TAMPEREEN TEKNILLINEN YLIOPISTO

Jussi Taipalmaa

IÄN JA SUKUPUOLEN ARVIOINTI VIDEOKUVASTA

Diplomityö

Tarkastajat:
yliopistonlehtori Heikki Huttunen ja
professori Hannu-Matti Järvinen
Tarkastajat ja aihe hyväksytty
Tieto- ja sähkötekniikan tiedekuntaneuvoston kokouksessa 7. lokakuuta 2015

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietotekniikan koulutusohjelma
TAIPALMAA, JUSSI: Iän ja sukupuolen arviointi videokuvasta
Diplomityö, 43 sivua
Toukokuu 2016
Pääaine: Pervasive Systems
Tarkastajat: yliopistonlehtori Heikki Huttunen ja professori Hannu-Matti Järvinen
Avainsanat: Automaattinen kasvoanalyysi, Hahmontunnistus, Iän arviointi, Koneoppiminen, Konvoluutiohermoverkot, Sukupuolen arviointi

Tässä diplomityössä toteutetaan automaattinen järjestelmä iän ja sukupuolen reaaliaikaiseen arviointiin videokuvasta. Tavoitteena on luoda järjestelmä, joka minimoi keskimääräisen virheen iän ja sukupuolen arvioinnissa, mutta samalla toteuttaa reaaliaikavaatimuksen.

Teoriaosiossa keskitytään AdaBoost-algoritmiin, Haar-piirteisiin ja kaskadiluokittimen muodostamiseen kasvojen paikannuksen osalta, ja konvoluutiohermoverkkoihin ja SVR-luokittimeen *(Support Vector Regression)* iän ja sukupuolen arvioinnin osalta. Toteutusosiossa esitellään vaatimukset täyttävän järjestelmän arkkitehtuuri ja tärkeimmät ominaisuudet.

Lopuksi tarkastellaan, kuinka hyvin järjestelmä toteuttaa annetut vaatimukset ja vertaillaan suorituskykyä muihin vastaavanlaisiin järjestelmiin. Lisäksi keskustellaan siitä, mihin suuntaan järjestelmän jatkokehityksessä tulisi lähteä.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Pervasive Computing
TAIPALMAA, JUSSI: Age and Gender Recognition from Video Stream
Master of Science Thesis, 43 pages
May 2016
Major: Pervasive Systems
Examiner: University lecturer Heikki Huttunen and Professor Hannu-Matti Järvinen
Keywords: Age Recognition, Automatic Face Analysis, Convolutional Neural Networks, Gender Recognition, Machine Learning, Pattern Recognition

This Master's thesis aims to implement an automatic real-time system for age and gender recognition from video stream. The main goal is to create a system that minimizes the mean absolute error in age and gender recognition and also fulfills the real-time requirements.

The theory section focuses on AdaBoost, Haar-features and generating a cascade of classifiers used on face detection, and convolutional neural networks and support vector regression, which are used on age and gender estimation. The implementation section describes a system architecture and main properties of a system that fulfills the requirements.

Finally, the performance of the system will be inspected and compared against other similar systems. A discussion about the extensions and development of the current system is also included into this study.

ALKUSANAT

Tämä diplomityö on tehty Tampereen teknillisen yliopiston signaalinkäsittelyn laitokselle lukuvuoden 2015–2016 aikana.

Haluan kiittää työn tarkastajaa yliopistonlehtori Heikki Huttusta mielenkiintoisen aiheen tarjoamisesta ja asiantuntevasta ohjauksesta. Lisäksi haluan kiittää myös työn toista tarkastajaa professori Hannu-Matti Järvistä asiantuntevista kommenteista.

Tampereella 20.5.2016

Jussi Taipalmaa

SISÄLTÖ

TERMIT JA NIIDEN MÄÄRITELMÄT

AdaBoost	Koneoppimisalgoritmi, joka muodostaa vahvan luokittimen joukosta heikkoja luokittimia.
Annotointi	Eli nimikointi on aineiston kuvaamista, luokittelua tai jäsentelyä systemaattisella tavalla.
Boosting	Englanninkielinen nimi, jota käytetään luokittimien yhdistelemisestä, joiden opetusjoukot muodostetaan uudelleenotannalla.
CNN	Konvoluutiohermoverkko (engl:. Convolutional Neural Network)
Dropout	Operaatio, joka tarkoittaa, että neuronin ulostuloksi asetetaan nolla tietyllä todennäköisyydellä.
Haar-piirre	Luokittelijan käyttämä kuvan suorakulmaisten alueiden pikselisummista laskettava tunnusluku.
Integraalikuva	Vakioajassa laskettava kuvan muunnos, jonka avulla Haar-piirteet voidaan laskea nopeasti.
Kaskadiluokitin	Puumainen rakenne peräkkäisiä heikkoja luokittimia.
Log-loss	Logritminen virhefunktio (engl.: Logarithmic Loss)
LRN	ReLU-operaation jälkeen suoritettava operaatio, jonka tehtävänä on helpottaa generalisaatiota. (engl.: Local Response Normalization)
MAE	Keskimääräinen virhe (engl.: Mean Absolute Error)
Max pooling	Operaatio, jolla ikkunan sisältä valitaan pikseli jonka sävyarvo on suurin.
ReLU	Konvoluutiohermoverkkojen yhteydessä käytettävä aktivaatiofunktio. (engl.: Rectified Linear Unit)
SVR	Regressiomalli, jota käytetään tässä työssä konvoluutiohermoverkkojen yhteydessä. (engl.: Support Vector Regression)
Tietovuo	Arkkitehtuurimalli, jossa väylä kuljettaa tietoa komponentilta toiselle. (engl.: Pipeline)

1. JOHDANTO

Viime vuosien kehitys tekoälyn ja koneoppimisen alueella on tehnyt mahdolliseksi sovellukset, joita ennen on pidetty koneelle erityisen vaikeina. Yksi näistä sovelluksista on henkilön iän ja sukupuolen määrittäminen valokuvan perusteella. Kyseinen sovellus on malliesimerkki tehtävästä, josta ihminen suoriutuu pääasiassa hyvin, mutta subjektiivisesti. Erityisesti selvästi eri ikäryhmiin tai eri etnisiin ryhmiin kuuluvista henkilöistä voi olla hankalaa tehdä tarkkaa ikäarviota. Kehitys koneoppimisessa sekä kuva-analyysissä kuitenkin mahdollistaa tarkkuuden viemisen uudelle tasolle, joka on tarkkuudeltaan jo lähellä ihmisen tekemää arviota ja joissakin tapauksissa jopa ihmisen arviota parempi.

Tässä diplomityössä käsitellään datalähtöiseen syväoppimiseen perustuvaa reaaliaikaista iän ja sukupuolen arviointimenetelmää. Yksinkertaistetusti voidaan määritellä, että järjestelmä on opetettu näyttämällä sille eri ikäisten ja sukupuolisten henkilöiden kuvia ja se on oppinut erottelemaan piirteet, jotka antavat viitteitä testihenkilön iästä ja sukupuolesta.

Tässä diplomityössä toteutettava järjestelmä koostuu neljästä pääkomponentista. Keskeisenä komponenttina on videokuvaa hallinnoiva komponentti. Se vastaa videokuvan käsittelystä ja tulosten esittämisestä käyttäjälle. Lisäksi kyseinen komponentti säilyttää kaikki ohjelman kannalta oleelliset tiedot tallessa muita komponentteja varten. Suoritusjärjestyksessä seuraavana komponenttina on kasvojen etsimistä ja paikannusta suorittava komponentti, joka nimensä mukaisesti suorittaa videokuvasta jatkuva-aikaista kasvojen etsimistä ja tallettaa tiedot videokuvakomponentille. Kun kasvot on havaittu, erilliset komponentit iän ja sukupuolen arvioinnille pyytävät kasvokuvaa käyttöönsä. Ne ajavat kuvan konvoluutiohermoverkon läpi ja välittävät saadut ikä- ja sukupuoliarviot takaisin videokuvakomponentille, joka esittää ne käyttäjälle.

Työn teoreettisena pohjana kasvojen etsinnälle ja paikannukselle toimii Paul Violan ja Michael Jonesin *Rapid Object Detection using a Boosted Cascade of Simple Feature* -tutkimus [20] ja iän ja sukupuolen arvioinnille Gil Levin ja Tal Hassnerin *Age and Gender Classification using Convolutional Neural Networks* -tutkimus [14]. Työn tavoitteena on tuottaa teorioiden pohjalta reaaliaikainen järjestelmä iän ja sukupuolen arviointia varten. Lisäksi tutkitaan konvoluutiohermoverkkojen opettamista, tavoitteena keskivirheen minimoiminen. Lisäarvoa tämä työ tuottaa juuri

reaaliaikaisuudella, joka perustuu arvioinnin suorittamiseen videokuvasta.

Diplomityössä perehdytään aluksi kasvojenetsinnän, sekä iän ja sukupuolen arvioinnin taustalla olevaan teoriaan (luku 2). Tämän jälkeen esitellään järjestelmä, jolla reaaliaikainen tunnistus on mahdollista, ja miten konvoluutiohermoverkot voidaan opettaa uudelleen. (luku 3). Luvussa 4 arvioidaan opettamisella aikaansaatua tarkkuutta ja järjestelmän reaaliaikaista suorituskykyä. Lopuksi luvussa 5 esitetään vielä mahdollisia suuntia järjestelmän jatkokehitykselle.

2. TEORIA

2.1 Kasvojenetsinnän teoreettinen tausta

Nykyinen taso konenäön eri osa-alueilla mahdollistaa useiden erilaisten kohteiden tunnistamisen ja paikantamisen kuvasta. Myöhemmin tässä diplomityössä kuvatussa järjestelmässä käytetään iän- ja sukupuolen arvioimiseen kuvaa kasvoista, joten sovelluksen kannalta tässä työssä keskitytään etsimään kuvasta ihmisen kasvot.

Kasvojen etsinnässä pyritään löytämään syötteenä olevassa kuvassa olevien kasvojen sijainti ja koko. On myös tärkeää pystyä tekemään ero sen välillä, onko kuvassa kasvoja vai ei. Tässä luvussa perehdytään sopivien piirteiden valintaan tunnistustehtävää varten, ja kaskadirakenteisen tunnistimen muodostamiseen.

Taustamateriaalina on hyödynnetty Tampereen teknillisen yliopiston signaalinkäsittelyn laitokselle aiemmin tehtyä kandidaatintyötä kasvojen paikannuksesta [7]. Kasvojenpaikannusjärjestelmän rakenne, johon tässä luvussa perehdytään, on Paul Violan ja Michael Jonesin kehittelemä [20].

2.1.1 Boosting-algoritmit

Luokittimien herkkyyttä reagoida yksittäisiin näytteisiin epätoivotusti, voidaan pyrkiä välttämään yhdistelemällä useampia luokittimia, joiden opetusjoukot muodostetaan uudelleenotannalla. Tätä menetelmää kutsutaan englanninkielisellä nimellä *boosting*.

Prosessin tarkoituksena on rakentaa iteratiivisesti sarja komponenttiluokittimia. Ideana on siis yhdistää luokittimien vahvuudet, jolloin saadaan yksittäistä luokitinta parempi tulos. Tavoitteeseen voidaan päästä esimerkiksi muokkaamalla opetusnäytteissä esiintyvien virheiden painokertoimia iteraatioiden välissä, jolloin eri iteraatioilla käytetään opetusnäytteitä uudelleen eri tavalla. Olennaisinta on ottaa mukaan kokonaisvirheen kannalta informatiivisimmat opetusnäytteet.

Boosting-algoritmin vaiheet ovat yksinkertaistetussa muodossa seuraavat:

- Oletetaan, että käytössä on joukko opetusnäytteitä:
  $D_{train} = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, jossa y_i on piirrevektoriin $\mathbf{x}_i$ liitetty luokka.

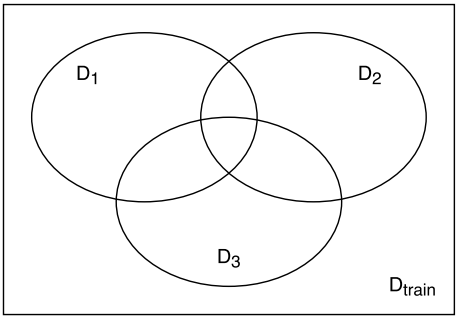
- Toistetaan $t = 1, 2, \ldots, T$

1. Valitaan opetusnäytteistä uudelleenotannalla $D_t \subset D_{train}$
2. Heikko luokitusalgoritmi muodostaa hypoteesin h_t joka minimoi luokitusvirheen joukolle D_t

- Lopuksi yhdistetään heikot hypoteesit vahvaksi luokittelusäännöksi, joka tuottaa paremman lopputuloksen kuin yksittäinen heikko luokitin. Lopputuloksena voi olla esimerkiksi heikkojen luokittimien äänestystulos.

Prosessin tarkoituksena on siis rakentaa yksi vahva luokitin yhdistelemällä heikkoja luokittimia. Heikolla luokittimella tarkoitetaan luokitinta, jonka tarkkuus on vähintään yhtä hyvä kuin arvaus eli siis kahden luokan luokitusongelmassa heikon luokittimen tarkkuus on vähintään 50 prosenttia. Heikoiksi luokittimiksi valitaan yleensä yksinkertaisia luokittimia, ja näin ollen heikot luokittimet ovat yleensä myös vaivattomasti muodostettavissa.

Uudelleen otannalla tarkoitetaan, että opetusnäytteistä valitaan jokaisella kierroksella n^* näytettä ($n^* < n$). Operaation visualisointi on esitetty kuvassa 2.1.



Kuva 2.1: Kolme uudelleenotannalla valittua opetusnäytteiden joukkoa joukosta D_{train}.

Vahva luokitusalgoritmi saadaan lopulta yhdistelemällä heikkojen luokituskomponenttien luokitustulos. Tällaisia malleja ovat esimerkiksi luokitustuloksen valinta äänestämällä tai kynnysarvon asettaminen heikkojen luokittimien ulostulojen painotetulle summalle. Komponenttiluokitin muodostetaan yhdistelemällä heikkoja luokittimia siten, että luodaan sarja luokittimia jotka ovat toinen toistaan vaativampia. Heikkojen luokittimien lopputulosta voidaan siis parantaa näytteillä, joiden valinnassa on painotettu edeltävien luokittimien tekemiä virheitä.

2.1.2 AdaBoost

Adaptive boosting eli AdaBoost on Freundin ja Schapiren esittelemä boosting-algoritmi, joka sopeutuu adaptiivisesti heikkojen luokittimien väärin luokittelemiin

hankaliin opetusnäytteisiin. Algoritmi muodostaa myös painokertoimet komponenttiluokittimille, joiden avulla niiden suhteellinen tarkkuus otetaan huomioon hypoteesien yhdistelyvaiheessa. [5]

Seuraavaksi esitellään algoritmi kahden luokan tapauksessa, koska ongelma on sama kuin tämän työn varsinaisessa käyttötarkoituksessa eli löydettiinkö kuvasta kasvot vai ei.

- Oletetaan, että käytössä on joukko opetusnäytteitä:
  $D_{train} = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, jossa $y_i \in \{-1, +1\}$ on piirrevektoriin $\mathbf{x}_i$ liitetty luokka.

- Alustetaan kaikille luokittimille sama painokerroin: $D_1(i) = \dfrac{1}{n}$, jossa n on luokittimien lukumäärä ja $i = 1, 2, \ldots, n$

- Toistetaan $t = 1, 2 \ldots, T$

  1. Muodostetaan painotetun opetusvirheen ε_j minimoiva heikko luokitin: $h_t = \arg\min_{h_j \in H} \varepsilon_j$
  2. Saadulla hypoteesilla $h_t : X \to \{-1, +1\}$ on opetusvirhe

     $$\varepsilon_t = P_{i \sim D_t}(h_t(x_i) \neq y_i) = \sum_{i=1}^{n} D_t(i)[y_i \neq h_j(x_i)] < 0,5,$$

     joka kuvaa todennäköisyyttä, että hypoteesi $h_t(x_i)$ on erisuuri kuin piirrevektoriin liitetty luokka y_i.
  3. Valitaan $\alpha_t = \dfrac{1}{2} \ln \left(\dfrac{1 - \varepsilon_t}{\varepsilon_t} \right)$
  4. Päivitetään:

     $$D_{t+1} = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} \text{ kun } h_t(x_i) = y_i \\ e^{\alpha_t} \text{ kun } h_t(x_i) \neq y_i \end{cases} = \frac{D_t(i) \exp(-\alpha_t y_t h_t(x_i))}{Z_t},$$

     jossa Z_t on normalisointivakio, jonka avulla D_t on jakauma.

- Lopullinen hypoteesi on:

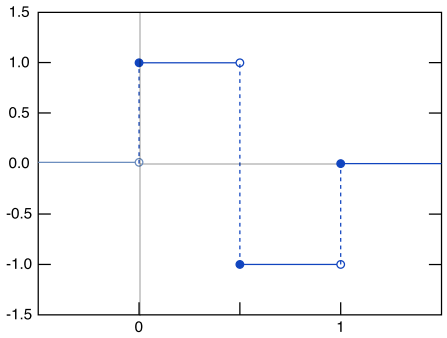
  $$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x_i) \right)$$

AdaBoost-algoritmin opetusvirhe on ylhäältä rajoitettu, ja raja lähestyy jokaisella iteraatiolla eksponentiaalisesti kohti nollaa. Kuitenkin käytännössä opetusongelman hankaluus kasvaa kierroksittain, ja heikko luokitin ei välttämättä pysty enää

tuottamaan satunnaista arvausta parempaa luokitustulosta, jolloin opetusalgoritmi on pysäytettävä. AdaBoost-algoritmin hyvinä puolina on, että sen ratkaisu toimii hyvin myös ennalta näkemättömällä aineistolla. Lisäksi AdaBoost-algoritmilla ei ole havaittu olevan taipumusta ylioppimiseen.

2.1.3 Haar-piirteet

Haar-piirteet *(Haar-like features)* ovat saaneet nimensä siitä, että ne muistuttavat Haar-funktiota (Kuva 2.2),

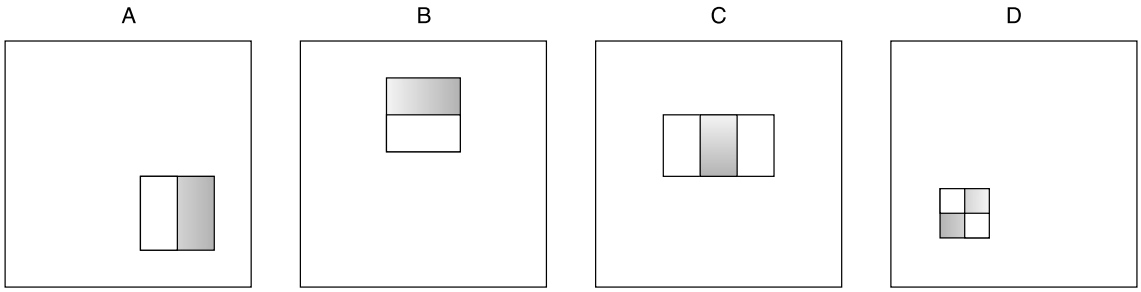


Kuva 2.2: Haar-funktio

joka määritellään seuraavalla tavalla:

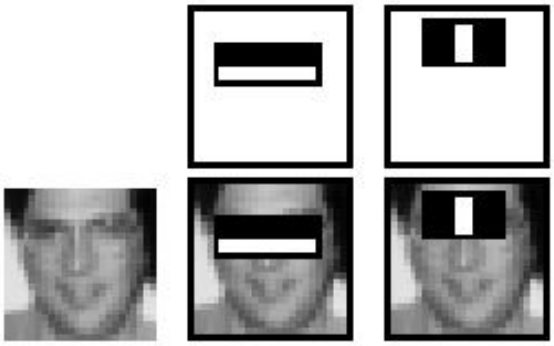
$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq t < 1, \\ 0, & \text{muulloin.} \end{cases} \tag{2.1}$$

Haar-piirteet esitetään kuten Violan ja Jonesin tutkimuksissa [20]. Haar-piirteiden laskeminen suoritetaan tarkasteluikkunassa, jota liikutetaan kuvan yllä. Tarkasteluikkunan kokoa myös kasvatetaan vaiheittain skaalauskertoimella. Ikkunan siirtäminen tapahtuu aluksi muutaman pikselin tarkkuudella ja myöhemmin skaalauskertoimella kerrottuna.



Kuva 2.3: Käytetyt Haar-piirteet

Itse piirteinä käytetään 2-ulotteisia Haar-piirteitä, jotka saadaan laskemalla erotuksia vähentämällä kuvassa olevien suorakulmaisten alueiden pikseliarvojen summia toisistaan. Piirteiden tarkoituksena on kerätä paikallista informaatiota reunamuodostelmista. Piirteitä on kolmentyppisiä. Niissä alueet ovat saman kokoisia, akseleiden suuntaisia, vierekkäisiä suorakulmion muotoisia alueita (Kuva 2.3).



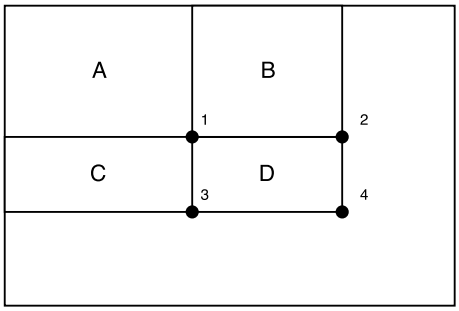
Kuva 2.4: Esimerkki Haar-piirteistä. Ylärivissä kaksi valittua piirrettä ja alarivissä piirteet on sijoitettu tyypillisen kasvokuvan päälle. Ensimmäinen piirre vertailee silmien alueen ja poskien yläosan välistä kontrastieroa. Toinen piirre tarkastelee silmien välissä sijaitsevan nenän alueen kontrastieroa silmien alueeseen. Kuva on peräisin lähteestä [20].

Kuvassa 2.4 on esitetty käytännön esimerkki Haar-piirteiden valitsemisesta. Kuvan ylärivissä kaksi erilaista Haar-piirrettä ja alemmalla rivillä piirteet on aseteltu tyypillisen kasvokuvan päälle. Ensimmäinen piirre mittaa silmien alueen ja poskien yläosan alueen välistä kontrastieroa. Piirre käyttää hyväkseen havaintoa, että silmien alue on yleensä poskien aluetta tummempi. Toinen piirre vastaavasti tarkastelee silmien alueen kontrastia verrattuna nenän alueeseen silmien välissä.

Kahden suorakulmion piirteissä alueet ovat leveys tai pystysuunnassa rinnakkain (A ja B). Kolmen suorakulmion piirteissä ulompien alueiden pikselisummista vähennetään keskimmäisen alueen pikselisumma (C). Neljän suorakulmion piirteissä diagonaalisten alueiden summat vähennetään toisistaan (D). Kuvan 2.3 esitystavalla voidaan yleisemmin ilmaista, että harmaiden suorakulmioiden pikselisummasta vähennetään valkoisten suorakulmioiden pikselisummat.

Piirteet voidaan laskea nopeasti käyttämällä esitysmuotoa, jota kutsutaan integraalikuvaksi *(Integral image)* (Kuva 2.5). Sen avulla voidaan jokainen tunnistimen tarvitseva suorakulmio-piirre laskea vakioaikaisesti. Piirteen skaalauksella tai sijainnilla ei ole vaikutusta, koska arvot saadaan indeksoimalla integraalikuvaa suorakulmion reunapisteissä. Kuvassa 2.5 suorakulmion D pikseleiden summa voidaan laskea neljän suorakulmion avulla. Pisteen 1 arvo saadaan suorakulmion A pikseleiden summasta. Piste 2 saadaan arvosta $A + B$, piste 3 saadaan arvosta $A + C$ ja piste 4 saadaan arvosta $A + B + C + D$. Alueen D pikseleiden summa saadaan siis laskettua

pisteiden summien avulla seuraavasti: $4 + 1 - (2 + 3)$.



Kuva 2.5: Pikseleiden summan laskeminen integraalikuvasta.

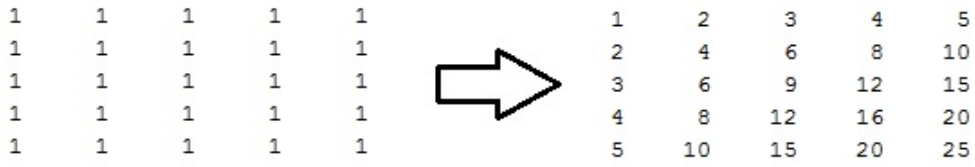
Integraalikuvan yksittäisen pikselin arvo sijainnissa x, y saadaan pisteen vasemmalla puolella ja yläpuolella olevien pikseleiden summana seuraavasti:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \tag{2.2}$$

jossa $ii(x, y)$ on integraalikuva ja $i(x, y)$ on alkuperäinen kuva. Käyttämällä seuraavaa rekursioparia:

$$\begin{aligned} s(x, y) &= s(x, y - 1) + i(x, y) \\ ii(x, y) &= ii(x - 1, y) + s(x, y), \end{aligned} \tag{2.3}$$

jossa $s(x, y)$ on kumulatiivinen rivisumma, $s(x, -1) = 0$ ja $ii(-1, y) = 0$, tarvitsee alkuperäisen kuvan pikselit käydä läpi vain yhden kerran integraalikuvan laskemiseksi. Integraalikuvan pikseliarvojen laskeminen on visualisoitu kuvassa 2.6.



Kuva 2.6: Vasemmalla on alkuperäinen kuva ja oikealla siitä muodostettu integraalikuva. Oikean kuvan pikseleiden arvot määräytyvät vasemmassa kuvassa saman pikselin ja sen vasemmalla ja yläpuolella olevien pikseleiden arvojen summasta.

2.1.4 Luokittimen käyttämien piirteiden valinta

Seuraavaksi esitellään Violan ja Jonesin kehittämä kaskadimallinen luokitin [20]. Menetelmässä käytetään AdaBoost-algoritmia poimimaan tunnistuksen kannalta merkittävimmät Haar-piirteet. Sopivien piirteiden löydyttyä muodostetaan eräänlainen puurakenne päätöksentekoon *(decision tree)*, jota kutsutaan tässä yhteydessä myös kaskadirakenteeksi. Suurin osa tarkasteluikkunoista ei sisällä kasvoja, joten resursseja säästetään siirtymällä alkupään komponenttiluokittimista eteenpäin tarpeeksi

nopeasti, kuitenkin säilyttämällä riittävän korkea herkkyys sen suhteen, ettei oikeita kasvoja hylätä liian nopeasti.

Oletetaan, että käytössä on opetusaineisto, joka koostuu sekä positiivisista ja negatiivisista esimerkeistä eli kuvista, jossa on kasvot ja kuvista, joissa ei ole kasvoja. Luokiteltavasta kuvasta muodostettavissa ali-ikkunoissa *(sub-window)* on jokaisessa yli 180 000 Haar-piirrettä, joka on paljon enemmän kuin kuvan pikselien määrä, joten ei ole mielekästä laskea kaikkia piirteitä.

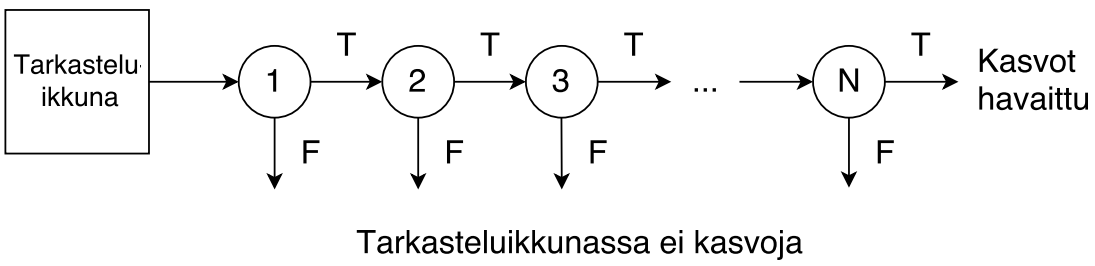
Oletuksena pidetään, että pienelläkin määrällä käytettäviä piirteitä voidaan muodostaa tehokas luokitin ja tavoitteena on löytää olennaisimmat piirteet. Oleellisimmat piirteet voidaan löytää AdaBoost-algoritmin avulla, käyttäen heikkoja luokittimia, joiden toiminta perustuu ainoastaan yhden Haar-piirteen arvoon. Tarkoituksena on löytää piirteet, jotka erottavat positiiviset ja negatiiviset esimerkit parhaiten toisistaan. Heikko komponenttiluokitin $h_j(x)$ koostuu yhdestä Haar-piirteestä f_j, kynnysarvosta θ_j ja etumerkistä p_j, joka määrää epäyhtälön suunnan:

$$h_j(x) = \begin{cases} 1, & \text{jos } p_j f_j(x) < p_j \theta_j \\ 0, & \text{muulloin,} \end{cases} \tag{2.4}$$

jossa x on opetusnäytteestä saatu ali-ikkuna. Minimivirheen tuottavan luokittimen löytäminen vaatii runsaasti aikaa, koska jokaiselle piirteelle lasketaan optimaalinen kynnysarvo ja tämän jälkeen valitaan paras piirre.

2.1.5 Kaskadirakenteen muodostaminen

Kaskadirakenteen ideana on hylätä suurin osa kuvista, joissa ei ole kasvoja, mahdollisimman aikaisessa vaiheessa yksinkertaisilla luokitinkomponenteilla. Tämän jälkeen monimutkaisemmat komponentit saavat käsiteltäväkseen vähemmän näytteitä, ja niillä pyritään hylkäämään aiemmissa vaiheissa virheellisesti oikeiksi *(false positive)* luokittuneet näytteet (Kuva 2.7).



Kuva 2.7: Luokitinkaskadi, jossa komponenttien monimutkaisuus kasvaa loppua kohti.

Positiivinen luokitustulos ensimmäisestä komponentista ohjaa tarkasteluikkunan

toiselle komponentille. Toisen komponentin positiivinen tulos ohjaa tarkasteluikkunan kolmannelle komponentille ja niin edelleen. Kun tarkasteluikkuna läpäisee kaikki komponentit positiivisella luokitustuloksella, on havaittu kasvot. Jos taas mikä tahansa komponentti antaa negatiivisen luokitustuloksen, kuvasta ei ole havaittu kasvoja ja tarkastelua ei jatketa eteenpäin. Näin ollen laskennallisesti kasvojen havaitseminen on enemmän aikaa vievä operaatio kuin kasvojen kuin se ettei kuvasta löydy kasvoja. Tämä on myös haluttu lopputulos, koska oletuksena suurin osa kuvista ei sisällä kasvoja ja komponenttien kompleksisuus kasvaa kohti kaskadirakenteen loppua. [20]

Kaskadirakenteen komponentit muodostetaan opettamalla luokittimet AdaBoost-algoritmilla, asettamalla kynnysarvoksi virheellisesti negatiiviseksi luokittuneiden *(false negative)* luokitustulosten minimointi. Kaskadirakenteessa myöhempänä olevien komponenttien opetukseen käytetään opetusnäytteitä, jotka luokittuivat positiivisesti aiemmissa komponenteissa. Tuloksena myöhemmät komponentit käsittelevät vaikeampaa luokitustehtävää kuin aiemmat komponentit, koska aiemman komponentin läpäisseet opetusnäytteet ovat haastavampia kuin tyypilliset opetusnäytteet.

2.2 Iän ja sukupuolen arvioinnin teoreettinen tausta

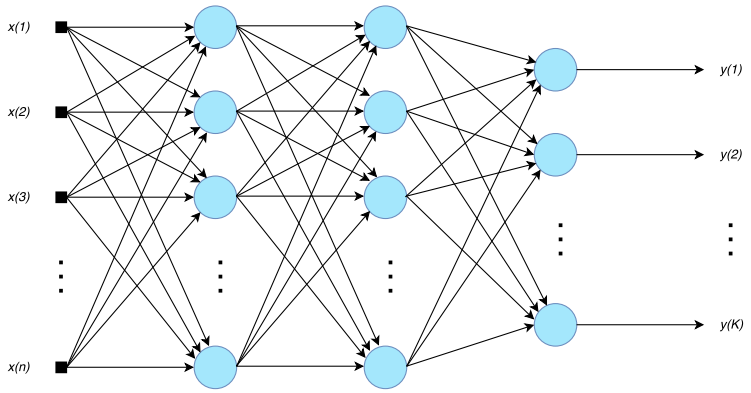
Kun kuvasta on pystytty määrittämään kasvojen sijainti ja koko, seuraavana tavoitteena on arvioida kuvassa olevan henkilön ikä ja sukupuoli. Tässä luvussa esitellään teoria konvoluutiohermoverkkojen käytölle kuvassa olevan henkilön iän ja sukupuolen arvioimiseksi. Aluksi luodaan lyhyt katsaus perinteisiin hermoverkkoihin, tämän jälkeen tutustutaan konvoluutiohermoverkkoon, joka on erityistapaus perinteisemmistä hermoverkoista. Lopuksi esitellään miten konvoluutiohermoverkkoja voidaan soveltaa iän ja sukupuolen arviointiin.

Teorian pohjana on hyödynnetty Tampereen teknillisen yliopiston signaalinkäsittelyn laitoksen kurssimateriaalia [8] ja aiemmin tehtyä kandidaatintyötä konvoluutiohermoverkoista [13]. Esiteltynä järjestelmä on käytetty Gil Levin ja Tal Hassnerin kehittämää konvoluutiohermoverkkoa iän ja sukupuolen arviointiin [14].

2.2.1 Hermoverkot

Ajatus hermoverkoista perustuu luonnollisiin hermoverkkoihin. Alkuperäisenä ajatuksena on siis ollut mallintaa ihmisaivojen tapaista päätöksentekoprosessia. Nykyisin kuitenkaan ei enää pyritä luonnollisten verkkojen toiminnan jäljittelyyn, vaan hermoverkkojen kehittäminen perustuu enemmän esimerkiksi tilastotieteeseen ja signaalinkäsittelyn teoriaan.

Perinteisesti hermoverkkojen voidaan ajatella koostuvan peräkkäisistä ja rinnakkaisista toisiinsa kytketyistä logistisista regressio-luokittimista. Hermoverkon jokainen solmu on siis logistinen regressiomalli, jota kutsutaan neuroniksi. Hermoverkoista on olemassa myös variantteja, joissa käytetään logistisen regression logistisen funktion sijasta jotakin muuta funktioita.



Kuva 2.8: Perinteinen monikerroksinen hermoverkko. Ensimmäisenä vasemmalla sisäänmenokerros ja oikealla ulostulokerros. Sisäänmenon ja ulostulon väliin jäävät piilokerrokset.

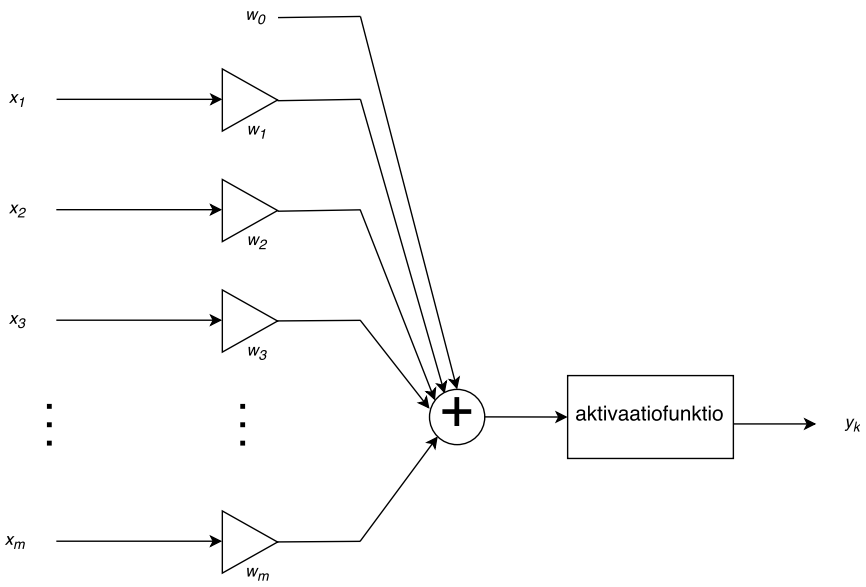
Kuvassa 2.8 on kuva keinotekoisesta hermoverkosta. Hermoverkko koostuu kerroksista *(layers)*, joiden jokaisesta neuronista on yhteys seuraavan kerroksen jokaiseen neuroniin. Jokainen neuroni suorittaa parametriensa mukaisen laskentaoperaation ja välittää saamansa tiedon eteenpäin seuraavalle kerrokselle.

Ensimmäisenä vasemmalla kuvassa on herätekerros *(input layer)*, jonka solmut välittävät herätteen lukuarvot $x(1), x(2), \ldots, x(M)$ hermoverkon ensimmäiselle kerrokselle. Varsinaisena herätteenä voi olla esimerkiksi piirrevektori tai vaikkapa kaikki kuvan pikselit. Vastaavasti kuvassa viimeisenä oikealla on kerroksena ulostulokerros *(output layer)*. Sisäänmenokerroksen ja ulostulokerroksen väliin kuvassa jää kolme kerrosta. Näitä kerroksia kutsutaan piilokerroksiksi *(hidden layer)*, ja piilokerroksen solmut suorittavat määrättyjä laskuoperaatioita ja välittävät saamansa lukuarvot eteenpäin kohti ulostulokerrosta.

Kuvassa 2.9 on kuvattu neuronin rakenne tarkemmin. Herätearvot $x_1, x_2, \ldots, x_m$ kerrotaan painoilla $w_1, w_2, \ldots, w_m$ ja saadut tulokset summataan yhteen. Tämän jälkeen tulee aktivaatiofunktio, joka simuloi luonnossa neuronien välittämää aktivaatioita eli siis hermosolu välittää tiedon eteenpäin riippuen siitä onko se aktivoitunut vai ei. Lisäksi summauksessa on aina mukana vakiotermi w_0.

Yksittäisen neuronin toimintaa voidaan kuvata kaavalla

$$y_k = \psi\left(w_0 + \sum_{k=1}^{m} w_k x_k\right), \tag{2.5}$$



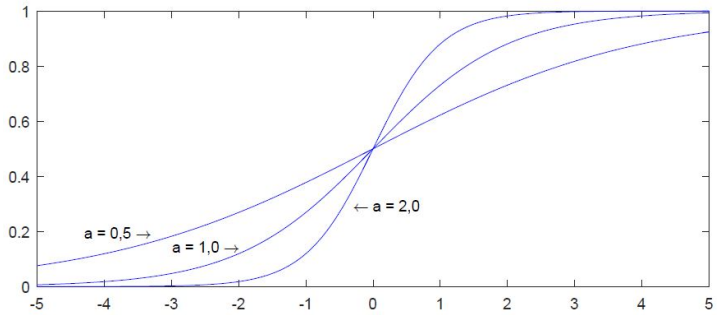
Kuva 2.9: Neuronin sisäinen rakenne. Neuroni kertoo kaikki sisääntulevat arvot painokertoimilla ja summaa yhteen. Tämän jälkeen aktivaatiofunktiolla määritetään, aktivoituuko neuroni vai ei.

jossa $\psi(\cdot)$ on aktivaatiofunktio.

Usein aktivaatiofunktiona käytetään jotakin S-kirjaimen muotoista *sigmoidi-funktioita*, josta on esimerkkinä *logistinen funktio*:

$$\psi(x) = \frac{1}{1 + e^{-ax}}, \tag{2.6}$$

jossa parametri a määrää miten jyrkästi funktio nousee origon lähellä. Logistisen funktion kuvaaja kolmella eri a:n arvolla on kuvassa 2.10.



Kuva 2.10: Logistisen funktion kuvaaja eri a:n arvoilla.

Kertoimen a kasvaessa funktio lähestyy jatkuvaa askelfunktioita

$$u(x) = \begin{cases} 0, \text{ kun } x < 0, \\ 1, \text{ kun } x \geq 0. \end{cases} \tag{2.7}$$

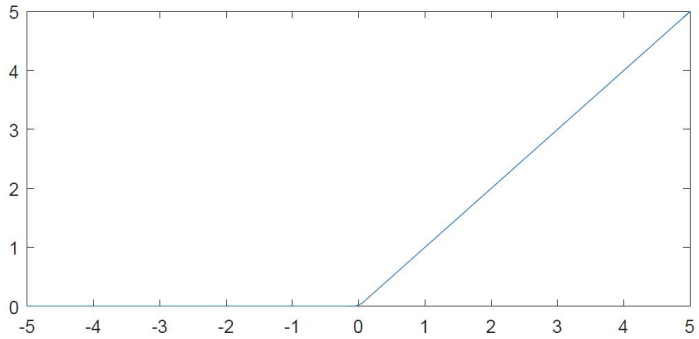
Aktivaatiofunktiona voidaan käyttää myös muita vastaavanlaisia funktioita. Tässä diplomityössä käytetään konvoluutiohermoverkkojen yhteydessä aktivaatiofunktiona ReLU-operaattoria *(Rectified Linear Unit)*:

$$f(x) = \max(0, x), \tag{2.8}$$

ReLU-operaattori on ramppisignaali, joka voidaan määritellä myös jatkuvan askelfunktion avulla:

$$r(x) = nu(x) = \begin{cases} 0, \text{ kun } x < 0, \\ n, \text{ kun } x \geq 0, \end{cases} \tag{2.9}$$

jossa $u(x)$ on jatkuva askelfunktio. ReLU-operaattori on esitetty kuvassa 2.11.



Kuva 2.11: ReLU-operaattorin kuvaaja välillä $-5 \leq x \leq 5$.

Aktivaatiofunktio päättää aktivoituuko neuroni vai ei, eli onko neuroniin sisään tulevien ärsykkeiden määrä tarpeeksi suuri aktiovoiman neuronin. Voidaan siis tarkastella, että toteuttaako herätteiden painotettu summa ehdon:

$$\sum_{k=1}^{m} w_k x_k \geq -w_0. \tag{2.10}$$

Neuroni siis aktivoituu, jos siihen kohdistuu riittävä määrä ärsykkeitä edellisen tason neuroneilta, ja edelleen kukin neuroni välittää tiedon aktivoitumisestaan seuraavan tason neuroneille ja näin vaikuttaa omalta osaltaan seuraavan tason neuronien aktivoitumiseen.

Hermoverkko koostuu yleensä hyvin suuresta määrästä yksinkertaisia laskentayksiköitä, joten kerroinmäärää varten tarvitaan automaattinen laskentamenetelmä. Yleisimmin käytetty painojen laskentamenetelmä on *back-propagation algorithm* (suom.: *takaisinlevitysmenetelmä*). Menetelmä pohjautuu LMS-algoritmiin *(Least Mean Squares)*, ja tässäkin tapauksessa virhefunktion minimi haetaan korjaamalla painoja gradientin minimin suuntaan.

Hermoverkon opettaminen on prosessi, joka suoritetaan ennen kuin verkko voidaan ottaa käyttöön. Yleensä opettaminen vie runsaasti prosessoriaikaa, tunteja tai jopa vuorokausia. Esimerkiksi tässä työssä käytettyjen itse opetettujen verkkojen opettamiseen kului Nvidia 5200M grafiikkaprosessorilla aikaa noin 30 tuntia.

Opetusta varten tarvitaan opetusjoukko, jossa on valmiiksi luokiteltuja esimerkkejä. Siten voidaan määrätä haluttu ulostulo $d(n)$ ja verrata sitä verkon ulostuloon $y(n)$. Halutaan siis laskea halutun ja todellisen ulostulon välinen virhe jokaisen opetusnäytteen jälkeen $e(n) = d(n) - y(n)$, kaikille $n = 1, 2, \ldots, K$. Itse virhefunktio on tavallisesti puolet virhevektorin neliösummasta:

$$\varepsilon = \frac{1}{2} \sum_{k=1}^{K} e^2(n). \tag{2.11}$$

Tässä työssä opetusnäytteinä käytetään kasvokuvia, joissa olevien henkilöiden ikä ja sukupuoli ovat tiedossa.

Virhefunktion ε arvo riippuu jokaisesta verkon painosta, ja on mahdollista johtaa kaavat funktion osittaisderivaatoille kunkin painon suhteen:

$$\frac{\partial \varepsilon}{\partial w_{jk}}, \tag{2.12}$$

missä w_{jk} tarkoittaa kerroksen j painoa k.

Vastaavasti voidaan käyttää myös jotakin logaritmista virhefunktioita, kuten esimerkiksi *logarithmic loss* [10]:

$$\text{log-loss} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(p_{ij}), \tag{2.13}$$

jossa N on näytteiden lukumäärä, M on luokkien lukumäärä, log on luonnollinen logaritmi. Muuttuja y_{ij} saa arvon 1, jos näyte i kuuluu luokkaan j, muulloin arvoksi tulee 0. Muuttuja p_{ij} kuvaa sitä, miten todennäköisesti näyte i kuuluu luokkaan j. Muuttujan p_{ij} arvot ovat väliltä varmasti tosi (1), tosi ja epätosi yhtäpitävät (0,5) ja varmasti epätosi (0). Logaritmin käyttö virheen laskemisessa johtaa siihen, että jos jokin luokitus merkitään varmasti todeksi (1) vaikka oikea tulos olisi epätosi, kasvaa virheen kokonaissumma äärettömäksi.

Kun virhe on laskettu, voidaan verkon painoja päivittää negatiivisen gradientin suuntaan, eli luonnollisesti suuntaan johon virhe pienenee. Kertoimien päivitys suoritetaan oikealta vasemmalle eli lopusta kohti alkua taso kerrallaan. Myöhäisempien tasojen painokertoimet vaikuttavat aikaisempien tasojen osittaisderivaattoihin, joten päivitys täytyy aloittaa ulostulon suunnasta.

Painot voidaan päivittää jokaisen esimerkin esittämisen jälkeen tai vasta kun kaikki opetusnäytteet on ajettu verkon läpi. Opetuksen tulos riippuu siitä, missä

järjestyksessä opetusnäytteet on ajettu hermoverkon läpi. Useimmiten järjestys on satunnainen, joten kahta samanlaista hermoverkkoa ei käytännössä voida saada peräkkäisillä opetuskerroilla.

Opetusjoukon lisäksi on hyvä olla käytössä myös pienempi testijoukko. Käytännössä testi joukko saadaan jakamalla opetukseen käytössä olevien näytteiden joukko opetusnäytteisiin ja testausnäytteisiin halutussa suhteessa. Käyttämällä erillisiä näytteitä opetukseen ja testaukseen saadaan parempi kuva siitä, miten hermoverkko toimisi todellisessa tilanteessa verrattuna siihen jos käytettäisiin samoja näytteitä opetukseen ja testaamiseen.

Opetuksen edetessä testijoukko ajetaan säännöllisin väliajoin hermoverkon läpi. Luokittelutuloksen virhe lasketaan, ja näin saadaan arvio siitä miten hyvin hermoverkko osaa luokitella ennen näkemättömiä näytteitä.

Testijoukkoa käyttämällä voidaan myös välttää verkon ylioppiminen *(overfitting)*, jota ilmenee erityisesti pienillä opetusjoukoilla. Ylioppimisessa hermoverkko alkaa oppia opetusjoukon individuaalit piirteet liian hyvin, eikä enää välttämättä sovellu yhtä hyvin opetusjoukon ulkopuolisten näytteiden luokitteluun.

2.2.2 Konvoluutiohermoverkot

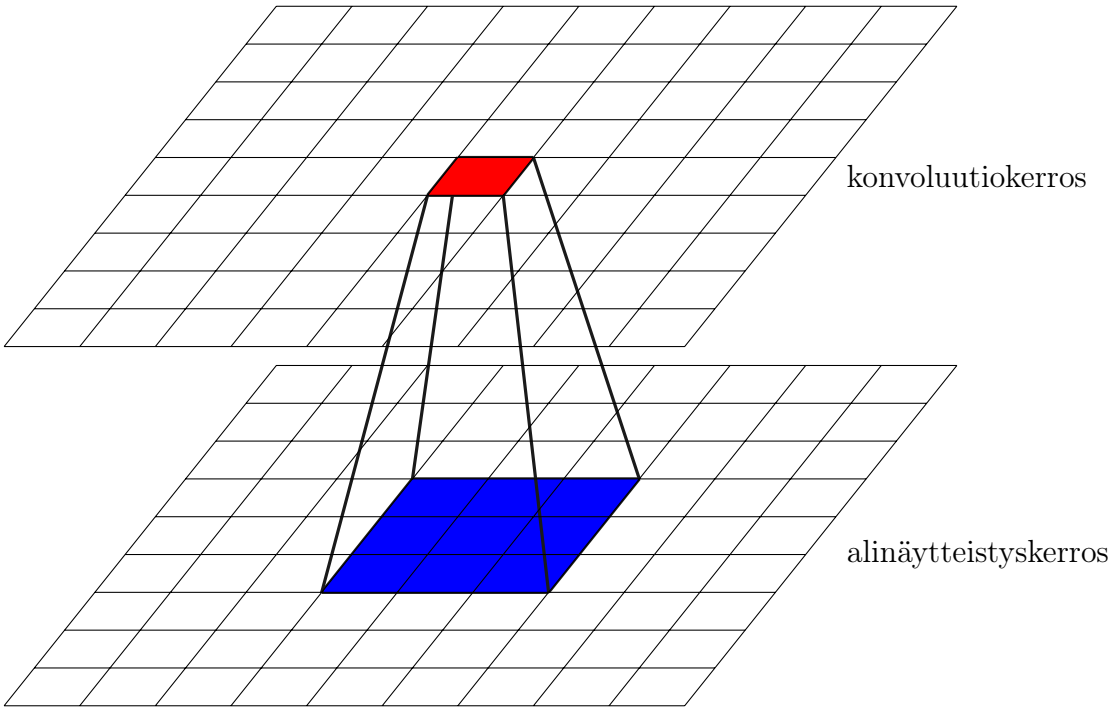
Kuten perinteinen hermoverkko pyrkii mallintamaan luonnollisten hermoverkkojen toimintaa, samoin on myös konvoluutiohermoverkkojen tapauksessa kuvankäsittelystä puhuttaessa. Tarkalleen ottaen pyritään hyödyntämään tietämystä ihmisen näköhermoston toiminnasta, sillä ihmisen näköaistia voidaan pitää huippukehittyneenä kuvankäsittelyjärjestelmänä.

Ensimmäisen kerran konvoluutiohermoverkon määritteli Le Cun et al. [12] vuonna 1990. Määritellyssä verkossa vuorottelee kaksi kerrosta: konvoluutiokerros ja alinäytteistyskerros *(subsampling layer)*. Konvoluutiokerros laskee edellisen alinäytteistyskerroksen ulostulosta kaksiulotteisen konvoluution:

$$g(y,x) = f * k = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(y+m, x+n)k(N-1-m, N-1-n), \qquad (2.14)$$

jossa f on $M \times M$ harmaasävykuva ja edellisen kerroksen ulostulo, ja k on $N \times N$ konvoluutiokerneli (kts. kuva 2.12). Tuloksena saadaan $(M-N+1)\times(M-N+1)$ kokoinen kuva. Yhtälö 2.14 on määrätty siten, että siinä ei sallita kertolaskua harmaasävykuvan indeksien ulkopuolella. Perinteiset hermoverkot käsittelevät kuvia yleensä yksiulotteisessa muodossa, konvoluutiohermoverkot puolestaan käsittelevät kuvia samassa muodossa kuin ihmisen näköjärjestelmä. Jokaisessa konvoluutiokerroksessa on useita konvoluutiokerneleitä, joiden kanssa sisääntuleva kuva konvoloidaan. Tuloksena saadaan K kappaletta erilisiä kuvia, joita kutsutaan nimellä *feature map*,

joka siis vastaa konvoluution ulostuloa $g = f * k$. Jokainen erillinen feature map siis erikoistuu verkon opetuksen yhteydessä tunnistamaan kuvasta tiettyä ominaisuutta oman konvoluutiokernelinsä avulla.



Kuva 2.12: Pikselin arvon laskeminen konvoluution avulla. Konvoluutiokernelin koko on 3×3. Alemman kerroksen pikselit (sininen alue) kerrotaan konvoluutiokernelin painokertoimilla ja summataan yhteen. Tärkeänä huomiona on, että sisääntuloalue on lokaali eikä koko kuvan alueelle ulottuva.

Konvoloimalla kuva monen konvoluutiokerroksen läpi, pystyy verkko havaitsemaan kuvasta koko ajan monimutkaisempia ja laajemmalle levinneitä ominaisuuksia verkon loppupäähän kuljettaessa. Alinäytteistyskerrosten tehtävänä on vähentää verkon herkkyyttä häiriöille eli ne toimivat kohinanpoistosuodattimina. Tehtävät operaatiot ovat käytännössä kuvan pienentämistä, ja tavallisesti tämä on vierekkäisten pikselien keskiarvottamista tai ns. *max pooling* -operaatio, jolloin ikkunan sisältä valitaan pikseli jonka sävyarvo on suurin.

Viimeisen konvoluutiokerroksen jälkeen verkon ulostulona on yksiulotteinen vektori, joka syötetään 1–2 kerroksiselle täysin kytketylle (perinteiselle) hermoverkolle. Näin saatu ulostulo on verkon lopullinen ulostulo.

Yhteenvetona voidaan todeta, että konvoluutiohermoverkko erottaa kuvasta tunnistuksen kannalta olennaisimmat piirteet, jotka voidaan helposti luokitella matalalla hermoverkolla. Näin voidaan vähentää optimoitavien parametrien määrää täysin kytkettyyn hermoverkkoon verrattuna.

2.2.3 Konvoluutiohermoverkot iän ja sukupuolen arvioinnissa.

Motivaationa iän ja sukupuolen tunnistukselle voidaan pitää faktaa, että esimerkiksi monissa kielissä on omat kielioppisääntönsä miespuolisille ja naispuolisille henkilöille, ja erilaisia tervehdyksiä eri ikäisille henkilöille. Iän ja sukupuolen tunnistusta voidaan myös pitää malliesimerkkinä tehtävästä, joka on helppo ihmiselle, mutta hankala tietokoneelle. Seuraavaksi esitellään ratkaisu, jossa on käytetty konvoluutiohermoverkkoja iän ja sukupuolen arviointiin valokuvasta [14].

Kuva 2.13: Konvoluutiohermoverkon arkkitehtuuri. Ensimmäinen konvoluutiokerros koostuu 96 suotimesta jotka käyttävät 7×7 pikselin kerneliä, toisella konvoluutiokerroksella on 256 suodinta jotka käyttävät 5×5 pikselin kerneliä, ja viimeisellä konvoluutiokerroksella on 384 suodinta jotka käyttävät 3×3 pikselin kerneliä. Näitä seuraa kaksi täysin kytkettyä kerrosta joilla on molemmilla 512 neuronia.

Esitellyssä ratkaisussa on käytetty samanlaista hermoverkon topologiaa sekä iän, että sukupuolen arvioinnissa. Hermoverkon toimintaidea on esitelty kuvassa 2.13. Verkko koostuu ainoastaan kolmesta konvoluutiokerroksesta ja kahdesta täysin kytketystä kerroksesta, joissa on suhteellisen vähäinen määrä neuroneja. Vertailun vuoksi, kuvien luokitteluun suunnitelluissa verkoissa on käytetty laajempia verkkoja [2] [11]. Esimerkkiarvoina on 5 konvoluutiokerrosta ja 4 096 neuronia täysin kytketyillä kerroksilla. Verkkojen koko on valittu tarkoituksella pienemmäksi ylioppimisen välttämiseksi. Toisena syynä pienempiin verkkoihin on tarkasteltava ongelma. Tässä mallissa iät on jaettu luokkiin siten, että mahdollisia luokkia on kahdeksan. Sukupuolen arvioinnissa luokkia on vain kaksi. Tämä on huomattavasti pienempi määrä verrattuna esimerkiksi ImageNet-ongelmaan, jossa luokkia on 1 000 [11]. Taulukossa 2.1 on esitelty, miten luokat on määritelty ja miten monta kuvaa kustakin luokasta oli käytössä.

Taulukko 2.1: Käytössä olleiden kuvien jakautuminen ikä- ja sukupuoli ryhmittäin.

	0–2	4–6	8–13	15–20	25–32	38–43	48–53	60–	Yhteensä
Mies	745	928	934	734	2 308	1 294	392	442	7 777
Nainen	682	1 234	1 360	919	2 589	1 056	433	427	8 700
Molemmat	1 427	2 162	2 294	1 653	4 897	2 350	825	869	16 477

Verkko käsittelee kuvat suoraan kolmivärikuvina. Aluksi kuvat skaalataan uudelleen kokoon 256×256. Tämän jälkeen kuvia rajataan siten, että kuvan keskipiste säilyy ennallaan ja talteen otetaan 227×227 pikselin kokoinen kuva. Tämä kuva syötetään verkolle.

Aktivaatiofunktiona käytetään ReLU-operaattoria, jonka ansiosta syvien konvoluutiohermoverkkojen opettaminen onnistuu moninkertaisesti nopeammin verrattuna perinteisiin aktivaatiofunktioihin [11]. ReLU-operaation jälkeen suoritetaan vielä *local response normalization -operaatio (LRN)*, jonka tehtävänä on helpottaa generalisaatiota ReLU -operaation jälkeen. Merkitköön $a^i_{x,y}$ neuronin aktivaatiota joka saadaan sijainnissa (x, y) kernelin i jälkeen lisäämällä ReLU-operaation epälineaarisuus, saadaan LRN-aktivaatio:

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}, \tag{2.15}$$

jossa n on samassa spatiaalipositiossa olevien "vierekkäisten" kernelien määrä ja N on kerroksessa olevien kernelien kokonaismäärä. Vakiot k, n, α ja β ovat hyperparametrejä jotka on saatu määriteltyä kokeilemalla, ja tässä tapauksessa käytetään arvoja $k = 2, n = 5, \alpha = 10^{-4}$ ja $\beta = 0.75$. [11]

Seuraavaksi esitellään tarkempi kuvaus käytetyn konvoluutiohermoverkon rakenteesta, joka on nähtävissä kuvasta 2.14. Verkon kolme konvoluutiokerrosta on määritelty seuraavalla tavalla:

1. Ensimmäinen konvoluutiokerros muodostuu $3 \times 7 \times 7$ pikselin suotimista, joita on 96 kappaletta. Tätä seuraa ReLU-operaatio ja max pooling -operaatio, joka poimii maksimiarvot 3×3 alueilta kahden pikselin askelvälillä. Tämän jälkeen toteutetaan vielä LRN-operaatio.

2. Toinen konvoluutio kerros saa sisäänmenonaan edellisen kerroksen $96 \times 28 \times 28$ ulostulon. Se sisältää 256 kappaletta $96 \times 5 \times 5$ pikselin suotimia. Tämän jälkeen suoritetaan ReLU-, max pooling- ja LRN-operaatiot samoilla parametreilla kuin aiemmalla kerroksella.

3. Viimeiselle konvoluutiokerrokselle tulee sisään $256 \times 14 \times 14$ kokoinen matriisi jolle suoritetaan 384 kppaletta $256 \times 3 \times 3$ pikselin kokoisia suodatuksia. Tätä seuraavat vielä ReLU- ja max pooling -operaatiot.

Tätä seuraavat täysin kytketyt kerrokset:

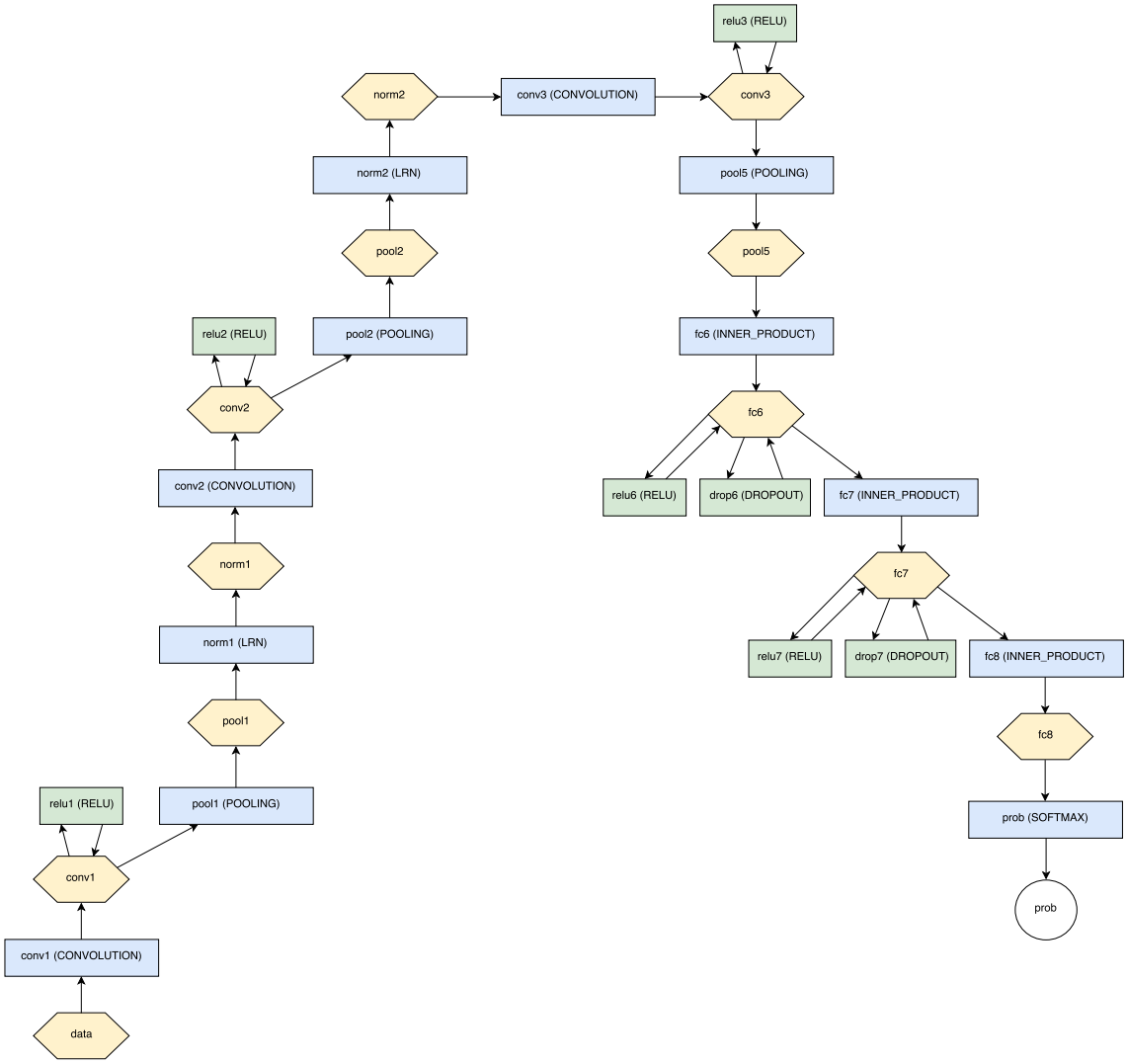
4. Ensimmäisen täysin kytketyn kerroksen sisäänmenona on viimeisen konvoluutiokerroksen ulostulos. Kerros koostuu 512 neuronista ja niitä seuraa ReLU-

operaatio ja dropout-operaatio arvolla 0,5, joka tarkoittaa, että neuronin ulostuloksi asetetaan nolla 50 prosentin todennäköisyydellä.

5. Toinen täysin kytketty kerros saa sisäänmenona ensimmäisen täysin kytketyn kerroksen 512-ulotteisen ulostulon. Myös toinen kerros koostuu 512 neuronista, ja niitä seuraavat ReLU- ja dropout-operaatiot.

6. Viimeinen täysin kytketty kerros mappaa aiemman kerroksen ulostulon oikeaan ikä-tai sukupuoliluokkaan.

Viimeisenä on soft-max-kerros, joka kertoo prosentuaalisen todennäköisyyden siitä, miten todennäköisesti näyte kuuluu mihinkin luokkaan. Itse lopputulokseksi valitaan lopulta luokka, jolla on suurin todennäköisyys.



Kuva 2.14: Verkon täydellinen arkkitehtuuri. Yksityiskohdat kuvattu tekstissä.

Tulosten mukaan suurimmat virheet testiaineiston kanssa tulivat kuvista, jotka

olivat haastavia hämärän kuvan tai alhaisen resoluution vuoksi. Sukupuolen arvioinnin puolella säännöllisimmät väärinarvioinnit tapahtuivat vauvojen ja nuorten lasten kohdalla, sillä sukupuolen määrittämisen kannalta tärkeimmät ominaisuudet eivät ole vielä esillä. Sukupuolen arvioinnissa oikeinluokittuneiden osuus oli 86,8 ± 1,4 prosenttia. Iän arvioinnissa arvio osui täsmälleen oikeaan luokkaan 50,7 ± 5,1 prosentin todennäköisyydellä ja vähintään edelliseen tai seuraavaan luokkaan 84,7 ± 2,2 prosentin todennäköisyydellä.

Yhteenvetona voidaan todeta, että konvoluutiohermoverkoilla voidaan parantaa iän ja sukupuolen arvioinnin tähänastisia tuloksia. Toiseksi tuloksista voidaan olettaa, että laajemmat ja huolitellummat järjestelmät, joille on käytössä paljon opetusdataa voivat saada aikaan parempia lopputuloksia.

2.2.4 Support Vector -regressio

Jos luokkien sijasta halutaan käyttää tarkkoja arvoja luokitteluongelman ratkaisemiseen, voidaan ongelmaa lähestyä regression avulla. Jos ajatellaan iän arviointia, on tässä työssä tehtävässä sovelluksessa käyttäjän kannalta mielekkäämpää antaa tarkka arvio käyttäjän iästä, kuin että käytettäisiin taulukossa 2.1 esitettyjä ikäjakaumaluokkia.

Tässä työssä on regressiomalliksi valittu SVR-luokitin *(Support Vector Regression)*, jolla on seuraavanlainen matemaattinen tausta [19]. Opetusvektoreilla $x_i \in \mathbb{R}^p$, $i = 1, 2, \ldots, n$ ja vektorilla $y \in \mathbb{R}^n$ SVR ratkaisee seuraavan ongelman:

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\zeta_i + \zeta_i^*) \tag{2.16}$$

ehdoilla:

$$\begin{gathered} y_i - w^T \phi(x_i) - b \leq \varepsilon + \zeta_i, \\ w^T \phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*, \\ \zeta_i, \zeta_i^* \leq 0, i = 1, 2, \ldots, n \end{gathered} \tag{2.17}$$

Sen duaali on

$$\min_{\alpha,\alpha^*} \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \varepsilon e^T (\alpha + \alpha^*) - y^T (\alpha - \alpha^*) \tag{2.18}$$

ehdoilla:

$$\begin{gathered} \eta^T (\alpha - \alpha^*) = 0 \\ 0 \leq \alpha, \alpha^* \leq C, i = 1, 2, \ldots, n \end{gathered} \tag{2.19}$$

jossa e on vektori ykkösiä, $C > 0$ on yläraja, Q on $n \times n$ positiivinen semidefiniitti matriisi ja $Q_{ij} \equiv K(x_i, x_j) = \phi(x_i)^T \phi x_j$ on kerneli.

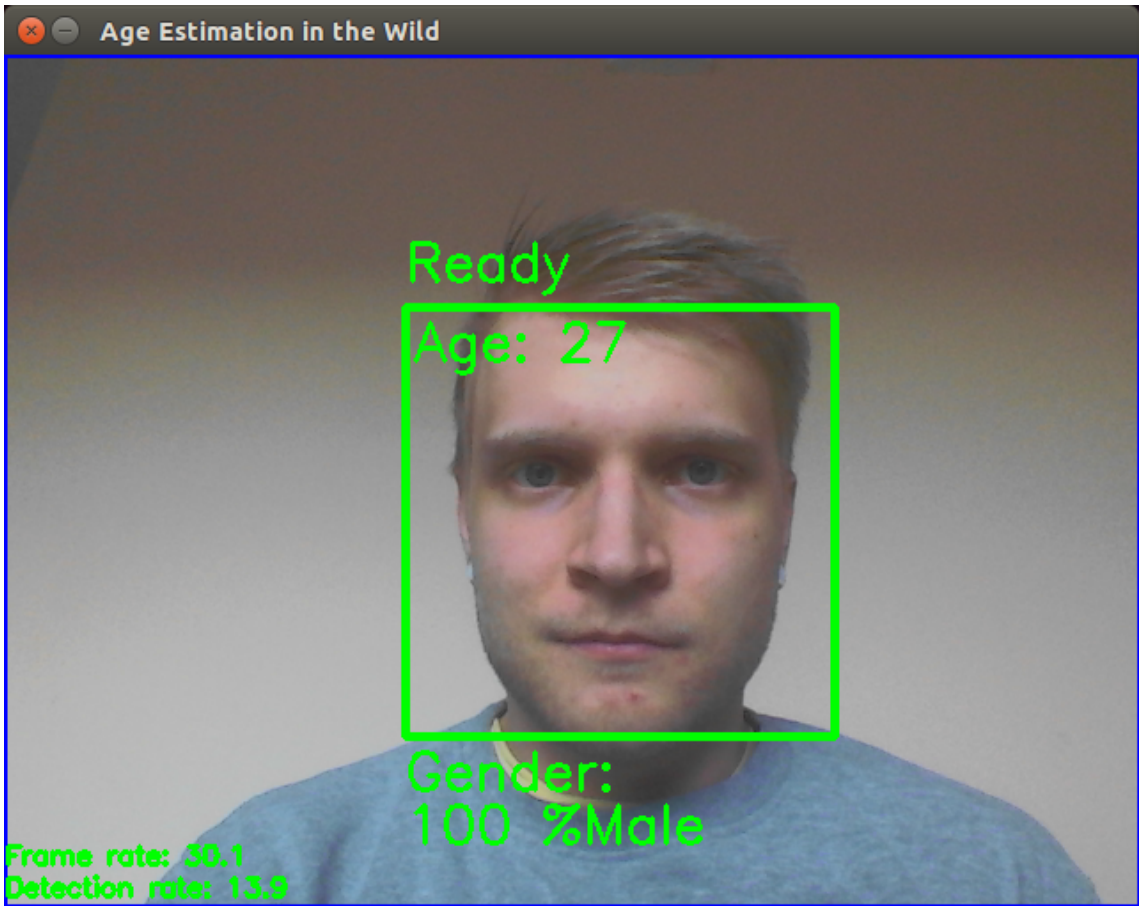
Päätösfunktioina toimii:

$$\sum_{i=0}^{n}(\alpha - \alpha^*)K(x_i, x_j) + \rho \tag{2.20}$$

Tavoitteena on luoda regressiomalli sisäänmenon ja halutun ulostulon välille, ja etsiä optimaalinen regressiomalli muunneltavien parametrien avulla.

3. JÄRJESTELMÄN KUVAUS

Tässä diplomityössä toteutettu järjestelmä on reaaliaikainen iän ja sukupuolen arviointijärjestelmä, joka suorittaa arvioinnin videokuvasta. Tunnistus suoritetaan rinnakkaisina prosesseina, joita ovat videokuvan käsittely, kasvojen etsintä, iän arviointi ja sukupuolen arviointi. Kasvojen etsintä suoritetaan kaskadirakenteisella luokittimella ja iän ja sukupuolen arviointiin käytetään konvoluutiohermoverkkoa.

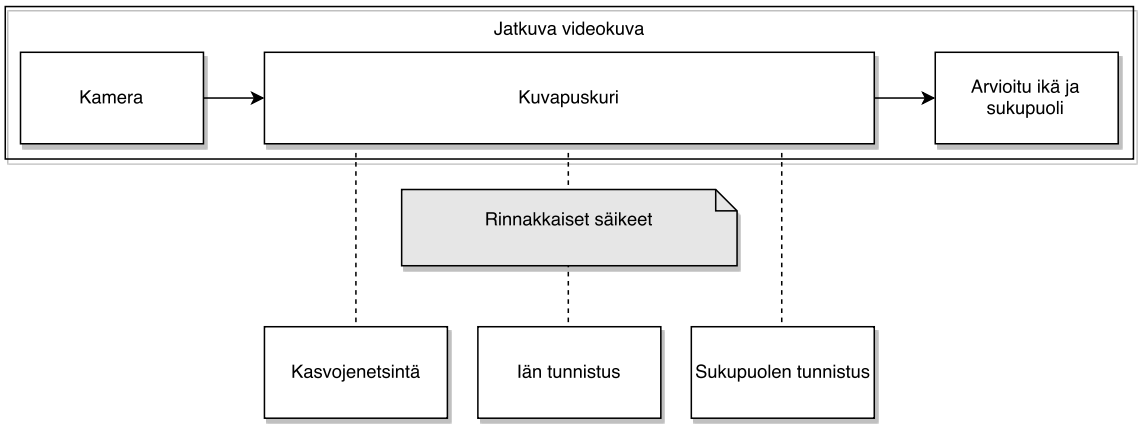


Kuva 3.1: Kuva käyttöliittymästä.

Kuvassa 3.1 on järjestelmän käyttöliittymäkuva, joka esittää onnistuneen arviointiprosessin lopputulosta. Kuvaan on merkitty järjestelmän tunnistamat kasvot ja estimointi löydettyjen kasvojen iästä ja sukupuolesta. Myöhemmin tässä luvussa esitellään järjestelmän arkkitehtuuri ja tehdään tarkempi kuvaus toteutetusta järjestelmästä.

3.1 Arkkitehtuuri

Järjestelmän toimintaperiaate noudattaa mallia, jossa järjestelmään menee sisään kuva, ja ulostulona saadaan arvio kuvassa olevan henkilön iästä ja sukupuolesta. Arkkitehtuurina käytetään siis tietovuoarkkitehtuuria, joka on esitelty kuvassa 3.2.



Kuva 3.2: Järjestelmän yleisarkkitehtuuri tietovuomallin avulla kuvattuna.

Keskeisenä komponenttina on jatkuvaa videokuvaa käsittelevä säie. Kyseinen säie kuvaa kohdetta videokameran välityksellä, ja ottaa videokuvasta jatkuvasti kuvakaappauksia, joita järjestelmän muut säikeet pyytävät käsiteltäviksi. Jatkuvan videokuvan säie toimii siis myös ikään kuin väylänä muiden säikeiden välillä. Videokuvasäikeen toisena tehtävänä on myös videokuvan ja muilta säikeiltä saadun tiedon eli löydettyjen kasvojen, iän ja sukupuolen esittäminen käyttäjälle. Koska videokuvauksen on oltava ohjelman käyttäjäystävällisen käytettävyyden vuoksi jatkuva-aikaista, on komponenttien toiminnassa hyödynnetty rinnakkaisuutta.

Muina itsenäisinä komponentteina toimivat kasvojenetsintä, iän tunnistus ja sukupuolen tunnistus. Kuten tietovuoarkkitehtuurille on olennaista, kaikki komponentit ovat toisistaan riippumattomia ja voidaan tarvittaessa korvata muilla saman toiminnallisuuden omaavilla komponenteilla. Tämä on myös tärkeä ominaisuus järjestelmän jatkokehityksen kannalta.

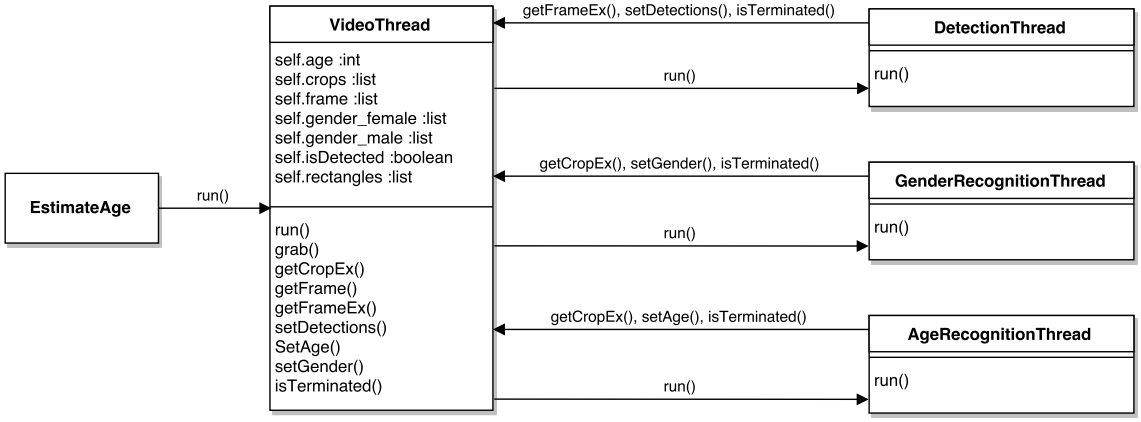
Kasvojen etsintä, nimensä mukaisesti, etsii videokuvasta otetuista kuvakaappauksista kasvoja. Kun kuvasta löydetään kasvot, ilmoitetaan kasvojen koordinaatit videokuvakomponentille, josta väylällä seuraavana oleva komponentti voi niitä hyödyntää. Vastaavasti jos kuvasta ei löydy kasvoja, siirrytään seuraavaan kuvakaappaukseen ja suoritetaan sama operaatio uudelleen.

Iän tunnistus ja sukupuolen tunnistus suoritetaan kun videokuvasta on löydetty kasvot. Kasvojen koordinaattien avulla rajataan kuvakaappausta siten, että tunnistettavasta kuvasta säilytetään pelkät kasvot, ja muu informaatio jätetään tarkastelematta. Molemmissa komponenteissa arviointi suoritetaan käyttämällä hermoverkkoja eli kuvan kasvoille estimoidaan ikä perustuen verkkojen oppimiin ominaisuuksiin

kasvojen piirteiden välisestä yhteydestä ikään ja sukupuoleen.

3.2 Kehitysnäkymä

Tässä luvussa on esitetty komponenttien yksityiskohtaisempi toiminta ohjelmistokehityksen näkökulmasta. Alla oleva luokkakaavio kuvaa järjestelmän rakennetta (Kuva 3.3).



Kuva 3.3: Järjestelmän luokkakaavio. EstimateAge-luokka ainoastaan käynnistää tunnistusprosessin ja varsinainen kommunikaatio tapahtuu muista luokista VideoThread-luokkaan ja takaisin.

EstimateAge-luokka käynnistää ohjelman. Se alustaa ja käynnistää VideoThread-säikeen suorituksen. Käynnistyttyään VideoThread-luokka käynnistää ohjelman loput säikeet ja ohjelman suoritus voi alkaa.

VideoThread-luokka sisältää säikeen joka ohjaa videonkäsittelyä. Videonkäsittely sisältää kuvakaappausten ottamisen kameralta, löydettyjen kasvojen sijainnin varastoinnin ja kasvoja vastaavien iän ja sukupuolen esittämisen käyttäjälle. Kun tuloksia esitetään käyttäjälle, voidaan tulokset esittää jokaisen tunnistusprosessin jälkeen tai vaihtoehtoisesti voidaan esittää keskiarvo useammasta peräkkäisestä samasta henkilöstä tehdystä tunnistuksesta. Kameran käytössä hyödynnetään OpenCV-kirjaston tarjoamaa rajapintaa videokuvan nauhoittamiseksi kamerasta [16]. Luokan tehtävänä on myös toimia tietovuon väylänä muille komponenteille. Luokan jäsenmuuttujat siis sisältävät kaiken oleellisen tiedon prosessin suorittamiseksi, ja muut komponentit ainoastaan kysyvät ja muuttavat VideoThread-luokan tiloja.

DetectionThread-luokka sisältää säikeen, joka vastaa kasvojen etsinnästä. Säie pyytää VideoThread-luokalta kuvakaappauksen ja saatu kehys muutetaan monivärisestä harmaasävyiseksi. Harmaasävyinen kehys ajetaan kaskadiluokittimelle, joka kertoo löydettyjen kasvojen koordinaatit. Kasvojen löydyttyä säie välittää löydettyjen kasvojen koordinaatit VideoThread-luokalle. Mikäli kuvasta ei löydy kasvoja, pyytää säie uutta kuvakaappausta VideoThread-luokalta.

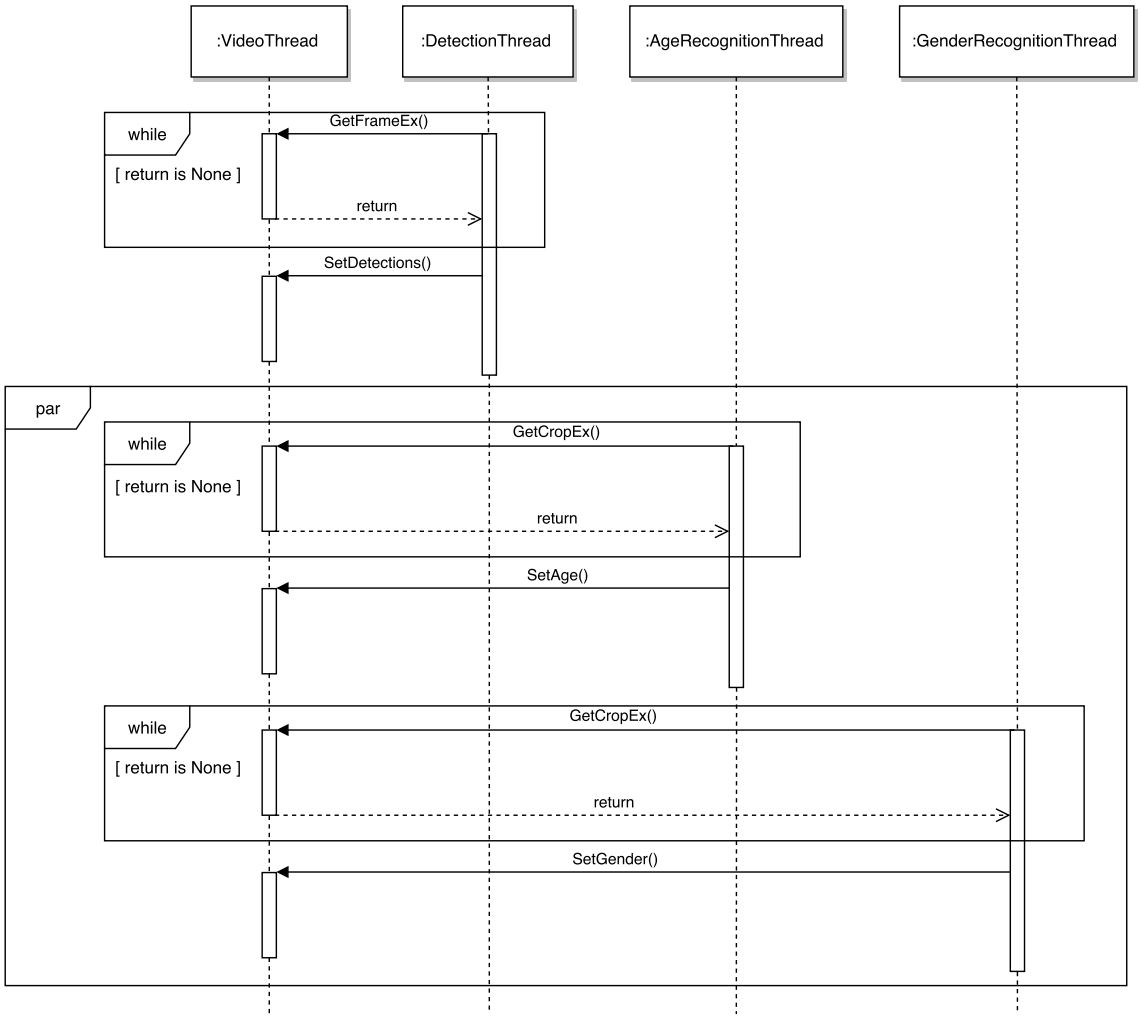
Kasvojen etsinnässä hyödynnetään OpenCV-kirjaston mukana tulevia valmiiksi opetettuja XML-muotoon tallennettuja kaskadiluokittimia. Tässä toteutuksessa on käytetty OpenCV-kirjaston kaskadiluokitinrajapintaa ja luokittimena on käytetty Rainer Lienhartin kehittämää *Stump-based 20×20 gentle adaboost frontal face detector* -luokitinta, joka on modifioitu versio luvussa 2 esitellystä Paul Violan ja Michael Jonesin kehittämästä kaskadiluokittimesta [15]. Luokittelussa käytettävä kaskadiluokittimen kuvauksen sisältävä XML-tiedosto linkitetään OpenCV-kirjastoon, ja näin saadaan alustettua luokitin, jota käytetään kosvojenetsintään.

AgeRecognitionThread- ja GenderRecognitionThread-luokkien toteutukset muistuttavat hyvin paljon toisiaan. Merkittävimpänä eroavaisuutena on, että toinen antaa estimaatin iästä ja toinen sukupuolesta. Perusideana molemmissa on kuitenkin, että pyydetään VideoThread-luokalta kasvokuva, ajetaan kuva hermoverkolle ja verkolta saatu arviointitulos palautetaan takaisin VideoThread-luokalle.

Molemmissa tapauksissa käytetään syväoppimiselle tarkoitettua Caffe-ohjelmistokehystä [9]. Näin voidaan hyödyntää luvussa 3 esiteltyjä Gil Levin ja Tal Hassnerin suunnittelemia konvoluutiohermoverkkoja, joista on saatavilla topologiamallit Caffelle [14]. Sukupuolen estimointiin luvun 3 konvoluutiohermoverkkoa käytetään sellaisenaan. Iän estimointiin luvun 3 konvoluutiohermoverkkoa on modifioitu siten, että luokkien sijasta etsitään regression avulla numeerinen arvo estimoitavalle iälle. Tämä onnistuu ohjaamalla verkon viimeisen täysin kytketyn kerroksen ulostulo SVR-luokittimelle *(Support Vector Regression)*, jolle on opetettu regressiomalli täysin kytketyltä kerrokselta saatavien arvojen ja opetusnäytteiden oikeiden ikien välille. SVR-luokitin on toteutettu käyttämällä hyväksi scikit-learn -alustaa [18]

3.3 Looginen näkymä

Seuraavaksi esitellään yksi järjestelmän alusta loppuun asti suorittama arviointioperaatio käyttämällä visualisointiin sekvenssikaaviota (Kuva 3.4).



Kuva 3.4: Sekvenssikaavio yhdestä järjestelmän suorittamasta tunnistusprosessista.

Ohjelman käynnistyttyä VideoThread-säie käynnistää kameran ja alkaa ottamaan kuvakaappauksia videokuvasta. DetectionThread-säie pyytää VideoThread-säikeeltä kuvakaappauskehyksiä käsiteltäväksi. Tämä tapahtuu silmukassa eli säie pyytää tasaisin väliajoin kehystä, kunnes uusi kehys on saatavilla. Saatuaan kehyksen DetectionThread-säie suorittaa oman kasvojenetsintäprosessinsa. Sen valmistuttua DetectionThread-säie välittää tiedon löydettyjen kasvojen koordinaateista VideoThread-säikeelle tai vastaavasti tiedon, että kasvoja ei löytynyt. VideoThread-säie tallettaa koordinaatit ja piirtää käyttöliittymään neliön löydettyjen kasvojen ympärille.

Ohjelman suoritusjärjestyksen kannalta on olennaista, että kasvojen etsintä tulee olla suoritettuna ennen siirtymistä iän ja sukupuolen arviointiin. Tämä onnistuu

siten, että DetectionThread-säie käsittelee kehyksiä joissa on koko kameran kuvaamisalue, ja AgeRecognitionThread- ja GenderRecognitionThread-säikeet käsittelevät ainoastaan rajattuja kehyksiä, jotka DetectionThread-säie on tunnistanut kasvoiksi.

AgeRecognitionThread- ja GenderRecognitionThread-säikeet ovat käytännössä hyvin samalla tavalla toteutettu ja eroavaisuutena on lähinnä se, että toisessa konvoluutiohermoverkko on opetettu luokittelemaan ikäryhmiä ja toisessa sukupuolia. Molemmat säikeet odottavat DetectionThread-säikeen tavoin silmukassa kunnes pyyntö VideoThread-säikeeltä tuo vastauksena rajatun kehyksen, jossa on kasvot. Tämän jälkeen molemmat suorittavat oman luokitteluprosessinsa ja palauttavat luokitustuloksensa eli tiedon iästä ja sukupuolesta VideoThread-säikeelle, joka puolestaan tulostaa tiedot käyttöliittymään aiemmin kasvojen ympärille piirretyn neliön läheisyyteen.

Ikää ja sukupuolta arvioivien säikeiden toteutus riippuu ainoastaan siitä, että on saatavilla rajattu kuva, jossa on kasvot. Muuten säikeet ovat täysin riippumattomia toisistaan sekä muista säikeistä. Tämä mahdollistaa säikeiden rinnakkaisen luonteen, ja nopeuttaa täten ohjelman suorittamista.

3.4 Rinnakkaisuus

Järjestelmässä VideoThread, DetectionThread, AgeRecognitionThread ja GenderRecognitionThread ovat rinnakkaisia säikeitä. Suoritusjärjestyksen puolesta DetectionThread-säie odottaa VideoThread-säikeeltä uutta kuvakaappausta ja AgeRecognitionThread- ja GenderRecognitionThread-säikeet odottavat DetectionThread-säikeeltä kuvakaappauksesta löydettyjä kasvokoordinaatteja. Koska säikeet käsittelevät yhteisiä muuttujia, on ohjelmassa käytetty poissulkemista *(mutual exclusion, mutex)* kriittisien alueiden suojaamiseksi ja lukkiutumisen estämiseksi.

Lukkiutumisella tarkoitetaan tilaa, jossa kaksi tai useampi prosessi ei pääse etenemään, koska ne odottavat toisiaan. Lukkiutumisella on neljä välttämätöntä ja samalla riittävää ehtoa [6]:

1. Poissulkemisehto. Säie varaa jonkin resurssin vain omaan käyttöönsä.
2. Irrottamattomuusehto. Vain säie itse voi vapauttaa varaamansa resurssin.
3. Varaus-odotusehto. Säie ei vapauta resurssejaan odottaessaan lisäresurssien vapautumista.
4. Silmukkaodotusehto. Säikeet odottavat toisiaan silmukassa.

Yhdenkin ehdon rikkominen riittää lukkiutumisen estämiseksi, joten kaikkia ehtoja ei tarvitse estää.

Tässä järjestelmässä rikkoutuu ainoastaan silmukkaodotusehto. Poissulkemisehto ja irrottamattomuusehto toteutuvat, sillä VideoThread-säie varaa resursseja vain omaan käyttöönsä ja ainoastaan säie itse kykenee vapauttamaan varaamansa resurssit. Lisäksi myös varaus-odotusehto toteutuu, koska poissulkemispyynnön aluksi ei vapauteta resursseja.

Rinnakkaiset ohjelmanosat joutuvat ohjelmaa suoritettaessa odottamaan toisiaan. Odotusta voidaan hallita synkronoinnilla. Tässä järjestelmässä synkronointi on toteutettu silmukassa aktiivisella odotuksella. Käytännössä tämä tapahtuu siten, että mikäli kysytty resurssi on varattu tai ei ole saatavilla, kysyvä säie nukutetaan 0,1 sekunnin ajaksi ennen kuin se kysyy resurssia uudelleen. Näin ollen nukuttamisella säästetään myös hieman suoritusaikaa, koska kyselyoperaatiota ei suoriteta jatkuvalla nopeudella. Tämä rikkoo myös silmukkaodotusehdon ja näin kyetään täyttämään vaatimukset järjestelmän lukkiutumisen estämiseksi.

Lisäongelman muodostaa se, mikäli ohjelman suoritus lopetetaan VideoThread-säikeessä kesken kun jokin muu säie suorittaa aktiivista odottamista. Tämä johtaisi siihen, että kysyvä säie odottaa vapaaksi resurssia, joka ei enää ole saatavilla, ja ohjelma jää lukkotilaan. Ongelma vältetään sillä, että VideoThread-luokan isTerminated()-metodilla välitetään muille säikeille tieto ohjelman suorituksen loppumisesta, ja täten säikeiden suorittaminen voidaan lopettaa onnistuneesti.

3.5 Suorituskykyvaatimukset

Ohjelman suorituskyvyn parantamiseksi on Caffe-ohjelmistokehykselle suunniteltujen konvoluutiohermoverkkojen ajamiseen käytetty grafiikkaprosessorilaskentaa, joka nopeuttaa huomattavasti hermoverkkojen käyttöä. Jotta Caffe pystyy hyödyntämään grafiikkaprosessorilaskentaa, täytyy käytössä olla Nvidia-näytönohjain, joka on yhteensopiva vähintään CUDA:n version 7.0 kanssa.

3.6 Opetus

Hermoverkot on mahdollista opettaa uudella opetusjoukolla, ja tässä tapauksessa opetuksessa on keskitytty iän arviointiin suunnitellun verkon uudelleenopettamiseen. Tämän lisäksi on perehdytty SVR-luokittimen opettamiseen.

3.6.1 Hermoverkkojen opetus

Materiaaliksi opetukseen tarvitaan kuvia kasvoista ja lisäksi tekstitiedosto, joka sisältää listauksen opetuksessa käytettävien kuvien tiedostonimistä ja kuvissa olevien henkilöiden iän. Opetusmateriaalina on pääosin käytetty internetistä saatuja valmiiksi kerättyjä opetusmateriaaleja, jotka on luotu iänarviointikilpailuja tai

-tutkimusta varten. Käytettyjä opetusmateriaaleja olivat *FG-NET Aging Database* [4], *ChaLearn Looking at people apparent age estimation* -kilpailussa käytetty opetusaineisto ja Israelin avoimen yliopiston *The OUI-Adience Face Image Project* -tietokanta [3].

Kun opetusaineistona käytetään valmiiksi saatavilla olevia tietokantoja, vaatii se yleensä jonkin verran esikäsittelyä. Tässä tapauksessa esikäsittelynä suoritettiin ohjelmallisesti kuvien kerääminen saman polun alle, joka on hyvin triviaali operaatio, ja kuvien label-tiedostojen kokoaminen yhteen tekstitiedostoon ja saattaminen samaan yksiselitteisesti tulkittavaan muotoon, jotta kuvia voidaan käyttää opetuksessa. Tämän lisäksi poistettiin käytöstä ohjelmallisesti label-tiedostoja hyväksikäyttäen kaikki kuvat, joissa oli useammat kuin yhdet kasvot. Käytännössä tämä tarkoitti sitä, että jos yhteen tiedostonimeen oli liitetty useampi ikätieto, ei kuvaa käytetty. Näin voitiin varmistua siitä, että label-tiedoston tiedot saatiin kerättyä kaikki samaan muotoon, ja että opetuksessa verkolle syötetään kuvia, joissa on vain yhdet kasvot. Kuvassa 3.5 on kuvakaappaus lopullisesta opetuksessa käytetystä annotaatiotiedostosta.

```
img278873.jpg;7;f
img662093.jpg;7;m
img980786.jpg;47;f
12078187746_6f26a262b1_o.jpg;35;f
12101664454_bd06b1358c_o.jpg;17;f
12101320335_881b4a4b40_o.jpg;10;m
image_1791.jpg;30;f
image_953.jpg;23;m
image_1801.jpg;23;f
image_2544.jpg;21;f
2015_01_17-145141.jpg;47;m
2015-01-17-145247.jpg;47;m
2015-01-26-164244.jpg;47;f
```

Kuva 3.5: Kuvakaappaus käytetystä annotaatiotiedostosta. Yhdellä rivillä on kuvan tiedostonimi, kuvassa olevan henkilön ikä ja sukupuoli.

Myös oman opetusmateriaalin kerääminen on mahdollista, joko keräämällä täysin uutta opetusmateriaalia tallentamalla ja annotoimalla järjestelmän tunnistamia ja rajaamia kasvoja, tai keräämällä ohjelmallisesti julkisesti saatavilla olevia kasvokuvia ja ikätietoja internetistä, esimerkiksi wikipediasta. Vaikka täysin oman opetusmateriaalin kerääminen olisi järjestelmän oppimisen kannalta paras vaihtoehto, on se ajallisesti hidasta, koska kuvat täytyy annotoida manuaalisesti. Tämän työn rajoissa on luotu mahdollisuus oman opetusmateriaalin keräämiselle, mutta sitä on käytetty opetuksessa varsin vähäisessä määrin, ja opetuksen painopiste on ollut valmiina saatavien materiaalien käytössä. Myöskään ohjelmallisesti kerättävään materiaaliin ei olla tässä vaiheessa keskitytty, vaikka sen mahdollisuus tiedostetaan.

Käytännössä opetus tapahtuu hyödyntämällä caffe-ohjelmistokehystä. Käytössä oleva opetusmateriaali jaetaan opetus -ja testausjoukkoihin siten, että 90 prosenttia kuvista käytetään opetukseen ja 10 prosenttia testaukseen. Lisäksi kaikki kuvat skaalataan uudelleen 256×256 pikselin kokoon. Ennen opetusta pitää määritellä vielä käytettävän verkon topologia, joka tässä tapauksessa on sama kuin Levin ja Hassnerin määrittelemässä konvoluutioverkossa [14]. Lisäksi luodaan vielä solvertiedosto, jossa määritellään muun muassa opetuksessa suoritettavien iteraatioiden lukumäärä, ja kuinka monen iteraation välein testiaineisto ajetaan verkon läpi.

3.6.2 SVR-luokittimen opetus

SVR-luokittimen opetuksessa käytetään samaa opetusmateriaalia kuin aiemmin mainitun hermoverkon opetuksessa. Erona on, että SVR-luokitin saa sisäänmenonsa hermoverkon viimeisen täysin kytketyn kerroksen ulostulosta 512-näytteisenä vektorina, ja sisäänmeno linkitetään SVR-luokittimen operaatioiden avulla haluttuun ulostuloon.

Luvussa 2.2.4 esitetty SVR-luokittimen matemaattinen toteutus onnistuu käytännössä helposti hyödyntämällä scikit-learn-alustaa [18], joka tarjoaa valmiin toteutuksen SVR-luokittimen käytölle. Toteutuksessa siis luodaan regressiomalli piirrevektorin ja halutun ulostulon, eli oikean iän välille.

Pienimmän virheen tuottava regressiomalli etsitään scikit-learn-alustan SVR-funktion avulla muuttamalla parametrin C arvoa. Muut parametrit pidetään oletusarvoisina.

Koko järjestelmän opetus tapahtuu siis kaksivaiheisesti. Aluksi opetetaan konvoluutiohermoverkko luokittelemaan kasvot tiettyihin ikäryhmiin. Tämän jälkeen opetetaan SVR-luokitin konvoluutiohermoverkon jatkeeksi siten, että otetaan konvoluutiohermoverkon viimeisen täysin kytketyn kerroksen ulostulo ja annetaan se SVR-luokittimen sisäänmenoksi ja haluttuna ulostulona käytetään luokkien sijasta tarkkaa ikää.

3.7 Opetusmateriaalin kerääminen

Kuten kohdassa 3.6.1 mainittiin, järjestelmään on toteutettu mahdollisuus tallentaa tunnistettuja kasvoja. Käytännössä tallennus tapahtuu siten, että ohjelmaa suorittava laite, kamera ja näyttöpääte sijoitetaan fyysisesti tilaan, jossa on ihmisiä. Näin järjestelmää testaavista ihmistä saadaan tallennettua kasvokuvat. Vaikka kuvat kerätään ainoastaan järjestelmän opetusta varten, on tärkeää että julkisia tiloja hyödynnettäessä varmistutaan kuvin tallentamisen laillisuudesta.

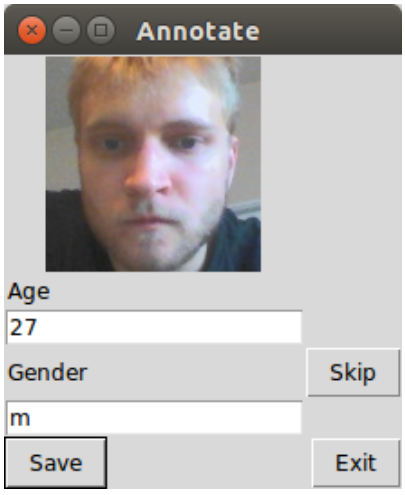
3.7.1 Tietojen tallennus

Ohjelmalla voidaan tallentaa DetectionThread-säikeen löytämiä kasvoja. Koska yhden henkilön kasvoista tehdään yleensä monta havaintoa, käytetään tallennusta silloin, kun käyttäjälle esitetään ikä -ja sukupuliarvio useamman kuvan keskiarvosta. Tallennus suoritetaan vasta keskiarvotuksen viimeiselle kasvokehykselle, ja näin vältetään se, että yhdestä käyttäjästä tallennettaisiin turhan monta kuvaa. Mikäli keskiarvotusta ei käytetä, voidaan sama tehdä esimerkiksi tallentamalla esimerkiksi joka kymmenes tehty kasvorajaus.

Kuvan tallentamisen suorittaa VideoThread-säie sen jälkeen, kun keskiarvotukseen tarvittavien kuvien lukumäärä on saavutettu. Kuvat tallennetaan jpg-formaatissa, ja kuvat nimetään yksilöllisesti käyttämällä aikaleimaa yksiselitteisessä muodossa `vuosi-kuukausi-päivä-tuntiminuuttisekunti.jpg` eli esimerkiksi `2016-01-15-123622.jpg`, täten vältetään, että useampia kuvia tallennettaisiin samalla nimellä.

3.7.2 Annotointi

Annotoinnilla tarkoitetaan tässä yhteydessä kuvien henkilöiden iän ja sukupuolen määrittämistä. Annotointi on tehtävä manuaalisesti, ja on täten työlästä kun tallennettuja kuvia on paljon. Kuvassa 3.6 on toteutetun annotointityökalun käyttöliittymä.



Kuva 3.6: Annotointityökalun käyttöliittymä.

Annotoinnissa siis täytetään käsin oikea ikä ja sukupuoli niille osoitettuihin kenttiin jokaiselle kuvalle. Työkalu esittää annotoijalle kuvia tallennettujen kuvien kansiosta ja annotoija syöttää kenttiin iän ja sukupuolen. Sallittuja arvoja iäksi on positiivinen kokonaisluku välillä 0–100 ja sukupuoleksi m *(male)*, f *(female)* ja u *(unidentified)*. Mikäli tallennettuihin kuviin on päätynyt kuva, jossa kasvot ovat epäselvät

tai ei ole kasvoja ollenkaan, voidaan kuva siirtää hylättyjen kuvien kansioon Skip-painikkeella. Annotoinnin onnistuessa kuva siirretään Save-painikkeella tallennettujen kuvien kansioon. Samalla tallennetaan annotointitiedot tekstitiedostoon muodossa `tiedostonimi;ikä;sukupuoli`. Näin annotoituja kuvia voidaan myöhemmin käyttää verkkojen opetusmateriaalina.

Koska kuvia on yleensä paljon ja annotointi on manuaalista, on käyttöliittymää optimoitu siten, että yhden kuvan annotointi olisi mahdollisimman nopeaa. Kohdistin sijaitsee oletusarvoisesti iänsyöttökentässä, tabulaattorilla voidaan siirtyä suoraan sukupuolensyöttökenttään ja näppäimistön Enter-näppäin on linkitetty save-painikkeeseen. Tallennuksen jälkeen kentät tyhjentyvät ja kohdistin siirtyy takaisin iänsyöttökenttään. Näin yksi annotointi voidaan suorittaa keskimäärin viidellä näppäimenpainalluksella.

4. TULOKSET

Seuraavaksi esitellään työllä aikaan saatuja tuloksia. Mittareina tarkastellaan iän- ja sukupuolenarvioinnin tarkkuutta ja järjestelmän suorituskykyä reaaliaikavaatimusten toteutumiseksi. Tutkimuksen painopiste oli reaaliaikavaatimuksen toteutumisessa ja iän arvioinnin tarkkuudessa.

4.1 Iän arviointi

Iän arvioinnissa käytettiin modifioitua versiota Levin ja Hassnerin määrittelemästä konvoluutiohermoverkosta [14], jossa luokittelu eri ikäryhmiin oli korvattu määrittämällä tarkka ikäarvio SVR-luokittimen avulla. Molemmat opetettiin käytössä olevalla opetusaineistolla. Hermoverkon opetus suoritettiin ensin, ja tämän jälkeen opetettiin SVR-luokitin samalla opetusjoukolla, mutta varsina syötteinä käytettiin hermoverkon viimeisen täysin kytketyn kerroksen 512-näytteistä ulostuloa. Iän arvioinnissa käytettyjen hermoverkkojen, sekä alkuperäisen luokitteluun käytetyn verkon, että modifioidun tarkan ikäarvon antavan verkon lopulliset rakenteet on esitetty kuvassa 4.3.

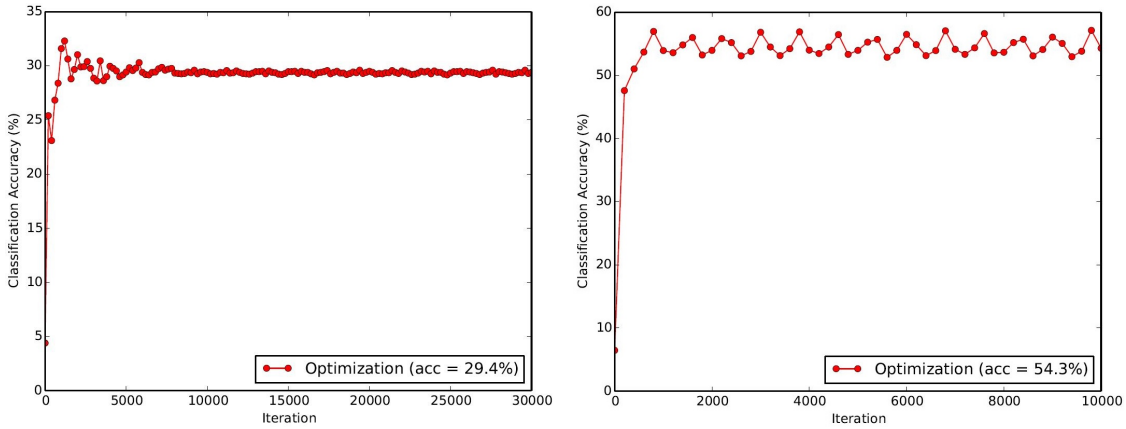
4.1.1 Luokitin

Luokitinta yritettiin opettaa useampaan otteeseen käyttäen opetuksessa erisuuruisia opetusaineistoja. Kuvassa 4.1 on vasemmanpuoleisessa kuviossa käytetty opetusaineistona FG-NET- ja ChaLearn-aineistoja [4] [1], ja oikeanpuoleisessa kuviossa on käytetty edellä mainittujen lisäksi myös Adience-tietokantaa [3] ja itsetallennettuja kuvia.

FG-NET- ja ChaLearn-aineistoissa kuvia oli yhteensä 3 478. Adience-tietokannasta valittiin käytettäväksi 7 215 kuvaa ja itsetallennettuja kuvia 469. Kokonaisuudessaan Adience-tietokannassa olisi ollut huomattavasti enemmän kuvia, mutta käyttöön valittiin ainoastaan kuvat, joissa oli vain yhdet kasvot.

Kuvioista voidaan tehdä johtopäätökset, että opetusmateriaalin määrän lisääminen on vaikuttanut positiivisesti luokittimen oikeinluokittuneiden näytteiden osuuteen. Myös suurta iteraatioiden lukumäärää on testattu, mutta näyttäisi siltä, että iteraatioiden lukumäärän kasvattaminen muutamaa tuhatta suuremmaksi ei paranna luokittimen tarkkuutta. Toisaalta liian suuri iteraatioiden määrä voi johtaa ylioppimiseen, joten käyttöön valittiin oikeanpuoleisen kaavion 2 000 iteraatiokierroksen

kohdalla opetettu verkko.

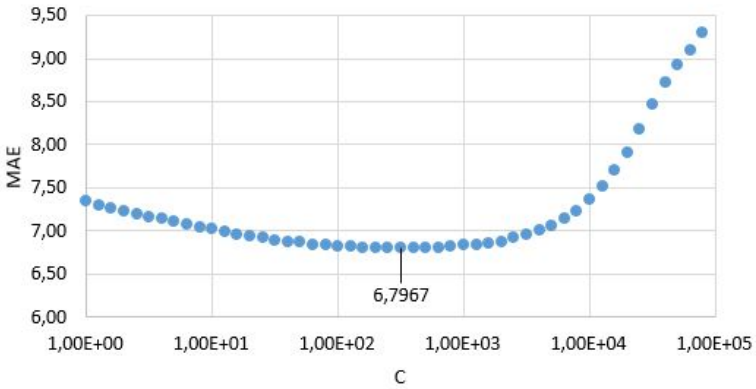


Kuva 4.1: Kaaviot verkkojen opetuksen tuloksista. Vasemmalla pienemmällä opetusmateriaalilla opetettu verkko, ja oikealla suuremmalla opetusmateriaalilla opetettu.

Kun analysoidaan itseopetetun verkon tarkkuutta, voidaan sanoa että on päästy samoihin tarkkuuslukemiin kuin Levin ja Hasnerin testeissä, jossa oikein luokittuneiden näytteiden osuus oli 50,7 $\pm$ 5,1 prosenttia [14]. Näin ollen voidaan olettaa, että ollaan saavutettu kyseisen verkon maksimaalinen suorituskyky, ja voidaan siirtyä opettamaan SVR-luokitinta, jolla saavutetaan numeeriset arvot luokkien sijasta.

4.1.2 SVR-luokitin

Kuten aiemmin todettiin, suoritetaan SVR-luokittimen avulla regressiomallin opetus 512-näytteisen vektorin ja oikean iän välille. Opetuksessa käytetään scikit-learn-alustan SVR-funktioita [18]. Muista Scikit-learn-alustan tarjoamista malleista testattiin SVR polynomisella kernelillä, LinearSVR, AdaBoostRegressor, GradientBoostingRegressor ja RandomForestRegressor. Normaali SVR-luokitin tuotti testeissä parhaan tuloksen ja sille suoritettiin optimointi.



Kuva 4.2: Regressiomallin virheen minimointi muuttamalla parametrin C arvoa.

Regressiomallin optimoinnissa ainoastaan SVR-funktion parametria C muutettiin ja loput parametrit pidettiin oletusarvoisina. Virhemetriikkana käytettiin keskivirhettä *(Mean Absolute Error)*.

Regressiomalli opetettiin parametrin C arvoilla $C = 10^a$, jossa a saa arvot väliltä $1 \leq a \leq 5$, jossa a:n arvoa kasvatetaan 0,1 suuruisilla askelilla. Optimointiprosessi on nähtävissä kuvasta 4.2. Pienin keskivirhe saavutettiin parametrin C arvolla $C = 10^{2,5}$ eli $C = 3{,}162 \times 10^2$. Tällöin keskivirhe oli 6,8 vuotta.

Taulukossa 4.1 on esitelty SVR-luokittimen testiaineistolle tuottamat keski-iät ja keskimääräiset virheet ikäryhmittäin. Vasemmanpuoleisessa sarakkeessa ovat ikäryhmät, keskimmäisessä sarakkeessa on kustakin ikäryhmästä tehtyjen arviointien keski-ikä ja oikeanpuoleisessa sarakkeessa on kunkin ikäryhmän keskimääräinen virhe.

Taulukko 4.1: SVR-luokittimen tuottama keskivirhe ja keski-ikä ikäryhmittäin.

Ikäryhmä	Ryhmän keski-ikä	MAE
60–	43	34.63
50–59	36	16.39
40–49	33	10.27
30–39	31	6.04
20–29	28	5.04
15–19	23	7.78
10–14	15	6.69
5–9	8	4.02
0–4	3	2.72

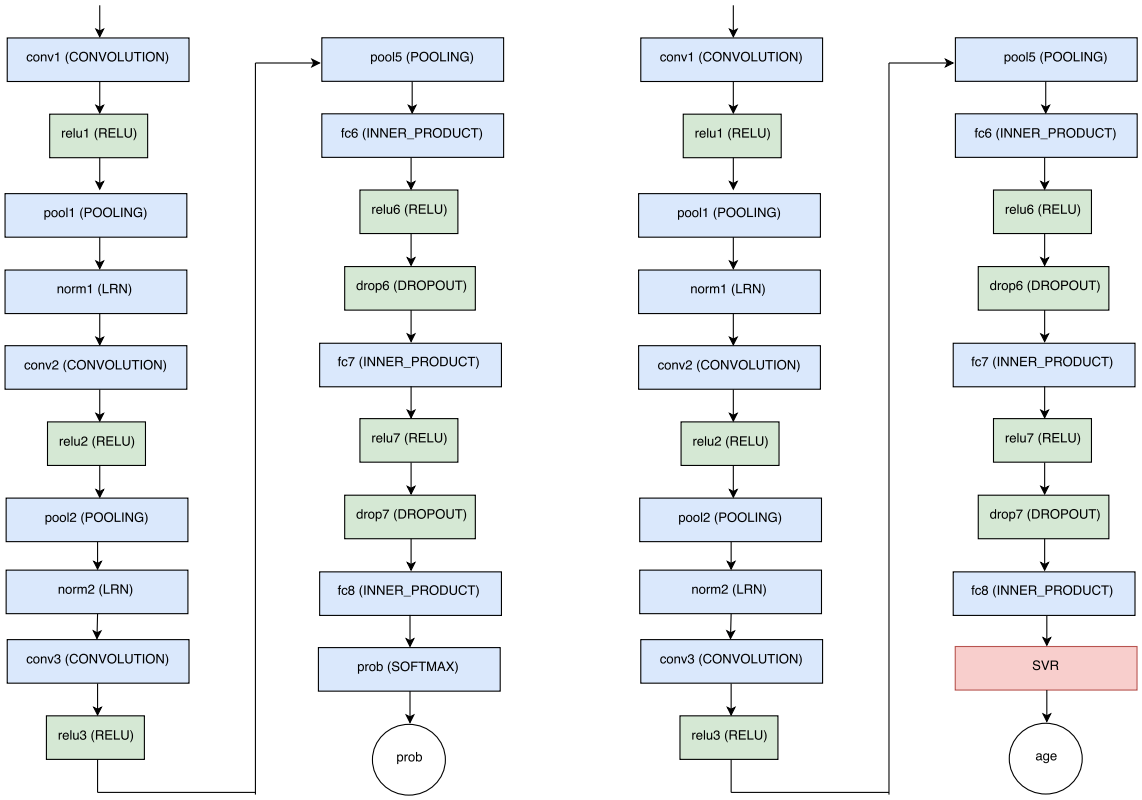
Taulukosta voidaan tehdä kaksi johtopäätöstä. Ensinnäkin ryhmien keski-ikä on nousevassa ja ennen kaikkea oikeassa järjestyksessä, joten voidaan olettaa että järjestelmä pystyy keskimäärin erottamaan vanhemman henkilön nuoremmasta henkilöstä. Toiseksi voidaan huomata, että ikäryhmien 40–49, 50–59 ja 60– keskivirhe on selkeästi koko järjestelmän keskivirhettä (6,8 vuotta) korkeampi. Koska kyse on hermoverkkojen opettamisesta, syy korkeampien ikien suureen keskivirheeseen jää osittain pimentoon. Eräänä syynä voisi kuitenkin olla, että koska opetusmateriaali on pääsääntöisesti kerätty internetin sosiaalisen median palveluista, on vanhempien ihmisten osuus opetusjoukosta selvästi vähäisempi, ja täten opetuksen pääpaino olisi siirtynyt nuorempien ihmisten kohdalle.

Tuloksia arvioidessa voidaan koko järjestelmän keskivirhettä pitää kohtalaisen hyvänä, koska tarkan iän arvioiminen valokuvasta ei ole välttämättä ihmisellekään kovin helppo tehtävä. Ihminen ei kuitenkaan tee selviä virheitä, kuten tässä toteutuksessa on käynyt korkeampien ikien kohdalla. Vertailun vuoksi *LAP challenge on apparent age estimation* kilpailun voittajan keskivirhe oli 3,2 vuotta [17]

4.2 Sukupuolen arviointi

Sukupuolen arvioinnin opetusta ei suoritettu erikseen omalla opetusdatalla. Koska verkon rakennetta ei muutettu mitenkään, ja Levin ja Hassnerin testeissä käytetty opetusmateriaali oli laajempi kuin oma käytössä ollut opetusmateriaali, voidaan tehdä oletus, että opetuksessa ei olisi päästy parempiin tuloksiin opettamalla verkkoa uudestaan. Lopullisiksi tuloksiksi siis jäivät Levin ja Hassnerin 86,8 ± 1,4 prosentin oikein luokitelluiden näytteiden osuus [14]. Käytetyn verkon rakenne on esitetty kuvassa 4.3.

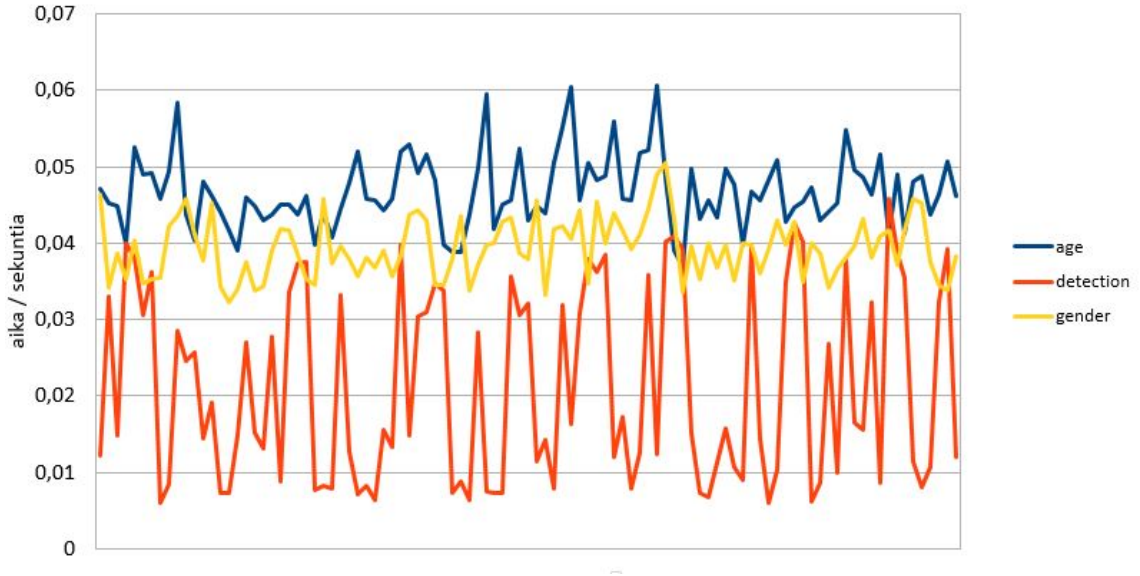
Levin ja Hassnerin arvion mukaan suuri osa väärin luokitelluista näytteistä on kuvia vauvoista tai todella nuorista lapsista, joilla selvät sukupuolen määräävät ominaisuudet eivät vielä ole kehittyneet erotettaviksi asti [14]. On vaikeaa arvioida miten tulokset olisivat muuttuneet, jos opetusmateriaalista olisi poistettu pienten lasten osuus. Kuitenkin jos tarkastellaan 86,8 ± 1,4 prosentin oikein luokitelluiden osuutta, voidaan todeta, että ei olla päästy samalle tasolle ihmisen kanssa, joka todennäköisesti osaa määrittää sukupuolen joitakin poikkeuksia lukuun ottamatta lähes 100 prosenttisella tarkkuudella, mutta toisaalta taas saavutettu tarkkuus on huomattavasti silkkaa arvausta parempi.



Kuva 4.3: Käytettyjen verkkojen rakenteet. Vasemmalla sukupuolen tunnistuksessa ja iän arvioinnissa luokittelutulokset antava verkko. Oikealla lopullinen iän arvioinnissa käytetty modifioitu verkko, joka antaa tarkan ikäarvion.

4.3 Suorituskyky

Työn varsinaisena tavoitteena oli tuottaa järjestelmä, joka toteuttaa iän- ja sukupuolen arvioinnin reaaliaikaisesti. Saavutettujen tulosten toisena tärkeänä kohteena on täten järjestelmän suorituskyky.



Kuva 4.4: Järjestelmän eri osuuksien laskenta-ajat grafiikkaprosessoria hyödyntäen. Y-akselilla yhteen operaatioon kulunut aika sekunteina ja x-akseli kuvaa ohjelman ajamisen aikana suoritettuja yksittäisiä operaatioita sadan operaation ajan.

Kuvassa 4.4 on esitetty sadan operaation verran testausalustalla suoritettua ja kellotettua ohjelman ajoa. GenderDetectionThread-säie suoriutuu omasta tehtävästään selvästi nopeimmin ja AgeRecognitionThread-säikeen suorittamiseen kuluu eniten aikaa. Säikeiden keskimääräiset suoritusajat on nähtävissä taulukosta 4.2.

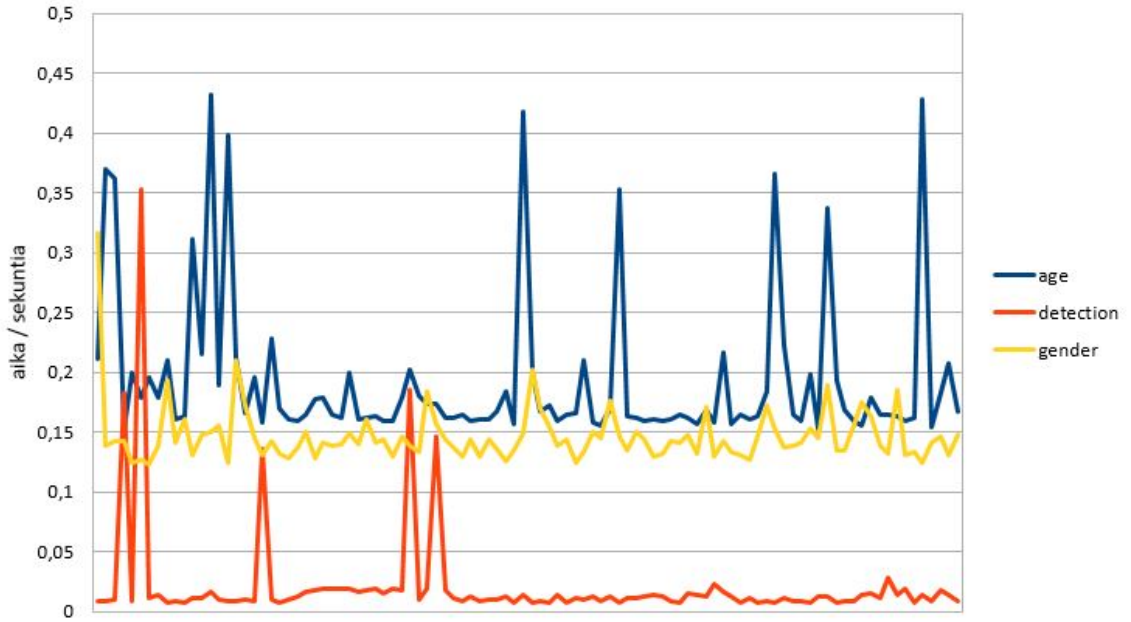
Taulukko 4.2: Järjestelmän eri osien keskimääräiset suoritusajat.

Prosessi	Keskimääräinen suoritusaika
DetectionThread	0,021 s
GenderRecognitionThred	0,039 s
AgeRecognitionThread	0,047 s

Koska GenderRecognitionThread- ja AgeRecognitionThread-säikeet voidaan suorittaa rinnakkaisuuden vuoksi samanaikaisesti, järjestelmän todellinen yhteen kokonaiseen operaation käyttämä aika on keskimäärin 0,021 + 0,047 = 0,068 sekuntia. Taajuudeksi muunnettuna järjestelmä voi suorittaa keskimäärin 14,7 kasvojenetsintä, iän arviointi ja sukupuolen arviointioperaatioita sekunnissa.

On tärkeää huomata, että edellä mainitut laskenta-ajat on saavutettu käyttämällä

grafiikkaprosessorilaskentaa. Caffe-ohjelmistokehys mahdollistaa grafiikkaprosessorin käytön hermoverkkojen ajamisessa, joten GenderRecognitionThread- ja AgeRecognitionThread-säikeet suoritetaan grafiikkaprosessorilla. DetctionThread-säie suoritetaan normaalisti tietokoneen prosessorilla.



Kuva 4.5: Järjestelmän eri osuuksien laskenta-ajat ilman grafiikkaprosessoria. Y-akselilla yhteen operaatioon kulunut aika sekunteina ja x-akseli kuvaa ohjelman ajamisen aikana suoritettuja yksittäisiä operaatitoita sadan operaation ajan. Huomaa y-akselin skaala verrattuna aikaisempaan kuvaan.

Mikäli grafiikkaprosessoria ei voida hyödyntää, laskenta-ajat moninkertaistuvat. Tämä on visualisoitu kuvassa 4.5 ja säikeiden keskimääräiset suoritusajat on listattu taulukossa 4.3.

Taulukko 4.3: Järjestelmän eri osien keskimääräiset suoritusajat.

Prosessi	Keskimääräinen suoritusaika
DetectionThread	0,022 s
GenderRecognitionThred	0,147 s
AgeRecognitionThread	0,194 s

DetectionThread-säikeen suorittamiseen kuluva aika pysyy käytännössä samana, mutta hermoverkkojen ajamiseen kuluva aika nelinkertaistuu. Tämä aiheuttaa taajuuden putoamisen 4,6 suoritettuun tunnistus- ja arviointioperaatioon sekunnissa. Teoriassa nopeuden putoaminen ei kuulosta pahalta, mutta käytännössä prosessorin kuormittaminen aiheuttaa sen, että VideoThread-säie ei saa enää tarpeeksi suoritusaikaa. Videon esittäminen muuttuu jaksottaiseksi eli videokuva ei näytä enää jatkuvalta, vaan etenee hyppäyksittäin.

Yhteenvetona voidaan todeta, että grafiikkaprosessorilaskenta on lähes pakollista mielekkään käyttäjäkokemuksen luomiseksi. Se mahdollistaa videokuvan ja tulosten esittämisen siten, että se näyttää reaaliaikaiselta käyttäjän näkökulmasta. Täten voidaan todeta, että järjestelmän reaaliaikaisuusvaatimus onnistuttiin toteuttamaan.

5. JATKOKEHITYS

Koska kaikki järjestelmän päätehtävät on toteutettu modulaarisesti erillisissä komponenteissa, helpottaa se järjestelmän mahdollista jatkokehitystä. Komponentit voidaan yksinkertaisesti korvata paremmilla komponenteilla, kunhan ne suorittavat saman tehtävän. Suorituskyvyltään parempia komponentteja valitessa tulee kuitenkin muistaa järjestelmän reaaliaikaisuuden asettamat vaatimukset, joten laskennallisesti liian raskaita komponentteja tulee välttää.

Järjestelmään voidaan myös luoda uusia toiminnallisuuksia uusien komponenttien avulla. Periaatteessa järjestelmään voidaan lisätä mikä tahansa automaattisen kasvoanalyysin osa-alue, yhtenä esimerkkinä on vaikkapa ilmeiden tunnistaminen eli esimerkiksi hymyileekö kuvassa ole henkilö.

Uusien toiminnallisuuksien lisääminen onnistuu periaatteessa helposti. Komponentti voidaan lisätä uudeksi säikeeksi GenderRecognitionThread- ja AgeRecognitionThread-säikeiden rinnalle. Koska järjestelmässä on hyödynnetty rinnakkaisuutta, uuden komponentin lisääminen ei vaikuttaisi merkittävästi myöskään yksittäisen prosessin suoritusaikaan, ja täten myös reaaliaikavaatimus täyttyisi.

Käytännössä uuden komponentin lisääminen onnistuu helposti, koska voidaan hyödyntää nykyistä ohjelmarakennetta. Rinnakkaisuudessa vaadittavassa vapautuvien resurssien odotuksessa kyetään hyödyntämään samaa rakennetta kuin jo olemassa olevissa säikeissä. Varsinkin hermoverkkoja käyttäviä uusia komponentteja lisätessä voidaan hyödyntää olemassa olevia komponenttirakenteita, sillä esimerkiksi AgeRecognitionThread ja GenderRecognitionThread eroavat toisistaan lähinnä luokan rakentajassa tehtävän verkon alustuksen ja lopuksi verkolta ulostulona saatavien arvojen käsittelyn kohdalla.

Mahdollisina järjestelmän suorituskykyä parantavina komponentteina voisivat olla komponentit, joilla iän- ja sukupuolenarvioinnin oikeinluokittuneiden kuvien osuutta saataisiin parannettua. Eräänä vaihtoehtona olisi kasvojen etsinnän yhteyteen liitettävä kasvojen yhdensuuntaistaminen *(face alignment)*. Näin kyettäisiin varmistumaan siitä, että hermoverkoille luokitukseen menevät kuvat ovat aina saman tyyppisiä eli edestä päin otettuja suoria kasvokuvia *(frontal face image)*. Tämä mahdollistaisi myös sen, että opetuksessa voitaisiin käyttää saman tyyppisiä kuvia, mikä vähentäisi luokitteluongelman kompleksisuutta. Myös silmälasit tuntuivat käytännön testeissä aiheuttavan järjestelmälle jonkin verran ongelmia, joten sekin olisi

potentiaalinen kohde jatkotutkimuksille.

Toisena mahdollisena kehityskohteena on iän arviointiin käytetyn komponentin korvaaminen esimerkiksi *LAP challenge on apparent age estimation* -kilpailun voittaneella järjestelmällä [17]. Toteutuksessa konvoluutiohermoverkkona on käytetty VVG-16 arkkitehtuurin 16-kerroksista verkkoa, joka koostuu 13 konvoluutiokerroksesta ja kahdesta 4 096 neuronin täysin kytketystä kerroksesta. Regression sijaan tämä verkko antaa ulostuloina 101 eri luokkaa, jotka siis vastaavat suoraan ikiä väliltä 0–100 vuotta.

Kuten kuvauksesta on huomattavissa, kyseinen verkko on huomattavasti raskaampi ajettava kuin tässä toteutuksessa käytetty verkko, ja se vaatii enemmän laskentatehoa. Testausalustana käytettiin Nvidia 5200M näytönohjainta 1 GB:n muistilla. Verkko oli kuitenkin liian raskas ajettava ja näytönohjaimen muisti loppuu kesken. Jos ajamista yritetään ilman grafiikkaprosessorilaskentaa, jumittaa se prosessorin täysin ja järjestelmä toimii todella hitaasti. Ajaminen kuitenkin onnistuu myös testauksessa käytetyllä alustalla, mutta se vaatii lähes kaikkien muiden testausalustalla tapahtuvien prosessien alasajon. Suositeltavaa siis olisi käyttää kyseisen verkon kanssa tehokkaampaa näytönohjainta vähintään 2 GB:n omalla muistilla.

Komponentin käyttöönotto on kuitenkin kannattavaa sillä se on opetettu huomattavasti laajemmalla opetusmateriaalilla, joka sisältää yli 500 000 ohjelmallisesti Wikipediasta ja IMDb:stä *(Internet Movie Database)* kerättyä kasvokuvaa. Kyseisellä menetelmällä on testeissä saavutettu 3,2 vuoden keskimääräinen virhe, joka on selkeästi parempi kuin järjestelmän tämänhetkinen 6,8 vuoden keskimääräinen virhe.

6. YHTEENVETO

Tässä diplomityössä toteutettiin automaattisen kasvoanalyysin järjestelmä, joka arvioi käyttäjän iän ja sukupuolen reaaliaikaisesti videokameran välityksellä. Järjestelmä on suunniteltu hyödyntäen tietovuoarkkitehtuuria ja komponenttien toimintojen rinnakkaista suoritusta. Videokuvaa hallinnoiva komponentti vastaa tietojen säilyttämisestä ja esittämisestä käyttäjälle, ja rinnakkaisuuden hallinnasta. Muut komponentit toimivat rinnakkain. Ne pyytävät videokuvakomponentilta tietoja ja tiedot saatuaan suorittavat omat operaationsa, joita ovat kasvojen etsintä, iän arviointi ja sukupuolen arviointi. Komponenteista koostuva järjestelmä antaa hyvät mahdollisuudet jatkokehitykselle, sillä komponentit ovat helposti korvattavissa, vastaavan tehtävän suorittavilla, paremmilla komponenteilla. Myös uusia ominaisuuksia tuovien komponenttien lisääminen onnistuu helposti ja järjestelmän rinnakkaisen luonteen vuoksi uusilla komponenteilla ei ole merkittävää vaikutusta ohjelman suoritusaikaan.

Työn taustalla olevina teorioina käytettiin kasvojen etsinnässä Paul Violan ja Michael Jonesin *Rapid Object Detection using a Boosted Cascade of Simple Features* -tutkimusta [20] ja iän ja sukupuolen arvioinnissa Gil Levin ja Tal Hassnerin *Age and Gender Classification using Convolutional Neural Networks* -tutkimusta [14]. Teorioiden hyvinä puolina ovat kasvojen etsinnässä minimoitu laskenta-aika säilyttäen korkea tarkkuus tunnistuksessa. Iän ja sukupuolen arvioinnissa konvoluutiohermoverkot mahdollistavat sen, että opetusaineistona ei tarvitse käyttää laboratorio-olosuhteissa otettuja kuvia vaan voidaan käyttää myös internetissä saatavilla olevia kuvajoukkoja ja luokitustulokset pysyvät silti korkeina.

Sukupuolen arvioinnissa Levin ja Hassnerin järjestelmää käytettiin sellaisenaan, mutta iän arvioinnissa järjestelmää modifioitiin liittämällä konvoluutiohermoverkon jatkeeksi SVR-luokitin *(Support Vector Regression)*. Alkuperäisessä toteutuksessa ikää arvioitiin ikäryhmittäin eli konvoluutiohermoverkon ulostulona saatiin todennäköisyydet siitä, kuinka todennäköisesti verkon läpi ajettu kuva kuuluu mihinkin ennalta määrättyyn ikäryhmään. Regression avulla saatiin ikäryhmien sijasta tarkka ikäarvio ja käytännössä tämä onnistui syöttämällä konvoluutiohermoverkon viimeisen täysin kytketyn kerroksen ulostulo SVR-luokittimelle.

SVR-luokittimen käyttö vaatii konvoluutiohermoverkon opettamista uudelleen, ja tämä suoritettiin käyttämällä noin 11 000 kuvaa. Opetus tapahtuu kaksiosaisesti.

Ensin opetetaan konvoluutiohermoverkko uudelleen. Tämän jälkeen opetetaan SVR-luokitin samalla opetusaineistolla käyttämällä varsinaisina opetusnäytteinä konvoluutiohermoverkon viimeisen täysikytketyn kerroksen ulostuloa ja tavoitearvoina kuvien henkilöiden tarkkaa ikää. Näin voidaan luoda regressiomalli sisäänmenon ja halutun ulostulon välille.

Järjestelmän lopullisina arviointitarkkuuksina saatiin iän tunnistuksessa 86,8 ± 1,4 prosentin oikeinluokittuneiden näytteiden osuus. Iän arvioinnissa käytettiin mittarina keskimääräistä virhettä *(Mean Absolute Error)*, joka tälle järjestelmälle oli 6,8 vuotta. Jos tulosta verrataan tämän hetken parhaimpiin kuuluvan järjestelmän keskimääräiseen virheeseen, 3,2 vuotta, voidaan tämän järjestelmän arviointitarkkuutta pitää kohtalaisena.

Työn varsinaisena tavoitteena oli toteuttaa järjestelmä, joka suoriutuu annetuista tehtävistä reaaliaikaisesti. Grafiikkaprosessoria hyödyntämällä saatiin paras suorituskyky ja tällöin operaatioiden keskimääräiset laskenta-ajat olivat taulukon 6.1 mukaiset.

Taulukko 6.1: Järjestelmän eri osien keskimääräiset suoritusajat.

Prosessi	Keskimääräinen suoritusaika
DetectionThread	0,021 s
GenderRecognitionThred	0,039 s
AgeRecognitionThread	0,047 s

Kun muistetaan ohjelman rinnakkainen suoritusjärjestys, on todellinen operaatioiden suorittamiseen kulunut aika sama kuin kasvojen etsintään ja hitaampaan arviointioperaatioon kulunut aika eli 0,021 + 0,047 = 0,068 sekuntia, joka on taajuudeksi muunnettuna 14,7 tunnistusoperaatiota sekunnissa. Täten voidaan todeta, että reaaliaikavaatimus saavutettiin.

REFERENCES

[1] Chalearn Looking at People Apparent age estimation dataset [Online]. Saatavilla: `http://gesture.chalearn.org/`

[2] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman. Return of the devil in details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531, 2014.

[3] E. Eidinger, R. Enbar, T. Hassner. Age and Gender Estimation of Unfiltered Faces. Transactions on Information Forensics and Security (IEEE-TIFS), special issue on Facial Biometrics in the Wild, Volume 9, Issue 12, pp. 2170–2179, December 2014.

[4] FG-NET Aging Database [Online]. Ei saatavilla.

[5] Y. Freund, R.E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55, 1, pp. 119–139, 1996.

[6] I. Haikala, H.-M. Järvinen. Käyttöjärjestelmät. Talentum, 2003. ISBN 951-762-837-4

[7] K. Herrala. Kasvojen paikannus. Kandidaatintyö. Tampereen teknillinen Yliopisto. Signaalinkäsittelyn laitos, 2009.

[8] H. Huttunen. Signaalinkäsittelyn perusteet. Tampereen teknillinen yliopisto. Signaalinkäsittelyn laitos. Opetusmoniste, 2014:1.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint arXiv:1408.5093, 2014.

[10] Kaggle [Online], Saatavilla: `www.kaggle.com`

[11] A. Krizhevsky, I. Sutskever, G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Neural Inform. Process. Syst., pp. 1725–1732, 2012.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, R. Hubbard, L. D. Jackel. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems 2, 1990.

[13] J. Lehmusvaara. Konvoluutioneuroverkot kirjain- ja numeromerkkien tunnistuksessa. Kandidaatintyö. Tampereen teknillinen Yliopisto. Signaalinkäsittelyn laitos. 2014

[14] G. Levi, T. Hassner. Age and Gender Classification Using Convolutional Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) workshops. June, 2015.

[15] R. Lienhart, J. Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, pp. 900–903, September, 2002.

[16] OpenCV 2.4.12.0 Documentation [Online].
Saatavilla: `http://docs.opencv.org/2.4/index.html`

[17] R. Rothe, R. Timofte, L. Van Gool. DEX: Deep EXpectation of apparent age from a single image. ICCV, ChaLearn Looking at People workshop, December, 2015.

[18] scikit-learn, Machine Learning in Python [Online].
Saatavilla: `http://scikit-learn.org/stable/`

[19] A. Smola, B. Schölkopf. A tutorial on support vector regression, 2004.

[20] P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 511–518, 2001.