robotin pään ohjaaminen tapahtuu neljän Dynamixel XL-430 servon avulla. Robotin silmiä pystytään ohjaamaan ihmissilmän tavoin molempia servoja yhtä aikaa käytettäessä. Kuvassa 7 on havainnollistettu InMoov-robotin pää RViz-simulaattorissa ja kuvassa 8 on esitettyä Oulun yliopistolla asuva fyysinen versio InMoov-robotista.



Kuva 7. InMoov robotin pää RViz-simulaattorissa.



Kuva 8. Oulun yliopistolla käytössä oleva InMoov-robotti.

5.2. Ohjelmistot

Tässä aliluvussa esitellään työn kehittämisessä käytetyt ohjelmistokirjastot, kasvojen tunnistukseen ja ilmeiden havainnoinnin toteutukseen valitut menetelmät, robotin ohjaukseen käytetty järjestelmä ja Keras-mallin kouluttamisessa hyödynnetty kuvatietokanta.

5.2.1. *OpenCV*

Ohjelmiston kehittäminen tapahtui Python-ohjelmointikielellä ja apuna käytettiin OpenCV:n [57, 58] avoimen lähdekoodin ohjelmistokirjastoa. OpenCV sisältää yli 2500 eri algoritmia, myös viimeisintä tekniikkaa konenäköön ja koneoppimiseen. Se tarjoaa valmiita toteutuksia yleisesti tietokonenäössä ja digitaalisessa kuvankäsittelyssä käytettäviin algoritmeihin. Esimerkiksi algoritmeja, joilla pystytään muun muassa tunnistamaan kasvoja, erilaisia kohteita ja liikkeitä sekä seuraamaan myös liikkuvia kohteita. Ohjelmakirjasto on toteutettu C++ -ohjelmointikielellä, joka mahdollistaa nopeamman suorituksen kuin puhtaasti Pythonilla kirjoitetut versiot. Kirjastoa käytetään paljon muun muassa tunnetuissa ohjelmistoyrityksissä ja tutkimuksissa.

5.2.2. *YuNet*

Kasvojen tunnistamiseen kuvasta tai videolta on perinteisesti käytetty OpenCV:n valmista Haar Cascade -menetelmää, mutta teknologia kehittyy koko ajan, ja suoritustehokkuus, havaitsemiskyky ja tarkkuus ovat vaatimuksia, jotka vaikuttavat mitä ohjelmaa käytetään. OpenCV:n YuNet, jonka Shiqi Yu kehitti vuonna 2018, ja avoimen lähdekoodin vuonna 2019, pohjautuu konvoluutioneuroverkkoihin. CNN-perustuvissa malleissa on aina tehtävä kompromissi tarkkuuden ja tehokkuuden välillä. YuNet on suunniteltu käytettävän sulautetuilla laitteilla, joten se toimii tosi tehokkaasti, mutta pystyy silti kilpailemaan muiden CNN-mallien kanssa. Toisin kuin muut CNN-mallit, kuten RetinaFace, TinaFace, MTCNN ja Dlib, YuNet ei tarvitse GPU:ta toimiakseen nopeasti, vaan se suoriutuu nopeasti CPU:lla [59, 60]. Se on tehokas ja voidaan ladata monille laitteille. YuNet voi saavuttaa 1000 kuvaa per sekunti tehokkuudessa ja silti pystyä tarkasti tunnistamaan kasvoja. Lisäksi se pystyy tunnistamaan vaikeitakin kasvokulmia, esimerkiksi sivuttaisia tai osittain piilossa olevia kasvoja.[61]

5.2.3. *YuNetin ja Haar Cascaden väliset erot*

Chen Zhanjie on tutkimuksessaan eri testeillä verrannut Haar Cascadea ja YuNetiä ja osoittanut, että YuNet on parempi kuin Haar Cascade erilaisten kasvojen havaitsemisessa, riippumatta siitä minkä kokoisia kasvokuvat ovat. Se vie vain 1/5 siitä ajasta, mikä menee perinteisillä malleilla, joihin Haar Cascade kuuluu. [61, 60] YuNetilläkin oli vaikeuksia tunnistaa pienempiä kuvia, mutta se oli kuitenkin parempi

verrattuna Haar Cascadeen, joka saattoi tehdä paljon virrehavaintoja, eikä pystynyt tunnistamaan pieniä kuvia tai osittain peitossa olevia kasvoja. Testauksesta esimerkki kuvassa 9, ja taulukossa 1 on havaintomäärien ja käytetyn ajan ero.



Kuva 9. Haar Cascaden ja YuNetin eroja kasvojen havaintomäärissä ja käytetystä ajasta.³

Taulukko 1. Havaintojen määrä ja käytetty aika tulokset. Testissä käytetty kuvaa 9, jonka resoluutio on 1280x853

Malli	Havaintojen määrä	Käytetty aika
Haar Cascade	19	301 ms
YuNet	53	84 ms

Jos sovellusta ollaan tekemässä sulautetulle laitteelle, niin mallin viemä tila voi olla ratkaiseva tekijä sitä valittaessa. haarcascade_frontalface_default.xml -malli vie 908 kilotavun verran tilaa kun taas face_detection_yunet_2022mar.onnx -malli vie 337 kilotavun verran. Tämä tilan viemä ero saattaa olla kriittinen sulautetuilla laitteilla, joilla voi olla hyvin vähän keskusmuistia. YuNet-malli vie siis voiton vanhasta Haar Cascade -mallista joka tavalla, eikä ole mitään syytä olla käyttämättä sitä Haar Cascade -mallin sijasta. [61]

5.2.4. TensorFlow ja Keras

TensorFlow on koneoppimistutkimuksia varten kehitetty avoimen lähdekoodin koneoppimisalusta, joka tukee prosessoreita, näytönohjaimia, mobiililaitteita ja selaimia. TensorFlow sisältää hyödyllisiä toimintoja, jotka auttavat yksinkertaistamaan

³Original image by Matheus Bertelli. license: free to use.

ratkaisuja luotaessa koneoppimismalleja. Python on suositeltu kieli Tensorflow ohjelmointirajapinnan (eng. Application Programming Interface, API) käyttämiseen, sillä on paras tuki ja täyteläisin toteutus. Tensorflow on itse toteutettu C++ -kielellä, joten se antaa paremman suoritusnopeuden Python ohjelmiin. [62, 63]. TensorFlow:n päällä on Keras [64], joka on Pythonilla kirjoitettu API syväoppimiskäytännön suorittamiseen. Sekä TensorFlow että Keras on kehitetty niin, että lopputulokseen pääseminen tapahtuu mahdollisimman yksinkertaisesti, mutta kuitenkin tehokkaasti ja nopeasti.

5.2.5. Robot Operating System, ROS

Robotin ohjausta varten oli käytössä ROS-järjestelmä tai teknisesti ROS2, mikä on päivitetty versio ROS1-järjestelmästä. ROS2:ssa on panostettu enemmän tietoturvallisuuteen, otettu huomioon eri verkkojen erilaiset rakenteet ja se soveltuu paremmin sulautetuille laitteille. ROS2 on tehty enemmän teollisuusstandardit mielessä ja sopii paremmin eri käyttötarkoituksiin. [65]

ROS [66, 67] on avoimen lähdekoodin käyttöjärjestelmä, jonka avulla pystytään ohjaamaan tietokoneella robotin toimintoja. Kommunikaatio robotin eri osien välillä pohjautuu tilaus- ja julkaisujärjestelmään, jossa komponentit lähettävät tietoa toisille kyseisen tiedon tilaajina toimiville komponenteille. Viestien kulku tapahtuu aiheuttajien tietoväylässä. Sekä lähettävät että vastaanottavat komponentit eivät tiedä tiedon alkuperää tai kohdetta, sillä viestien ohjauksesta vastaa isäntäkomponentti (eng. master component). Muut komponentit ovat rekisteröityneet isäntäkomponenttiin. ROS-ympäristö mahdollistaa käyttöjärjestelmältä odotettavia palveluita, esimerkiksi matalan tason laitteistokontrollin, toiminnallisuuden käyttöönoton ja prosessien välillä tapahtuvan viestien välityksen sekä pakettien käsittelyn. ROS:ssa ohjelmistokomponenttien järjestämiseen käytetään paketteja, jotka sisältävät muun muassa ROS-prosesseja, ohjelmistokirjastoja, dataa, konfiguraatitiedostoja tai muita tiedostoja muodostaen näin yhdessä toimivan kokonaisuuden. ROS-prosessit muodostavat ROS:in toiminnan. [68]

5.2.6. FER-2013

Mallien opetukseen käytetyistä kuvatietokannoista, joita on saatavilla monia erilaisia, ja jotka voivat sisältää miljoonia kuvia, Facial Emotion Recognition (FER) FER-2013-datasetti vaikutti sopivimmilta tähän projektiin. Lisäksi sitä oli käytetty vastaavanlaisissa tunteiden tunnistamiseen liittyvissä tutkimuksissa. Se on julkaistu kansainvälisessä koneoppimisen konferenssissa International Conference on Machine Learning (ICML) vuonna 2013 [69], jolloin luotiin kilpailua varten malli, jonka avulla pystyttiin tunnistamaan kuvien tunteet mahdollisimman tarkkaan.

Tietokannassa jokainen kuva on luokiteltu tunnekkategorioihin, jotka koostuvat 48 pikseliä x 48 pikseliä olevista mustavalkokuvista. FER-13 käsittää seitsemän erilaista mikroilmettä: viha, inho, ilo, neutraali, pelko, suru ja yllättyneisyys. Kuvia on kaikkiaan seitsemälle eri tunteelle 35887, ja niissä on kuvia sekä validointiin että opettamiseen [70]. Kasvojen ilmeiden tunnistuksen tutkimuksessa mm. Ian

Goodfellow [69] käytti kasvojen tunteiden tunnistukseen FER-2013-kuvatietokantaa, ja sai tarkkuudeksi 65,5 %. Yksityisissä FER-13 kuvilla tehdyillä tarkkuustesteillä oli vaihtelua 67.484 ja 71.162 %:in välillä [69]. Kuvassa 10 on esimerkki FER-13 tietokannan tunteiden ilmaisuista.



Kuva 10. FER-2013 avoimesta tietokannasta.⁴

5.3. Konenäkösovelluksen toteuttaminen

Konenäkösovelluksen toteuttamisessa oli kaksi päävaihetta, jotka muodostavat toimivan liukuhihnaimaisen sovelluksen toiminnan. Ensimmäisenä erotetaan kasvojen kuva web-kameran kuvasta, ja toisessa vaiheessa tehdään analyysi, jonka perusteella päätellään, mikä tunne kasvoilla mahdollisesti on. Näiden vaiheiden toteuttamiseksi tarvittiin OpenCV:tä ja Kerasta. Nämä kaksi laajaa ohjelmakirjastoa olivat ainoat tarvittavat. Ensimmäisessä vaiheessa käytettiin OpenCV:stä YuNet-mallia, jolla tunnistettiin ja erotettiin kasvo kuvasta Kerasta varten. Seuraavassa vaiheessa käytettiin Kerasta tekemään analyysi ensimmäisen vaiheen jälkeen saatavalle kuvalle. Robottipään servoja ohjataan toisen vaiheen analyysin tuottamien tietojen mukaan.

5.3.1. Kasvojen tunnistus kuvasta

Ensimmäisessä vaiheessa käytettiin valmiita painoja YuNet CNN-mallille. CNN-verkonpainot tarjosi suoraan OpenCV-ohjelmakirjasto, eli projektia varten ei koulutettu erikseen mallia erottamaan kasvoja kuvasta. Mallilla saatiin erotettua kasvojen kuva web-kameran kuvasta. Tämän jälkeen kasvojen kuvalle tehtiin jatkokäsittelytoimenpiteitä OpenCV:n avulla. Kuva täytyi muuttaa 48x48 resoluutioiseksi ja harmaansävyiseksi, jotta se pystytään käsittelemään seuraavassa vaiheessa.

5.3.2. Tunteiden tunnistamisohjelma

Toisessa vaiheessa käytettävän Keras-mallin toteutus tehtiin alusta alkaen. Kouluttamiseen käytettiin FER-2013 tietokannasta saatavia kuvia. FER-2013

⁴Kuvat: <https://www.kaggle.com/datasets/msmbare/fer2013> (Database Contents Licence (ODbL) v1.0)

tietokanta on jaettu kolmeen osaa: koulutus kuvat, julkiset kuvat, ja yksityiset kuvat. Koulutuksessa käytettiin koulutus kuvia, validoinnissa julkisia kuvia ja mallin arvioinnissa yksityisiä kuvia. Ilmeitä tunnistava Keras-malli luotiin tekemällä 10-kerroksinen konvoluutionaalinen neuroverkko. Mallin arkkitehtuuri on esitetty kuvassa 11. Mallin arkkitehtuuri pidettiin yksinkertaisena johtuen huomioon otettavista laitevaatimuksista. Kuvan 11 mukaisesti malli koostuu neljästä konvoluutiokerroksesta, neljästä näytteistyskerroksesta ja kahdesta täysin yhdistetystä kerroksesta. Kerrosten välillä suoritettiin erän normalisointi (eng. Batch Normalization) nopeuttamaan mallin kouluttamista ja lisättiin pudotuskerrokset (eng. Dropout Layer) vähentämään mallin ylikouluttautumista koulutuskuville.



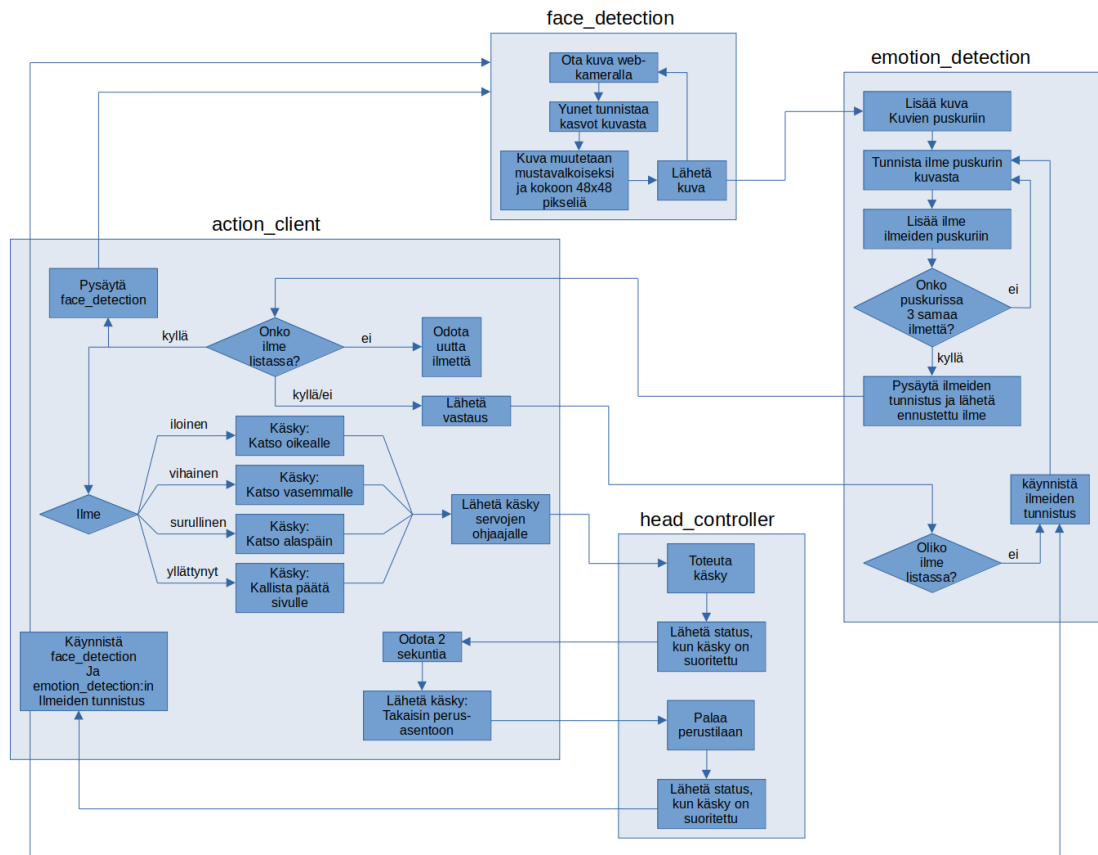
Kuva 11. Käytetty konvoluutionaalisen neuroverkon arkkitehtuuri.

Mallin koulutuksessa hyperparametreja ja ohjelman asetuksia optimoitiin ajamalla malliohjelmia useita kertoja, kunnes tarkkuus alkoi olla halutulla tasolla. Kun päästiin vaiheeseen, jossa mallin rakenteeseen ja opetusdataan tehdyt muutokset olivat varsin pieniä, mallin kehittäminen voitiin lopettaa. Viimeisenä vaiheena valmista mallia pakattiin karsimalla (eng. Pruning) osa mallin parametreista pois huolehtien samalla siitä, että mallin tarkkuus ei kärsinyt huomattavasti pakkaamisesta.

5.4. ROS:lla yhdistetty kokonaisuus

ROS toimi tässä työssä liimana, joka yhdisti konenäkösuuden ja robotin ohjaamisen. Se tarjosi julkaisija- ja tilaajajärjestelmän, jonka avulla eri osakokonaisuudet pystyivät kommunikoimaan keskenään. Ohjelma jakaantui kolmeen eri suoritusyksikköön (eng. Node): "face detection", "emotion detection" ja "action client". Face detection node vastaa web-kameran hallinnasta ja kasvokuvan erottamisesta web-kameran kuvasta. Tämä node lähettää käsitellyn kuvan emotion detection nodelle, jonka tehtävänä on päätellä oikea tunne kasvoilta. Viimeisenä emotion detection lähettää kuvan action client nodelle, joka on tämän ohjelman päänode. Sen tehtävänä on ohjata muita nodeja, toimia tilakoneen tavoin, ja estää niitä tekemästä turhaa työtä, kuten ottaa kuvia kasvoista, vaikka robotti on jo suorittamassa jotain liikettä. Se vastasi myös liikkumiskäskeyjen lähettämisestä, kun se vastaanotti havaittuja ilmeitä. Ohjelmistokokonaisuuden rakenne on esillä kuvassa 12.

Ohjelmistokokonaisuuden jakaminen useaan eri nodeen toi mukaan sekä etuja että haittoja. Suuri haaste on eri ajoitusten kanssa, sillä kasvojentunnistus oli käytännössä



Kuva 12. Kaavio eri nodejen yhteistoiminnasta ROS:lla.

paljon nopeampi suorittaa yhdelle kuvalle kuin ilmeentunnistus. Kasvojentunnistus vei aikaa testeissä useita kertoja vähemmän aikaa kuin ilmeentunnistus. Kaikkein hitain osuus on robottipään servojen liikuttaminen, johon kuluu useita sekunteja, eli kymmeniä kertoja enemmän kuin kasvojentunnistukseen. Tämä tarkoittaa sitä, että jos kaikkia nodeja ajettaisiin rinnakkain ilman odotuksia, servot saisivat uusia käskyjä, ennen kuin edellinen on vielä suoritettu loppuun. Varhainen versio ohjelmistokokonaisuudesta toimi niin, että action client odotti servojen pääsevän loppuun toiminnoissaan ennen seuraavan käskyn lähettämistä, mutta muuten kaikkia nodeja ajettiin rinnakkain koko ajan. Tämä sai aikaan sen ongelman, että tietokoneen prosessori oli koko ajan maksimikäytöllä, ja järjestelmän käyttö muuttui hitaaksi ja vaikeaksi. Tästä syystä järjestelmään lisättiin muiden nodejen pysäytykset ja käynnistykset action client nodeen. Tämä ratkaisu ei ole täydellinen, sillä viestien lähettämiseen eri nodejen välillä kuluu myös aikaa, ja kasvojentunnistus ehtii tunnistaa kasvat parista kuvasta turhaan, ennen kuin se vastaanottaa viestin pysäyttää tunnistus. Useamman noden ratkaisu oli hyvä siinä mielessä, että eri nodeja voitiin kehittää ja testata toisistaan erillään. Jo varhaisessa vaiheessa kehitystä siirryttiin Haar Cascade -kasvojentunnistuksesta YuNetiin, eikä tämä päivitys ei vaatinut muutoksia muihin nodeihin. Useamman erillisen noden ansiosta suorituskykyä testatessa voitiin ajaa pelkkiä kasvojen ja ilmeiden tunnistuksia erillään action clientistä ja robotin simulaatioympäristöstä, joka on prosessorille raskas ohjelma suorittaa.

6. TESTAUS

Tässä luvussa esitellään, miten testaus suoritettiin, testien tulokset ja niiden analysointi. Testit suoritettiin InMoov-robotin RViz-simulaattorin ja projektintekijöiden kotitietokoneiden avulla käyttäen web-kameraa. Konenäkösovellus testattiin lopulta toimivaksi, ja se pystyi tunnistamaan kameran edessä olevat kasvat. Tavoitteena testauksessa oli myös saada selville, kuinka hyvin robotti pystyi tunnistamaan eri tunteet kasvojen ilmeistä. Tunteista inho jätettiin pois, koska sille oli kaikista vähiten opetuskuvia saatavilla tietokannassa, vain 436, joten se aiheutti paljon virhetulkintoja. Muille ilmeille opetuskuvia sen sijaan oli useita tuhansia. Ohjelmistoa ajettiin myös Oulun yliopistolla fyysisellä robotilla. Tällöin varmistettiin, että ohjelmisto toimii robotin silmiin asennetun kameran ja päätä liikuttavien servojen kanssa.

6.1. Ohjelmistokokonaisuuden testaus

Tässä työssä testattiin yksittäisten komponenttien toimintaa ROS-ympäristöön liitettynä ja näiden komponenttien toimintaa yhtenäisenä kokonaisuutena. Työn komponentteja olivat kasvojen tunnistusohjelma, ilmeen tunnistusohjelma, ja näiden yhteisvaikutuksella toimiva robotin päätä liikuttava kokonaisuus. Testaaminen jakautui kahteen osioon: ohjelmiston osien käyttämään aikaan järjestelmässä ja ilmeitä tunnistavan Keras-mallin tarkkuuden tutkiminen ROS-ympäristössä. Tämän lisäksi testattiin ja varmistettiin robotin pään liikkumisen toimivuus tunnistettujen ilmeiden seurauksena.

6.1.1. Ohjelmiston suorituskyky

Ohjelmiston osien suorituskykyä ROS-ympäristössä tarkasteltiin seuraamalla jokaisen yksikön käyttämää aikaa täydellä toimintakierroksella. Viestien lähettäminen toisille nodeille sekä vastauksen odottaminen jätettiin kuitenkin ajanoton ulkopuolelle. Suoritusajan testaamisessa käytettiin Intel Core i5-5200U -prosessoria sekä web-kameraa resoluutiolla 1280x720. Ohjelmistoa ajettiin Oracle VirtualBox -ohjelmalla virtuaalikoneessa, jolle oli asennettu Ubuntu 20.04 -käyttöjärjestelmä. Taulukon 2 mukaisesti kasvojentunnistus yksikön toiminta vei keskimäärin 181,8 millisekuntia. Ilmeen tunnistus yksikön tehdessä kolme peräkkäistä saman ilmeen tunnistusta monisäe ajona, yksikön suorittamiseen meni keskimäärin 1300,1 millisekuntia. Action client -yksikön toiminta kasvojentunnistussyksikön käynnistämisestä käskyn lähettämiseen servoja kontrolloivalle palvelulle, eli aika, joka kuvaa järjestelmän täydellisen kierroksen kestoa, vei kokonaisuudessaan keskimääräisesti 1671,4 millisekuntia.

Taulukko 2. Jokaisen noden viemä keskimääräinen suoritus aika täydellä kierroksella

Face detection node	Emotion recognition node	Action client
181.8 ms	1300.1 ms	1671.4 ms

Ohjelman suorituskykyä testatessa tarkasteltiin myös pelkän YuNet-kasvontunnistuksen sekä Keras-ilmeentunnistuksen viemää aikaa yhdelle kuvalle yhdellä säikeellä ajettuna. Testiajo toistettiin sekä RViz-simulaatioympäristön kanssa, että ilman sitä. Robotin simulointi käytti paljon prosessorin suorituskykyä ja suoritusajat olivat merkittävästi nopeammat silloin, kun RViz-ympäristöä ei ajettu taustalla.

Taulukko 3. Jokaisen noden viemä keskimääräinen suoritus aika täydellä kierroksella

	YuNet face detection	Keras emotion detection
With RViz	169.7 ms	953.6 ms
Without RViz	98.1 ms	604.1 ms

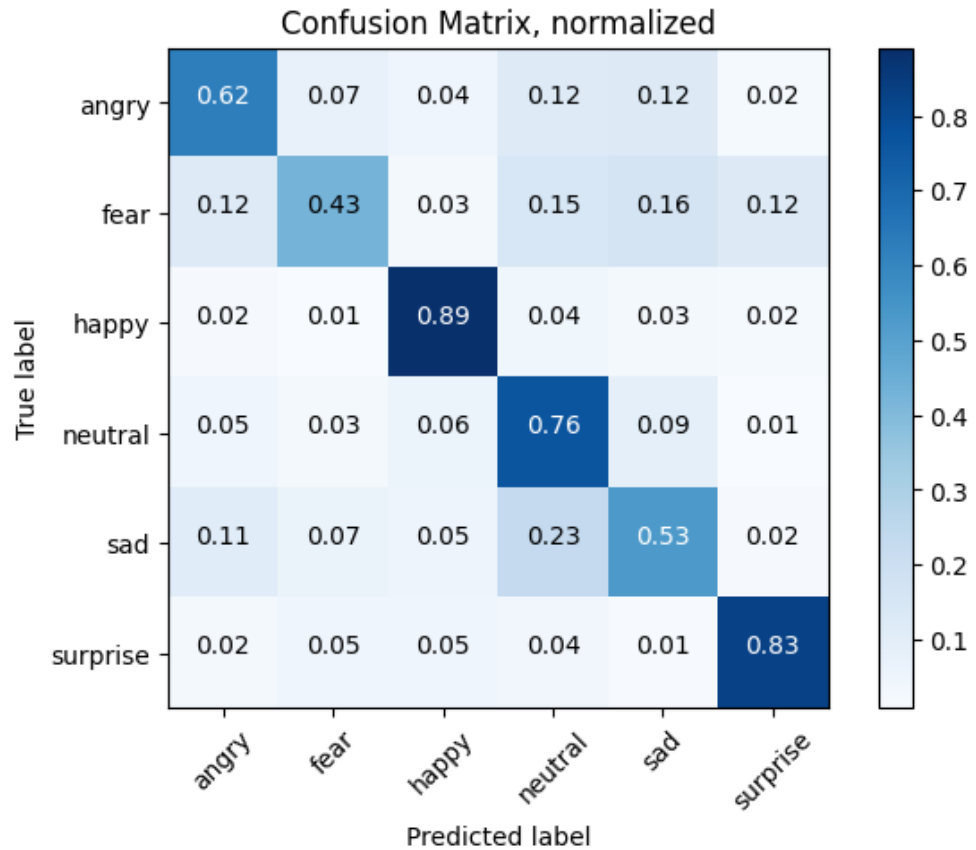
6.1.2. Ilmeentunnistusohjelman tarkkuus

Mallin tarkkuutta mitattiin aluksi testaamalla malli FER-2013 yksityiskuvista tehdyllä testisetillä, jota ei ollut käytetty mallin kouluttamisessa tai validoinnissa. Kuvan 13 mukaisesti koulutettu malli kykeni tunnistamaan iloisen, yllättyneen ja neutraalin ilmeen kohtalaisella tarkkuudella, mutta ilmeet viha, pelko ja suru olivat mallille hieman haastavampia ja malli sekoitti niiden luokittelun usein keskenään. Näiden ilmeiden luokittelu sekoittui myös neutraaliin ilmeeseen huomattavissa määrin.

Mallin toimivuutta testattiin myös käytännössä osana koko työn ohjelmisto kokonaisuutta ROS-ympäristössä. Testaus suoritettiin projektintekijöiden kotitietokoneilla hyödyntäen RViz-simulaattoria Inmoov-robotin simuloimiseen. Testauksessa koehenkilö oli metrin etäisyydellä web-kamerasta mallille opetettu ilme yllään. Järjestelmä käynnistettiin ja kirjattiin ylös ilmeentunnistussyksikön lähettämä viesti action client -yksikölle. Tämä toistettiin jokaiselle malliin koulutetulle ilmeelle viisi kertaa valoisassa sekä hämärässä tilassa. Koehenkilöitä testauksessa oli kokonaisuudessaan 6 kappaletta. Testauksessa ilmenneet erot tilan valoisuudessa on esitetty taulukossa 4. Taulukon kaikki sarakkeet esittävät mallin tarkkuutta, mutta niillä on eri merkitys. Taulukossa esitetty tarkkuus-sarake (eng. Accuracy) kertoo, kuinka suuren osan malli luokitteli syöte kuvista oikein. täsmällisyys-sarake (eng. Precision) kertoo keskimääräisen luvun, kun luokka on luokiteltuna oikein suhteutettuna luokan saamiin luokitteluihin yhteensä. Kolmas sarake, eli tunnistuskyky (eng. Recall), jota voi myös herkkyydeksi sanoa, kertoo kuinka monesti malli luokitteli luokan keskimääräisesti oikein suhteutettuna luokan kokonaismäärään. Viimeisessä sarakkeessa F1-score kertoo harmonisen keskiarvon precision- ja recall-sarakkeista. Kokonaisuudessaan järjestelmän luokkakohtaista ilmeiden tunnistustarkkuutta on havainnollistettu konfuusiomatriisissa joka löytyy kuvasta 14.

Taulukko 4. Ilmeen tunnistuksen tarkkuuslukuja

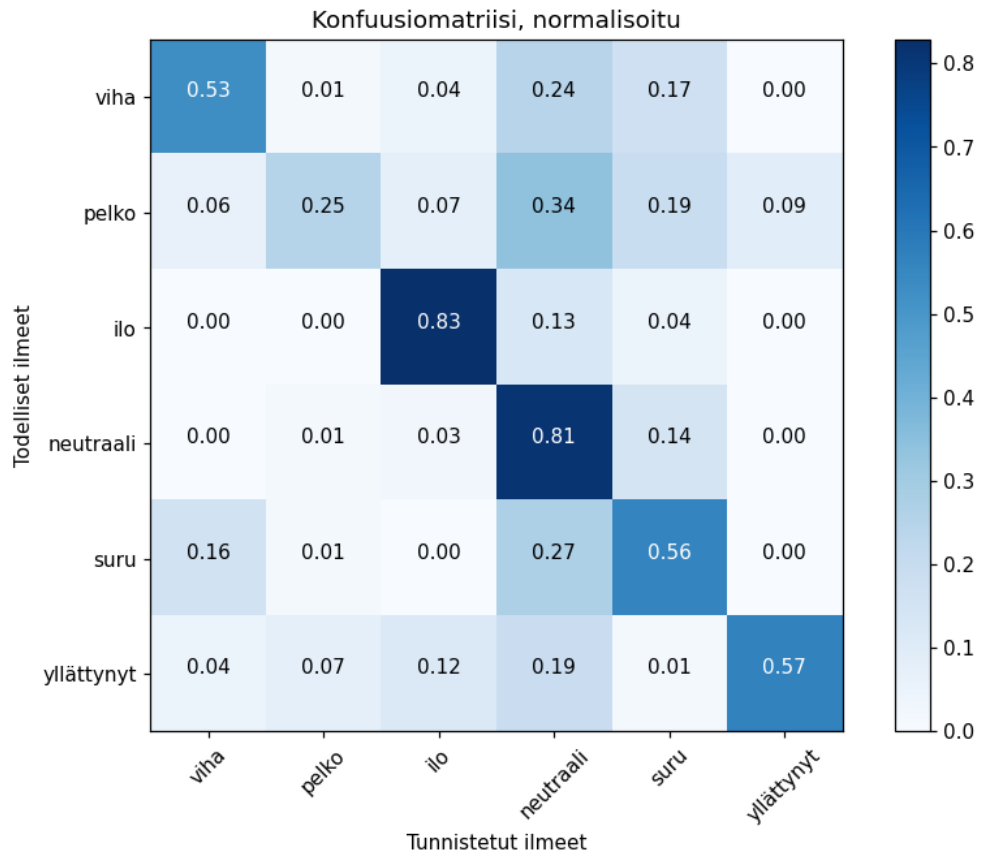
Testset	Lighting	Accuracy	Precision	Recall	F1-score
FER-2013	-	0.6941	0.6910	0.6776	0.6731
Robot	High brightness	0.6095	0.6725	0.6095	0.6067
Robot	Low brightness	0.5749	0.5392	0.4857	0.4766
Robot	Combined	0.5923	0.5565	0.5041	0.4997



Kuva 13. Normalisoitu konfuusiomatriisi ilmeitä luokittelevasta Keras-mallista FER-2013 testisetistä.

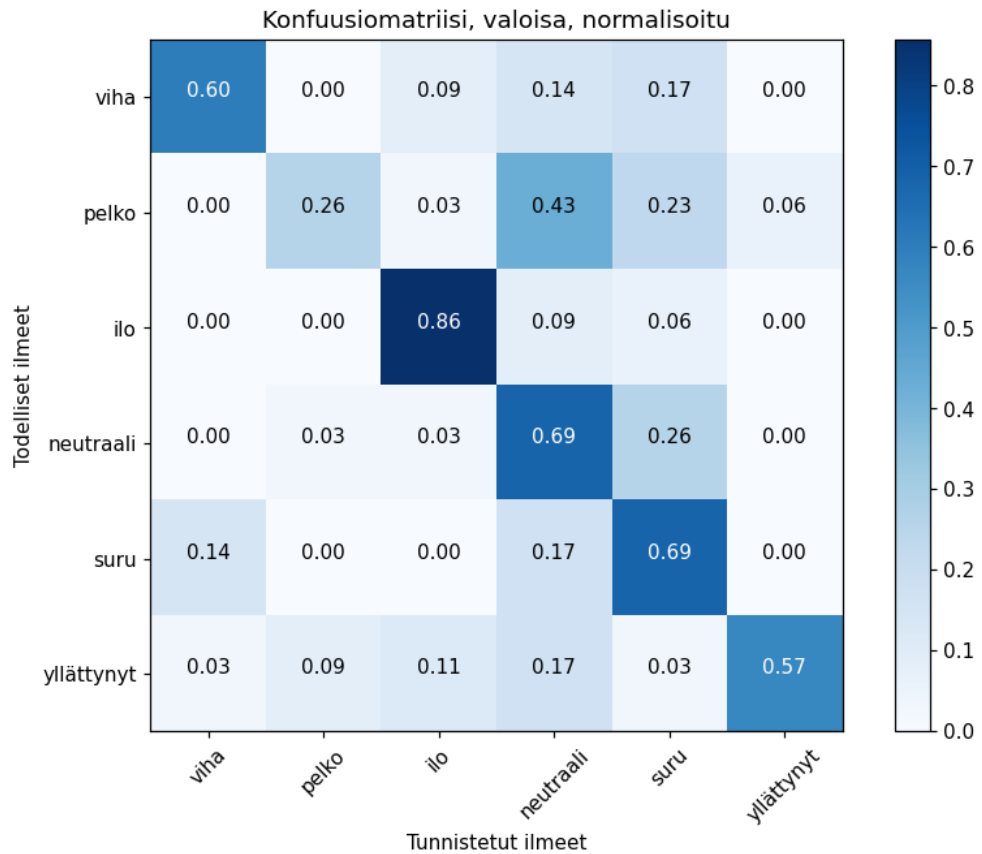
6.2. Ohjelmiston ajo fyysisellä robotilla

Testattaessa ohjelmiston toimivuutta Oulun Yliopiston 3d-tulostetulla fyysisellä robotilla ohjelmistoa ajettiin samaan tapaan kuin simuloitulla robotilla Ubuntu 20.04 -käyttöjärjestelmällä varustetun virtuaalikoneen sisällä. Ohjelmisto lähetti käskyjä servoille U2D2 -ohjaimen kautta, joka oli yhdistetty virtuaalikoneeseen USB-väylän kautta. Fyysisen robotin pään hallintaan liittyvä palvelin ja pään servojen kontrolleri käynnistettiin virtuaalikoneessa simuloitun robotin sijaan. Siirtyminen simuloitulla robotilla testaamisesta fyysisellä robotilla testaamiseen ei vaatinut muutoksia itse ohjelmistokokonaisuuden koodiin muuten kuin siten, että servoille lähetettäviä arvoja piti muokata ottamaan huomioon robottipään servojen asennot, jotka eivät olleet perusasennossa 0.0 kuten simuloitulla robotilla. Fyysisellä robotilla testattiin pään kääntymistä sivusuunnassa, jolloin servolle lähetettäviin arvoihin piti lisätä 0.6 jotta perusasennossa robotin pää katsoo suoraan. Muiden projektissa käytettyjen servojen osalta testattiin se, että järjestelmä tunnistaa servot ja että viestien lähettäminen onnistuu pään kallistusta sääteleviin kolmeen XL430-servoon. Servojen hallinta tehtiin U2D2-ohjaimen kautta, joka oli yhdistetty Power Hub -lisälevyyn. Power Hub -levy toimitti servoille virtaa, joka saatiin säädettyästä, tasavirtaa tuottavasta virtalähteestä. Fyysinen robottipää käyttää kahta erilaista servomallia, joilla oli kummallakin eri vaatimus virran jännitteelle. XL430-servot toimivat jännitteillä 6,5-12 voltia, kun



Kuva 14. Normalisoitu konfuusiomatriisi ilmeitä luokittelevasta Keras-mallista omalla testidatalla.

taas XL320-servot vaativat jännitteen välillä 6-8,4 voltia. Power Hub -levyyn pystyi kytkemään kaksi servoa 3-pinnisellä liitännällä, kun taas robottipäässä oli käytössä seitsemän servoa, joten servot kytkettiin kiinni toisiinsa sarjakytkennällä. Osoittautui hyväksi periaatteeksi kytkeä ensin suuremmalla jännitteellä toimivat XL430-servot kiinni Power Hubiin ja sitten pienemmällä jännitteellä toimivat XL320-servot sarjaan XL430-servoihin. Tutkiessa servoille menevää jännitettä Dynamixel Wizard 2.0 -ohjelmalla havaittiin, että kytkiessä servoja sarjakytkennällä käyttöjännite laskee sarjassa myöhemmin vuorossa olevilla servoilla. Testatessa robottia ensimmäinen U2D2 Power Hubin ulostulo oli kytketty sarjaan, jossa oli ensin kaksi XL430-servoa ja sitten kolme XL320-servoa. Toinen Power Hubin ulostulo oli kytketty sarjaan, jossa oli kaksi XL430-servoa. Virtalähde oli säädetty tuottamaan sähköä 8,3 voltin jännitteellä. Jännite valittiin siten, että se on sallittujen rajojen sisällä kummallekin servotyypille. Koska XL320-servot olivat sarjakytkennän loppupäässä, niille tuleva jännite oli käytännössä vähän pienempi, noin 8 voltia. Robotin web-kamera oli mallia Microsoft Lifecam HD 3000, joka oli asennettu robotin pään silmään. Tämä kamera yhdistettiin USB-liitännällä virtuaalikoneeseen. Ohjelmistokokonaisuutta testattiin neutraalilla, vihaisella ja iloisella ilmeellä. Ohjelmisto toimi kuten haluttiin; vihainen ja iloinen ilme saivat robotin kääntämään päätään ja palaamaan sen jälkeen takaisin perusasentoon. Havaittuaan neutraalin ilmeen järjestelmä ei lähettänyt käskyä päättää liikuttaville servoille.



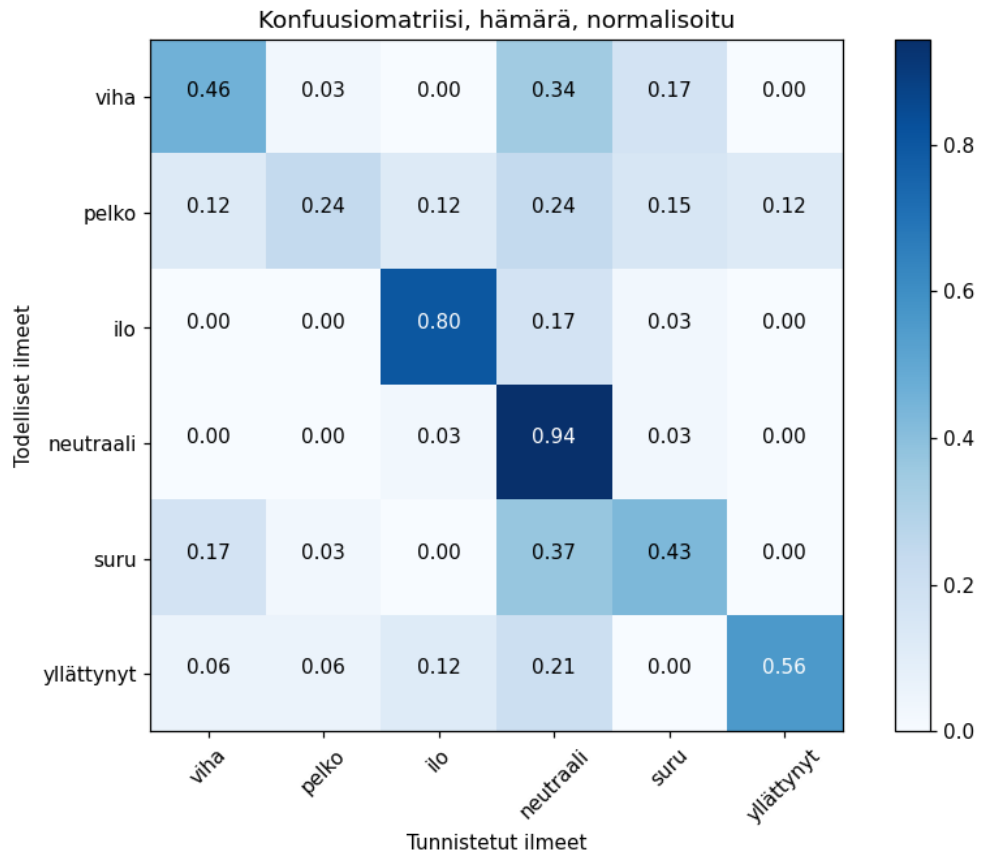
Kuva 15. Normalisoitu konfuusiomatriisi ilmeitä luokittelevasta Keras-mallista omalla testidatalla, valoisassa ympäristössä.

6.3. Tulosten analysointi

Kokonaisuudessaan ohjelmisto toimi kuten se oli ohjelmoitu toimimaan. Kolmen saman peräkkäisen tunnistetun ilmeen johdosta robotti käänsi päätänsä ja palasi takaisin aloitustilaansa, mikäli kyseinen ilme oli ohjelmoitu aktivoimaan liike. Tätä tapahtumaa seurattiin action clientin lähettämistä viesteistä terminaalista sekä visuaalisesti RViz-simulaattorin avulla. Ohjelmisto todettiin toimivaksi myös fyysisellä robotilla.

Kasvojen tunnistusohjelma toimi hyvin. Taulukon 1 mukaisesti YuNetin käyttämä aika 1280x853 resoluution kuvassa kasvojen tunnistamiseen oli 84 millisekuntia, mikä on huomattavasti pienempi kuin Haar Cascaden käyttämä 301 millisekuntia. Samassa testissä YuNetin todettiin nopeamman suoritusajan lisäksi antavan vähemmän virheellisiä kasvojen tunnistuksia Haar Cascadeen verrattuna. Ajettaessa YuNetiä ROS-ympäristössä web-kameran ottamilla kuvilla ja muiden nodejen kanssa yhtä aikaa kasvojen tunnistus toimi nopeasti ja luotettavasti, joten valittu lähestymistapa kasvojen tunnistuksen toteutukseen oli varsin onnistunut.

Ilmeiden tunnistusohjelma toimi käytännössä kohtalaisesti ROS-ympäristössä. Ilmeiden tunnistusyksikkö vaati kolmen saman peräkkäisen ilmeen tunnistamiseen keskimäärin 1300,1 millisekuntia kolmella säikeellä toteutettuna. Itse mallia oli pakattu vain karsimalla, mutta esimerkiksi käyttämällä TensorFlow Lite-kirjastoa,



Kuva 16. Normalisoitu konfuusiomatriisi ilmeitä luokittelevasta Keras-mallista omalla testidatalla, hämärässä ympäristössä.

joka on suunnattu enemmän mobiili ja sulautetuille järjestelmille kuin TensorFlow pääkirjasto, ja muita optimointitekniikoita, yksikön käyttämää aikaa laskentaan olisi voitu vielä tästä pienentää. Kokonaisuudessaan tässä järjestelmässä 1300,1 millisekuntia on siedettävä tulos jättäen kuitenkin tilaa järjestelmän ja mallin jatkokehitykselle.

Koulutetun konenäkömallin tarkkuus ilmeen tunnistuksessa oli yleisellä tasolla tyydyttävä, mutta tiettyjen ilmeiden osalta tarkka. Korkein tarkkuus oli iloisen ilmeen tunnistamisessa, missä saavutettiin oikea tulos 83 %:ssa tapauksista valmiin testidatan kanssa ja 86 %:ssa tapauksista omalla testidatalla. Malli osasi tunnistaa myös neutraalin ilmeen, surun ja vihan suhteellisen luotettavasti sekä valmiissa testidatassa, että käytännön testeissä.

Mallin tarkkuus kirkkaassa ja hämärässä valaistuksessa ilmeittäin on esitetty konfuusiomatriiseissa kuvissa 15 ja 16. Tulokset ovat lähellä toisiaan, mutta valoisassa ympäristössä konenäkömalli suoriutuu paremmin kaikkien muiden ilmeiden kuin neutraalin ilmeen osalta. Hämärässä valaistuksessa konenäkömalli tunnisti neutraalin ilmeen 94 %:n tarkkuudella oikein silloin, kun ilme oli myös todellisuudessa neutraali. Malli kuitenkin tunnisti hämärässä valaistuksessa huomattavan usein myös muita ilmeitä virheellisesti neutraaliksi, mikä selittää tätä tulosta. Kokonaisuutena siis kirkas valaistus tuotti paremman tarkkuuden. Kuitenkin ainakin yhden testaajan osalta kirkas valaistus tuotti huonomman tuloksen kuin hämärä valaistus. Tämä johtuu

todennäköisesti siitä, että tuon testin aikana kirkas valaistus tuli vahvasti vasemmalta sivulta. Myöhemmin web-kameran kuvaa tarkastellessa kameralla oli vaikeuksia valkotasapainon säätämisen kanssa, sillä osa kasvoista oli kirkkaasti valaistu suoralla valonsäteellä, kun taas osa kasvoista olivat valonsäteen varjossa.

Yhteenvetona järjestelmän testauksesta voidaan todeta, että ohjelmisto toimi kokonaisuudessaan hyvin. Yksikköjen viemää aikaa olisi voitu vielä hienosäätää kokonaisuuden rakennetta muuttamalla ja mallia optimoimalla, mutta kyseinen toteutus oli silti hyvin toimiva projektintekijöiden kotitietokoneilla. Huomioon otettavaa on kuitenkin myös RViz-simulaattorin käyttö, joka myös söi laskentatehoa prosessorilta testauksia tehdessä. Valoisuudella havaittiin olevan pientä vaikutusta mallin tarkkuuteen. Mallin koulutuksessa käytettiin maltillisesti tietojen lisäämistä (eng. data augmentation) koulutuskuviin, joka voi selittää mallin soveltuvuutta myös erilaisille valaistusolosuhteille kohtalaisissa määrin. Käytännön testeissä saavutettiin 59 %:n tarkkuus. On huomioitava, että datan kerääminen suoritettiin näyttelemällä pyydettyä ilmettä ja suppea 6 kappaleen koehenkilöjoukko koostui ei-näyttelijöistä, jolloin mahdollisesti ilme on korostettu tai hillitty verrattuna todelliseen reaalioloissa esiintyvään ilmeeseen ja malli toimi tarkemmin tilanteissa, joissa koehenkilö korosti näyteltävää ilmettä.

7. POHDINTA

Merkittävä askel tekoälyn kanssa kommunikoidessa olisi ottaa huomioon keskustelukumppanin kasvonilmeillä toteuttama sanaton viestintä, ja sopeuttaa vastausta sen perusteella. Jotta tekoäly osaisi tulkita ihmisen kommunikointia tarkasti, sen tulee osata ottaa huomioon ja tulkita ilmeiden ja kehonkielen vaikutusta puheeseen.

Konetunteita ja tunnelaskentaa kehitetään eri puolilla maailmaa, ja yksi tällaisista tutkijayhteisöistä on Oulun yliopiston konenäköryhmä, joka kehittää tekniikkaa, joka pystyy näkemisen lisäksi ymmärtämään myös ihmisen toimintaa. Älykamera pystyy ilmoittamaan tunnistuksen lisäksi ihmisen tunnetilan. Tutkimuksissa on pystytty tunnistamaan näyttelijöiden ilmeitä videokuvasta ja kuuden perustunnetilan, kuten viha, inho, suru, onni, pelko ja hämmästyminen tunnistaminen onnistuukin varsin hyvin laboratorioissa. Yleisessä käytössä olevia kasvojen tunnistusmenetelmiä käytetään pääasiassa turvallisuustarkoituksissa ja varmennuskeinona sisäänkirjautumisessa esimerkiksi älypuhelimelle ja tietokoneelle. Kasvojen avulla tunnistettaessa on kuitenkin syytä huomioida mahdollisuus väärinkäyttöön, eli voidaan käyttää esimerkiksi kuvaa toisen ihmisen kasvoista tunnistuksessa ja täten saadaan pääsy toisen henkilön tietoihin. Kiinnostus kuitenkin kasvaa myös eri aloille, ja yrityksistä mm. Sony ja Omron ovat jo kehittäneet kaupallisia sovelluksia. Myös uusia malleja kasvojen tunnistamiseen kehitetään, näistä uusimpia on OpenCV:n YuNet, jossa on parannettu havaitsemiskykyä, nopeutta ja tarkkuutta. Kehitys tekoälyn alalla on nopeaa, joten tekoälyä hyödyntävän ohjelmiston arkkitehtuuri kannattaa suunnitella modulaariseksi ja helposti muokattavaksi. Tämä helpottaa kehittyneempien tekoälymallien ja -kirjastojen päivittämistä ohjelmistoon kun niitä tulee saataville. Tässä projektissa tätä periaatetta hyödynnettiin jakamalla tunnistuksen eri osia omiin suoritusyksiköihinsä.

Tunteiden tunnistaminen nykyisten järjestelmien avulla edellyttää, että kuvaolosuhteet ovat optimaaliset. Todellisessa tilanteessa kasvot eivät aina ole suoraan kasvotusten robotin kanssa, eikä katsekontakti toteudu samalla tavalla kuin kameran edessä. Nähdäkseen ihmisen robotin pään tulee osoittaa oikeaan suuntaan olla oikealla korkeudella. Pitkän tai lyhyen käyttäjän kasvot saattavat jäädä kameran katvealueeseen. Ihminen saattaa myös istua tai käyttää pyörätuolia. Myös välimatka kamerasta vaikuttaa tunnistamiseen, eli kohde pitää olla riittävän lähellä. Muita tilanteeseen, ja siten tulkintaan vaikuttavia tekijöitä ovat muun muassa reaali maailmassa muuttuvien olosuhteiden vuoksi valaistuksen vaihtelut ja kasvonilmeiden selkeys. Valaistuksen vaihtelut ja liian tummat tai vastaavasti vaaleat värisävyt voivat vaikeuttaa tunnistusta. Vaikka tunnetilojen tunnistamisprosentti useammassa tutkimuksessa on hyvä, tunteiden tunnistaminen on haastavaa, koska kaikkien ihmisten tunteiden ilmaisu ei aina ole yhtä voimakasta kasvonilmeillä. Ihmisten ilmeet voivat olla myös erilaisia, eivätkä kaikki ilmeet välttämättä eroa kovin paljon neutraalista ilmeestä. Tällöin on mahdollista sekoittaa tunnetiloja toisiinsa.

Ihmisen tunnetilat ovat myös diskreettien perustunteiden lisäksi yhdistelmiä erilaisista tuntemuksista. Tulevaisuuden sosiaaliset robotit voivat ottaa huomioon kasvojen ilmeiden tuottamaa sanatonta viestintää eri tunteiden yhdistelmänä ollessaan ihmisen kanssa vuorovaikutuksessa. Samoin tekoälyn luomat vastaukset ihmiselle voidaan GPT:n aikakaudella tuottaa luovasti tilanteen vaatimalla tyyllillä ja sisällöllä.

8. PROJEKTIN KUVAUS

Projektin työskentely jakaantui eri työvaiheisiin. Alussa sovittiin teoriaosuuden jaosta, ja kukin kolmesta tutustui oman aihealueensa lähdekirjallisuuteen ja kirjoittamiseen. Yhteyttä pidettiin enimmäkseen Discordin välityksellä. Konenäkösovelluksen tekemisessä ja koneoppimisen toteuttamisessa työskentely, jossa kehitettiin tarkoitukseen sopivat mallit ja kirjoitettiin toteutusta, tapahtui välillä itsenäisesti, välillä yhdessä, ja myös Githubin avulla jakamalla. Suurin osa työajasta kului kurssin alussa lähteiden etsimiseen ja teoriakappaleiden kirjoittamiseen, ja sen jälkeen käytännön toteutukseen ja kirjoittamiseen. Konenäkötoteutuksen tekeminen oli haastavaa ja aikaa vievää. Taulukossa 5 on kuvattu tuntijakauma tekijöiden kesken projektin eri vaiheissa.

Taulukko 5. Projektintekijöiden tuntijakaumat

Tekijä	Tutkielman kirjoitus	Ohjelmointi	Testaus	Yhteensä
Petri Jaakkola	80	150	30	260
Mikael Rinkinen	76	81	36	193
Janne Vihriälä	70	150	30	250
Kaikki tekijät	226	381	96	703

9. YHTEENVETO

Tekoälyn kehityksen myötä konenäkösovelluksien käyttö on laajentunut eri aloille helpottamaan ihmisen toimintaa ja automatisoimaan toimintoja. Konenäön käyttö edellyttää koneoppimista, jossa ihmisen oppimismenetelmää muistuttavia neuroverkkoja hyödyntämällä koneelle opetetaan tiettyä dataa, kunnes päästään haluttuun johtopäätökseen.

Projektityössä toteutettiin InMoov-robotin päähän kehitetty konenäkösovellus, jossa kone opetettiin FER-13 tietokannasta vapaasti saatavan kuvamateriaalin avulla tunnistamaan kuvista ihmisen kasvojen perusilmeitä, jotka kuvaavat ihmisen perustunnetiloja.

Lopputuloksena voidaan todeta, että robotti kykenee tavallisen kuluttajan käytössä olevan laitteiston avulla tunnistamaan ihmisen perustunnetiloja varsin hyvin. Tunnistamisen lisäksi robotti ohjelmoitiin kääntämään päätä eri suuntiin eri ilmeiden perusteella. Esimerkiksi jos käyttäjällä on vihainen ilme kasvoillaan, robotti kääntää päänsä pois päin ja sitten takaisin vähän ajan päästä.

Jatkokehityksenä projektiin voisi yhdistää muita samalla kurssilla tehtyjä töitä. Esimerkiksi robotti puhuu nopeampaa, kun havaitsee vihaisen näköiset kasvot tai matkii käyttäjän kasvoja avaamalla suun auki, kun käyttäjä on yllätynyt. Tähän työhön voi aika helposti liittää muita töitä, ja siten luoda luontevampaa ihmisen ja robotin välistä kommunikaatiota. Muutoksia tarvitsisi tehdä kahteen nodeen. Face detection node avaa oman kahvan käyttämään web-kameraa, kun taas useat muut työt kuuntelevat raakaa web-kamera nodea, joka julkaisee kuvia tietylle aihealueelle tietoväylään. Tämä pitää vaihtaa, jos muut yhdistettävät projektit käyttävät myös web-kameraa. action client nodeen pitäisi vain luoda uusia palveluita, tai julkaisijoita sitä mukaa, mitä siihen yhdistää. Nykyinen tapa estää järjestelmää tekemästä turhaa työtä action clientin lähettämällä käynnistys- ja pysäytyskäskyillä vähentää tarpeettomia kasvojentunnistuksia merkittävästi, mutta ei kokonaan. Toisenlaisella viestirakenteella voisi optimoida järjestelmää pidemmälle. Esimerkiksi face detection node voisi aloittaa kasvojen tunnistuksen vasta saatuaan pyynnön toimittaa uusi kuva.

Projektin tapa yhdistää ennustettuja perustunteita ennalta määrättyihin toimintoihin edustaa vanhankantaista suhtautumista tekoälyyn. Uusi, GPT-mallien aikakauden versio projektista voisi ennustaa yhden päätunteen sijasta jokaisen tunnetilan vahvuutta ilmeestä. Sitten robotin liikkeen voisi tuottaa tekoälyn avulla niin, että liikkeessä otetaan huomioon eri tunnetiloja ja niiden vahvuuksia ja servojen liikkeet luodaan ennalta määrittelemättömällä tavalla tunnetilojen yhdistelmän perusteella. Tässä tapauksessa laitteen koodin pitäisi varmistaa ennen käskyjen lähettämistä servoille, että tuotettu liikerata on sallituissa rajoissa.

Projektityön tekeminen opetti käytännössä tekoälyn ja konenäön toimintaan liittyviä asioita. Lisäksi projekti laajensi ymmärrystä siitä, kuinka sosiaalisten robottien toiminta on useiden tekijöiden summa, ja toiminnallisuuden takaamiseksi huomioon otettavia asioita on paljon. Aiheena tunteiden tunnistaminen oli mielenkiintoinen. Haasteellisinta ja paljon aikaa vievää oli se, että mallin luominen ja opetusasetukset saatiin toimimaan konenäön toteutuksessa. Myös ROS aiheutti pientä päänvaivaa, koska se on kehittyvä projekti ja siitä puuttuu kontrollointia helpottavaa toiminnallisuutta. Nämä ongelmat pystyttiin ratkaisemaan, mutta loppuohjelmasta ei tullut niin tehokasta kuin se teoriassa voisi olla.

10. VIITTEET

- [1] D. Li and Y. Du, *Artificial intelligence with uncertainty*. CRC press, 2017. s. 22. DOI: doi.org/10.1201/9781315366951. Luettu 18.1.2022.
- [2] J.-J. Zhu, J. Jiang, M. Yang, and Z. J. Ren, “Chatgpt and environmental research,” *Environmental Science & Technology*, 2023. <https://doi.org/10.1021/acs.est.3c01818>.
- [3] M. Grincalaitis, “Chatgpt vs gtp-3 by merunas.” <https://merunasgrincalaitis.medium.com/chatgpt-vs-gpt-3-by-merun-4c282c83d50a>, 2022. Luettu 6.6.2023.
- [4] A. Hughes, “Ghatpot: Everything you need to know about openai’s gpt4-tool,” *BBC, Science Focus Magazine*, 2023. <https://www.sciencefocus.com/future-technology/gpt-3/>.
- [5] T. Mäenpää, M. Niskanen, H. Pylkkö, S. Ropponen, and O. Silven, “Konenäön hyödyntämismahdollisuudet teiden ylläpidossa ja hoidossa,” *Tiehallinnon selvityksiä*, 2008. ss. 11-13.
- [6] A. Hornberg, *Handbook of machine and computer vision: the guide for developers and users*. John Wiley & Sons, 2017. ss. 170-171.
- [7] C. Steger, M. Ulrich, and C. Wiedemann, *Machine vision algorithms and applications*. John Wiley & Sons, 2018. Luettu 20.1.2022.
- [8] M. Pietikäinen and O. Silven, *Tekoälyn haasteet, koeoppimisesta ja konenäöstä tunnetekoälyyn, Konenäkö*. Oulun yliopisto, Sähkötekniikan osasto, Oulu: Konenäön ja signaalianalyysin keskus, 2021. URL:<http://jultika.oulu.fi/files/isbn9789526232027.pdf> Luettu 7.2.2022.
- [9] L. Van Aerschot, T. Turja, and T. Särkikoski, “Roboteista tehokkuutta ja helpotusta hoitotyöhön? työntekijät empivät, mutta teknologia ei pelota,” *Yhteiskuntapolitiikka*, 2017. ss. 630–638.
- [10] T. Turja and M. Niemelä, “Robotin ja hoitajan yhteistyöllä tehokkuutta ja mielekkyyttä hoitotyöhön,” *Sosiaalipsykologi*, vol. 2018, no. 2, 2018. ss. 12-17.
- [11] J. Lehtinen, “Robottiikka vaatii monitieteellisyyttä,” *Tieteessä tapahtuu*, vol. 33, no. 5, 2015. URL:<https://journal.fi/tt/article/view/52748>. Luettu 21.1.2022.
- [12] T. Mirtti, J. Näpänkangas, *et al.*, “Tekoäly patologian kudosleikkeiden tulkinnassa,” *Duodecim*, 2020. URL:<https://www.duodecimlehti.fi/duo15745>. Luettu 21.1.2020.
- [13] Dustin.fi, “tietokonenäkö-tekoälyn kulmakivi.” URL:<https://www.dustin.fi/solutions/tietopankki/archive/tietokonenaekoe-tekoaelyn-kulmakivi>, 2018. Luettu 20.1.2022.

- [14] E. R. Davies, *Computer and machine vision: theory, algorithms, practicalities*. Academic Press, 2012. s. 13.
- [15] S. S. Biswas, "Role of chat gpt in public health," *Annals of Biomedical Engineering*, pp. 1–2, 2023.
- [16] M. Mori, "Bukimi no tani [the uncanny valley]," *Energy*, vol. 7, 1970. ss. 33–35.
- [17] B. G. Batchelor, "Coming to terms with machine vision and computer vision—they are not the same," *Advanced imaging*, vol. 14, no. 1, 1999. ss. 22–26, ss. 583–585.
- [18] S. PARKER, "Ihmiskeho suuri ensyklopedia," *Helsinki: Readme. fi*, 2015. ss. 120–121.
- [19] J. Häkkinen and G. Nyman, "Ihmisen kolmiulotteisen stereohavaitsemisen mekanismit," *Psykologia*, vol. 33, pp. 293–310, 01 1998.
- [20] R. Gonzalez and R. Woods, *Digital Image Processing*. Pearson/Prentice Hall, 2008. Luettu 5.2.2022.
- [21] L. R. Young and D. Sheena, "Survey of eye movement recording methods," *Behavior research methods & instrumentation*, vol. 7, no. 5. Springer, 1975. ss. 397–429.
- [22] A. T. Duchowski, *Eye tracking methodology: Theory and practice*. Springer, 2017. ss. 40–42.
- [23] L. Shapiro and G. Stockman, "'computer vision', str. 137, 150," *Prentice Hall*, 2001. s. 1, s. 279.
- [24] T. Huang, "Computer Vision: Evolution And Promise," *1996 CERN School of Computing*, 1996.
- [25] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [26] A. Owen-Hill, "Robot vision vs computer vision: What's the difference?," *Robotics Tomorrow*, 2016. URL:<https://blog.robotiq.com/robot-vision-vs-computer-vision-whats-the-difference>. Luettu 28.1.2022.
- [27] A. Kaehler and G. Bradski, *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. "O'Reilly Media, Inc.", 2016. Luettu 10.2.2022.
- [28] A. Jaruga-Rozdolska, "Artificial intelligence as part of future practices in the architect's work: Midjourney generative tool as part of a process of creating an architectural form," *Architectus*, no. 3 (71), 2022.
- [29] "Discord interface." URL:<https://docs.midjourney.com/docs/midjourney-discord>, 2023. Luettu 4.6.2023.

- [30] “Midjourney describe.” URL:<https://docs.midjourney.com/docs/describe>, 2023. Luettu 10.6.2023.
- [31] Y. Chen, J. Wang, S. Chen, Z. Shi, and J. Cai, “Facial motion prior networks for facial expression recognition,” in *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019. ss. 1-4.
- [32] Y. He and S. Chen, “Person-independent facial expression recognition based on improved local binary pattern and higher-order singular value decomposition,” *IEEE Access*, vol. 8, 2020. ss. 190184–190193.
- [33] S. Shimo, “Risks of bias in ai-based emotional analysis technology from diversity perspectives,” in *2020 IEEE International Symposium on Technology and Society (ISTAS)*, 2020. ss. 66–68.
- [34] D. Y. Choi, D. H. Kim, and B. C. Song, “Recognizing fine facial micro-expressions using two-dimensional landmark feature,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018. ss. 1962-1966.
- [35] C. Huang, H. Ai, Y. Li, and S. Lao, “High-performance rotation invariant multiview face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, 2007. ss. 671-686.
- [36] N. H. Jabber and I. A. Hashim, “Robust eye features extraction based on eye angles for efficient gaze classification system,” in *2018 Third Scientific Conference of Electrical Engineering (SCEE)*, 2018. ss. 13-18.
- [37] S. L. Happy and A. Routray, “Automatic facial expression recognition using features of salient facial patches,” *IEEE Transactions on Affective Computing*, vol. 6, no. 1, 2015. ss. 1-12.
- [38] J. Ren, X. Jiang, and J. Yuan, “Face and facial expressions recognition and analysis,” in *Magnenat-Thalman N., Yuan J., Thalman D., You BJ. (eds) Context Aware Human-Robot and Human-Agent Interaction. Human-Computer Interaction Series. Springer, Cham*, 2016. ss. 3-29.
- [39] M. K. Hasan, M. S. Ahsan, Abdullah-Al-Mamun, S. H. S. Newaz, and G. M. Lee, “Human face detection techniques: A comprehensive review and future research directions,” *Electronics*, vol. 10, no. 19, 2021. DOI = 10.3390/electronics10192354.
- [40] H. Hatem, Z. Beiji, and R. Majeed, “A survey of feature base methods for human face detection,” *International Journal of Control and Automation*, vol. 8, 05 2015. ss. 61-78.
- [41] S. Rajan, P. C, S. Devaraj, and N. Madian, “A novel deep learning model for facial expression recognition based on maximum boosted cnn and lstm,” *IET Image Processing*, vol. 14, 02 2020. DOI = 10.1049/iet-ipr.2019.1188.
- [42] Y. Huang, F. Chen, S. Lv, and X. Wang, “Facial expression recognition: A survey,” *Symmetry*, vol. 11, no. 10, 2019. DOI = 10.3390/sym11101189.

- [43] M. Iqtait, F. S. Mohamad, and M. Mamat, "Feature extraction for face recognition via active shape model (ASM) and active appearance model (AAM)," *IOP Conference Series: Materials Science and Engineering*, vol. 332, mar 2018. s. 012032.
- [44] C. E. IBM, "Convolutional neural networks." URL:<https://www.ibm.com/cloud/learn/convolutional-neural-networks>, 2020.
- [45] H. Tuominen and P. Neittaanmäki, *Tekoälyn perusteita ja sovelluksia, Keinotekoiset neuroverkot*. Informaatioteknologian tiedekunta. Jyväskylän yliopisto. Jyväskylä, 2019. URL:<http://urn.fi/URN:ISBN:978-951-39-7796-2>, ss. 6-7, 22-54.
- [46] J. D. Kelleher, *Deep learning*. MIT press, 2019. URL:<https://mitpress.mit.edu/9780262537551/deep-learning/>, ss. 14-17, 78-81.
- [47] "About quori." URL:<http://www.quori.org/about#about-quori>, 2023. Luettu 19.6.2023.
- [48] "Nao: Personal robot teaching assistant." URL:<https://us.softbankrobotics.com/nao>, 2023. Luettu 19.6.2023.
- [49] "Paro therapeutic robot." URL:<http://www.parorobots.com/>, 2023. Luettu 19.6.2023.
- [50] C. M. Martinez-Martin E., Escalona F., "Socially assistive robots for older adults and people with autism: An overview," *Volume 9, Issue 2, February 2020, Article number 367*, 2020.
- [51] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley [from the field]," *IEEE Robotics Automation Magazine*, vol. 19, no. 2, 2012. ss. 98-100.
- [52] D. Mazzei, N. Lazzeri, D. Hanson, and D. De Rossi, "Hefes: An hybrid engine for facial expressions synthesis to control human-like androids and avatars," in *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2012. ss. 195-200.
- [53] A. Habib, S. K. Das, I.-C. Bogdan, D. Hanson, and D. O. Popa, "Learning human-like facial expressions for android phillip k. dick," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, 2014. ss. 1159-1165.
- [54] S. P. Reddy Karri and B. Santhosh Kumar, "Deep learning techniques for implementation of chatbots," in *2020 International Conference on Computer Communication and Informatics (ICCCI)*, 2020. ss. 1-50.
- [55] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, "A brief overview of chatgpt: The history, status quo and potential future development," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1122-1136, 2023.
- [56] G. Langevin, "Inmoov-robot." URL:<https://www.youtube.com/watch?v=H4Z09edx52E>, 2015. Luettu 15.4.2022.

- [57] “Opencv.” URL:<https://opencv.org/about/>, 2022. Luettu 25.4.2022.
- [58] G. Bradski, “The OpenCV Library,” *Dr. Dobbs’s Journal of Software Tools*, 2000.
- [59] T. T. T. M. T), “Yunet — ultra-high-performance face detection in opencv — a good solution for real-time poc, demo, face applications..” URL:<https://medium.com/@silkworm/yunet-ultra-high-performance-face-detection-in-opencv-a-good-solution-for-real-time-poc-b01063e251d5>, 2021. Luettu: 5.5.2023.
- [60] W. Wu, H. Peng, and S. Yu, “Yunet: A tiny millisecond-level face detector,” *Machine Intelligence Research*, 04 2023.
- [61] P. Nelson, “Opencv face detection:cascade classifier vs.yunet.” URL:<https://opencv.org/opencv-face-detection-cascade-classifier-vs-yunet/>, 2022. Luettu 5.5.2023.
- [62] G. Zaccane, *Getting started with TensorFlow*. Packt Publishing Birmingham, 2016. ss.11-12.
- [63] “Api documentation.” URL:https://www.tensorflow.org/api_docs/, 2021. Luettu 10.5.2023.
- [64] “Keras.” URL:<https://keras.io/about/>, 2015. Luettu 20.9.2022.
- [65] “Ros2 vs. ros1— key differences and which one is better?.” URL:<https://medium.com/@oelmofty/ros2-how-is-it-better-than-ros1-881632e1979a>, 2022. Luettu 12.5.2023.
- [66] “Ros, about robot operating system..” URL:<https://www.ros.org/about-ros/>, 2021. Luettu 25.4.2022.
- [67] “What is ros?.” URL:<http://www.clearpathrobotics.com/assets/guides/kinetic/ros/Intro%20to%20the%20Robot%20Operating%20System.html#what-is-ros>. Luettu 30.4.2022.
- [68] “Ros 2 middleware interface.” URL:https://design.ros2.org/articles/ros_middleware_interface.html, 2017. Luettu 30.4.2022.
- [69] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, *et al.*, “Challenges in representation learning: A report on three machine learning contests,” in *International conference on neural information processing*, Springer, 2013. ss. 3–4.
- [70] C. Pramerdorfer and M. Kampel, “Facial expression recognition using convolutional neural networks: state of the art,” *arXiv preprint arXiv:1612.02903*, 2016. s. 2.

11. LIITE

Liite 1 Lähdekoodi: <https://github.com/ouspg/SOP-Robot>