Tampereen yliopisto

Santeri Koskinen

PUULAJITUNNISTUS KONVOLUUTIONEUROVERKOILLA

Diplomityö
Informaatioteknologian ja viestinnän tiedekunta
Tarkastaja: Professori Ari Visa
Tarkastaja: Diplomi-insinööri Timo Rouvinen
Kesäkuu 2021

TIIVISTELMÄ

Santeri Koskinen: Puulajitunnistus konvoluutioneuroverkoilla
Diplomityö
Tampereen yliopisto
Tietotekniikan DI-ohjelma
Kesäkuu 2021

Tässä diplomityössä tutkitaan neuroverkon parametrien sekä koulutusaineiston laadun vaikutusta automaattiseen puulajitunnistukseen. Tutkimusta varten työssä koulutetaan puun runkojen kuvia ja metadataa käyttävä konvoluutioneuroverkko tunnistamaan näytteitä neljän Suomen metsissä yleisesti esiintyvän puulajin väliltä. Neuroverkon koulutusta ja tulosten mittaamista varten käytetään yksityistä 23 600 runkokuvasta per puulaji koostuvaa aineistoa.

Tuloksena saadaan 72,5 % luokittelutarkkuus testiaineistolle. Konvoluutioneuroverkon syötteenä toimivien kuvien koon kasvatus ja metadatan käyttö todetaan luokittelutulosta parantaviksi menetelmiksi. Kuvien esiprosessointimenetelmistä kuvien skaalaaminen välille [0, 1] osoittautuu toimivimmaksi menetelmäksi. Koulutusaineistosta huonojen kuvien poistolla ei saavuteta merkittävää parannusta luokittelutulokseen.

Avainsanat: neuroverkko, puu, konvoluutio, runko

ABSTRACT

Santeri Koskinen: Tree species recognition with convolutional neural networks
Master's Thesis
Tampere University
Master's Degree Programme in Information Technology
June 2021

This master's thesis researches the impact neural network parameters and training data quality have on automatic tree species recognition. For this research, a convolutional neural network using images of tree trunks and their metadata is trained to classify samples between four different tree species commonly found in Finnish forests. A private dataset consisting of 23,600 tree trunk images for each tree species is used for training the net and validating the results.

A 72.5 % classification accuracy for the testing dataset is achieved. Increasing the size of the images used by the convolutional neural network as well as the use of metadata are found to increase classification accuracy. For the image preprocessing methods, scaling images to range [0, 1] is found out to be the most effective method. The removal of bad images from the training dataset does not increase the classification accuracy substantially.

Keywords: neural network, tree, convolution, trunk

ALKUSANAT

Kiitän työn ohjaajana ja tarkastajana toiminutta professori Ari Visaa asiantuntevista kommenteista työn eri vaiheissa sekä joustavasta aikataulusta ja ohjauksesta näin etätyöaikana. Kiitokset myös Tampereen yliopistolle laadukkaasta kurssitarjonnasta ohjelmistotekniikan ja signaalinkäsittelyn aloilta.

Lisäksi haluan kiittää Trestima Oy:tä työn aiheen tarjoamisesta. Erityisesti kiitän työn ohjaajana ja tarkastajana toiminutta Timo Rouvista sekä ohjelmistoarkkitehti Antti Kivimäkeä käytännön toteutuksen ohjaamisesta ja ideoista.

Tampereella, 28. kesäkuuta 2021

Santeri Koskinen

SISÄLLYSLUETTELO

LYHENTEET JA MERKINNÄT

CNN	engl. Convolutional Neural Network, konvoluutioneuroverkko
GPS	engl. Global Positioning System, satelliittidataa käyttävä paikannusjärjestelmä
HSV-avaruus	väriavaruus, jossa erilliset värikanavat sävylle (hue), värikylläisyyden (saturation) ja kirkkauden arvolle (value)
JSON	engl. JavaScript Object Notation, yleinen tiedostomuoto, jossa tieto on esitetty avain–attribuutti-pareina
MLP	engl. Multilayer Perceptron, neuroverkkojen alalaji, jossa perseptroneja on kytketty peräkkäisiin kerroksiin
RGB-kuva	värikuva, joka koostuu punaisesta (red), vihreästä (green) ja sinisestä (blue) kanavasta
px	pikseli eli kuvapiste

1. JOHDANTO

Puiden automaattinen tunnistaminen tekee nopeammaksi, helpommaksi ja halvemmaksi selvittää, mitä puita metsässä sijaitsee verrattuna ihmisasiantuntijan manuaaliseen tarkastustyöhön. Metsäalueen puiden lajijakauman selvittäminen on tärkeää, sillä puulajien markkinahinnat poikkeavat toisistaan merkittävästi. Esimerkiksi kuusitukeista on maksettu vuonna 2019 Suomessa keskimäärin jopa neljäsosan enemmän kuutiometriltä kuin koivutukeista (1). Metsien tarvitsemat hoitotoimenpiteet riippuvat myös puulajeista ja niiden määrästä (2). Metsäalueen puiden lajijakauman selvittäminen onkin siis metsäalueen hinnan muodostamiseen oleellinen tieto. Aihe on erityisen tärkeä Suomessa, sillä noin 75 % Suomen pinta-alasta on metsää (3). Lisäksi metsäteollisuuden tuotteet ovat merkittävässä asemassa Suomen vientituotteita tarkasteltaessa, muodostaen yli 20 % vientituotteiden kokonaisarvosta (3).

Puiden automaattista tunnistamista varten työssä koulutetaan konvoluutioneuroverkko, joka ottaa syötteenään käyttäjän antaman kuvan puun rungosta sekä muuta tekstimuotoista dataa kuvanottohetkestä, ja antaa tuloksenaan puulajin nimen, jota syötteen kuva järjestelmän mukaan edustaa. Työssä tutkitaan neljän yleisimmän suomalaisen puulajin tunnistamista sekä osassa tutkimuksissa käytetään lisälajia Suomen ulkopuolelta, jolloin mahdollisia luokkia on viisi. Työssä käytettävät kuvat on otettu tavallisella kännykkäkameralla.

Yhdentoista itävaltalaispuulajin tunnistamisessa ihmisasiantuntija on saavuttanut 77,8 % luokittelutarkkuuden puun runkojen ja neulasten kuvista tunnistamalla (4). Tässä työssä ei ole käytössä neulasten kuvia, mutta tunnistettavia luokkia on vähemmän. Jotta koulutettavalla järjestelmällä olisi käyttöä, tulisi luokittelun onnistua edellä mainitun ihmisasiantuntijan lailla.

Puulajien tunnistamista rungoista on tutkittu aikaisemmin konvoluutioneuroverkoilla (5) ja muilla koneoppimismetodeilla (6). Tässä työssä tutkimuskysymykseksi nouseekin, voidaanko runkoja tunnistaa luotettavasti, kun tietoaineisto koostuu helposti kerättävistä, mutta aikaisempiin tutkimuksiin verrattuna huonolaatuisista kännykkäkameran kuvista. Isoa puualuetta kuvattaessa puulajien inventaarion tekeminen on aikaa vievää työtä, jolloin inventaarion tekemistä nopeuttaisi mahdollisuus käyttää epätarkempia, usean puun sisältävää kuvaa yksittäisten puiden tarkan kuvaamisen sijaan.

Tutkimuksen päätavoite on selvittää, minkälainen tietoaineiston käsittely tuottaa korkeimman luokittelutuloksen. Ongelmaa yritetään ratkaista työssä erilaisia kuvien esiprosessointimenetelmiä käyttäen, koulutusaineistoa suodattaen sekä kuvainformaation lisäksi kuvien metadataa käyttäen. Työssä tutkitaan syötteiden laadun vaikutuksen lisäksi, miten neuroverkon rakenne ja parametrit vaikuttavat luokittelutuloksiin.

Tämä diplomityö koostuu teoreettisesta kirjallisuuskatsauksesta puuntunnistuksessa käytettäviin menetelmiin sekä käytännön koneoppimisluokittelijan toteutuksen menetelmien raportoinnista. Työn seuraavassa kappaleessa käsitellään teoreettista pohjaa menetelmille, joita työn kokeiluissa käytettiin. Kolmannessa kappaleessa tarkastellaan tarkemmin aikaisempia tutkimuksia aiheesta. Neljännessä kappaleessa käsitellään tutkimusmenetelmiä ja viidennessä kappaleessa niitä käyttämällä saatuja tuloksia. Kuudennessa kappaleessa pohditaan saatujen tuloksien syitä ja esitetään parannusehdotuksia. Työn viimeinen kappale toimii yhteenvetona.

2. TEORIA

Tietokoneet on luotu toteuttamaan ihmisen sille antamia käskyjä. Näin ollen, jotta tietokone voisi toteuttaa käskyjä, on ihmisen luotava sitä varten käskyt tietokoneen ymmärtämään muotoon. Tämä tarkoittaa, että mikäli halutaan toteuttaa esimerkiksi ohjelma, joka valitsee kuvajoukon seasta vain ne kuvat, joissa esiintyy kissa, täytyy ihmisen ensiksi luoda formaali matemaattinen käsite siitä, mikä tekee kuvan sisällöstä kissan. Ihmiselle kissan kuvaaminen toiselle ihmiselle ei yleisesti tuota ongelmia, sillä ihmisellä on iän ja kokemuksen myötä kehittynyt konsepti, siitä miten asiat voivat toimia luonnossa. Esimerkiksi jos ongelmana on etsiä kissan silmät, ihminen todennäköisesti etsii niitä kissan pään alueelta, eikä esimerkiksi kissan hännän alueelta tai varsinkaan kissan ruumiin ulkopuoliselta alueelta, mutta tietokoneelle tämä ei ole itsestään selvää. Ongelma on mahdollisesti vielä ihmisen mallinnettavissa, mikäli tehtävänä on aina etsiä sama kissa samassa asennossa, mutta menee nopeasti lähes mahdottomaksi, jos algoritmin pitäisi tunnistaa kaiken värisiä kissoja, koosta, asennosta ja valaistuksesta riippumatta.

Konseptien mallintaminen tietokoneelle ymmärrettävään muotoon on siis vaikeata, ja olisikin yksinkertaisempaa, jos tietokonetta voisi opettaa kuin ihmistä yleistämään mallin esimerkkien avulla. Koneoppiminen pyrkii tekemään tämän mahdolliseksi. Koneoppiminen ei rajoitu pelkkään kuvantunnistukseen, vaan tutkimusta ja käytännön sovelluksia löytyy monilta aloilta, joissa käytetty data on tietokoneen luettavassa muodossa, kuten puheentunnistuksessa (7), tekstin analysoinnissa (8) ja pörssikurssien ennustamisessa (9).

2.1 Digitaalinen kuvankäsittely

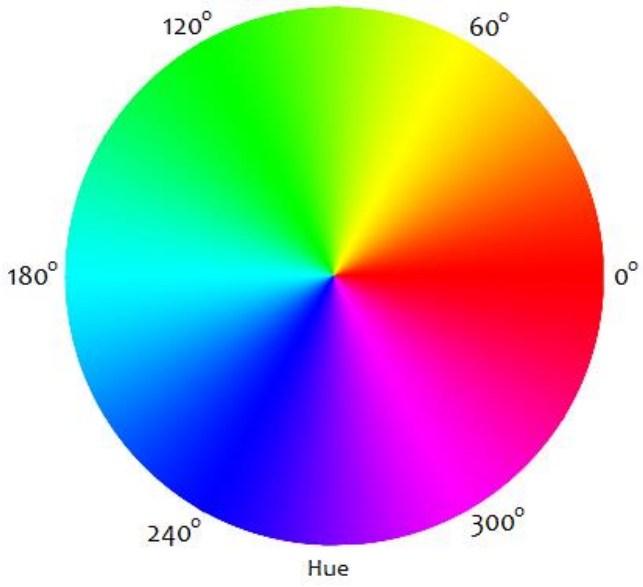
Tietokoneella käsiteltävät digitaaliset kuvat koostuvat kuvapisteistä eli pikseleistä. Kuvat sisältävät lähes poikkeuksetta useamman kuin yhden pikselin, ja merkintätapaa *n* × *m* px käytetään kuvaamaan kuvan ulottuvuuksia, jossa positiivinen kokonaisluku n kuvaa pikselien määrää leveyssuunnassa ja m vastaavasti korkeussuunnassa. Yksinkertaisimmillaan pikselit koostuvat yhdestä ennalta määritellylle lukuvälille sijoittuvasta lukuarvosta, jolloin pikselit edustavat kuvan harmaasävyarvoja eli kyseessä on mustavalkokuva. Käytettyjen bittien määrä määrittää montako väritasoa kuvassa pystytään esittämään, esimerkiksi 8 bittiä pikseliä kohden tarkoittaa $2^8 = 256$ väritasoa, jolloin jokainen pikseli saa kokonaislukuarvon väliltä [0, 255] esittäen kyseisen pikselin valon intensiteettiä.

Värikuvien esittäminen vaatii useamman lukuarvon jokaista pikseliä varten. Yleinen tapa esittää värikuvia digitaalisessa muodossa on kolmikanavainen RGB-väriavaruuden esitys, jossa jokaiselle pikselille annetaan punaisuuden (R, red), vihreyden (G, green) ja sinisyyden (B, blue) arvo. Kyseisiä värejä sekoittamalla oikeassa suhteessa voidaan luoda muut värit. RGB-kuvan pikselit sisältävät siis kolminkertaisen bittien määrän harmaasävykuvaan nähden. Värikuvia on mahdollista muuttaa harmaasävykuviksi RGB-komponenttien arvojen painotettua summaa käyttäen kaavaa

$$gray = 0{,}2989 * R + 0{,}5870 * G + 0{,}1140 * B \qquad (1)$$

jokaiselle kuvan pikselille, jolloin saadaan kolmikanavainen värikuva muutettua yksikanavaiseksi (10). Joissain tapauksissa kuvilla voi olla kolmen värikanavan lisäksi myös kanavia ilmaisemaan esimerkiksi kohteiden syvyystietoa tai opasiteettia (11).

RGB-väriavaruuden lisäksi on olemassa myös muita väriavaruuksia, joihin RGB-kuvat pystytään muuntamaan muunnoskaavoja käyttäen. HSV-väriavaruuden esityksessä kuvalla on värisävyn (H, hue), värikylläisyyden (S, saturation) ja kirkkauden arvon (V, value) kanavat (12). HSV-väriavaruuden etuna voidaan pitää H-kanavan vuoksi mahdollisuutta asettaa rajat väreille, toisin kuin RGB-avaruudessa, missä esimerkiksi vihreän värin määrittely riippuu G-komponentin lisäksi myös R- ja B-komponenteista.



Kuva 1: HSV-avaruuden H-kanava (13)

Muunnos RGB-avaruudesta HSV-avaruuteen tapahtuu komponenteittain kaavoilla

$$H = \begin{cases} \theta, & B \leq G \\ 360 - \theta, & B > G \end{cases}, \theta = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}\right\} \quad (2)$$

$$S = 1 - \frac{\min(R,G,B)}{\max(R,G,B)} \quad (3)$$

$$V = \max(R,G,B) \quad (4)$$

, joissa huomattavaa on H-komponenttia käsiteltäessä punaisen värin sijaitsevan sekä pienimmissä että suurimmissa asteissa (12).

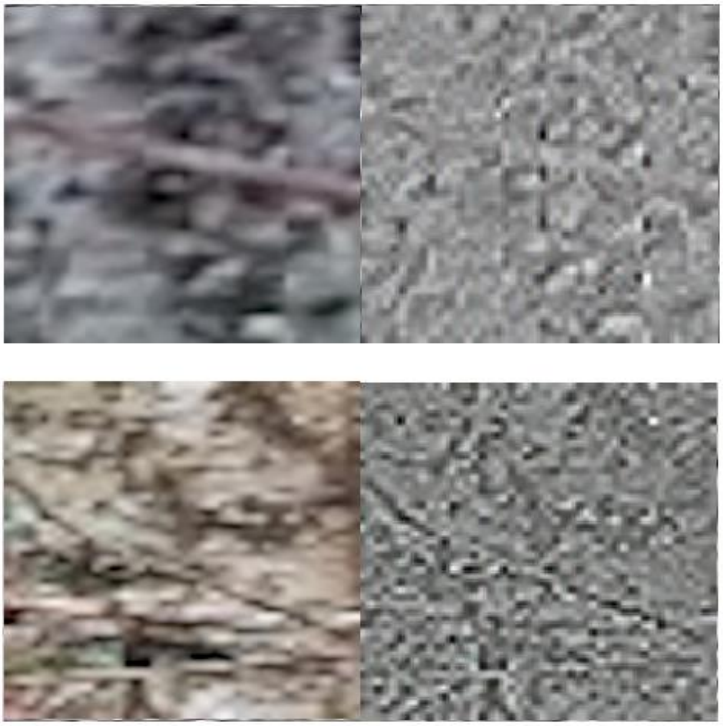
CIELAB-väriavaruus pyrkii esittämään värien muutokset ihmisen näköaistin kannalta lineaarisesti (12). Kuten RGB- ja HSV-avaruudet, CIELAB koostuu kolmesta kanavasta: valoisuutta kuvaavasta L*-kanavasta, punaisuutta ja vihreyttä mittavasta a*-kanavasta sekä keltaisuutta ja sinisyyttä mittaavasta b*-kanavasta (14).

Prosessia, jossa kuvan pikseleiden arvoja muutetaan niin, että kuva joko näyttää ihmissilmälle paremmalta tai soveltuu paremmin jatkokäsiteltäväksi esimerkiksi konenäön sovelluksille, kutsutaan kuvien ehostamiseksi (15). Histogrammi on tapa esittää kuvan pikseliarvojen esiintymisjakauma, ja näin ollen histogrammia muokkaamalla voidaan vaikuttaa siihen miltä kuva näyttää (12). Histogrammin tasoitus on kuvanehostustekniikka, joka levittää histogrammin käytettävälle arvoalueelle parantaen kuvan kontrastia (15).

ZCA (Zero Component Analysis)-vaalentaminen on ehostustekniikka, jota voidaan käyttää kun on olemassa samankokoisia kuvia sisältävä kuva-aineisto. ZCA-vaalentaminen korostaa kuvissa ilmeneviä nopeita intensiteetin muutoksia eli reunoja. ZCA-vaalentamista varten kuvan pikselit skaalataan välille [0, 1], vähennetään pikseleittäin kuva-aineistosta laskettu vastaavat keskiarvot ja lasketaan saadulle matriisille X’ pääakselihajotelma (SVD, singular value decomposition). Tämän jälkeen ZCA-vaalennettu kuva saadaan SVD:stä saatuja U-, S- ja V-matriiseja hyödyntämällä kaavalla

$$X_{ZCA} = U \cdot \mathrm{diag}\left(\frac{1}{\sqrt{\mathrm{diag}(S) + \epsilon}}\right) U^T \cdot X' \quad (5)$$

, jossa diag() on diagonaalimatriisi ja ϵ on valittavissa oleva parametri vaalentamiskerroin (16).



Kuva 2: ZCA-vaalennus koivun rungoille. Vasemmalla alkuperäinen kuva, oikealla ZCA-vaalennettu

Signaalinkäsittelyssä digitaalisella suodattamisella tarkoitetaan digitaalisen signaalin muokkaamista viemällä se järjestelmän läpi, joka suorittaa signaalille matemaattisia operaatioita. Näitä signaalia muuttavia järjestelmiä kutsutaan suodattimiksi ja usein suodatuksen tarkoituksena on vaimentaa signaalista epätoivottuja taajuuksia tai korostaa tiettyjä taajuusalueita (17). Suodattimien toiminta suunnitellaan sovelluskohteen mukaan ja suodattimen tyypin määrittää suodattimen tuottama impulssivaste. Suodattimet muuttavat signaalia suorittamalla niille konvoluution.

Konvoluutioksi kutsutaan tulosta, joka saadaan suodattamalla signaali suodattimella. Konvoluutio on määritetty kaikille funktioille (18) ja digitaalisen kuvankäsittelyn kannalta tärkeä konvoluutiofunktio on kaksiulotteinen diskreetti konvoluutio. Kaksiulotteinen diskreetti konvoluutio w saadaan laskettua taulukkomaisen rakenteen, kuten kuvan sarakkeessa k, rivillä l sijaitsevalle alkiolle kaavalla

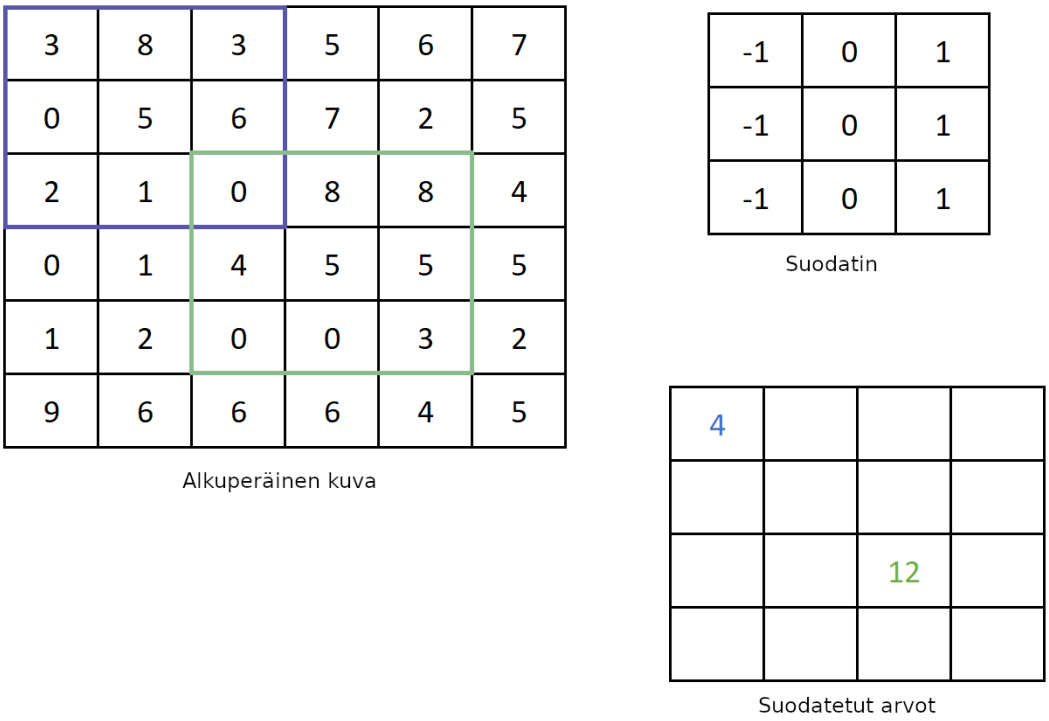
$$w(k,l) = \sum_{i \in M} \sum_{j \in N} h(i,j)s(k-i, l-j) \tag{6}$$

, missä on M ennalta määritetty konvoluutioikkunan leveys ja N konvoluutioikkunan korkeus, funktio h on suodattimen impulssivaste ja s on suodatettava signaali (17).

Digitaalisessa kuvankäsittelyssä tilatason suodattimilla voidaan kaksiulotteisena signaalina toimivia kuvia ehostaa tai kuvista voidaan etsiä jatkokäsittelyn kannalta mielenkiintoisia kohtia kuten reunoja. Tilatason suodattimet ovat *n* × *m* pikselin kokoisia, tyypillisesti selvästi kuvan ulottuvuuksia pienempiä ikkunoita, joita asetetaan kuvassa jokaiseen mahdolliseen kohtaan (19). Tämän jälkeen ikkunoille ja niiden alle jääville alikuville lasketaan diskreetti konvoluutio, jolloin jokaisesta ikkuna-alikuvaparista saadaan uusi arvo tuloskuvaan. Esimerkki digitaalisen konvoluution laskemisesta kahdelle kuvan arvolle on esitetty kuvassa 3.

Useimmiten ikkunan ulottuvuudet ovat parittomia lukuja ja konvoluution arvo saadaan ikkunan alle jäävän alueen keskimmäiselle pikselille (17). Ikkunan ollessa suurempi kuin 1 × 1 px, saatu tuloskuva on ulottuvuuksiltaan pienempi kuin alkuperäinen kuva, sillä suodatusta ei voida tehdä kuvan reuna-alueilla ikkunan mennessä kuvan reunojen yli. Tämä voi olla sovelluskohteesta riippuen ongelmallista ja alkuperäisen kuvan kokoisen tuloksen saaminen vaatiikin täyttämismenetelmien käyttöä.

Täyttämismenetelmät lisäävät kuvan reunoille arvoja, jolloin konvoluutioikkuna mahtuu laskemaan arvon kaikista alkuperäisen kuvan arvoista. Reunojen täyttäminen nollilla on yksinkertainen täyttämismenetelmä, jossa kuva laajennetaan lisäämällä $\frac{n-1}{2}$ kappaletta nollasarakkeita kuvan ensimmäisen sarakkeen edelle ja viimeisen sarakkeen jälkeen. Vastaavasti korkeussuunnassa lisätään $\frac{m-1}{2}$ kappaletta nollarivejä (20). Muut täyttämismenetelmät toimivat samalla tavalla, mutta nollien lisäämisen sijaan esimerkiksi peilataan alkuperäisen kuvan reunoja tarvittavan määrän lisäämiseksi.



Kuva 3: Digitaalinen konvoluutio. Tuloskuva on pienempi kuin alkuperäinen kuva, sillä täyttämismenetelmiä ei ole käytetty.

Joskus digitaalista konvoluutiota kuville suoritettaessa halutaan tarkoituksella saada pienempi tuloskuva kuin alkuperäinen kuva, esimerkiksi jatkoprosessoinnin nopeuttamisen takia (20). Tällöin konvoluutioikkunaa voidaan liikuttaa enemmän kuin yksi pikseli kerrallaan, jolloin konvoluutiota ei suoriteta alkuperäisen kuvan jokaisessa kohdassa.

Digitaalisia kuvia voidaan luoda tietokoneilla erilaisia piirto-ohjelmia käyttäen tai tuomalla reaalimaailmasta kohteiden esityksiä digitaaliseen muotoon. Jälkimmäisessä tapauksessa kolmiulotteisen kohteen kuvauksen kaksiulotteiseksi kuvaksi tekee digitaalikamera. Kameroiden suorittamat kohteiden muunnokset kaksiulotteiseksi kuviksi eivät ole yksiselitteisiä, vaan ne riippuvat käytettävistä kameroista ja niiden kalibroinnista (21). Kameroiden onnistuneella kalibroinnilla onkin konenäkösovellusten tuloksiin suuri vaikutus (22).

2.2 Koneoppiminen

Koneoppiminen (Machine Learning, ML) on eräs tekoälyn (Artificial Intelligence, AI) osa-alue. Koneoppimisessa perustana on idea siitä, että tietokonealgoritmin toiminta parantuisi automaattisesti kokemuksen myötä, jolloin algoritmin kehittäjän ei tarvitse manuaalisesti varautua jokaiseen mahdolliseen lopputulemaan (23). Koneoppimisen voidaan katsoa alkaneen 1950-luvulla oppivien koneiden konseptien myötä, mutta ensimmäiset

varsinaiset yleiskäyttöiset algoritmit kehitettiin vuosia myöhemmin. Esimerkiksi koneoppimisessa vieläkin suositut tukivektorikoneet ja satunnaismetsät ovat vuodelta 1995 (24).

Koneoppiminen voidaan jakaa kolmeen eri kategoriaan: ohjattuun oppimiseen (supervised learning), ohjaamattomaan oppimiseen (unsupervised learning) ja vahvistusoppimiseen (reinforcement learning). Ohjatussa oppimisessa pyritään esimerkkidataa käyttäen luomaan malli, joka toimii mahdollisimman hyvin myös silloin, kun mallille syötetään dataa, jota se ei ole ennen käsitellyt. Ohjattu oppiminen voidaan jakaa edelleen kahteen eri alikategoriaan: luokitteluun ja regressioon.

Luokittelussa mallin tarkoitus on tehdä päätös siitä, mihin ennalta määritellyistä luokista mallille syötetty näyte kuuluu (24). Tässä diplomityössä käytetyt menetelmät kuuluvat luokittelun piiriin, sillä tarkoituksena on kouluttaa malli esimerkkidataa käyttäen ja tämän jälkeen pyrkiä mahdollisimman tarkasti luokittelemaan mallille tuntemattomat näytteet ennalta määritettyihin puulajeihin, kuten koivuun tai mäntyyn. Luokittelumenetelmät vaativat merkittyä esimerkkidataa, joten näytteiden keräämisen lisäksi on merkittävä ennen mallille syöttämistä, mitä luokkaa kyseinen näyte edustaa.

Regressiota käytetään, kun tavoitteena on ennustaa lukuarvoja mahdollisimman lähelle oikeaa tulosta (24). Esimerkiksi, jos tavoitteena olisi ennustaa puiden ikää, olisi vuoden päähän oikeasta iästä tehty arvaus parempi arvio kuin kymmenen vuoden päähän osuva arvaus. Tämä poikkeaa luokittelusta, jossa kuusen luokittelu haavaksi on yhtä väärin kuin luokittelu koivuksi.

Ohjaamattomassa oppimisessa tavoitteena on tietoaineiston jakaminen toisistaan poikkeaviin ryhmiin, joissa näytteet ovat keskenään samankaltaisia (25). Esimerkiksi metsän puut voisi ryhmitellä niiden lehtien lukumäärän mukaan, jolloin tulokseksi voitaisiin saada lehtipuiden ryhmä ja havupuiden ryhmä.

Vahvistusoppimisessa tavoitteena on antaa järjestelmän oppia itsenäisesti palkitsemalla oikeasta toiminnasta ja rankaisemalla huonosta toiminnasta (24). Esimerkiksi automaattisesti lentävää lennokkia, jonka tarkoitus on ottaa kuvia metsästä, voisi palkita kuvien ottamisesta, jos kuvassa näkyy puu ja rangaista puita päin lentämisestä ja kuvista, joissa ei ole puita.

Luokittelussa tunnistettavien luokkien malli yritetään luoda esimerkkidataa käyttäen. Tätä varten on kerättävä lukuisia näytteitä jokaisesta luokiteltavasta luokasta. Näytteiden keräämisen jälkeen kerätty datakokoelma jaetaan koulutusdataan, testidataan ja joskus erilliseen validointidataan (24). Koneoppimismalli koulutetaan koulutusdataa käyttäen ja mallin toimintaa arvioidaan mallin ennennäkemätöntä testidataa käyttäen, jolloin saa-

daan arvioitua mallin toimintaa yleisessä tapauksessa (26). Erillistä validointidataa käytettäessä validointidatalla mitataan mallin toimintaa koulutusvaiheessa, jolloin koulutuksessa iteraatiokierrosten välissä katsotaan, miten malli suoriutuu ennen näkemättömän datan luokittelusta. Testidataa käytetään tässä tapauksessa vain koko koulutuksen jälkeen mittaamaan parhaan iteraation suoriutumista riippumattomasti. Validointidatan ja testidatan tulisi sisältää näytteitä samassa suhteessa kuin mitä ne ovat koulutusdatassa. Lisäksi mallin halutaan yleistyvän mahdollisimman hyvin, joten jokaisen käytettävistä aineistoista tulisi sisältää monipuolisesti luokan edustajia. Esimerkiksi jos data-aineisto sisältää hyvässä ja huonossa valaistusolosuhteissa otettuja kuvia, tulisi myös koulutus-validointi- ja testiaineistojen sisältää kuvia samassa suhteessa. Käytännössä aineistot muodostetaan ottamalla satunnaisnäytteitä koko data-aineistosta (24), tarvittaessa luokittain, jolloin luokkien suhde pysyy samana joka aineistossa. Datan jakaminen 80:20 suhteessa koulutusaineiston ja testiaineiston välille on yleisesti käytetty jakosuhde (24). Erillistä validointiaineistoa käytettäessä voidaan tietoaineisto jakaa esimerkiksi 70:20:10 suhteessa koulutus- validointi- ja testiaineistoon (27).

Liian pienen koulutusaineiston käyttö tai epäonnistunut koneoppimismenetelmien valinta johtaa tilanteeseen, jossa koulutettava malli jää liian yksinkertaiseksi, jolloin saadut tulokset ovat huonoja sekä koulutusvaiheessa että testausvaiheessa (24). Joskus malli voi myös muodostua liian monimutkaiseksi, jolloin koulutusvaiheessa saadaan hyviä tuloksia, mutta testausvaiheessa tulokset ovat selvästi huonompia. Tällöin mallin toimivuus yleisissä tapauksissa on huono ja koneoppimisjärjestelmää on yksinkertaistettava (24) tai käytettävän datan määrää on lisättävä (28).

Mallin toiminnan mittaamista varten testiaineistolle suoritetaan luokittelu ja verrataan mallin antamia luokkia näytteiden todellisiin luokkiin. Näin saadaan jokaiselle luokalle erikseen neljä eri lukuarvoa. Oikein arvatut tapaukset ovat ne tapaukset, joissa luokan edustajat arvattiin kyseisen luokan edustajiksi (True Positive, TP) ja tapaukset, joissa luokan ulkopuoliset edustajat arvattiin muuksi luokiksi (True Negative, TN). Väärin arvatuissa tapaukset jakautuvat luokan ulkopuolisiin tapauksiin, jotka arvattiin luokan edustajiksi (False Positive, FP) ja luokan edustajiin, jotka arvattiin luokkaan kuulumattomiksi (False Negative, FN) (24). Tämän jälkeen mallin toimintaa voidaan mitata laskemalla mallin tarkkuus kaavalla

$$\text{mallin tarkkuus} = \frac{TP + \text{TN}}{TP + FP + FN + TN} \tag{7}$$

, joka tuottaa luvun väliltä [0, 1]. Joillakin sovellusaloilla voi mallin tarkkuuden sijasta olla hyödyllisempää käyttää muita mittareita parhaan mallin löytämiseen, esimerkiksi lääketieteellisellä alalla löytämättömien sairauksien (FN) määrän minimoiminen voi olla tärkeämpää luokittajalle kuin terveiden potilaiden luokitteleminen sairaiksi (FP).

Sekoitusmatriisi on tapa esittää luokittelun tuloksia. Matriisin vaakarivit kuvaavat oikeita luokkia, pystysarakkeet puolestaan luokittelussa tehtäviä arvauksia. Seuraavasta esimerkkimatriisista on luettavissa, että testiaineisto sisälsi 18 näytettä luokkaa kohden, sillä jokaisen vaakarivin näytteiden summa on 18. Oikein arvatut näytteet voidaan lukea matriisin diagonaalilta. Sekoitusmatriisi on hyödyllinen, kun halutaan selvittää mitkä luokat luokittelija sekoittaa keskenään. Esimerkkimatriisissa luokittelija arvaa haavan koivuksi 7 kertaa, mutta eukalyptuksen koivuksi vain yhden kerran, joten luokittelija suoriutuu paremmin koivun ja eukalyptuksen erottamisesta kuin koivun ja haavan erottamisesta.

Taulukko 1: Esimerkkisekoitusmatriisi

	Haapa	Koivu	Eukalyptus
Haapa	10	7	1
Koivu	5	13	0
Eukalyptus	1	1	16

Tukivektorikone on yleisesti käytetty koneoppimisluokittelija, joka yrittää jakaa näytteet ryhmiin lineaarisen tason luomalla, niin että ryhmät toisistaan erottavasta tasosta on mahdollisimman pitkä matka luokkien lähimpiin näytteisiin (29). Tapauksissa, joissa luokkia ei voida erottaa toisistaan, kuvataan näytteet epälineaarista kernel-funktiota käyttäen korkeampiuloitteisempaan avaruuteen, jonka jälkeen lineaarisen tason luomista yritetään uudelleen. Tarvittaessa voidaan sallia osan näytteistä väärin luokittelemisen määrittämällä regularisointivakion *C*, joka määrittää kuinka paljon väärin luokittelemisesta rangaistaan tukivektorikoneen tason luonnin päätöksenteossa (30). Tukivektorikonetta on käytetty monella sovellusalalla etenkin konenäön yhteydessä, kuten puuntunnistuksessa (31) (32), kalojen tunnistuksessa (33) ja lääketieteellisessä kuvantamisessa (29).

Konenäön sovelluksissa kuvia ei aina käytetä sellaisinaan, vaan usein kuvien pikseliarvoja käyttäen muodostetaan kuvan sisältöä kuvaavia arvojen joukkoa (11). Näitä ominaisuusvektoreita (feature descriptor) ovat yksinkertaisimmillaan kuvien pikseleistä lasketut tilastolliset suureet, kuten keskimääräinen harmaasävytaso. Tällaisten, koko kuvan sisällöstä laskettavien globaalien ominaisuusvektorien ulottuvuuksien määrä on usein selvästi pienempi alkuperäiseen kuvaan verrattuna, mikä nopeuttaa jatkotoimenpiteiden suorittamista. Pelkät globaalit ominaisuusvektorit eivät kuitenkaan ole konenäön sovelluksiin usein riittäviä, sillä kuvissa arvojen sijainti on yleisesti tärkeää (11). Lokaalit ominaisuusvektorit pyrkivät ottamaan huomioon kuvan arvojen lisäksi myös niiden sijainnin.

Hyvän ominaisuusvektorin tulisi poiketa muiden kohteiden ominaisuusvektoreista, jotta ominaisuusvektorin perusteella voidaan paikantaa oikea kohde. Hyvän ominaisuusvektorin tulisi myös kuvata kohdetta myös silloin, kun kohteen valaistusolosuhteet muuttuvat, kohde liikkuu tai kääntyy kuvassa. Ominaisuusvektorin tulisi ottaa huomioon myös kohteen koon muuttuminen kuvassa, esimerkiksi syvyyssuunnassa liikkumisen takia. Lisäksi käytännön sovelluksia varten hyvän ominaisuusvektorin tulisi olla nopeasti tietokoneelle laskettavissa (34). Ennen konvoluutioneuroverkkoja ominaisuusvektoreiden muodostus riippui sovelluskohteesta suuresti (11). Esimerkiksi lokaalien kuva-alueiden kontrasti- ja tilainformaatiota hyödyntäviä LBP-ominaisuusvektoreita (Local Binary Pattern) on käytetty onnistuneesti pintarakenteiden tunnistamisessa (35).

Koneoppimisluokittelijoiden vaatimien tietoaineistojen kerääminen on usein työlästä, varsinkin kun käytetään kuvamateriaalia, sillä kuvien ottamisen lisäksi ihmisen pitää merkitä kuvat oikeisiin luokkiin, jotta koulutus voidaan tehdä. Julkisia, kaikille saatavilla olevia tietoaineistoja, kuten CIFAR-10:tä (36) käyttäen koneoppimisjärjestelmien kouluttaminen on mahdollista kaikille vaivattomasti. Kattavien aineistojen saatavuus on kuitenkin yleisesti ongelma koneoppimisessa (23) ja monilla sovellusalueilla data joudutaan usein keräämään ennen kuin koneoppimista päästään tekemään.

Yleisesti koneoppimisjärjestelmät eivät ole monikäyttöisiä, vaan ne koulutetaan suoriutumaan tietystä ongelmasta, esimerkiksi ihmisen iän arvioinnista. Koneoppimisjärjestelmien mahdolliseksi tulevaisuudeksi on kuitenkin ideoitu monikäyttöisiä järjestelmiä, jotka voisivat ihmisen tavoin hyödyntää eri sovellusalalta opittuja konsepteja uuden sovellusalan oppimisessa (23).

2.3 Neuroverkot

Keinotekoiset neuroverkot (Artificial Neural Network) ovat alun perin ihmisen hermoston toiminnan perusteella kehitettyjä koneoppimismenetelmiä, jotka nimestään huolimatta

eivät mallinna ihmisen hermostoa neurotieteelliseltä kannalta kovinkaan tarkasti (20). Neuroverkot koostuvat neuroneista, jotka ottavat syötteenään painoilla kerrottuna syötedatan arvot ja tuottavat sovelluskohteesta riippuen yhden (regressio) tai useamman (luokittelu) arvon. Neuroverkon tuottamaa tulosta verrataan oikeaan tulokseen, jonka perusteella syötepainoja muokataan, tarkoituksena vähentää luokittelussa tapahtuvaa virhettä. Muokkaamalla syötepainoja iteratiivisesti suuntaan, johon tehty virhe vähenee, neuroverkon tuottama tulos paranee.

Neuroverkkojen kehitys alkoi klassisten koneoppimismenetelmien tapaan 1950-luvulla, mutta ovat nousseet suosioon vasta 2000-luvulla. Syynä neuroverkkojen hitaalle kehitykselle on neuroverkkojen vaatima, klassisiin menetelmiin verrattuna selvästi suurempi opetusaineiston määrä ja laitteistolta vaadittava laskentateho. Tällä vuosituhannella Big Data on mahdollistanut helpomman tietoaineistojen keräämisen monella alalla ja erillisten näytönohjainten nopea kehitys on kasvattanut käytettävissä olevaa laskentatehoa neuroverkkojen vaatimuksiin sopivaksi (20). Monet kaupalliset toimijat kuten Amazon (37), Google (38) ja Microsoft (39) ovat myös alkaneet tarjoamaan tehokkaiden laskentakoneiden vuokrausta pilvipalveluissaan, tehden monimutkaisten neuroverkkojen koulutuksen mahdolliseksi ilman tarvetta omistaa kallista laitteistoa. Laskentatehon tarjoamista hajautetusti on myös ideoitu käyttäen lohkoketjuja (40).

Neuroverkkoja käyttämällä saadaan yleensä parempia tuloksia kuin klassisilla koneoppimismenetelmillä, sillä neuroverkkojen koulutukseen vaikuttaa vähemmän koulutusaineiston koko (41). Neuroverkkojen huonona puolena voidaan pitää korkean koulutusajan lisäksi myös verkon parametrien vaikutuksen epämääräisyyttä (41), mikä vaikeuttaa neuroverkon luokitteluperusteiden ymmärtämistä.

Neuroverkot koostuvat yksinkertaisimmillaan yhdestä syötekerroksesta ja tulostuskerroksesta. Tällaista neuroverkkoa kutsutaan perseptroniksi. Perseptronit yksinään käytettynä eivät ole klassisia koneoppimismenetelmiä, kuten tukivektorikoneita monimutkaisempia ja monet klassiset koneoppimisalgoritmit voidaankin luoda neuroverkkoja käyttäen (20). Neuroverkkojen syvyys on määritelty laskentaa suorittavien kerrosten määrästä, joten perseptroni on yksikerroksinen neuroverkko, vaikka se sisältääkin syöte- ja tulostuskerroksen.

Neuroverkon kouluttamista varten tulee luokat merkitsevät tunnisteet (label) muuttaa numeeriseen muotoon, esimerkiksi binääriluokittelussa voitaisiin luokan koivu edustajia merkitä luokalla -1 ja luokan mänty edustajia luokalla 1. Monilajiluokittelussa on tapana

muuttaa tunnisteet niin sanotulla one-hot koodauksella, jossa tunnisteet muutetaan luokkien lukumäärän pituisiksi vektoreiksi, joissa ykkösen sijainti määrittää luokan ja muut alkiot ovat nollia (20).

Syötekerros ottaa sisään *d* kappaletta ominaisuuksia syötedatasta, esimerkiksi mustavalkokuvan jokainen pikseli voi toimia ominaisuutena syötekerrosta varten. Jokaista ominaisuutta $[x_1, \dots, x_d]$ varten luodaan alkutilanteessa ominaisuuden kertoimeksi reaalilukuarvo $[w_1, \dots, w_d]$, joita kutsutaan ominaisuuden painoiksi tai painokertoimiksi. Ottamalla summan vastaavilla painoilla kerrotuista ominaisuuksista saadaan ulos yksittäinen arvo *v* jokaista kerroksen neuronia kohden (20). Tämä arvo syötetään kynnystämistä tekevään aktivointifunktioon, jolloin saadaan lopullinen tulos $\hat{y}$, jota verrataan koulutuksessa tunnettuun todelliseen arvoon y regressiota tehdessä tai luokan tunnisteen arvoon luokittelua tehdessä (11). Neuroverkon häviö lasketaan syöttämällä neuroverkon tuottama tulos $\hat{y}$ ja todellinen arvo y häviöfunktioon, joka tuottaa yksittäisen arvon riippuen kuinka lähelle oikeaa tulosta arvattiin (20). Lopulta neuroverkon painot päivitetään vastavirta-algoritmia käyttäen ja kun koko koulutusaineisto on näytetty kerran neuroverkolle, on suoritettu yksi kokonainen koulutusjakso (epoch). Jaksoja toistamalla ennalta määritellyn määrän, esimerkiksi 1000 kertaa, painot siirtyvät kohti mallia, joka tuottaa mahdollisimman paljon oikeita arvauksia. Käytännössä koulutukseen käytettävää aikaa saa vähennettyä lopettamalla koulutuksen, kun validointidatan perusteella laskettu suure, kuten luokittelutarkkuus ei enää parane. Lokaalien minimien välttämiseksi koulutusta ei kuitenkaan kannata lopettaa heti, jos tulos ei parane, vaan tarkastella muutosta useamman koulutusjakson ikkunalla.

Aktivointifunktion valinta vaikuttaa neuroverkon koulutuksen ja käytännön luokittelun kestoon ja tarkkuuteen. Aktivointifunktioiden epälineaarisuus on monikerroksisissa neuroverkoissa tärkeä verkon suorituskykyä parantava ominaisuus (20). Lisäksi painokerrointen päivitystä varten aktivointifunktioiden tulisi olla jatkuvia ja derivoituvia (11). Yhdessä monikerroksisessa neuroverkossa on mahdollista käyttää useampaa erilaista aktivointifunktiota, sillä aktivointifunktiot valitaan neuroverkon kerrosten perusteella. Usein luokittelun tuloksena halutaan jakauma luokkien todennäköisyydestä, joka saadaan lisäämällä laskentakerrosten perään normalisointia tekevä, softmax-aktivointifunktiota käyttävä kerros (11).

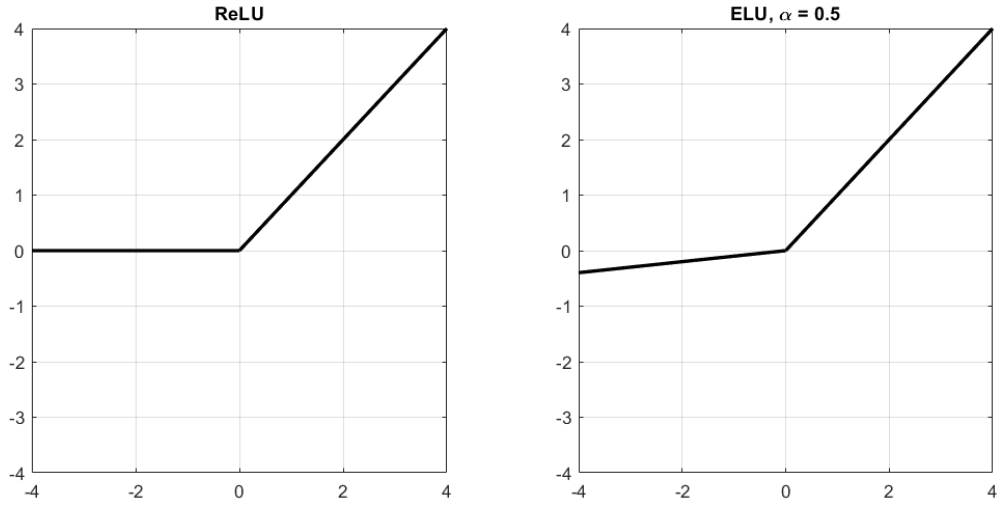
Aktivointifunktioita on olemassa monia, joista epälineaarinen ReLU (Rectified Linear Unit) on suosittu (20). ReLU on yksinkertainen paloittain määritelty funktio

$$f(v) = \begin{cases} 0, & v < 0 \\ v, & v \geq 0 \end{cases} = \max(0, v) \tag{8}$$

jota käytetään usein moderneissa neuroverkoissa. ReLU:a läheisesti muistuttavaa ELU:a (Exponential Linear Unit) käyttämällä neuroverkko voidaan kouluttaa nopeammin kuin ReLU:lla (42). ELU:n funktio on

$$f(v) = \begin{cases} \alpha(\exp(v) - 1), & v < 0 \\ v, & v \geq 0 \end{cases} \quad (9)$$

, missä $\alpha > 0$ on negatiivisia arvoja kontrolloiva hyperparametri ja exp() on eksponenttifunktio. Toisin kuin ReLU, joka tuottaa vain nollaa ja sitä suurempia arvoja, ELU tuottaa myös negatiivisia arvoja. Tästä johtuen ELU:n tuottamat arvot ovat lähempänä nollakeskiarvoa kuin ReLU:n, minkä on tutkittu nopeuttavan neuroverkkojen koulutukseen kuluvaa aikaa (43).



Kuva 4: ReLU ja ELU aktivointifunktiot

ELU:a käyttävät neuroverkot ovat osoittautuneet konenäön sovelluksissa usein käytettävissä konvoluutioneuroverkoissa huomattavasti vastaavia ReLU:a käyttäviä neuroverkkoja tarkemmiksi, kun koulutukseen käytettävät iteraatiomäärät ovat verkoille samat. ELU:a käyttävien verkkojen suorittama luokittelu on sen sijaan hieman ReLU-verkkoja hitaampaa. (42)

Softmax-aktivointifunktio antaa luokittelussa jokaiselle luokalle todennäköisyyden käyttäen syötteenään laskentakerrosten tulosta. Softmaxin funktio on

$$\Phi(\bar{v})_i = \frac{\exp(v_i)}{\sum_{j=1}^{k} \exp(v_j)} \quad \forall_i \in \{1, \dots, k\} \quad (10)$$

, missä *k* on luokkien lukumäärä ja $\bar{v}$ on *k* reaalilukuarvosta koostuva vektori $\bar{v} = [v_1, \dots, v_k]$ (20). Tulosvektori sisältää näin reaalilukuarvon [0, 1] jokaiselle luokalle, jotka

kuvaavat luokittelun todennäköisyyttä. Luokittelun tulos saadaan ottamalla arvoista suurin ja luokittelun varmuutta voidaan arvioida vertaamalla suurimman luokan arvoa johonkin vakioon tai seuraavaksi suurimpaan luokan arvoon.

Häviöfunktio on tapa mitata neuroverkon toimintaa ja hyvin suoriutuvan neuroverkon luomiseksi onkin tärkeää saada häviö mahdollisimman pieneksi. Häviöfunktion valinta riippuu sovelluskohteesta ja yleisesti valinta tehdään käytettävien luokkien lukumäärän perusteella (20). Binääritapauksessa häviö saadaan käyttäen logistista regressiota

$$L = \ln(1 + \exp(-y \cdot \hat{y}) \tag{11}$$

, missä *y* saa luokasta riippuen arvon -1 tai 1 ja $\hat{y}$ on binääriluokittelun tuloksena saatava yksittäinen arvo (20). Kun luokkia on enemmän kuin kaksi, häviö saadaan käyttämällä kategorista ristientropiaa

$$L = -\ln\left(\frac{\exp(v_r)}{\sum_{j=1}^{k} \exp(v_j)}\right) \tag{12}$$

, missä käytetään hyväksi kaavassa (10) määriteltyä softmax-funktiota, alaindeksin r kuvatessa oikean tunnisteen indeksiä.

Häviöfunktioista saatavan arvon absoluuttinen suuruus riippuu sovelluskohteesta ja muusta verkon toteutuksesta, joten luku ei ole verrannollinen eri sovelluskohteiden välillä. Sen sijaan tärkeä on, että häviö vähenee koulutuksen edetessä.

Neuronikerroksia voidaan kuitenkin yhdistää toisiinsa ja näin luoda monikerroksisia neuroverkkoja, joita kutsutaan myös monikerroksisiksi perseptroneiksi (Multilayer Perceptron, MLP) (11). Tällöin jokaisen kerroksen ulostulo toimii seuraavan kerroksen jokaisen neuronin syötteenä. Kuten yksikerroksisen perseptronin tapauksessa, jokaiselle syötteelle on oma painokerroin myös silloin kun syötteenä toimii edellisen kerroksen ulostulo, jolloin jokaisen kerroksen väliin saadaan edellisen kerroksen neuronien lukumäärä kertaa kyseisen kerroksen lukumäärä kappaletta painokertoimia (20). Yksikerroksiset neuroverkot pystyvät ratkaisemaan vain lineaarisesti separoituvia ongelmia, toisin kuin monikerroksisilla neuroverkoilla, joilla pystytään käsittelemään myös ongelmia, jotka eivät ole lineaarisesti separoitavia. Tämä on mahdollista kerrosten välissä olevien epälineaaristen aktivointifunktioiden kuten ReLU:n ansiosta, jotka muuttavat neuronien lineaariset laskentatulokset epälineaarisiksi (11).

Syöte- ja ulostulokerroksen välisiä laskentaa suorittavia kerroksia kutsutaan yleisesti piilokerroksiksi (11). Piilokerrosten määrän kasvattaminen lisää neuroverkon mahdollista suorituskykyä merkittävästi, mutta samalla myös neuroverkon kouluttamiseen kuluva aika kasvaa. Neuroverkkoja, joissa on yli kolme piilokerrosta, kutsutaan joskus syviksi

neuroverkoiksi ja näiden neuroverkkojen kouluttamiseen erillinen näytönohjain on käytännössä pakollinen (24). Tavallisesti neuroverkon syötteet liikkuvat neuroverkoissa vain yhteen suuntaan, jolloin neuroverkkoa kutsutaan eteenpäin kytketyksi neuroverkoksi. Joissain tapauksissa neuroverkot voivat kuitenkin sisältää kytkentöjä myös taaksepäin, jolloin kyseessä on takaisinkytketty neuroverkko (24). Tässä työssä käsitellään vain eteenpäin kytkettyjä neuroverkkoja.

Vastavirta-algoritmi (back-propagation algorithm) on neuroverkoissa painokerrointen päivityksessä käytettävä algoritmi (11). Vastavirta-algoritmi pyrkii selvittämään, mitkä neuroverkon painoista vaikuttavat saadun virheen määrään eniten ja muuttamaan painoja, niin että virhe vähenee. Laskemalla saadulle virheelle osittaisderivaatan jokaisen painokertoimen suhteen, saadaan selville yksittäisen painokertoimen vaikutus virheeseen. Painokerroin vaikuttaa epäsuorasti kaikkiin sen jälkeisiin kerroksiin, joten aikaisempien kerrosten osittaisderivaattoja laskettaessa on otettava huomioon kyseisen kerroksen jälkeisten kerrosten osittaisderivaatat. Peräkkäisiä kerroksia yhdistettynä funktiona käsittelemällä voidaan käyttää derivaatan ketjusääntöä ja näin laskea osittaisderivaatat aikaisemmille kerroksille (20).

Vastavirta-algoritmin suorittama gradientin laskenta vaatii tietokoneelta paljon muistia varsinkin syvissä neuroverkoissa, sillä opittavien parametrien määrä kasvaa nopeasti neuroverkon rakenteen monimutkaistuessa (20). Tämän takia koko koulutusaineistoa ei voida usein syöttää kerralla gradientin laskemiseksi, vaan koulutus tehdään ottamalla koulutusaineistosta pienempi satunnaisesti valittu erä (mini-batch) kerrallaan ja laskemalla gradientti ja päivittämällä painokertoimet joka erän jälkeen (11). Painokertoimien päivityksiä suoritetaan joka koulutusjaksossa, kunnes koko koulutusaineisto on käyty läpi, eli erän koko on käänteen verrannollinen suoritettavien päivitysten määrään. Erän koon valitseminen on neuroverkon toteuttajan päätettävissä ja tapausta, jossa erän koko on yksi, kutsutaan stokastiseksi gradienttilaskeutumiseksi (Stochastic Gradient Descent, SGD) (20). Eräkoon suuruus vaikuttaa neuroverkon tarkkuuteen ja koulutusnopeuteen, pienen eräkoon käyttämisen olevan hitaampi laskea (11) ja tuottavan usein koulutusvaiheessa epätarkempia tuloksia (20). Suuremmalla eräkoolla puolestaan tarvittava muistin määrä on myös suurempi, joten eräkoko on valittava käytettävän laitteiston perusteella (20).

Vastavirta-algoritmi ei yksinään määrää paljonko painokertoimia muutetaan, vaan saatu gradientti kerrotaan oppimisnopeutta (learning rate) kuvaavalla kertoimella $\propto > 0$. Oppimisnopeuden asettaminen liian pieneksi hidastaa koulutusta, sillä jokainen koulutusjakso muuttaa painoja vain hieman oikeaan suuntaan. Liian pieni oppimisnopeus johtaa

usein myös tilanteeseen, jossa globaalia minimiä ei löydetä, vaan algoritmi pysähtyy lokaaliin minimikohtaan. Toisaalta oppimisnopeuden asettaminen liian suureksi johtaa tilanteeseen, jossa virheen minimiä ei ikinä löydetä, vaan virhe jää poukkoilemaan edestakaisin minimikohdan ympärille (20). Näin ollen oppimisnopeuden valinta on tärkeä osa neuroverkkoja suunniteltaessa ja ongelmaan on kehitetty ratkaisuksi muuttuvat oppimisnopeuskertoimet.

Yksinkertaisimmillaan muuttuvat oppimisnopeuskertoimet pienenevät koulutusjaksojen etenemisen perusteella, kertomalla sen hetkisen oppimisnopeuden vakiokertoimella

$0 < l < 1$ joka *n*:s jakso. Edistyneemmät tavat muuttaa oppimisnopeuskerrointa käyttävät hyväkseen koulutuksen aikana saatavan validointiaineiston häviön suuruutta pienentämällä oppimisnopeutta, kun häviö ei enää pienene. Oppimisnopeuskertoimen muuttamisessa voidaan lisäksi käyttää häviön ”liikemäärään” pohjautuvia menetelmiä, joissa oppimisnopeutta kasvatetaan, kun häviö vähenee useamman iteraation peräkkäin ja vähennetään häviön kasvaessa, jolloin oppimisnopeus pysyy tarpeeksi suurena lokaalien minimien ylittämiseksi (20).

2.4 Konvoluutioneuroverkot

Konvoluutioneuroverkot (Convolutional Neural Network, CNN) ovat keinotekoisten eteenpäin kytkettyjen neuroverkkojen alalaji, joille yhteistä on vähintään yhden, mutta usein useamman digitaalista konvoluutiota suorittavan konvoluutiokerroksen sisältäminen (20). Konvoluutioneuroverkkoja käytetään useimmiten, kun syötteenä on kuva, mutta muitakin käyttötapauksia on. Yhteistä käyttötapauksille on ruudukkomainen data, jossa syötetaulukon alkioiden järjestys vaikuttaa syötteen tulkintaan (44). Esimerkiksi lauseen sanojen järjestys riippuu edeltävästä ja seuraavasta sanasta kielioppisääntöjen mukaisesti ja vaihtamalla esimerkiksi jokaisen lauseen viimeisen sanan ensimmäisen sanan kanssa, voi lauseen merkitys muuttua. Vastaesimerkkinä järjestelemättömän datan, kuten asunnon huoneluvun ja hinnan sarakkeiden vaihtaminen keskenään ei muuta taulukon kokonaisuuden tulkintaa.

Konvoluutioneuroverkot ovat usein syvempiä kuin konvoluutiokerroksia sisältämättömät neuroverkot. Laskentatehon kehittymisen myötä syvien neuroverkkojen kouluttamisen mahdollisuus on vaikuttanut varsinkin konvoluutioneuroverkkojen kehitykseen suuresti (20). Konvoluutioneuroverkkoja voidaan luokittelun lisäksi käyttää myös monissa muissa konenäön sovelluksissa, kuten syötekuvan kanssa samanlaisten kuvien etsimisessä tai kohteiden paikantamisessa kuvista tai videoista (20).

Konvoluutioneuroverkot koostuvat eteenpäin kytkettyjen neuroverkkojen tavoin syötekerroksesta, yhdestä tai useammasta piilokerroksesta ja tulostuskerroksesta. Syöte- ja tulostuskerrokset ovat toiminnaltaan samoja molemmissa verkoissa ja konvoluutioneuroverkot poikkeavatkin vain joidenkin piilokerrosten osalta eteenpäin kytketyistä neuroverkoista. Konvoluutioneuroverkkojen erityiset piilokerrokset koostuvat vähintään konvoluutiokerroksesta ja mahdollisesti myös yhdistämiskerroksesta.

Konvoluutiokerrosten tehtävä neuroverkoissa on suorittaa kerroksen syötteelle luvussa 2.1 esitelty digitaalinen konvoluutio. Konvoluutioneuroverkoissa kerrosten syötteet ovat kuitenkin kolmiulotteista dataa, myös mustavalkokuvia käsiteltäessä, joten konvoluutioikkunankin tulee olla kolmiulotteinen. Konvoluutioikkunoiden syvyys on aina sama kuin syötteen syvyys, esimerkiksi RGB-värikuvien tapauksessa ensimmäisen kerroksen syvyys on 3. Konvoluutio suoritetaan ennalta määritellyllä määrällä p suodattimia, jolloin tuloksena saadaan p kappaletta kaksiulotteisia tulosmatriiseita, joiden koko on sama kuin syötteellä, jos konvoluutiossa käytettiin reunojen täyttämistä tai konvoluutioikkunan koon ollessa 1 × 1 px, muussa tapauksessa alkuperäistä pienempi. Tulosmatriisin alkiot syötetään aktivointifunktioon, kuten ReLU:un, joka muuttaa vain alkioiden arvoja, ei tulosmatriisin kokoa. Näitä aktivoituja tulosmatriiseita kutsutaan ominaisuuskartoiksi (feature map). (20)

Ominaisuuskartat ovat konvoluutioneuroverkon tapa kuvata kuvista suodattimilla saatavia ominaisuusvektoreita. Luotavien ominaisuuskarttojen määrä ei ole rajoitettu alkuperäisen syötteen kanavien lukumäärän perusteella, joten kolmikanavaisesta värikuvasta voidaan siis luoda esimerkiksi 32 ominaisuuskarttaa (19). Verkon syötettä lähellä olevien kerrosten suodattimet tunnistavat alkeellisia muotoja, kuten reunoja tai läikkämäisiä muotoja (11).

Yleisesti konvoluutioneuroverkoilla ei tarvitse itse muodostaa ominaisuusvektoreita toisin kuin klassisia koneoppimismenetelmiä käytettäessä, koska syvien konvoluutioverkkojen oppimat ominaisuusvektorit tuottavat paremmat tulokset kuin aiemmat käsintehdyt ominaisuusvektorit, kuten LBP ja SIFT (45). Tämän takia konvoluutioneuroverkkojen kouluttaminen vaatii usein vähemmän sovellusalan osaamista kuin klassisten menetelmien käyttäminen (46). Konvoluutioneuroverkoissa suodattimien alkiot ovat opittava asia, niiden voidaan siis katsoa vastaavan eteenpäin kytkettyjen neuroverkkojen painokertoimia (19).

Kuten eteenpäin kytketyissä neuroverkoissa, edellisen kerroksen syötteet toimivat myös konvoluutioneuroverkoissa seuraavan kerroksen syötteinä. Tästä johtuen aiempien ker-

rosten ominaisuuskartat toimivat seuraavien kerrosten syötteinä, ja jos täyttämismenetelmiä ei käytetä, konvoluutiokerrosten syötteiden koko pienenee verkon syventyessä. Tämä on usein konvoluutioneuroverkoissa kuitenkin haluttu ominaisuus, sillä syötteen pieneneminen tarkoittaa syvempien kerrosten käsittelevän suurempaa aluetta alkuperäisestä neuroverkkoon syötetystä kuvasta, vaikka konvoluutioikkunan koko pysyisikin samana joka kerroksella. Näin ollen syvemmät konvoluutiokerrokset pystyvät yhdistämään aiempien kerrosten alkeellisia muotoja ja tunnistamaan monimutkaisempia muotoja verkon syvetessä (20).

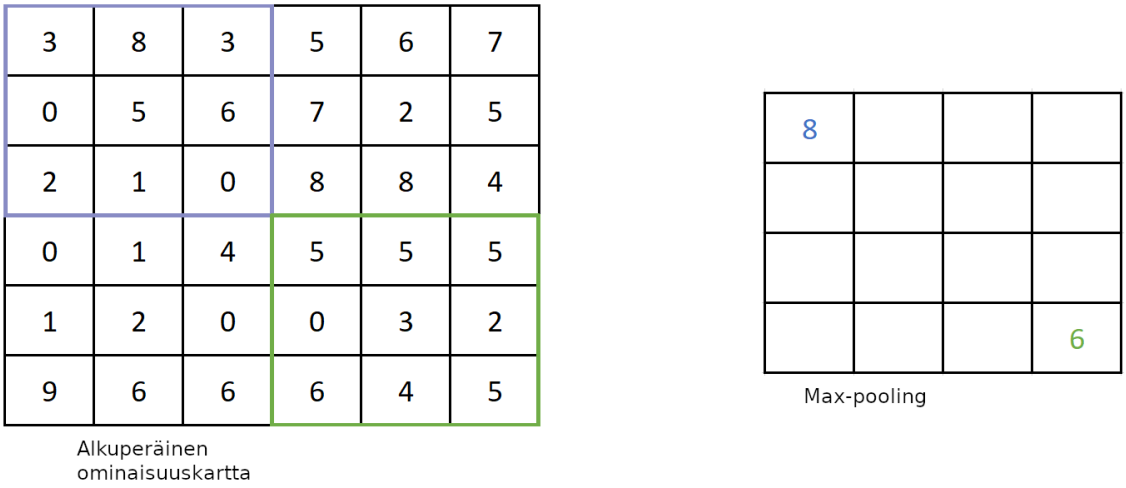
Toisin kuin konvoluutiota käyttämättömissä eteenpäin kytketyissä neuroverkoissa, konvoluutioneuroverkoissa jokainen edelliskerroksen syöte ei kytkeydy jokaiseen seuraavan kerroksen neuroniin (47). Tämän takia konvoluutiokerrosten väliset yhteydet ottavat huomioon alkioiden sijainnin syötteessä, toisin kuin eteenpäin kytkettyjen neuroverkkojen täysin yhdistetyt neuronikerrokset (11).

Yhdistämiskerros (Pooling layer) toimii hieman luvussa 2.1 esitellyn digitaalisen konvoluution tavoin. Yhdistämiskerrokset pienentävät kuvaa verkon syventyessä käyttämällä kaksiulotteisen digitaalisen konvoluution tavoin syötteen päälle laitettavia ikkunoita ja suorittamalla ikkunoiden alle jääville arvoille yhdistämisfunktion, jolla saadaan arvo tulosominaisuuskarttaan ikkunan keskipistettä vastaavaan kohtaan. Yleisiä yhdistämisfunktioita ovat max-pooling, joka yksinkertaisesti ottaa ikkunan alle jäävistä ominaisuuskartan arvoista suurimman, lisäten näin neuroverkon epälineaarisuutta, sekä average-pooling, joka laskee ikkunan alle jäävistä arvoista keskiarvon (20).

Yhdistämiskerroksen ulostulon ulottuvuudet ovat syötteen ulottuvuuksia pienemmät reunojen poisjäämisen takia, kuten täyttämismenetelmiä käyttämättömässä konvoluutiossa (11). Tästä syystä yhdistämiskerrosten käyttö mahdollistaa suurempien kokonaisuuksien hahmottamisen. Lisäksi aktivaatiokarttojen pieneneminen verkon syvetessä vähentää neuroverkon parametreja, mikä puolestaan vähentää laskentaan kuluvaa aikaa (11). Yhdistämiskerrokset myös vähentävät datan sijainnin merkitystä ominaisuuskartassa, jolloin myöskään objektin sijainnilla alkuperäisessä kuvassa ei ole niin suurta merkitystä luokittelun kannalta (20).

Yhdistämiskerroksia ei aina käytetä jokaisen konvoluutiokerroksen jälkeen varsinkaan syvissä neuroverkoissa, sillä jokainen yhdistämiskerros vähentää seuraavan konvoluutiokerroksen syötteen kokoa, tehden konvoluution suorittamisesta mahdotonta syötteen ollessa liian pieni (20). Joissakin konvoluutioneuroverkon toteutuksissa yhdistämiskerroksia ei käytetä lainkaan, vaan kuvan pienentäminen suoritetaan pelkästään konvoluutiokerrosten avulla liikuttamalla konvoluutioikkunaa yli yhden pikselin verran kerrallaan

(20). Seuraavassa kuvassa on esitetty kaksi esimerkkitapausta max-poolingin arvon muodostamisesta.



Kuva 5: Max-poolingin tuottamia arvoja

Konvoluutioneuroverkoissa on konvoluutiota suorittavien piilokerrosten lisäksi myös tavallisissa eteenpäin kytketyissä neuroverkoissa käytettäviä piilokerroksia. Nämä piilokerrokset sijaitsevat konvoluutiopiilokerrosten jälkeen ja ne toimivat kuten muissakin neuroverkoissa, lisäten opittavien painokerrointen määrää ja tuottaen todennäköisyydet luokille esimerkiksi softmax-aktivointifunktioita hyödyntäen (20). Vaikka konvoluutiokerrokset sisältävät usein vain pienen osan neuroverkon kaikista oppimista parametreista, on niiden vaikutuksen todettu olevan suuri luokittelutulokseen, tehden konvoluutiokerroksista tehokkaan osan neuroverkkoja (45).

Konvoluutioneuroverkoissa voidaan usein luoda koulutusaineistossa käytettävistä kuvista koneellisesti erilaisia versioita ja näin parantaa verkon kykyä yleistää (20). Esimerkiksi koulutusjaksoissa satunnainen kuvien kääntäminen ja peilaaminen, liikuttaminen vaaka- tai pystysuunnassa ja kontrastin muutokset ovat mahdollisia tapoja lisätä koulutusaineiston varianssia (48). Koneellisia muunnoksia lisätessä täytyy kuitenkin olla varovainen, sillä esimerkiksi tekstintunnistuksessa b-kirjaimen peilaaminen tuottaa d-kirjaimen, mikä sekoittaa kouluttamista.

Koulutusnäytteiden ei tarvitse koostua vain yhdestä tietotyypistä, vaan näyte voi sisältää esimerkiksi kuvan ja siihen liittyvää tekstimuotoista dataa. Metadatan käytöllä konvoluutioneuroverkoissa kuvien lisänä on saatu luokittelutarkkuutta nostettua planktonien tunnistuksessa (41). Metadata tuo lisäparametreja verkon opittavaksi ja näin kasvattaa jokaiseen koulutusjaksoon kuluvaa aikaa, mutta voi myös vähentää tarvittavien koulutusjaksojen määrää. Lisäksi neuroverkon metadatasta riippumattomat parametrit vaikuttavat huomattavasti enemmän koulutusaikaan kuin metadatan parametrit (41). Metadatan

pitäisi tuoda luokittelun kannalta lisäinformaatiota, joka auttaa näytteiden erottelemisessa toisistaan, esimerkiksi jos kaikkien näytteiden kasvupaikka on sama, kasvupaikkatieto ei tuo apua tunnistukseen.

Konvoluutiokerrokset toimivat neuroverkoissa kuvan ominaisuusvektoreiden tuottajina. Koska neuroverkkojen ensimmäisten konvoluutiokerrosten tehtävä on tunnistaa kuvista yksinkertaisia muotoja, joita esiintyy useimmissa kuvissa, on mahdollista käyttää toiseen käyttötarkoitukseen koulutetun neuroverkon alkuosaa uuden neuroverkon koulutuksen pohjana (20). Tämä tapa alustaa uusi neuroverkko vanhan painokertoimilla on erityisen hyödyllinen silloin, kun uuden verkon kouluttamiseen on käytettävissä vain pieni määrä dataa, mutta vanha verkko on koulutettu suurella datamäärällä (11). Tällöin uuden verkon alustetut aikaisempien kerrosten painokertoimet voidaan joko jättää vakioiksi, jolloin koulutus ei muuta kuin syvempien kerrosten painokertoimia (20). Toinen vaihtoehto on sallia alustettujen painokerrointen muuttaminen, jolloin virheen minimi löydetään usein aikaisemmin kuin satunnaisilla luvuilla alustetuilla painokertoimilla (11). Yleisesti, käyttämällä toisen neuroverkon painokertoimia uuden verkon pohjana voidaan siis vähentää neuroverkon koulutukseen kuluvaa aikaa.

Joskus konvoluutioneuroverkkoja ei käytetä luokitteluun ollenkaan, vaan verkon alkuosaa käytetään ominaisuusvektorien keräämiseen, jotka syötetään tunnistettavaksi toiselle koneoppimisluokittelijalle kuten tukivektorikoneelle (49). Pelkästään konvoluutioneuroverkon alkuosaa käyttämällä, konvoluutioneuroverkot soveltuvat luokittelun ja regression lisäksi myös muihin konenäön sovelluksiin ja monen koneoppimisluokittelijan järjestelmiin.

2.5 Puiden tunnistaminen

Puulajien tunnistaminen konenäöllä on yleisesti vaikea tehtävä. Saman puulajin yksilöt voivat näyttää hyvinkin erilaisilta riippuen puun ominaisuuksista kuten iästä, terveydestä ja kasvupaikasta. Toisaalta eri puulajeilla voi olla samannäköinen kaarna, mikä tekee rungon perusteella lajin tunnistamisesta vaikeaa (6).

Kuvien käyttämisen lisäksi metsien omistajilla on olemassa usein lisätietoa muusta puiden kasvuun vaikuttavasta tiedosta, kuten metsän iästä ja sijainnista, jota voidaan hyödyntää lisätietona lajien tunnistamisessa. Kasvupaikan huomioon ottamista vaikeuttaa puiden kasvaminen eri tavalla riippuen ympäröivästä kasvillisuudesta tai sen puutteesta (5), joten pelkkä maantieteellinen sijainti ei välttämättä määritä puiden ulkonäköä. Lisäksi kuvanottohetki vaikuttaa puun rungon ulkonäköön suuresti, esimerkiksi kellonaika, sää

ja vuodenaika muuttavat kuvaa usein merkittävästi. Reaalimaailmassa runkojen tunnistusta vaikeuttaa myös mahdolliset puut osittain peittävät objektit, kuten lehdet ja muu kasvusto. Kuvat voivat olla myös epätarkkoja, jos kamera on liikkunut kuvanottohetkellä. Monta puuta sisältävissä kuvissa kameran kohdennusetäisyyden takia runkojen tarkkuus saattaa myös poiketa merkittävästi saman kuvan sisällä, jos puut ovat eri etäisyydellä kamerasta.

Kuva 6: Peittävä kasvusto tai lumi voi vaikeuttaa lajien tunnistamista puun rungoista

Puulajien tunnistusta voidaan tehdä rungon lisäksi myös lehdistä tai neulasista (4). Tämä ei kuitenkaan toimi yleisesti, sillä vuodenajasta ja paikasta riippuen puilla ei välttämättä ole lehtiä. Näin ollen runko on ainoa osa puusta, joka on aina näkyvissä ja siten paras kohta tunnistaa puun laji. Runko myös pysyy tunnistettavana senkin jälkeen, kun puu on leikattu tukeiksi, mikä mahdollistaa saman tunnistusjärjestelmän käyttämisen myös jatkokäsittelyssä.

Ihmisasiantuntija käyttää puulajien tunnistuksessa rungoista pääsääntöisesti kaarnan väriä ja kuviointia (50). Kaarnan väri kuitenkin muuttuu vuodenaikojen mukaan, joten se ei ole aina luotettavin tunnistuksen kohde (4). Oletettavaa on siis konenäönkin painottavan tunnistuksessa kaarnan kuviointia tietoaineiston ollessa tarpeeksi monipuolinen.

Ihmisasiantuntijan paikan päällä tekemän tunnistuksen ja konenäön suorittaman puulajitunnistuksen suuri ero on ihmisasiantuntijan kyky käyttää muutakin tietoa puusta kuin vain rungon ulkonäköä. Esimerkiksi ihmiselle on mahdollista käyttää tunnistuksessa apuna ympäröivää kasvustoa, puun kokoa tai puun kaarnan tuntua. Näin ollen asiantuntijalla on usein paljon enemmän tietoa käytettävissään kuin mitä koneoppimisjärjestel-

mälle on edes mahdollista antaa. Paikan päällä olevaan asiantuntijaan ei myöskään vaikuta kameran aiheuttamat värien vääristymät tai huonot kuvakulmat, jotka estävät rungon näkemisen.

3. AIEMMAT TUTKIMUKSET AIHEESTA JA AINEISTO

Koneoppimista ja neuroverkkoja on tutkittu jo vuosikymmeniä, mutta käytännöllisten koneoppimista hyödyntävien sovellusten kehitys on lähtenyt käyntiin vasta tällä vuosituhannella (23). Ala on nopeasti kehittyvä ja varsinkin konvoluutioneuroverkkojen kehitys on riippuvainen myös yleisestä tietokoneiden laskentatehon kehityksestä. Historiallisesti neuroverkkojen kehitys on saanut ideoita neurotieteestä, joten uudet löydökset aivojen toiminnasta voivat tuoda uusia ideoita myös neuroverkkojen kehitykseen.

3.1 Aineiston valinta

Nopeasti muuttuvana alana, aineistoa valitessa suosittiin uudempia, viimeisen kymmenen vuoden aikana kirjoitettuja lähteitä varsinkin neuroverkkojen osalta. Toisaalta koneoppimisen ja kuvankäsittelyn matemaattiset perusteet eivät ole muuttuneet ajan saatossa, joten näiltä osin kelpuutettiin myös vanhemmat lähteet. Yleisesti lähdeaineistoa valitessa yritettiin suosia paljon viitattuja lähteitä. Lähteet ovat suurimmaksi osaksi tieteellisiä julkaisuja, joissakin tapauksissa valmiin toteutuksen esittelemisen vuoksi lähteinä käytettiin myös lähteitä tieteellisten julkaisujen ulkopuolelta.

3.2 Aiemmat aiheeseen liittyvät tutkimukset

Puulajitunnistusta tehdään usein ilmasta otettujen kuvia käyttämällä lasermittauksia hyödyntämällä. Ballantin tutkimusryhmän tutkimuksessa (51) tutkittiin tukivektorikone- ja satunnaismetsäluokittelijoilla puulajien luokittelua hyperspektrikuvia käyttäen. Tutkimuksessa saavutetaan yli 90 % luokittelutarkkuus kahdeksan puulajin aineistolle. Tuloksista käy myös ilmi tukivektorikoneen olevan satunnaismetsää parempi luokittelija varsinkin, kun koulutusnäytteiden määrä on alhainen. Ilmakuvien ansiosta yksittäisten puiden sijainti voidaan piirtää kartalle, joten puulajikartan segmentointi on mahdollista.

Puulajitunnistusta on tehty myös puun latvoista otettuja korkearesoluutioisia satelliittikuvia syötteenä käyttäen. Yanin tutkimusryhmän (46) suorittamassa kokeessa konvoluutioneuroverkkojen suoriutuivat kuuden puulajin luokittelutehtävästä merkittävästi satunnaismetsä- ja tukivektorikoneluokittelijoita paremmin, saavuttaen 82,7 %:in luokittelutarkkuuden.

Korkearesoluutioisia, 224 × 224 px suuruisia, puun rungon kuvia konvoluutioneuroverkkojen kanssa käyttänyt Carpentierin tutkimusryhmä (5) saavutti 23 puulajin aineistolle

korkean 97,81 %:n tarkkuuden käyttämällä luokittelussa useampaa kuvaa yksittäisestä puusta. Yli 800 000 kuvan aineiston keränneessä tutkimuksessa käy ilmi saman puun rungon eri kohdista otettujen kuvien parantavan luokittelutulosta varsinkin, kun kuvat on otettu rungon eri kohdista.

4. TUTKIMUSMENETELMÄT

Työssä tutkittiin viiden puulajin luokittelua. Käytetyt luokat olivat haapa, koivu, eukalyptus, mänty ja kuusi. Näistä eukalyptusta lukuun ottamatta kaikki ovat yleisiä Suomen metsissä esiintyviä puulajeja. Puiden luokittelu siis tapahtui näihin viidellä edellä mainittuun luokkaan, eikä esimerkiksi koivuja luokiteltu edelleen raudus- ja hieskoivuiksi.

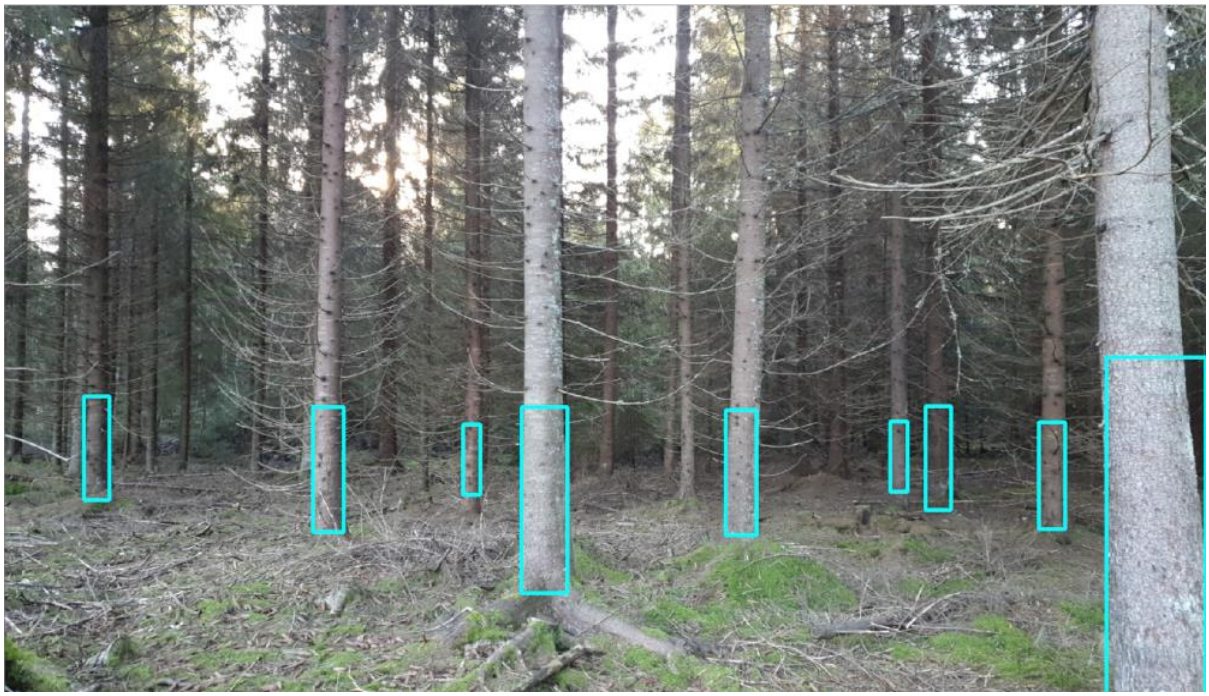
4.1 Käytettävä aineisto

Työssä käytettiin Trestima Oy:n keräämää valokuvista ja niiden metadatasta muodostuvaa tietoaineistoa. Tietoaineiston kuvat olivat valokuvia metsistä, joissa pääsääntöisesti oli monta puuta yhtä valokuvaa kohden. Kuvat olivat tavallisilla, kuluttajakäyttöön suunnatuilla kännykkäkameroilla otettuja kuvia, joiden resoluutio oli useimmiten 1280 × 720 px tai 1600 × 900 px. Lisäksi kuvista oli annettu erillisinä JSON-tiedostoina kuvakoordinaatit, joista kävi ilmi runkojen sijainnit ja puulajit kuvissa, sekä lisätietoa, kuten kuvan GPS-koordinaatit ja aika. Ennen kuin kuvia voitiin syöttää koneoppimisjärjestelmälle, rungot piti leikata valokuvista niin, että yksittäisessä kuvassa oli vain yhden puun rungon kuva. Rungot, joiden kuvakoordinaattien muodostamat alueet menivät päällekkäin, jätettiin pois aineistosta, sillä niiden luokasta ei ollut varmuutta. Runkokuvia saatiin leikkausten jälkeen yhteensä käyttöön 118 000 kappaletta tasajakautuneesti viiden eri luokan välille.

Puun runkojen kuvat leikattiin irti useita puita sisältävistä metsäkuvista, mikä tarkoitti runkojen leveyksien poikkeavan toisistaan huomattavasti. Pienimpien runkojen leveys oli 4 pikseliä, suurimpien runkojen leveys jopa 121 pikseliä. Runkojen leveyksien suuri pikselimääräinen varianssi johtui puiden leveyksien vaihtelun lisäksi kuvanottoetäisyyksien vaihtelusta. Neuroverkko vaatii syötteiden pysyvän vakiokokoisina, joten kuvat skaalattiin kokoon 24 × 48 px neuroverkon syötekerrosta ennen, mikä tarkoitti syötekerroksessa 1 152-ulotteista ominaisuusvektoria. Kuvien skaalaus pienimmän leveyden mukaan olisi tarkoittanut 4 × 8 px suuruisia kuvia, joissa ei ole neuroverkolle tarpeeksi tietoa luokittelun oppimiseen. Kuvien skaalaaminen alkuperäistään suuremmaksi tekee kuvista epätarkempia, epätarkkuuden kasvaessa mitä suuremmiksi kuvat skaalataan.

JSON-tiedostoista saadut runkojen x-koordinaatit kattoivat koko rungon leveyden, mutta y-koordinaatit vain muutaman pikselin korkeuden rungosta. Jos rungot olisi leikattu käyttäen annettuja y-koordinaatteja, olisi neuroverkkoon syötettävät kuvat olleet usein liian pieniä, jotta niistä olisi saanut koulutettua toimivaa mallia. Ottamalla y-koordinaateista

keskiarvo ja vähentämällä saadusta y-koordinaatista rungon leveyden monikerta saatiin uusi yläraja runkokuvalle. Vastaavasti lisäämällä rungon leveyden monikerta y-koordinaattien keskiarvoon saatiin alaraja runkokuvalle. Näin ollen rungoista saatiin kuvat, joiden kaikkien mitat olivat keskenään vaihtelevat, mutta kuvasuhde sama, mikä teki kuvien skaalaamisesta samankokoisiksi neuroverkkoa varten suoraviivaista ja mahdollisimman vähän vääristymiä aiheuttavaa.



Kuva 7: Runkojen irrotus kuvasta JSON-tietojen perusteella. Runkojen korkeus on tässä nelinkertainen niiden leveyteen verrattuna.

Testejä pääsääntöisesti rakennettiin toistensa päälle niin, että sen hetkisen parhaimman tuloksen tuottavia menetelmiä käytettiin pohjana uusien menetelmien testaamiselle. Esimerkiksi aliluvussa 5.3 käsiteltävistä skaalaus- ja normalisointimenetelmistä valittiin tarkimman testitarkkuuden tuottavat menetelmät, joita käytettiin myös myöhemmissä aliluvuissa, ellei toisin mainita.

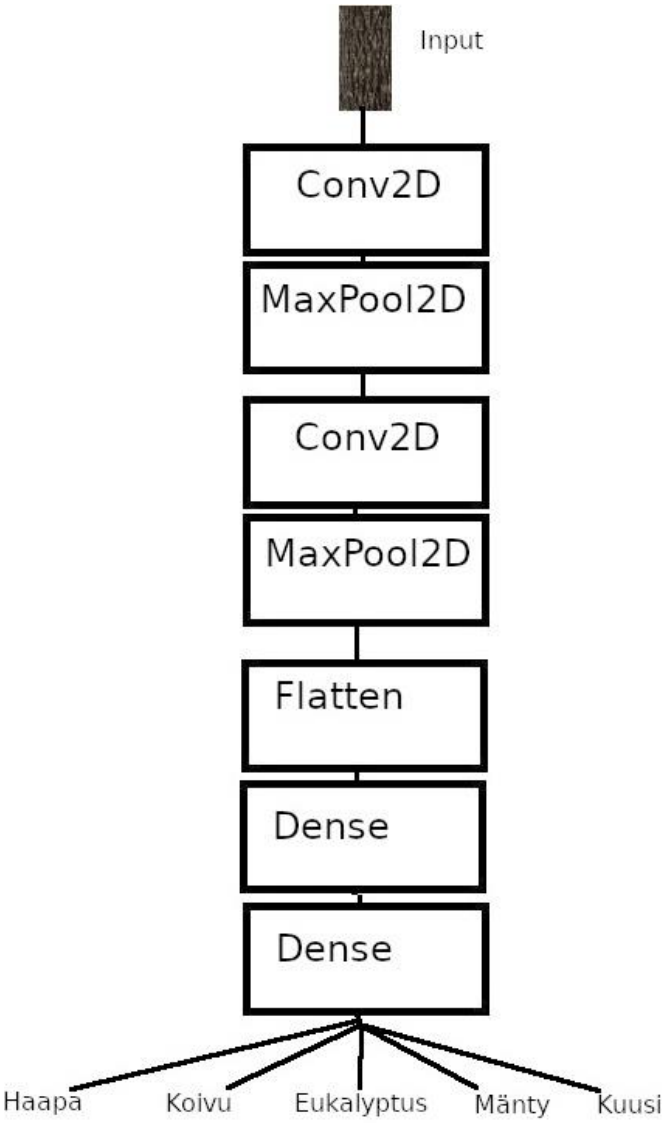
4.2 Tutkimusympäristö

Koneoppimisluokittelijat toteutettiin käyttäen Python-ohjelmointikieltä. Neuroverkkoluokittelijoiden luomisessa käytettiin avoimen lähdekoodin TensorFlow-kirjaston versiota 2.1.0 (52). Tukivektorikoneluokittelijoiden toteuttamisessa käytettiin avoimen lähdekoodin Scikit-learn-kirjaston versiota 0.21.3 (53).

TensorFlow-luokittelijat koulutettiin käyttäen erillistä Nvidia GTX 1070-näytönohjainta laskennassa. Testit voisi teoriassa myös toteuttaa ilman erillistä näytönohjainta, mutta tällöin koulutukseen kuluva aika moninkertaistuisi.

4.3 Neuroverkon toteutus

Ensimmäiseksi versioksi neuroverkosta toteutettiin kahdesta konvoluutiotasosta koostuva luokittelija, jonka rakenne on esitelty alla olevassa kuvassa.



Kuva 8: Toteutetun neuroverkon rakenne

Konvoluutio- ja max-pooling-kerrosten ikkunoiden kooksi valittiin 2 × 2 px, ja näiden välissä tapahtuviksi aktivointifunktioiksi ReLU. Flatten-kerros toimi konvoluutiokerrosten ja loppuneuroverkon välisenä siltana muuttamalla viimeisen konvoluutiokerroksen tuloksen yksiulotteiseksi vektoriksi. Dense-kerrokset olivat konvoluutiota suorittamattomia neuroverkon kerroksia. Ensimmäisen dense-kerroksen tarkoitus oli parantaa luokittelutulosta oppimalla painokertoimia, ja se koostui 128 neuronista ja ReLU-aktivointifunktiosta. Jälkimmäisen dense-kerroksen tehtävä oli tuottaa luokittelun todennäköisyydet, ja se koostui käytettävien luokkien lukumäärän mukaisesti viidestä neuronista ja softmax-aktivointifunktiosta.

Häviöfunktioksi neuroverkolle valittiin kategorinen ristientropia. Konvoluutiokerrosten luomien ominaisuuskarttojen määräksi valittiin 32 jokaiselle konvoluutiokerrokselle. Optimoijana käytettiin adaptiivista Adam-optimoijaa oppimisnopeudella 0,0001. Näiden parametrien vaikutusta luokittelutulokseen käsitellään aliluvussa 5.7 Konvoluutioverkon parametrit.

Suoritettavien koulutusjaksojen määräksi maksimimääräksi asetettiin 1000. Koulutus kuitenkin lopetettiin automaattisesti, jos validointitarkkuus ei noussut 30 jakson ikkunalla tarkasteltuna. Käytännössä koulutusjakojen maksimia ei saavutettu yhdelläkään koulutuskerralla ja koulutus pysähtyi pisimmilläänkin koulutusjaksolla 215.

5. TULOKSET

Saadut luokittelutulokset riippuivat käytetystä tietoaineiston suodatusmenetelmistä sekä esiprosessointimenetelmistä. Tulokset on esitelty kronologisessa testien suoritusjärjestyksessä.

5.1 Kuvien jakaminen koulutus-, validointi- ja testausaineistoihin

Käyttäen tietoaineiston 23 600 näytettä per luokka, näytteet jaettiin koneoppimisluokittelijoiden koulutusta varten aliaineistoihin. Kuvat jaettiin ensiksi 90:5:5 suhteella koulutus-, validointi- ja testausaineistoihin. Tällä jaolla saatiin testaustarkkuudeksi neuroverkosta 0,268, mikä on hieman korkeampi kuin tulos 0,200, joka saataisiin arvaamalla kaikki näytteet yhdeksi luokaksi tai keskimäärin arvaamalla näytteiden luokka satunnaisesti.

Jakamalla kuvat uusiksi suhteessa 60:20:20 saatiin testaustarkkuudeksi 0,452. Sen lisäksi, että luku on huomattavasti edellisen jaon tulosta korkeampi, voidaan tulosta myös pitää luotettavampana, sillä suurempi testiaineisto vähentää yksittäisen kuvan vaikutusta luokittelutulokseen. Tätä aineiston jakosuhdetta käytettiin kaikissa lopuissa testeissä neuroverkoilla ja myös tukivektorikonetta koulutettaessa.

Syötteenä kokeiltiin käyttää myös mustavalkoiseksi muunnettuja kuvia, jolloin tulokseksi saatiin 0,484. Saatu tulos oli korkeampi kuin vastaava värikuvilla saatu tulos, mutta koska tässä vaiheessa ei vielä ollut mitään esiprosessointia kuville, tutkimuksia jatkettiin myös värikuvia käyttämällä.

Tuloksista nähdään neuroverkon huono yleistämiskyky testiaineiston kanssa, jos tietoaineiston jaossa on jätetty liian vähän näytteitä validointia varten. Neuroverkon koulutukseen kuluva aika oli tässä vaiheessa noin 1,5 h.

5.2 Tukivektorikone

Vertailun vuoksi puiden luokittelua kokeiltiin myös kouluttamalla yksinkertainen tukivektorikoneluokittelija yksi vastaan muut-periaatteella. Käyttämällä samaa tietoaineistoa kuin neuroverkkojen koulutuksessa, saatiin luokittelutarkkuudeksi mustavalkokuvilla 0,281. Alhaisen luokittelutarkkuuden lisäksi tukivektorikone ei ollut erityisen hyvä erottamaan minkään yksittäisen luokan edustajia, kuten seuraavasta sekoitusmatriisista nähdään.

Taulukko 2: Sekoitusmatriisi tukivektorikoneen tuloksille

	Haapa	Koivu	Eukalyptus	Mänty	Kuusi
Haapa	471	421	778	766	764
Koivu	295	473	708	424	1300
Eukalyptus	208	338	1622	357	675
Mänty	315	379	739	973	794
Kuusi	241	365	873	785	936

Jos tukivektorikone olisi ollut erityisen tarkka erottamaan jotkut kaksi luokkaa toisistaan, luokittelijaa olisi voinut hyödyntää neuroverkkojen lisänä rakentamalla useampaa eri luokittelijaa käyttävän luokittelujärjestelmän. Tasaisesti alhaisten luokittelutarkkuuksien takia tämä ei kuitenkaan vaikuttanut toimivalta idealta, ja tukivektorikoneiden tutkiminen ongelman yhteydessä lopetettiin.

Tukivektorikoneen tuloksia olisi voinut parantaa syötteenä raakakuvien sijaan kuvista luotavien käsintehtyjen ominaisuusvektoreiden käyttäminen. Tämä olisi kuitenkin vaatinut paljon aikaa ja sovelluskohtaista osaamista, jonka lisäksi kirjallisuuden perusteella tulokset olisivat siltikin jääneet huonommiksi kuin konvoluutioneuroverkoilla.

5.3 Skaalaus ja normalisointimenetelmät

Raakakuvat koostuivat kolmesta värikanavasta, jotka saivat kokonaislukuarvoja väliltä [0, 255]. Jakamalla kaikki kuvan värikanavien arvot suurimmalla mahdollisella arvolla 255:llä, saatiin liukulukumuotoinen esitys, jossa jokainen värikanava koostui desimaaliluvuista väliltä [0, 1]. Tämä niin sanottu ominaisuusskaalaus nosti testaustarkkuuden 0,561:een, mikä on noin kymmenen prosenttiyksikön kasvuna erittäin merkittävä parannus. Ominaisuusskaalauksen merkitystä testattiin myös yksikanavaisille mustavalkoku-

ville, jolloin testaustarkkuudeksi saatiin 0,538. Värikuvien ominaisuusskaalaus tuotti selvästi tarkemman luokittelutarkkuuden, joten loput tutkimusmenetelmät suoritettiin näitä kuvia käyttäen.

Ominaisuusskaalauksen lisäksi jokaiselle ominaisuudelle, eli tässä tapauksessa kuvien pikseleille, laskettiin keskihajonta koko koulutusaineiston yli. Tämän jälkeen tietoaineiston kuvien pikseliarvot jaettiin niitä vastaavilla keskihajonnoilla. Keskihajonnoilla normalisoiduilla kuvilla testaustarkkuudeksi saatiin näin 0,548. Käyttämällä keskihajonnan normalisoinnin sijaan ZCA-vaalentamista testaustarkkuudeksi saatiin 0,513. Keskihajontanormalisointi ja ZCA-vaalentaminen tuottivat huonomman luokittelutuloksen kuin jos menetelmiä ei olisi käytetty, joten niitä ei käytetty enää myöhemmissä tutkimuksen vaiheissa.

5.4 Väriavaruudet

Kuvien muuntaminen RGB-väriavaruudesta yleisesti käytettyihin HSV- ja CIELAB-avaruuksiin vaikutti testitarkkuuteen tulosta huonontavasti. HSV-avaruuteen muunnetuilla kuvilla kaikkia kanavia syötteenä käyttäen testitarkkuudeksi saatiin 0,509. Histogrammin tasoittamisen vaikutus kokeiltiin tekemällä se V-kanavalle, jolloin luokittelutulokseksi saatiin edellisen kanssa lähes identtinen 0,510. Luokittelutarkkuutta testattiin myös pudottamalla osa kanavista pois, sillä valaistuksen intensiteettien vaihtelu olisi voinut vaikuttaa luokittelutulokseen tulosta heikentävästi. Pelkkää H-kanavaa käyttäen testitarkkuudeksi saatiin 0,322, ja H- ja S-kanavia käyttäen testitarkkuudeksi saatiin 0,318. Kaikki saadut tulokset olivat RGB-avaruuden vastaavia tuloksia huonompia ja varsinkin kanavien poisjättäminen heikensi tulosta merkittävästi.

CIELAB-avaruudessa luokittelutarkkuudet olivat selkeästi alhaisempia. Kaikkia CIELAB:in kolmea kanavaa käyttäen luokittelutarkkuudeksi saatiin 0,250. Jättämällä intensiteettiä kuvaava L-kanava pois saatiin luokittelutarkkuudeksi edellisen tuloksen kanssa lähes identtinen 0,249.

Väriavaruuksien muunnokset eivät siis tuottaneet parempaa luokittelutarkkuutta ja näin ollen tietoaineistojen kuvat pidettiin RGB-avaruuden esityksinä.

5.5 Koulutusaineiston suodatus

Neuroverkon koulutusta kokeiltiin myös poistamalla koulutusaineistoista kuvia, jotka mahdollisesti huonontaisivat verkon lajittelukykyä. Kuvia poistettiin aineistosta niiden alkuperäisen leveyden, kontrastin ja vihreiden pikseleiden määrän mukaan.

Epätarkkojen kuvien käytössä nousi mahdolliseksi ongelmaksi luokittelijan olevan kykenemätön luokittelemaan kuvia toisistaan, jos kuvat näyttivät ylöspäin skaalauksen jälkeen samoilta. Tämän takia neuroverkon koulutusta kokeiltiin poistamalla koulutusaineistosta alle 12 pikseliä leveät kuvat, jolloin kuvien skaalaaminen 24 pikselin levyiseksi vaatisi enimmillään kaksinkertaisen skaalauksen. Tämän jälkeen koulutusaineiston lajijakauma tasattiin poistamalla satunnaisesti näytteitä luokista, kunnes jokaisella luokalla oli sama määrä koulutusnäytteitä. Huomattavaa on myös, että näytteet poistettiin vain koulutusaineistosta, ei validointi- tai testiaineistosta, sillä luokittelijan tarkkuutta haluttiin testata myös pienillä puilla. Tällä suodatetulla koulutusaineistolla saatiin testitarkkuudeksi 0,563, mikä oli merkityksetön parannus verrattuna edelliseen, kaikki näytteet sisältävällä aineistolla saatuun tulokseen 0,561. Kuvien leveyden perusteella koulutusaineiston suodatusta kokeiltiin myös poistamalla alle 24 pikseliä leveät kuvat, jolloin testaustarkkuudeksi saatiin 0,531. Aineiston suodatuksessa alle 12 pikselin levyisten kuvien poisto pidettiin myöhempiä menetelmiä testattaessa, sillä vaikka tulos ei merkittävästi parantunut, koulutusaineisto pieneni, joka puolestaan vähensi koulutukseen kuluvaa aikaa.

Koulutusaineistoa koitettiin monipuolistaa lisäämällä koulutusaineistoon satunnaisesti vaakasuunnassa peilattuja kuvia olemassa olevista kuvista. Tällä menetelmällä testaustarkkuus saatiin nousemaan 0,572:een.

Koulutusaineiston suodatusta jatkettiin yrittämällä suodattaa lehtien ja kasvillisuuden peittämiä runkoja pois aineistosta. Tätä varten kuvat muutettiin väliaikaisesti HSV-väriavaruuteen ja määritettiin rajat vihreälle värille, joiden avulla yritettiin määrittää, koska runko on lehden peittämä. Silmämääräisesti määrittelemällä vihreyden alarajaksi saatiin HSV-arvo (40, 100, 0) ja ylärajaksi (80, 255, 80). Tämän jälkeen jokaisen kuvan jokainen pikseliarvo tarkistettiin, kuuluuko se vihreiden pikseleiden joukkoon. Lopulta koulutusaineistosta poistettiin ne kuvat, joissa yli 50 % pikseleistä määritettiin vihreiksi. Luokittelutulokseksi saatiin vihreät kuvat koulutusaineistosta poistamalla 0,572, joten tulos ei parantunut, mutta koulutusaineiston koko pieneni, joten tätä suodatusta käytettiin lopuissa testimenetelmissä.

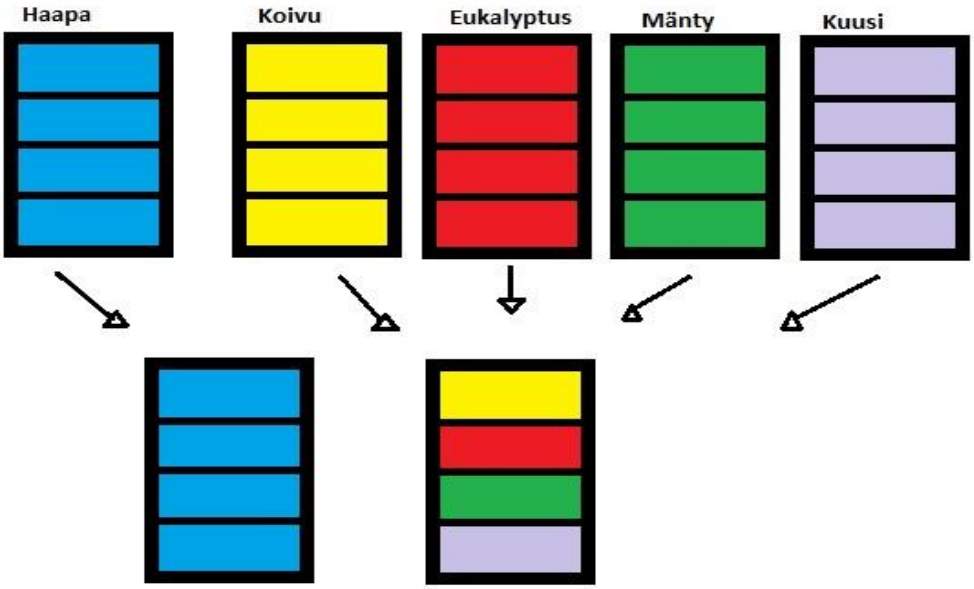
Kuva 9: HSV-avaruuden avulla määritettyjä vihreän sävyjä lehtien erottamista varten

Kuville laskettiin myös kontrasti, jonka perusteella koulutusaineistoa yritettiin suodattaa. Tavoitteena oli poistaa huonosta valaistuksesta johtuvat liian vaaleat tai tummat kuvat, jotka näyttäisivät samoilta luokasta riippumatta. Ennen ominaisskaalauksen tekemistä, kuvista etsittiin suurimman ja pienimmän intensiteetin omaava pikseli ja jos näiden erotus oli alle 50, merkittiin kuva huonokontrastiseksi ja poistettiin koulutusaineistosta. Luokittelutulokseksi saatiin tällä koulutusaineistolla 0,575. Kaikkien koulutusaineiston suodatusmenetelmien jälkeen koulutusaineiston kooksi jäi 11 085 kuvaa luokkaa kohden.

Väärin luokiteltuja kuvia tutkimalla havaittiin, että lumen peittämät rungot olivat vaikeasti tunnistettavia, sillä ne menivät usein sekaisin koivujen kanssa. Lumen peittämien runkojen poistaminen aineistosta koneellisesti on kuitenkin vaikea ongelma, sillä esimerkiksi valkoisten pikselien määrää tutkimalla tullaan helposti poistaneeksi myös valkorunkoisia puita, mikä ei lehtien tunnistuksen kanssa ollut niin suuri ongelma.

5.6 Neuroverkkojen käyttö binääriluokittelijoina

Luokittelua kokeiltiin kouluttamalla oma neuroverkko jokaiselle viidelle luokalle yksi vastaan muut-periaatetta käyttäen, jossa neuroverkon ainoa tehtävä oli arvata, kuuluuko näyte kyseiseen luokkaan. Kokonaiskoulutusaineisto pidettiin samankokoisena kuin aikaisemmissa testeissä, joten jokaisen luokan koulutus- ja validointiaineisto jaettiin neljään osaan. Tämän jälkeen neuroverkot koulutettiin luokittain käyttämällä kaikkia kyseisen luokan koulutus- ja validointiaineistoa yhtenä luokkana ja neljäsosaa muista luokista toisena luokkana.



Kuva 10: Esimerkki haavan koulutusaineiston muodostamisesta binääriluokittelijalle

Seuraavassa taulukossa on esitetty jokaiselle verkolle saadut tulokset. Huomattavaa on, että koska luokittelija luokitteli kahden eri luokan välillä, jotka olivat yhtä suuria, on satunnaisen arvauksen todennäköisyys 0,5. Aiemmissa testeissä todennäköisyys satunnaisarvaukselle oli 0,2, joten tulokset eivät ole suoraan verrannollisia.

Taulukko 3: Yksi vastaan muut -luokittelijat

Luokittelija	Testitarkkuus
Haapa	0,671
Koivu	0,784
Eukalyptus	0,832
Mänty	0,695
Kuusi	0,712

Yksittäisten binääriluokittelijoiden kouluttamisen jälkeen, luokittelijat yhdistettiin yhdeksi luokittelijaksi ottamalla suurimman varmuuden antavan luokittelijan luokka lopulliseksi arvaukseksi. Tämän viiteen eri luokkaan luokittelevan binäärineuroverkkojen järjestelmän testitarkkuudeksi saatiin 0,556, mikä oli hieman sen hetkistä parasta luokittelutulosta huonompi. Johtopäätöksenä todettiin yhden, suoraan viiteen luokkaan luokittelevan neuroverkon olevan sekä yksinkertaisempi että tarkempi luokittelija, joten tutkimusta jatkettiin yhtä neuroverkkoa käyttäen.

5.7 Konvoluutioverkon parametrit

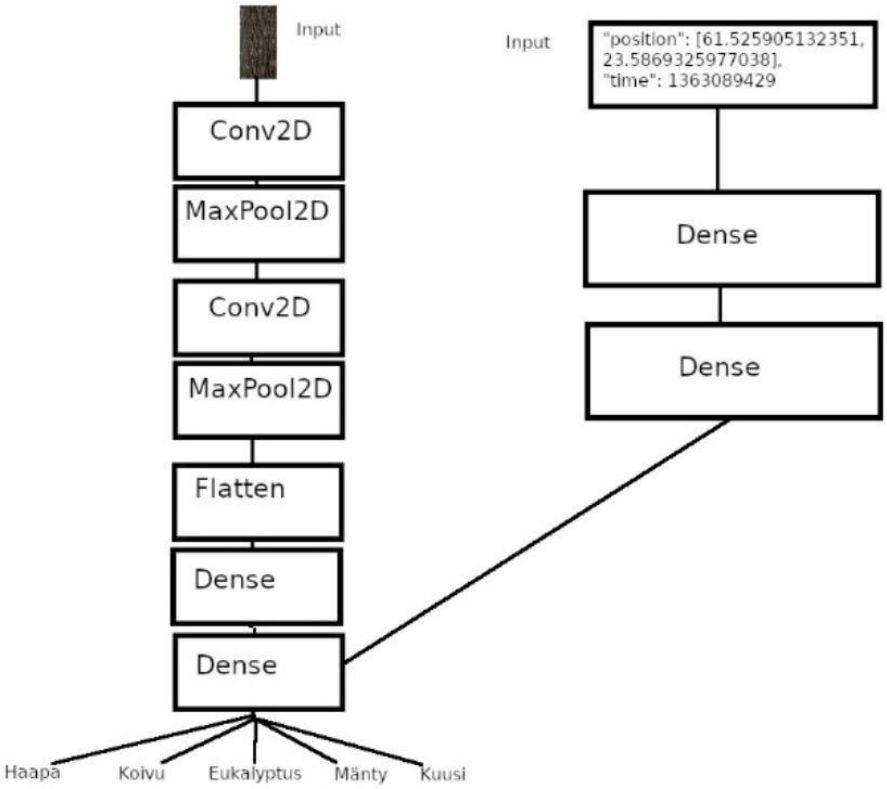
Konvoluutio- ja max-pooling-ikkunoiden kasvattaminen 3 × 2 px kokoisiksi nosti testitarkkuuden 0,589:ään. Konvoluutiokerroksissa käytetyn aktivointifunktio ReLU:n vaihtaminen ELU:un puolestaan nosti tarkkuuden 0,605:een. Tässä vaiheessa neuroverkon koulutukseen kuluva aika oli kasvanut noin kahteen tuntiin. Parametrejä olisi voinut kokeilla huomattavasti kattavamminkin, mutta koulutuksen kestosta ja ajallisista rajoituksista johtuen kokeiluissa siirryttiin eteenpäin.

5.8 Kuvien koko

Koska alkuperäisissä metsäkuvissa oli annettu puun runkojen leveyskoordinaatit täydellisenä, mutta korkeudesta vain osa, pystyi leikattavien runkokuvien korkeuden määrittämään jossain määrin vapaasti. Kasvattamalla runkokuvien korkeutta alkuperäisestä 1:2 suhteesta 1:4 suhteeseen, pysyivät rungot yhä leikkausikkunoiden sisässä ja näin ollen käytettyjen runkokuvien ulottuvuuksiksi saatiin 24 × 96 px. Kuvien korkeuden kasvattaminen kaksinkertaiseksi edelliseen verrattuna vaikutti merkittävästi luokittelutarkkuuteen, sillä uudeksi testitarkkuudeksi saatiin 0,641. Koulutukseen kuluva aika kasvoi tässä kohtaa noin kolmeen tuntiin per koulutuskerta, johtuen syötteiden koon ja näin myös neuroverkossa opittavien parametrien määrän kasvusta. Puut eivät aina kasva täysin suoraan, joten leikkausikkunoiden koon kasvattaminen vielä suuremmiksi olisi koulutuksen keston pidentämisen lisäksi mahdollisesti tuonut kuviin lajeista riippumatonta taustaa huonontamaan luokittelutulosta.

5.9 Metadatan hyödyntäminen

Kuvista oli käytettävissä pikseli-informaation lisäksi myös tietoa kuvanottohetkestä ja -paikasta. Tämän metadatan hyödyntämistä varten neuroverkkoa laajennettiin luomalla erillinen tekstimuotoista dataa syötteenä käyttävä monikerroksinen perseptroni. Kuvan pikseli-informaatiota käsittelevä CNN-haara ja tekstimuotoista informaatiota käsittelevä MLP-haara yhdistettiin lopullisen luokittelupäätöksen tekemistä varten seuraavan kuvan mukaisesti.



Kuva 11: Neuroverkon rakenne kuvien lisäksi metadataa käyttäen

MLP-haara koostui kahdesta peräkkäisestä neuronikerroksesta, joista ensimmäisessä oli kahdeksan ja toisessa viisi neuronia. Aktivointifunktiona molemmilla kerroksilla toimi ReLU. MLP-haara oli näin ollen huomattavasti CNN-haaraa kevyempi, mikä on perusteltua, sillä MLP-haaran syötedata oli selvästi pienempiulotteista kuin CNN-haaran 2304-ulotteinen data.

Käyttämällä metadatainformaationa kuvien GPS-koordinaatteja, saatiin uudeksi luokittelutarkkuudeksi yli kahdeksan prosenttiyksikköä edellistä parasta tulosta tarkempi 0,726. Kuten seuraavasta sekoitusmatriisista huomataan, luokittelija tunnisti eukalyptukset selvästi muita luokkia tarkemmin.

Taulukko 4: Sekoitusmatriisi paikkainformaatiota käyttäen

	Haapa	Koivu	Euka-lyptus	Mänty	Kuusi
Haapa	1879	451	4	392	474
Koivu	204	2477	1	203	315
Euka-lyptus	3	0	3192	4	1
Mänty	348	344	40	1963	505
Kuusi	301	413	2	385	2099

Eukalyptus tunnistettiin hyvin, sillä eukalyptusta ei kasva Suomessa luonnossa toisin kuin neljää muuta lajia. Koska eukalyptuksen tunnistaminen Suomen lajeista on harvoin käytännön sovelluksissa ilmenevä käyttötapaus, tiputettiin eukalyptus pois aineistosta ja loput testit suoritettiin neljää Suomessa esiintyvää puulajia käyttäen.

Ilman metadatan käyttämistä testitarkkuudeksi neljää luokkaa käyttäen saatiin 0,647. GPS-koordinaattien lisääminen syötteeksi paransi tulosta 0,663:een. Tuloksista huomataan paikkainformaation vaikuttavan luokittelutulokseen positiivisesti myös silloin, kun kaikkien luokkien olemassaolo alueella on todennäköistä. Tämä tulos ei ole yllättävä, sillä myös Suomen rajojen sisällä puulajien kasvu riippuu kasvupaikasta.

Käyttämällä kuvien aikatietoja paikkainformaation sijaan saatiin luokittelutulokseksi 0,652, kun metadatana ollut Unix-time-muotoiset muuttujat muutettiin niitä vastaaviksi kuukausiksi. Tarkoituksena kuukausiksi muuntamisessa oli saada neuroverkko oppimaan runkojen ulkonäön riippuvan vuodenajasta. Käyttämällä metadatana sekä kuukausia että GPS-koordinaatteja tulos on hieman parempi kuin pelkästään toista käyttämällä, 0,667.

Viimeisinä kokeiluina luokittelutarkkuutta yritettiin parantaa neuroverkon rakennetta muuttamalla. Lisäämällä kolmas konvoluutio- ja max-pooling-kerros konvoluutiota suorittavaan haaraan tulokseksi saatiin 0,714, mikä oli selkeä parannus edelliseen parhaaseen tulokseen. Neljäs konvoluutiokerros sen sijaan ei enää tuonut parannusta ja luokit-

telutulokseksi saatiinkin edellistä alempi 0,686. Ominaisuuskarttojen määrään lisääminen 48:aan nosti tarkkuuden 0,721:een ja 64:ään 0,725:een, mikä jäi kaikkien tutkimuksessa suoritettujen testien parhaaksi tulokseksi. Ominaisuuskarttojen määrällä 80 tulos oli huonompi 0,720. Koko neuroverkon koulutettavien parametrien määrä oli parhaalla tuloksella 149 548.

Neuroverkko tuottaa luokittelussa jokaiselle luokalle todennäköisyyden ja näin ollen luokittelutarkkuutta voidaan kasvattaa jättämällä epävarmimmat tulokset luokittelematta. Vaatimalla vähintään 0,9 varmuuden arvatusta luokasta saadaan luokittelutarkkuudeksi 0,950, mutta tällöin vain 31,6 % testinäytteistä luokitellaan. Maltillisemmalla 0,6 varmuudella saavutetaan 0,832 tarkkuus ja testinäytteistä 70,5 % luokitellaan. Näistä tuloksista nähdään, että toteutetun neuroverkon ollessa erittäin varma luokittelutuloksesta, on tulos lähes aina myös oikein. Tätä tietoa voisi hyödyntää joissain sovelluksissa, esimerkiksi antamalla ihmisasiantuntijan tai toisen koneoppimisjärjestelmän tunnistaa epävarmat tapaukset.

6. POHDINTOJA

Työssä käsiteltiin useita kuvien ja neuroverkkojen menetelmiä, joita käyttäen puun runkojen luokittelutulosta saatiin kasvatettua selvästi yli satunnaisen arvailun. Menetelmiä voisi kuitenkin vielä kehittää eteenpäin.

6.1 Johtopäätökset menetelmistä

Yleisesti tiedon muuttaminen toiseen muotoon ei lisännyt saatua luokittelutarkkuutta, poikkeuksena kuitenkin kuvien normalisointi välille [0,1] paransi tuloksia huomattavasti. Uuden tiedon lisääminen sen sijaan yleisesti paransi tulosta, mikä näkyi etenkin kuvien kokoa kasvatettaessa. Kuvien kokoa kasvattamalla voisi olla mahdollista parantaa luokittelutulosta edelleen, mutta tämä vaatisi tietoaineiston uudelleenkeräämistä. Korkeampiresoluutioisten kuvien ottaminen tietoaineistoa kerättäessä olisi saavutettavissa yksinkertaisesti ottamalla kuvat lähempää. Tällöin kuvaamiseen käytettävä aika kuitenkin kasvaisi, sillä yhteen kuvaan mahtuisi vähemmän puita kerralla, mikä voi olla ongelma sovelluskohteesta ja puualueen koosta riippuen.

Myös neuroverkon parametreillä ja syvyydellä oli suuri vaikutus luokittelun tulokseen ja koulutuksen kestoon. Neuroverkon parametrejä enemmän tutkimalla voisi olla mahdollista parantaa tarkkuutta entisestään. Parametrien optimoiminen on paljon laskentatehoa ja aikaa vaativa prosessi, jonka takia tässä työssä ei ehditty kattavasti selvittämään optimaalisia parametrejä.

Metadata vaikuttaa hyödylliseltä lisältä puulajitunnistuksessa ja esimerkiksi kuvien ottopaikkojen puulajien todennäköisyysjakaumat voisivat parantaa tulosta entisestään. Ongelmana metadatan käytölle voi kuitenkin olla sen saatavuus, sillä useat julkiset koneoppimiseen tarkoitetut kuvatietoaineistot sisältävät pelkästään syötteenä käytettävät kuvat ja niitä vastaavat luokat. Metadatan olisi myös hyvä muodostua automaattisesti kuvanottohetkellä, eikä riippua käyttäjän syötteestä, jotta sen kerääminen olisi helpompaa ja virheettömämpää. Ylipäänsä tietoaineiston keräämisen olisi hyvä tapahtua mahdollisimman automaattisesti, jotta ihmisvaikutuksen aiheuttamat aineiston vinoumat pysyvät mahdollisimman pienenä.

Kokonaisuutena kännykkäkameralla otetuista metsäkuvista on mahdollista tunnistaa suurin osa metsän puista konvoluutioneuroverkkoja käyttäen, mutta luokittelutulos 0,725 jää silti liian alhaiseksi moneen kaupalliseen sovellukseen.

Taulukon 4 perusteella voidaan todeta, että neuroverkko suosii hieman näytteiden arvaamista kuusiksi. Tämä nähdään väärien arvausten suurimman määrän olevan eukalyptuksen tapausta lukuun ottamatta kuusisarakkeessa. Ei kuitenkaan vaikuta siltä, että lehtipuut ja havupuut olisivat erityisen vaikeita erottaa toisistaan, sillä kuuset arvattiin useammin väärin koivuiksi kuin männyiksi.

6.2 Jatkokehitysehdotukset

Konvoluutioneuroverkkoja käytettäessä luokitteluun, aineiston laatu vaikuttaa tulokseen enemmän kuin näytteiden määrä, mikä huomattiin aineistoa suodatettaessa luokittelutarkkuuksien pysyvän likimäärin ennallaan, vaikka koulutusaineistoa pienennettiin. Täten mahdollisuuksien mukaan aineiston laatua voisi yrittää parantaa muillakin menetelmillä, vaikka se johtaisi pienempään käytettävään aineistoon.

Jatkokehitystä varten runkokuvien leikkaamista metsäkuvista voisi parantaa. Kuvan 7 oikeanpuolisimman puun leikkausikkunaa tarkastelemalla huomataan, että leikkausikkunan oikeassa yläkulmassa on kuvaan tullut mukaan rungon lisäksi myös taustaa. Puut eivät aina kasva täysin suoraan, joten ongelman ratkaisemiseksi pitäisi selvittää puiden kallistumissuunta. Joissakin tapauksissa puut voivat lisäksi kaareutua takaisin alkuperäiseen kasvusuuntaan, mikä tekee ongelmasta vaikean ratkaistavaksi.

Syvyyskanavan tuominen kuviin voisi antaa luokittelijalle lisätietoa esimerkiksi puiden iästä. Tällä hetkellä kuvassa olevat rungot voivat olla pieniä, koska puut ovat pieniä, mutta myös puiden ollessa kaukana kuvanottopaikasta. Tämä vaatisi kuitenkin kehittyneempää laitteistoa näytteiden keräämiseen, mikä voisi olla mahdollista tiettyihin sovelluksiin. Tavoitteena työssä oli kuitenkin luoda luokittelija, joka toimisi tavallisilla kännykkäkameroilla otetuilla kuvilla, joten ainakaan nykyteknologialla ei syvyysinformaatiota voida tuoda kuluttajakäyttöön suunnattuihin mobiilisovelluksiin.

Työssä koulutettua verkkoa voidaan kolme konvoluutiokerrosta sisältävänä pitää syvänä konvoluutioneuroverkkona. Verkon syvyyden kasvattaminen vaikutti positiivisesti luokittelutulokseen, joten vieläkin syvempiä verkkoja voisi tutkia. Ongelmana nykyisellä toteutuksella verkon syventämiseen on syötteiden muuttuminen liian pieneksi verkon loppupäässä, jolloin konvoluutio-operaatioita ei voida enää suorittaa. Ratkaisuksi syötteen liialliseen pienenemiseen verkon rakennetta voitaisiin muuttaa niin, että max-pooling suoritettaisiin harvemmin, esimerkiksi joka toisen konvoluutiokerroksen jälkeen, jolloin syötteen koko pienenisi hitaammin kerrosten edetessä.

Metadatan vaikutus luokittelun tulokseen näkyi varsinkin eukalyptuksen korkealla tunnistusmäärällä muista lajeista. Suuri luokittelutuloksen parannus saatiin puiden sijaintitieto

lisättäessä. Näin ollen metadataa kannattaa ehdottomasti hyödyntää luokittelussa, jos mahdollista, varsinkin jos lajien väliset metadatat poikkeavat toisistaan paljon, jolloin luokittelua ei tarvitse tehdä pelkästään visuaalisen datan perusteella.

Lisää metadataa voisi saada puulajien jakaumien perusteella. Koska puulajien esiintymistodennäköisyys riippuu kasvupaikasta ja työssä puun runkojen kuvat leikattiin suuremmista metsäkuvista, voisi olla mahdollista hyödyntää muiden, samassa kuvasta peräisin olevien runkojen luokittelutuloksia. Esimerkiksi, jos kaikki muut kuvan rungot on luokiteltu kuusiksi, kasvaa todennäköisyys siitä, että kyseessä on alue, jolla esiintyy paljon kuusia. Näin ollen todennäköisyys sille, että viimeinenkin runko kuvasta on kuusi, on korkeampi kuin alueella, jolla kasvaa puulajeja sekaisin. Tällaisen menetelmän toteuttaminen voisi auttaa luokittelijaa varsinkin tilanteissa, joissa yksi puun rungoista on huonosti näkyvissä.

7. YHTEENVETO

Konvoluutioneuroverkoilla puulajitunnistusta tehtäessä koulutusaineiston laatu vaikuttaa luokittelutulokseen enemmän kuin koulutusaineiston koko. Neuroverkon suunnittelulla ja kuvien prosessointimenetelmillä voidaan vaikuttaa suuresti saatavaan lopputulokseen. Konvoluutioneuroverkko voittaa selvästi satunnaisen puulajien arvailun, joten myös matalaresoluutioisista puun rungoista on mahdollista irrottaa hyödyllisiä luokittelussa käytettäviä ominaisuusvektoreita. Kuitenkin matalaresoluutiosilla kuvilla puidentunnistus jää kaupallisten sovellusten tarkkuusvaatimuksista, joten konvoluutioneuroverkko ei näissä olosuhteissa yksinään riitä luotettavaan puulajitunnistukseen. Kuvien lisäksi muut datanlähteet voivat parantaa tietyissä olosuhteissa luokittelutulosta suuresti, joten lisädataa kannattaa hyödyntää, jos sitä on helposti saatavilla. Jatkokehitystä kannattaisi kohdistaa automaattisiin metadatan keräystapoihin ja jatkoprosessointiin.

LÄHTEET

1. **Viitanen, J.;Mutanen, A. ja Karvinen, S.** *Metsäsektorin suhdannekatsaus 2020-2021.* s.l. : Luonnonvarakeskus, 2020.

2. **Kangas, A. ja Maltamo, M.** *Forest Inventory Methodology and Applications.* Dordrecht, Alankomaat : Springer Netherlands, 2006.

3. **Lier, M.;ym.** *Suomen metsät 2019.* s.l. : Luonnonvarakeskus, 2019.

4. **Fiel, S. ja Sablatnig, R.** *Automated identification of tree species from images of the bark, leaves or needles.* 2010.

5. **Carpentier, M.;Giguere, P. ja Gaudreault, J.** *Tree Species Identification from Bark Images Using Convolutional Neural Networks.* s.l. : IEEE, 2018.

6. **Boudra, S.;Yahiaoui, I. ja Behloul, A.** *A set of statistical radial binary patterns for tree species identification based on bark images.* 2020. Multimedia tools and applications.

7. **Agarwalla, S. ja Sarma, K. K.** *Machine learning based sample extraction for automatic speech recognition using dialectal Assamese speech.* s.l., Yhdysvallat : Elsevie Ltd, 2016.

8. **Shamir, L.** *Compound quantitative analysis of text using machine learning .* 2020.

9. **Jiang, M.;ym.** *The two-stage machine learning ensemble models for stock price prediction by combining mode decomposition, extreme learning machine and improved harmony search algorithm.* s.l. : Springer, 2020.

10. **MathWorks.** MATLAB rgb2gray. [Online] [Viitattu: 19. Helmikuu 2021.] https://se.mathworks.com/help/matlab/ref/rgb2gray.html.

11. **Venkatesan, R. ja Li, B.** *Convolutional Neural Networks in Visual Computing: A Concise Guide.* Portland : CRC Press, 2018.

12. **Peltonen, S.** *SGN-12001: Johdatus kuvan- ja videonkäsittelyyn.* s.l. : Tampereen Teknillinen Yliopisto, 2018.

13. **Nathan, J. B. ja Slobodyan, E.** *Driving Pixel Display Strips.* s.l. : Cornell University, 2018.

14. **Korifi, R.;ym.** *CIEL*a*b* color space predictive models for colorimetry devices – Analysis of perfume quality.* s.l. : Talanta(Oxford), 2013.

15. **Ibrahim, H. ja Kong, N. S. P.** *Brightness Preserving Dynamic Histogram Equalization for Image Contrast Enhancement.* s.l. : IEEE Transactions on Consumer Electronics, 2007.

16. **Pal, K. K. ja Sudeep, K. S.** *Preprocessing for Image Classification by Convolutional Neural Networks.* s.l., Intia : IEEE, 2016.

17. **Najim, M.** *Digital filters design for signal and image processing.* 1, Newport Beach, Kalifornia : ISTE Ltd, 2006.

18. **Byrne, C. L.** *Signal Processing : A Mathematical Approach (2nd Edition).* s.l. : CRC Press, 2014.

19. **Michelucci, U.** *Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection.* Berkeley, Kalifornia : Apress L. P, 2019.

20. **Aggarwal, C. C.** *Neural Networks and Deep Learning: A Textbook.* s.l. : Springer International Publishing AG, 2018.

21. **Hartley, R. ja Zisserman, A.** *Multiple view geometry in computer vision.* Cambridge, Yhdistynyt kuningaskunta : Cambridge University Press, 2003.

22. **Heikkila, J. ja Silven, O.** *A four-step camera calibration procedure with implicit image correction.* s.l. : IEEE, 1997. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, p.1106-1112.

23. **Jordan, M. I. ja Mitchell, T. M.** *Machine learning: Trends, perspectives, and prospects.* 2015.

24. **Chandramouli, S.;Das, A. K. ja Dutt, S.** *Machine Learning.* s.l. : Pearson Education India, 2018.

25. **Jones, A.;Kruger, C. ja Johnston, B.** *The Unsupervised Learning Workshop.* s.l. : Packt Publishing, 2020.

26. **Alpaydin, E.** *Machine Learning : The New AI.* Cambridge, MA : The MIT Press, 2016.

27. **Mueller, J. P. ja Massaron, L.** *Machine learning for dummies .* Hoboken, New Jersey, Yhdysvallat : John Wiley & Sons, Inc., 2016.

28. **Alpaydin, E.** *Introduction to machine learning .* Cambridge, Massachusetts, Yhdysvallat : The MIT Press, 2014.

29. **Boyle, B. H.** *Support Vector Machines: Data Analysis, Machine Learning, and Applications.* New York : Nova Science Publishers, 2011.

30. **Song, X.;Duan, Z. ja Jiang, X.** *Comparison of artificial neural networks and support vector machine classifiers for land cover classification in Northern China using a SPOT-5 HRG image.* 2012. International journal of remote sensing 33.10.

31. **Rajagopal, H.;ym.** *Application of image quality assessment module to motion-blurred wood images for wood species identification system.* 2019. Wood science and technology 53.4 .

32. **Martins, J.;ym.** *A Database for Automatic Classification of Forest Species.* s.l. : Springer-Verlag, 2013. Machine vision and applications 24.3.

33. **Hu, J.;ym.** *Fish species classification by color, texture and multi-class support vector machine using computer vision.* 2012. Computers and electronics in agriculture 88.

34. **Robert, M.;Dallaire, P. ja Giguère, P.** *Tree bark re-identification using a deep-learning feature descriptor.* Quebec, Kanada : Université Laval, 2018.

35. **Pietikäinen, M.;Ojala, T. ja Xu, Z.** *Rotation-invariant texture classification using feature distributions.* s.l. : Elsevier Ltd, 2000. Pattern Recognition.

36. **Krizhevsky, A.** The CIFAR-10 dataset. [Online] [Viitattu: 24. Huhtikuu 2021.] https://www.cs.toronto.edu/~kriz/cifar.html.

37. **Amazon Web Services, Inc.** Amazon SageMaker. [Online] [Viitattu: 19. Maaliskuu 2021.] https://aws.amazon.com/sagemaker/.

38. **Google.** Google Cloud. [Online] [Viitattu: 19. Maaliskuu 2021.] https://cloud.google.com/gpu/.

39. **Microsoft.** Azure Machine Learning. [Online] [Viitattu: 19. Maaliskuu 2021.] https://azure.microsoft.com/en-us/services/machine-learning/.

40. **Kumar, N.** *Blockchain, Big Data and Machine Learning : Trends and Applications.* Boca Raton, Florida, Yhdysvallat : CRC Press, 2021.

41. **Ellen, J. S.;Graff, C. A. ja Ohman, M. D.** *Improving plankton image classification using context metadata.* Hoboken, New Jersey, Yhdysvallat : John Wiley & Sons, Inc, 2019. Limnology and oceanography, methods, 2019-08, Vol.17 (8), p.439-461.

42. **Clevert, D.;Unterthiner, T. ja Hochreiter, S.** *Fast and Accurate Deep Network Learning by Exponential Linear Units.* 2015.

43. **Orr, G. B. ja Müller, K.-R.** *Neural Networks: Tricks of the Trade.* Berliini : Springer Berlin Heidelberg, 1998.

44. **Zafar,, I.;ym.** *Hands-On Convolutional Neural Networks with TensorFlow: Solve Computer Vision Problems with Modeling in TensorFlow and Python.* s.l. : Birmingham: Packt Publishing, Limited, 2018.

45. **Krizhevsky, A.;Sutskever, I. ja Hinton, G. E.** *Imagenet classification with deep convolutional neural networks.* 2012. Proceedings of NIPS.

46. **Yan, S.;Jing, L. ja Wang, H.** *A New Individual Tree Species Recognition Method Based on a Convolutional Neural Network and High-Spatial Resolution Remote Sensing Imagery.* Basel, Sveitsi : MDPI AG, 2021.

47. **Di, W.;Bhardwaj, A. ja Wei, J.** *Deep learning essentials: your hands-on guide to the fundamentals of deep learning and neural network modeling.* Birmingham : Packt Publishing, 2018.

48. **Google.** Tensorflow - Data augmentation. [Online] [Viitattu: 20. Huhtikuu 2021.] https://www.tensorflow.org/tutorials/images/data_augmentation.

49. **Itakura, K. ja Hosoi, F.** *Simple and Effective Tool for Estimating Tree Trunk Diameters and Tree Species Classification.* 2020. Applied optics. Optical technology and biomedical optics 59, no.2.

50. **Bertrand, S.;Cerutti, G. ja Tougne, L.** *Bark recognition to improve leaf-based classification in didactic tree species identification.* 2017. VISIGRAPP 2017 - Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications.

51. **Ballanti, L.;ym.** *Tree Species Classification Using Hyperspectral Imagery: A Comparison of Two Classifiers.* s.l. : MDPI, 2016.

52. **Google.** TensorFlow. [Online] [Viitattu: 14. Tammikuu 2021.] https://www.tensorflow.org/.

53. **Scikit-learn.** Scikit-learn. [Online] [Viitattu: 14. Tammikuu 2021.] https://scikit-learn.org/dev/index.html.