

Alex Talvio

**KOIRAN LONKKANIVELDYSPLASIAN
DIAGNOSOINTI
KONVOLUUTIONEUROVERKKOA
HYÖDYNTÄEN**

TIIVISTELMÄ

Alex Talvio: Koiran lonkkaniveldysplasian diagnosointi konvoluutioneuroverkkoa hyödyntäen

Pro gradu -tutkielma

Tampereen yliopisto

Tietojenkäsittelytieteiden tutkinto-ohjelma

Tammikuu 2022

Tämä tutkielma käsittelee koirien lonkkaniveldysplasian diagnosointia konenäön ja etenkin syväoppimisen mallien avulla. Tutkimuksessa käytettiin 53 081 lonkkanivelröntgenkuvaa, joihin on yhdistetty niiden FCI:n ohjeiden mukaiset tulokset. Tutkimus tuotti kahdentyyppisiä malleja, objektin tunnistus ja luokittelu. Objektin tunnistusmalleja käytettiin lonkkanivelten rajaukseen röntgenkuvista. Objektin tunnistusmalli toimi lähes moitteettomasti ja kykeni löytämään ja rajaamaan oikein lonkkanivelen lähes jokaisesta kuvasta. Lonkkanivelobjektin tunnistusmallin tuottamia rajattuja lonkkanivelkuvia käytettiin lonkkaniveldysplasia luokittelijan datana. Korkeimman suorituskyvyn lonkkaniveldysplasian luokittelussa sai EfficientNetB2-arkkitehtuuriin perustuva malli, joka saavutti F1-luvun 0.746. Tämän mallin luokkakohtaiset tarkkuudet lonkkaniveldysplasian luokittelussa olivat A (normaali): 0.869, B (rajatapaus): 0.606, C (lievä): 0.657, D (kohtalainen): 0.721, E (vaikea): 0.684. Jokaisessa luokassa tarkkuus ± 1 luokan tarkkuudella eli oikeaan tai naapuriluokkaan oli 0.98–1. Mikäli tarkastellaan tapauksia, joista luokittelija on varma (softmax-aktiivointi > 0.9) saadaan osajoukon tarkkuudeksi 0.923, joka kattaa noin 28 % aineistosta.

1	Johdanto	1
2	Lonkkaniveldysplasia	2
2.1	Syyt	2
2.2	Oireet ja merkit	2
2.3	Hoito	2
2.4	Seulonta	3
2.4.1	Fédération Cynologique Internationale (FCI)	3
2.4.2	Orthopedic Foundation for Animals (OFA)	4
2.4.3	British Veterinary Association (BVA)	4
2.4.4	Eri arvostelujärjestelmien hyvät ja huonot puolet	4
2.4.5	Muunnos järjestelmästä toiseen	5
2.5	Perinnöllisyys ja ilmaantuvuus	5
2.6	Kehitys	7
3	Data	10
3.1	Aineiston puutteet	10
3.1.1	Tutkimuksissa useita kuvia	10
3.1.2	Puolimerkit	11
3.2	Lopullinen aineiston koko	12
3.3	Lonkkien laatujauma	13
3.4	Aineiston luotettavuus	14
3.4.1	Paneelidata	14
3.4.2	Analyysi	15
4	Syväoppiminen ja konenäkö	17
4.1	Konenäön tehtävät	17
4.2	Konenäön sovelluskohteita	17
4.3	Konenäköjärjestelmän vaiheet	18
4.3.1	Syöte	18
4.3.2	Esikäsittely	18
4.3.3	Piirteiden erotus	19
4.3.4	Koneoppimisen malli	19
4.4	Konvoluutioneuroverkko	20
4.4.1	Arkkitehtuuri	20
4.4.2	Takaisinsyöttävä käänös (Backward Propagation)	23
4.5	ImageNet	25
4.6	Arviointi	26
4.6.1	Arvuaatiometriikat luokittelussa	26

4.6.2	Epätasapainoinen data	27
4.6.3	Moniluokkainen luokittelu	28
4.6.4	Datan jakaminen osiin	28
4.6.5	k-ristiinvalidointi (k-fold cross validation)	28
4.7	Ylisovittautuminen (overfitting)	29
4.7.1	Ylisovittautumisen ehkäisy	29
5	Laitteisto, kuvien esikäsittely ja rajaukset	32
5.1	Laitteisto	32
5.2	Tiedostomuoto	32
5.3	Normalisointi	32
5.4	AHE ja CLAHE	34
6	Lonkkanivelobjektin tunnistus	35
6.1	Data	35
6.2	Rajalaatikon oikeellisuus	36
6.3	Prosessi	37
6.3.1	Ensimmäiset ongelmat	37
6.3.2	Opetusaineiston kasvatus	38
6.3.3	Laajat testit	38
6.4	Tulokset	39
6.4.1	Testi 1. Väärin suunnatut kuvat	39
6.4.2	Testi 2. Väärät tutkimukset	40
6.4.3	Testi 3. Toiminta oikean aineiston kanssa	40
6.5	Pohdinta	41
7	Automaattinen lonkkalausuja	42
7.1	Data	42
7.2	Luokittelu vai regressio?	43
7.3	Mallit ja arkkitehtuuri	43
7.3.1	Mallien evaluointi	44
7.3.2	Konvoluutioneuroverkon arkkitehtuuri	44
7.4	Ylisovittautuminen	44
7.5	Tulokset	46
7.5.1	Luokkakohtaiset tarkkuudet	46
7.5.2	Ennustetun luokan todennäköisyys	49
7.5.3	Aktivointi	51
7.5.4	Varianssi	52
7.5.5	Vertailu toiseen tutkimukseen	53
7.6	Pohdinta	54
8	Yhteenveto	57
9	Viiteluettelo	58

1 Johdanto

Konenäkötutkimus on edennyt erittäin nopeasti viimeisen vuosikymmenen aikana. Konenäkötutkimuksen tavoite on luoda tietokoneille kyky ymmärtää ja tulkita kuvadataa ihmisen tavoin. Tämä tavoite on saavutettu useissa eri sovelluskohteista, ja konenäkö-järjestelmät tuottavat usein lupaavia tuloksia ja kykenevät haastamaan tai jopa ylittämään ihmisen suorituskyvyn useissa kuvan luokittelutehtävissä [Buetti-Dinh *et al.* 2019; Elgandy 2020 s.10].

Monet konenäköjärjestelmät ovat osoittautuneet tehokkaiksi lääketieteellisessä sekä eläinlääketieteellisessä kuviin perustuvassa diagnosoinnissa. Laadukkaan diagnoosin antaminen radiologisista kuvista, kuten röntgenkuvasta osoittautuu usein haastavaksi jopa kokeneille radiologeillekin. Tästä huolimatta on kyetty luomaan tehokkaita konenäköjärjestelmiä, muun muassa keuhkosyövän [Li *et al.* 2020] diagnosointi lääketieteessä ja sydämenlaajentuman diagnosointi eläinlääketieteessä [Burti *et al.* 2020]. Tällaisten järjestelmien on tarkoitus toimia työkaluina nopeuttamaan ja tehostamaan diagnosointia. Tällaisia järjestelmiä kutsutaan termillä Computer aided-detection eli CAD.

Lonkkanivelen kasvuhäiriö (Lonkkaniveldysplasia, LD) on yleisin koirilla esiintyvä luuston ja nivelten kasvuhäiriö. Pitkälle edenneen lonkkanivelen kasvuhäiriö johtaa usein useisiin negatiivisiin terveysvaikutuksiin, kuten nivelrikko, joka heikentää koiran elämänlaatua merkittävästi. Lonkkanivelen kasvuhäiriö on todettu periytyväksi ominaisuudeksi, jonka vuoksi sitä seulotaan laajasti monissa roduissa, jottei jalostuksessa käytetä heikkotasoisia yksilöitä. [Kennelliitto 2013]

LD-seulonta tapahtuu virallisen lonkkalausannon kautta, jossa radiologi arvioi röntgenkuvasta koiran molempien lonkkanivelten ”laadun” viisiasteisella asteikolla A-E. Tämä on manuaalinen prosessi. Tutkielmani tarkoitus on analysoida lonkkanivellausuntoaineistoa, sekä selvittää, kuinka konenäköä olisi mahdollista hyödyntää prosessin automatisoinnissa. Käytössäni on myös suuri määrä virallisia LD-seulontakuvia ja niihin liittyviä tietoja, jonka avulla loin syväoppimisen mallin suorittamaan koiran lonkkanivelen laadun arviointia automaattisesti. Luomani malli osoittautui lupaavaksi ja jatkokehityksellä sitä voisi hyödyntää CAD- työkaluna.

Luvussa 2 käsittelen lonkkaniveldysplasiaa ja sen diagnosointia. Luvussa 3 kuvaillaan ja analysoidaan tutkielman dataa. Luvussa 4 paneudutaan melko lyhyesti syväoppimiseen ja konenäköön, niiden historiaan ja toimintaan. Luvussa 5 tarkastellaan lyhyesti tutkimuksen tekemiseen käytettyä laitteistoa, sovelluksia ja kirjastoja. Luvussa 6 kerrotaan datan esikäsittelystä ja datan rajauksista. Luku 7 käsittelee automaattista lonkkanivelen rajausta Mask-RCNN-mallin avulla. Luku 8 käsittelee automaattista lonkkalausujaa. Luku 9 on lyhyt yhteenveto.

2 Lonkkaniveldysplasia

Lonkkaniveldysplasia on kategorisoitu kehityshäiriöksi, joka syntyy lonkkanivelen normaalista poikkeavasta muodosta ja löysyydestä, jotka puolestaan johtavat nivelen epänormaaliin kulumiseen. Syntyessään koiran lonkkanivel vaikuttaa normaalilta ja ensimmäiset viitteet LD:aan ovat havaittavasti aikaisintaan 3–5 kk iässä. [Evidensia 2019]

2.1 Syyt

Nykytiedon valossa lonkkaniveldysplasia on sairaus, johon vaikuttaa useat eri tekijät. Sen ilmaantuvuuteen ja vakavuuteen vaikuttavat useat periytyvät geenit sekä erilaiset ympäristötekijät. Ympäristötekijöitä ovat muun muassa ruokavalio, ylipaino ja yllirasittuminen kasvun aikana [ACVS 2021]. Geenit ovat kuitenkin suurin yksittäinen tekijä. Geneettinen puoli ei ole vielä täysin selvillä ja on muun muassa epäselvää, mitkä kaikki geenit tarkalleen ottaen vaikuttavat LD:hen [Ohlerth *et al.* 2019].

2.2 Oireet ja merkit

Koiran ikä vaikuttaa merkittävästi havaittaviin oireisiin. Yleisempiä oireita ovat muun muassa takajalan ontuminen, painon siirto eturaajoille, lihassmassan vähentyminen takajaloilta ja kipu lonkissa. Tauti kehittyy ajan kanssa usein päättyen nivelrikkoon. On myöskin mahdollista, ettei koira näkyvästi oirehdi taudista ollenkaan sen kehittyessä. Oireet itsessään eivät riitä taudin diagnosointiin. Diagnoosiin suoritetaan pääasiallisesti röntgenkuvan avulla. [ACVS 2021]

2.3 Hoito

LD:n aiheuttamaa vahinkoa ei voi korjata luonnollisesti, ja mahdolliset hoitomenetelmät rajautuvat kivun hoitamiseen kipulääkkein tai kirurgiseen operaatioon. Erilaisia leikkauksia LD:n korjaamiseen tai kivun ehkäisyyn on aina pienistä operaatioista lonkkaproteesiin. Yksinkertaisimmat operaatiot voidaan tehdä nuorille vielä kasvaville koirille, kuten alle 20 viikon ikäiselle häpyluun ruston lämpöluudutus eli JPS-leikkaus tai alle 6 kk ikäiselle lonkkamaljakon kääntöleikkaus DPO. [Evidensia 2019]

Koiran ikääntyessä vaihtoehdot LD:n kirurgiseen operointiin vähenee merkittävästi. Tehokkain jällejäävistä operaatioista on lonkkaproteesi, jossa reisiluun pään ja kaulan tilalle asetetaan metallinen proteesi ja lonkkamaljakko korvataan muovisella kupilla. Lonkkaproteesin asentaminen on kuitenkin kallis operaatio ja maksaa useita tuhansia euroja lonkkaa kohti. Lisäksi LD tyypillisesti esiintyy molemminpuolisesti, eikä lonkkaproteeseja voida asentaa molemmille puolille samaan aikaan. Proteesin asennuksen jälkeen tulee pitää 3–6 kuukauden tauko ennen toisen asennusta. Muun muassa JPS ja DPO voidaan suorittaa molemmille puolille samaan aikaan. [MVS 2021].

LD tyypillisesti alkaa kehittyä 3–5 kk iässä, mutta tässä kohtaa koirat ovat usein oireettomia ja liikkuvat ja leikkivät normaalisti. Koira alkaa usein oireilla vasta sukukypsyyden jälkeen, jolloin helpoimmat kirurgiset operaatiot ovat poissuljettuja. [MVS 2021]

2.4 Seulonta

LD:n periytyvyyden ja haastavan hoitamisen takia koirienjalostuksessa hyödynnetään LD:n seulontaa. Seulonta mahdollistaa koirien jalostamisen vain sellaisilla koirilla, joilla on hyvät lonkat, kasvat- taen todennäköisyyttä, että jälkipovi omaa myös hyvät lonkat. [Kennelliitto 2013]

LD-seulonta tapahtuu ottamalla röntgenkuva lonkista koiran ollessa 1–2 vuotta vanha. Kuvausikä kuitenkin vaihtelee roduittain. Seuraavaksi röntgenkuva lähetetään radiologin arvioitavaksi, joka palauttaa tuloksen molempien lonkkien laadusta. Lonkkien arvosteluasteikko vaihtelee alueellisen skee- man mukaan. Maailman kolme yleisintä arvostelukeemaa ovat FCI, OFA ja BVA.

2.4.1 Fédération Cynologique Internationale (FCI)

Suuressa osassa Eurooppaa, myös Suomessa, noudatetaan Fédération cynologique internationales (FCI) määrittelemiä ohjeita koirien lonkkanivelten arviointiin. FCI:n arvosteluasteikko koostuu vii- destä luokasta A-E. Asteikossa A-B viittaavat normaaleihin terveisiin lonkkiin ja C-E viittaavaa lonk- kiin, jossa lonkkaniveldysplasiaan on havaittavissa.

A	Normaali The femoral head and the acetabulum are congruent. The craniolateral acetabular rim appears sharp and slightly rounded. The joint space is narrow and even. The Norberg angle is about 105°. In excellent hip joints the craniolateral rim encircles the femoral head somewhat more in caudolateral direction.
B	Rajatapaus The femoral head and the acetabulum are slightly incongruent and the Norberg angle is about 105° or The femoral head and the acetabulum are congruent and the Norberg angle is less than 105°.
C	Lievä lonkkaniveldysplasia The femoral head and the acetabulum are incongruent, the Norberg angle is about 100° and/or there is slight flattening of the craniolateral acetabular rim. No more than slight signs of osteoarthritis on the cranial, caudal, or dorsal acetabular edge or on the femoral head and neck may be present.
D	Kohtalainen lonkkaniveldysplasia There is obvious incongruity between the femoral head and the acetabulum with subluxation. The Norberg angle is more than 90° (only as a reference). Flattening of the craniolateral rim and/or osteoarthrotic signs are present.
E	Vaikea lonkkaniveldysplasia Marked dysplastic changes of the hip joints, such as luxation or distinct subluxation are present. The Norberg angle is less than 90°. Obvious flattening of the cranial acetabular edge, deformation of the femoral head (mushroom shaped, flattening) or other signs of osteoarthritis are noted.

Taulukko 1. FCI:n [Dortmund 1991] ohjeet lonkkien laadun arviointiin.

FCI-lonkkalausuntoon riittää yksi hyväksytty arvioija. Arviointi tapahtuu röntgenkuvaa tutkimalla ja erottamalla tai mittaamalla erilaisia lonkan piirteitä. Tämän jälkeen valitaan taulukosta erotettuja piir- teitä parhaiten kuvaava luokka. Taulukossa 1 on kunkin luokan viralliset kriteerit. [Flückiger 2007]

2.4.2 Orthopedic Foundation for Animals (OFA)

Yhdysvalloissa ja Kanadassa on käytössä Orthopedic Foundation for Animals (OFA) järjestön määrittämät ohjeet lonkkanivelten arviointiin. OFA käyttää seitsenluokkaista arvosteluasteikkoa, joka on jaettu kahteen ryhmään. Normaaliin lonkkaniveleen viittaavat [excellent, good, fair] ja lonkkaniveldysplasiaan [mild, moderate, severe]. Näiden lisäksi on vielä yksi luokka “Borderline”, johon menee rajatapaukset, joita ei kyetä tarpeeksi suurella varmuudella luokittelemaan kumpaankaan ryhmään. Näissä rajatapauksissa OFA suosittelee koiran uudelleenkuvausta kuuden kuukauden jälkeen, kun tauti on mahdollisesti edennyt. [OFA 2021]

OFA-lausunnon antaa itsenäisesti kolme koulutettua radiologia, joiden antamat lausunnot yhdistetään yhdeksi. Esimerkiksi jos radiologi 1 antaa arvioksi “Excellent”, radiologi 2 “Good” ja radiologi 3 “Fair”, tällöin lopullinen tulos on “Good”. Kun OFA-radiologien lausuntoja vertailtiin keskenään, havaittiin kaikkien kolmen lonkan arvioivan radiologin olevan 94.9 % tapauksista samaa mieltä lonkan ryhmästä, eli onko kyseessä normaali, rajatapaus vai lonkkaniveldysplasia. Lisäksi 73.5 % tapauksissa kaikki kolme lausujaa antoivat täsmälleen saman tuloksen lonkan laadusta [Excellent, Good, Fair, Borderline, Mild, Moderate, Severe]. [OFA 2021]

2.4.3 British Veterinary Association (BVA)

British Veterinary Association (BVA):n määrittelemiä ohjeita lonkkien arviointiin käytetään Isossa-Britanniassa, Irlannissa, Australiassa ja Uudessa-Seelannissa. BVA käyttää arvosteluasteikkoa 0–106, jossa 0 on normaalit lonkat ja 106 vaikea lonkkaniveldysplasia. Kummastakin lonkkanivelestä tarkastetaan 9 ominaisuutta ja kukin ominaisuus pisteutetään luvulla 0–6 (paitsi yksi 0–5), edelleen 0 on ideaali ja 6 heikoin tulos. Molemmat lonkat saavat pisteet väliltä 0–53 ja niiden yhteenlaskettu summa on lopullinen arvio, jota sitten verrataan kyseisen rodun mediaanipisteisiin. Pisteityksen suorittaa kahdesta kolmeen tarkastajaa, jotka yhdessä päättävät kunkin lonkan pisteet. [Dennis 2012]

2.4.4 Eri arvostelujärjestelmien hyvät ja huonot puolet

FCI:n hyväksymien lausujien ongelmana on se, että heidän koulutuksensa ja pätevyytensä saattaa vaihdella erittäin paljon. Osa heistä ovat itseoppineita, kun toiset korkeasti koulutettuja radiologeja. Tämän vuoksi heidän antamien lausuntojen laatu voi vaihdella merkittävästi. Tämä on etenkin huomattavissa maakohtaisesti, kun joissain maissa lausujat ovat korkeasti koulutettuja ja toisissa pitkälti itseoppineita. Tämän takia FCI-lausunnot eivät ole välttämättä täysin vertailukelpoisia toisiinsa. [Flückiger 2007]

OFA-lausunnot ovat korkealaatuisia ja lausujat ovat kaikki kokeneita radiologeja. OFA-lausunnot vaikuttavat luotettavalta. Lonkan laadusta annetaan riippumattomasti usein yksimielinen tulos, joka kasvattaa lausuntojen luotettavuutta

BVA:n lausujat ovat myös kokeneita asiantuntijoita. He kuitenkin työskentelevät yhdessä eivätkä itsenäisesti, kuten OFA:n lausujat. Itsenäiset tulokset olisivat vakuuttavampia ja mahdollistaisi tulosten laadun arvioinnin, kun voidaan tarkastella, antavatko lausujat yksimielisiä tuloksia vai eivät. BVA:n käyttämä numeroasteikko on itsessään melko epäselvä, mutta mikäli kunkin arvioitavan ominaisuuden pisteytys on selvitetävissä, kertoisi se selkeästi, kuinka tulokseen on päädytty. Tällaista ominaisuutta ei ole FCI- ja OFA-skeemoissa.

2.4.5 Muunnos järjestelmästä toiseen

Lausuntojen tulosten muunnos järjestelmästä toiseen on tärkeä ominaisuus. FCI:n ja OFA väliset muunnokset ovat melko suoraviivaisia. BVA-lausunnot puolestaan ovat hieman haastavaa muuntaa FCI- tai OFA-asteikkoon. Luotettavan muunnoksen näihin deskriptiivisiin asteikoihin voi tehdä vain, jos tietää kunkin yhdeksän ominaisuuden saamat pisteet. Tällöinkin vaaditaan ammattilainen tulkitsemaan pisteitä. Tästä huolimatta suuntaa antavia muunnostauluja on luotu kuten OFA:n taulukko 2.

OFA	FCI (Europe)	BVA/KC (UK/Australia) (score = sum of both hips)
Excellent	A-1	0-4 (no > 3/hip)
Good	A-2	5-10 (no > 6/hip)
Fair	B-1	11-18
Borderline	B-2	19-25
Mild	C	26-35
Moderate	D	36-50
Severe	E	51-106

Taulukko 2. OFA:n muunnostaulu lonkkaniveltulosten muunnos skeemasta toiseen.

2.5 Perinnöllisyys ja ilmaantuvuus

LD:n on todettu olevan periytyvä ominaisuus. On kuitenkin epäselvyyttä, miten suuri merkitys periytyvillä geeneillä on ympäristötekijöihin verrattuna [Ohlerth *et al.* 2019]. LD:n ilmaantuvuudessa ja vakavuudessa on havaittavissa erittäin isoja eroja rotujen välillä. Yleinen konsensus kuitenkin on, että LD:n ilmaantuvuus ja vakavuus korreloi koiran painon kanssa [Evidensia 2019, AKC 2021]. Muun muassa AKC:n [2021] mukaan erityisesti isot ja jättirodut ovat alttiita lonkkaniveldysplasialle, kuten tanskandoggi, bernhardinkoira, labradorinnoutaja ja saksanpaimenkoira. Selvittääkseni väitteen todenmukaisuutta keräsin aineistoa vuosien 2013 ja 2018 välillä Suomessa syntyneistä koirista.

Rotu	Paino (kg)	Syntyneitä	Tutkittu	C/D/E
Labradorinnoutaja	32	12083	56%	19%
Ranskanbulldoggi	11	2826	19%	65%
Saksanpaimenkoira	35	7883	51%	32%
Kultainennoutaja	32	7339	54%	31%
Englanningbulldoggi	23	1102	8%	94%
Beagle	11	2939	4%	52%
Rottweiler	55	2137	61%	24%
Australianpaimenkoira	27	1826	64%	16%
Tanskandoggi	70	1504	31%	38%
Siperianhusky	24	3635	10%	10%
Welsh corgi pembroke	11	1519	45%	80%
Shetlanninlammaskoira	11	6136	47%	21%
Jackrussellinterrieri	6	6239	4%	29%
Jämtlanninpystykorva	32	8855	21%	9%
Mopsi	7	1987	0.03%	100%
Chihuahua	2	7422	0.04%	24%
Dobermanni	43	992	45%	34%
Mastiffi	90	149	39%	52%
Berninpaimenkoira	40	2659	54%	41%
Bernhardinkoira	70	574	48%	37%
Collie	27	3261	46%	21%

Taulukko 3. Koirarotujen paino (uroksen keskipaino), syntyneiden määrä, tutkittu osuus ja C/D/E-lonkkien osuus tutkituista koirista vuosina 2013–2018 syntyneistä koirista. [KoiraNet 2021]

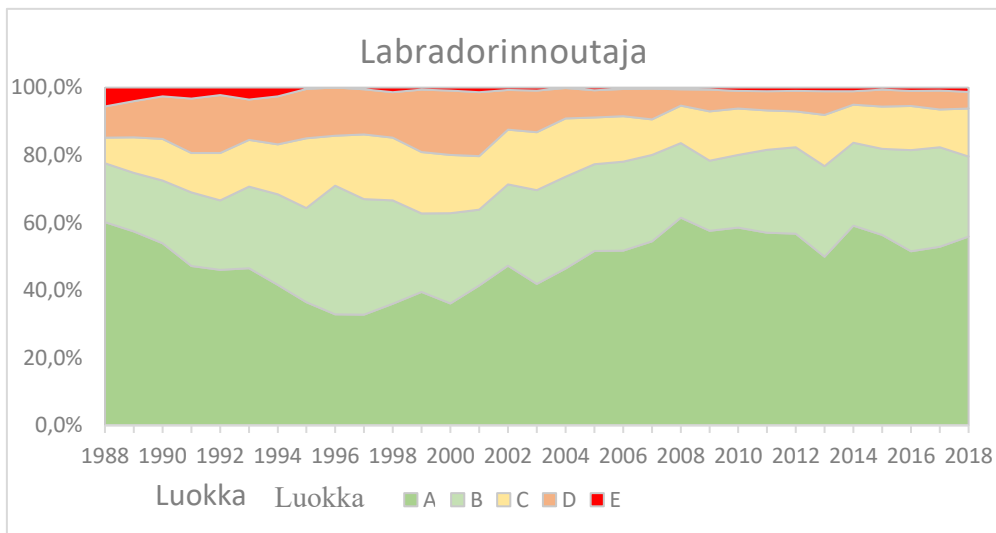
Taulukon 3 rodut on valittu mielivaltaisesti. Tavoitteena kuitenkin oli valita yleisiä rotuja eri kokoluokista, jotta koiran painon ja LD-ilmaantuvuuden suhde olisi havaittavissa. Taulukosta kuitenkin käy ilmi, ettei painon ja LD:n välinen korrelaatio ole erityisen vahva eikä koiran paino itsessään riittävä selittäväksi tekijäksi. Esimerkiksi väite, että labradorinnoutajat ovat erityisen alttiita LD:lle on mielestäni virheellinen, kuitenkin C/D/E-lonkkia on vain noin 19 % tutkituista yksilöistä, joka on neljänneksi vähiten tässä vertailussa.

Monet etenkin pienet rodut kärsivät erittäin alhaisesta tutkimusprosentista, jolloin todellinen LD:stä kärsivien yksilöiden määrä jää epäselväksi. Muutamissa pienimmissä roduissa on kuitenkin vastikään havahduttu tutkimaan enemmän koiria LD:n varalta. Muun muassa rodun Welsh corgi pembroke tutkittujen koirien osuus oli vain noin 10 % ennen vuotta 2009 syntyneistä verrattuna 2013–2018 välillä

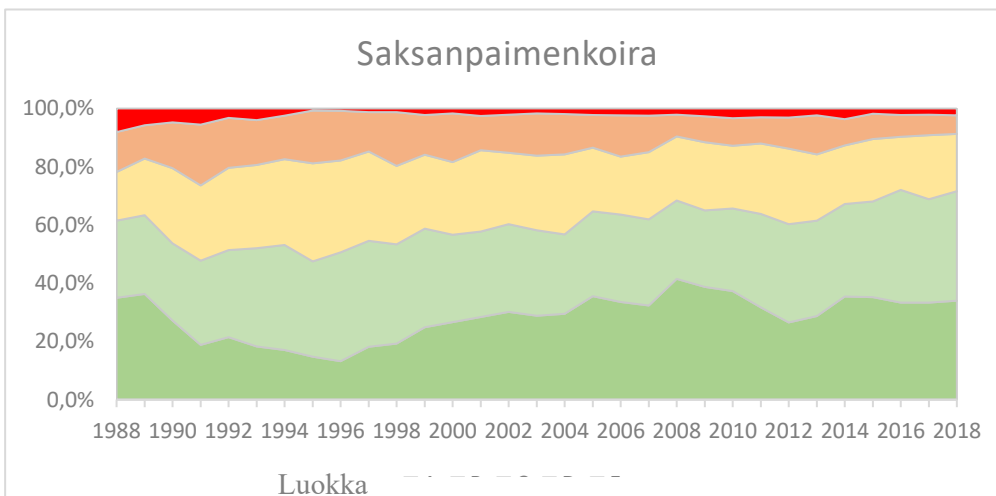
syntyneiden 45 % tutkittujen osuuteen. Vastaavaa kehitystä on myös havaittavissa englanninbulldogilla, joiden tutkittujen osuus ennen vuotta 2016 oli viiden prosentin luokkaa, mutta vuonna 2019 jo 21 %. Kaikissa roduissa ei kuitenkaan ole käynyt samoin. Esimerkiksi mopsin tilanne on huolestuttava, kun tutkittujen osuus on erittäin pieni eikä määrä vaikuta kasvavan. Lisäksi tässä vähäisessä tutkimusmäärässäkin 100 % tutkituista saa tuloksena C, D tai E-lonkat. Pieni tutkimusmäärä kuitenkin rajaa sitä, kuinka vahvoja johtopäätöksiä mopsin tilanteesta taulukon perusteella voi tehdä. Kuitenkin saamaa mieltä mopsien tilanteesta on myös OFA:n tilastot [OFA 2021], jonka mukaan mopsilla on keskimäärin huonoimmat lonkat kaikista roduista.

2.6 Kehitys

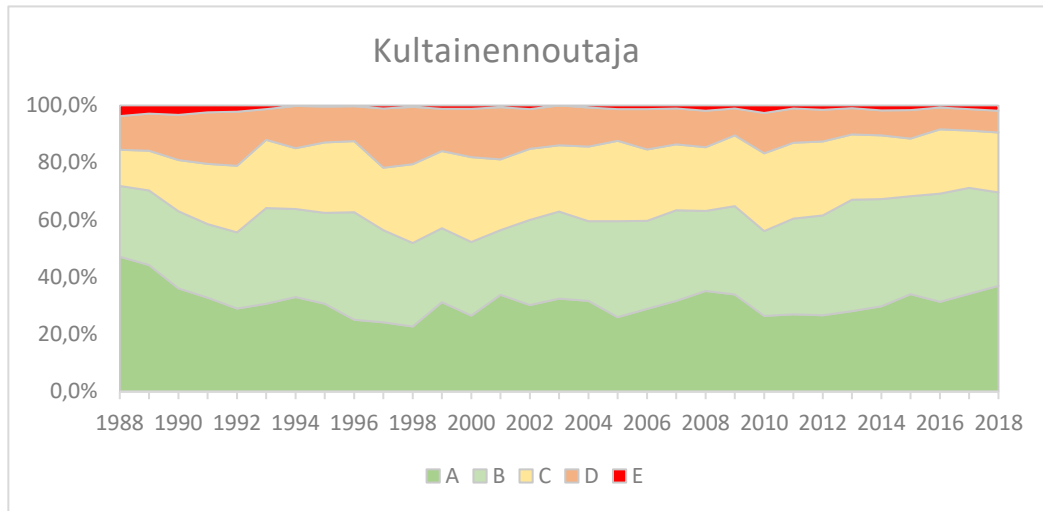
Valikoivan jalostuksen tehokkuutta voidaan arvioida tarkastelemalla rotujen lonkkatulosten kehitystä vuosien varrella.



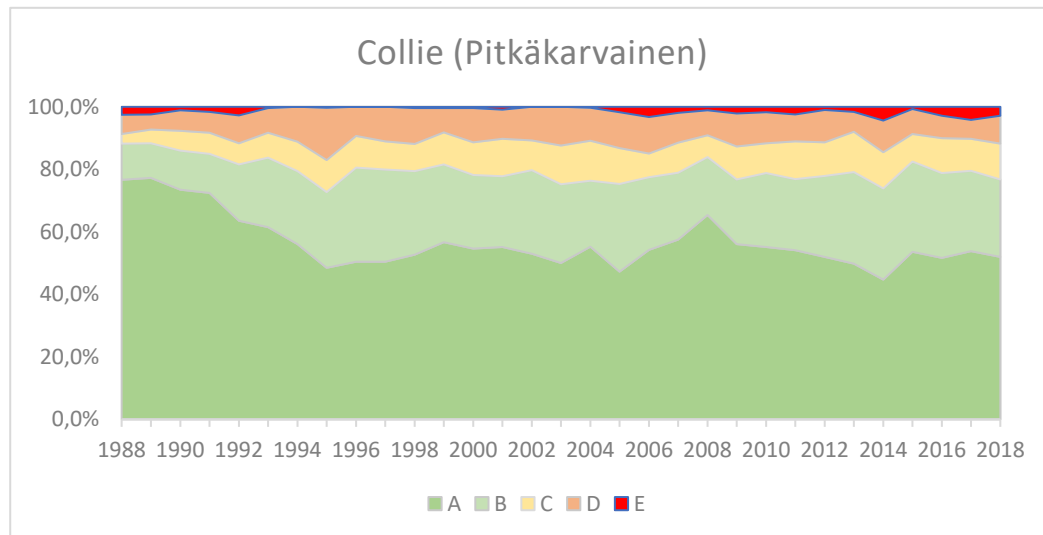
Kuva 1. Labradorinnoutajan lonkkatulosten kehitys vuosina 1988–2018.



Kuva 2. Saksanpaimenkoiran lonkkatulosten kehitys vuosina 1988–2018.



Kuva 3. Kultaisenoutajan lonkkatulosten kehitys vuosina 1988–2018.



Kuva 4. Pitkäkarvaisen collien lonkkatulosten kehitys vuosina 1988–2018.

Kuvista 1–3 nähdään, että labradorinnoutajan, saksanpaimenkoiran ja kultaisenoutajan kehitys on ollut samankaltaista. 1990-luvulla hyvien lonkkien määrä tippui reippaasti, jonka jälkeen tilanne on lähtenyt parantumaan 2000-luvun alussa nousten A-lonkkien osalta kutakuinkin samaan tilanteeseen vuonna 2018 kuin vuonna 1988. Näissä roduissa huonoitien tapausten määrä on kuitenkin vähentynyt merkittävästi ja D- ja E-tapauksia vaihtunut C:ksi. Kuvasta 4 voidaan huomata pitkäkarvaisen collien lonkkatulosten kehityksen poikkeavan merkittävästi näistä muista kolmesta rodusta. 1990-luvun laskusta ei ole palauduttu. Etenkin A-lonkkien määrä on tippunut erittäin paljon. Vuonna 1988 A-lonkkien osuus oli 77 % ja vuonna 2018 vain 52 %. A-lonkkien osuuden pienentyessä muiden osuus on kasvanut. B-lonkkien osuus on kasvanut 11 % -> 25 % ja huonojen C/D/E osuus 10 % -> 23 %.

Eräs selitys lonkkatulosten huonontumiseen 1990-luvulla on laman aiheuttama koiranpentubuumi. Kasvavan kysynnän vuoksi jalostukseen käytettiin terveydeltään heikompiteoisia yksilöitä, mikä heikensi jälkipolven lonkkien laatua. Koirien lonkkatulokset parantuvat vain, kun jalostukseen käytetään rodun keskitasoa parempia yksilöitä. [Mäki 2021]

Rotuja, joille vastaavaa pitkän ajan kehityksen seurausta on mahdollista tehdä, on hyvin rajallinen määrä. Rodun on oltava suosittu pitkältä ajanjaksolta, lisäksi tutkittujen osuus tulee olla melko korkea. Kovinkaan moni rotu ei täytä näitä kriteerejä. Tämän vuoksi kuvien 1-4 valitut rodut olivat mitä olivat.

3 Data

Tutkimuksen suorittamiseksi käytössäni oli Suomen Kennelliiton tarjoama aineisto, joka sisältää röntgenkuvia ja niihin liittyvät tiedot useista erilaisista tutkimuksista. Tämän tutkielman kannalta ainoastaan lonkkanivelröntgenkuvat ja niihin liittyvät tiedot ovat relevantteja.

Kokonaisuudessaan lonkkanivelröntgenkuvia on aineistossa 125 600 kappaletta. Kuvat on liitetty tutkimuksiin, ja kukin tutkimus voi sisältää useita kuvia. Tämän vuoksi on mielekkäämpää ajatella aineiston kokoa uniikkien tutkimusten määrällä, joka on noin 87 000 tutkimusta.

Jokaisesta röntgenkuvasta on saatavilla seuraavat tiedot:

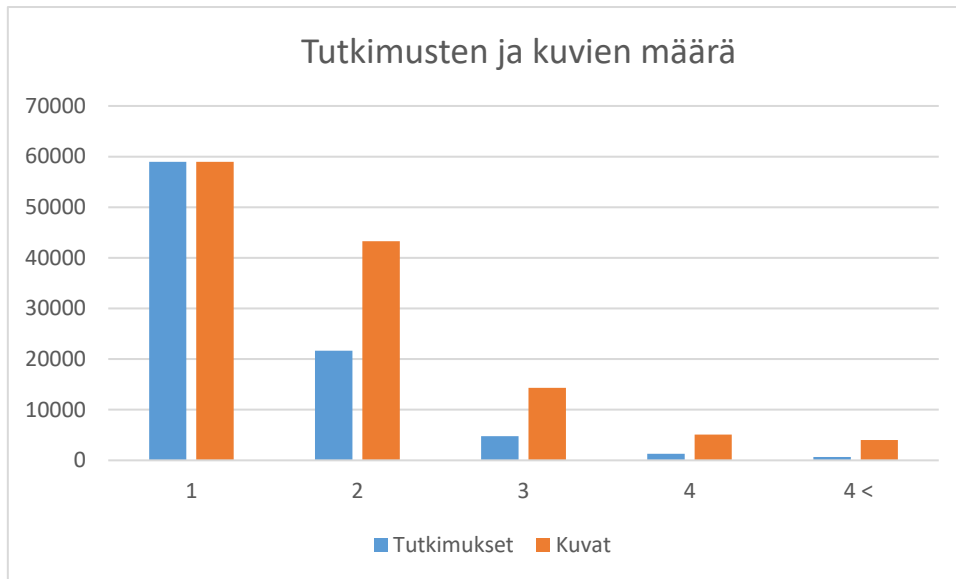
1. Tutkimustyyppi: Rajataan ainoastaan lonkkanivelututkimuksiin.
2. Tutkimuspäivä: Päivämäärä, jolloin tutkimus suoritettiin.
3. Arkistnumero: Tutkimuksen tunnistenumero arkistointia varten.
4. Kuvausikä: Kuvatun koiran ikä viikkoina kuvaushetkellä.
5. Rotu: Koiran rotu.
6. Tutkija: Kuvan eläinlääkäriin tunnistenumero.
7. Lausuja: Lausunnon antaneen eläinlääkäriin tunnistenumero.
8. Vasen: Vasemman lonkan laatu.
9. Oikea: Oikean lonkan laatu.
10. Tiedosto: Kuvatiedoston tiedoston sijainti (DICOM-muodossa).
11. Pakkaus: Onko kuva pakatussa muodossa vai ei.

3.1 Aineiston puutteet

3.1.1 Tutkimuksissa useita kuvia

Useisiin lonkkanivelututkimuksiin on liitetty useita kuvia. Usein, kun tutkimukseen on liitetty useampi kuin yksi kuva, se kertoo kuvan ottaneen eläinlääkäriin epävarmuudesta kyseisten kuvien laatuun ja sopivuuteen lonkkalausunnon antamiseksi. Tällöin kuvien laadun ja sopivuuden arviointi jää lausunnon antavalle eläinlääkärille. Ongelma syntyy, kun kaikki kuvat jäävät aineistoon, vaikka kuva ei täyttäisi sille asetettuja kriteereitä. Jälkeenpäin on mahdotonta tietää, mitä kuvaa käyttäen lausunnon antanut eläinlääkäri on antanut lausunnon. Tätä ongelmaa ei voi täysin ratkaista näin jälkeenpäin. Paras tilanne olisi, jos kukin tutkimus sisältäisi vain yhden kuvan tai vaihtoehtoisesti tiedon siitä, mitä kuvaa käyttäen lausunto on annettu. Nyt ongelman ratkaisuun tai kiertämiseen jää jäljelle muutama vaihtoehto.

1. Käytetään ainoastaan tutkimuksia, joissa on vain yksi kuva. Tällöin voidaan olettaa kuvan täyttävän kaikki kriteerit ja tiedetään, minkä kuvan perusteella lausunto on annettu.
2. Käytetään kaikkia kuvia ja hyväksytään riski, että kuvien laadussa voi olla eroavuuksia ja joukossa on huonoja kuvia.
3. Koitetaan seuloa tutkimuksista paras kuva.



Taulukko 4. Tutkimusten määrää joissa 1,2,3,4 tai enemmän kuvia per tutkimus. Oranssit pylväät kertovat kuvien kokonaismäärän näissä luokissa.

Jos käyttöön valitaan ainoastaan tutkimukset, joissa yksi kuva, jäisi käyttöön vain noin 50 % alkuperäisestä kuvamäärästä taulukossa 4. Tämä on liian vähän. Hyvien kuvien seulonta on puolestaan melko haastavaa, eikä kuvan laatua tai sopivuutta ole kovinkaan yksinkertaista määrittellä. Jäljelle jää vaihtoehto 2, jolloin käytetään kaikkia kuvia silläkin riskillä, että joukkoon saattaisi tulla heikkolaa-tuisia kuvia. Oletuksena kuitenkin on, että niiden määrä on melko pieni, jolloin se ei vaikuta järjes-telmien suorituskykyyn kovinkaan merkittävästi.

3.1.2 Puolimerkit

Eräs merkittävä ongelma aineiston käyttöön liittyen on myös se, ettei voi suoraan tietää kuvasta, kumpi lonkka on vasen ja kumpi oikea. Koiran vasen ja oikea puoli merkitään ainoastaan itse rönt-genkuvaan niin sanotulla puolimerkillä, jolloin vasemman ja oikean puolen voi tunnistaa vain kuvaa katsomalla (kuva 5). Tämä rajoittaa aineiston käytön vain sellaisiin kuviin, joissa kummatkin puolet ovat saaneet saman lonkkalausannon tuloksen, eikä täten puolella ole merkitystä. Noin 21 % tutki-muksissa on eri tulokset vasemmalle ja oikealle puolelle. Näin käytössä olevan aineiston määrä enim-millään tippuu noin 71 000 tutkimukseen.



Kuva 5. Esimerkki puolimerkistä lonkkakuvassa.

Tutkimuksen alussa tarkoitukseni oli luoda objektin paikannus- ja tunnistusjärjestelmä puolimerkeille, jotta saisin käyttöön kaikki lonkkatutkimukset. Nopeasti kuitenkin kävi ilmi, että tällainen järjestelmä saattaa olla erittäinkin vaikea toteuttaa. Ongelmana on pitkälti se, ettei puolimerkinnälle ole vakiintunut mitään yhtä tiettyä tapaa. Sen sijaan puolimerkintä käytännöt voivat erota yllättävänkin paljon eri eläinlääkärikliniikoiden välillä. Puolimerkin tulee kuitenkin olla kansainvälisesti tunnistettavissa, mikä onneksi rajoittaa erilaisia tapoja merkitä puolta. Tämän vuoksi tyypilliset hyväksytyt puolimerkit ovat muodoltaan vasemmalle puolelle [L, S, SIN] ja oikealle puolelle [R, D, DEX] (kuva 6). Vaikeus syntyy kuitenkin siitä, kun puolimerkki kuitenkin voi itsessään olla osittain kuvan ulkopuolella, pelikuvana tai missä vain kulmassa (kuva 6).



Kuva 6. Muutamia esimerkkejä erilaisista puolimerkeistä.

3.2 Lopullinen aineiston koko

Viimeiseksi aineistossa on väärin merkittyjä tutkimuksia. Esimerkiksi lonkkaniveltutkimuksessa lonkkanivelkuvan sijaan on tutkimukseen liitetty selkäkuva. Nämä väärin merkityt tutkimukset ovat

ongelmallisia, ja ne tulee löytää ja poistaa oikeiden lonkkanivelitutkimusten joukosta. Tein seulontaa aluksi manuaalisesti selaamalla kuvia, mutta melko nopeasti luovuin tästä lähestymistavasta sen puuduttavuuden ja hitauden vuoksi. Lisäksi manuaalinen selailu ei ollut loppujen lopuksi kovinkaan tarkkaa ja joitain vääriä kuvia meni silmiäni ohitse. Tämän vuoksi automatisoin väärin kuvien seulonnan myöhemmin esittelemäni lonkkanivelen hahmontunnistusjärjestelmän avulla. Järjestelmä ei päästänyt yhtään virheellistä kuvaa lopulliseen lonkkanivelkuvien joukkoon (tai ainakaan en ole löytänyt).

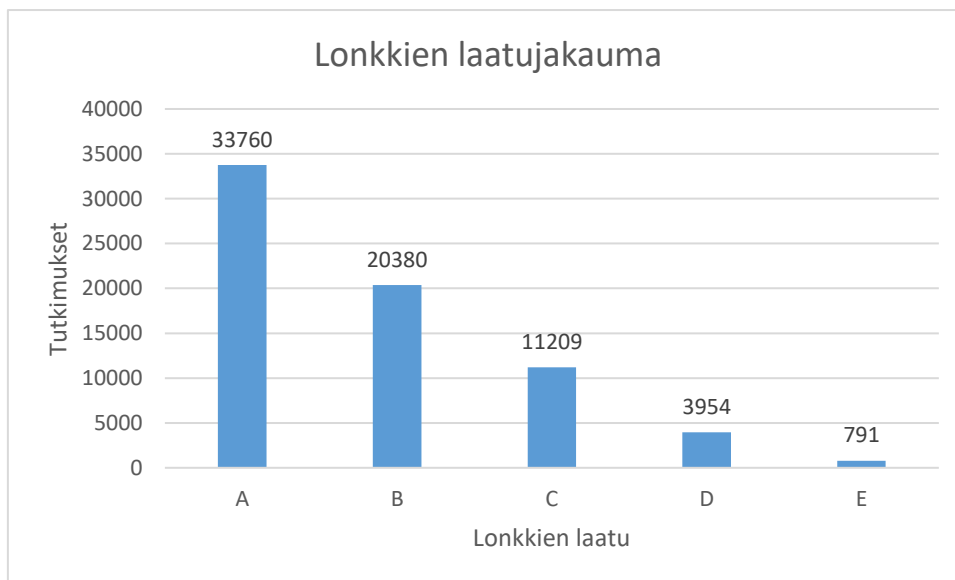
Lisäksi aineistossa on vielä noin 650 tutkimusta, joissa lausunnoksi on annettu '-' tai 'operoitu'. Kun jompikumpi edellisistä on annettu lausunnoksi lonkalle, tarkoittaa tämä sitä, ettei kyseisessä tutkimuksessa voitu antaa lausuntoa johtuen vaikkapa siitä, että koiran lonkkia on kirurgisesti operoitu. Nämä tapaukset rajasin lopullisen joukon ulkopuolelle.

Kun koko aineistosta on poistettu virheellisesti merkityt tutkimukset ja tutkimukset, joissa vasen ja oikea puoli ovat saaneet eri tuloksen, jäljelle jää noin 70 000 uniikkia tutkimusta.

3.3 Lonkkien laatujauma

Jokaiseen tutkimukseen on siis liitetty tieto koiran kummankin lonkan laadusta asteikolla [A, B, C, D, E], jossa A normaali ja E vakava lonkkadysplasia. Aineiston lonkkien laatu jakaantuu epätasaisesti, kun on muun muassa tutkimuksia, joissa molemmilla lonkilla on annettu laaduksi A, on yli 42 kertaa enemmän kuin tutkimuksia, joissa on annettu lonkkien laaduksi E (taulukko 5).

Tämä aineisto ei kuitenkaan kuvaa koko koirapopulaatiota Suomessa.



Taulukko 5. Tutkimusten määrä, joissa molemmat lonkat A, B, C, D tai E.

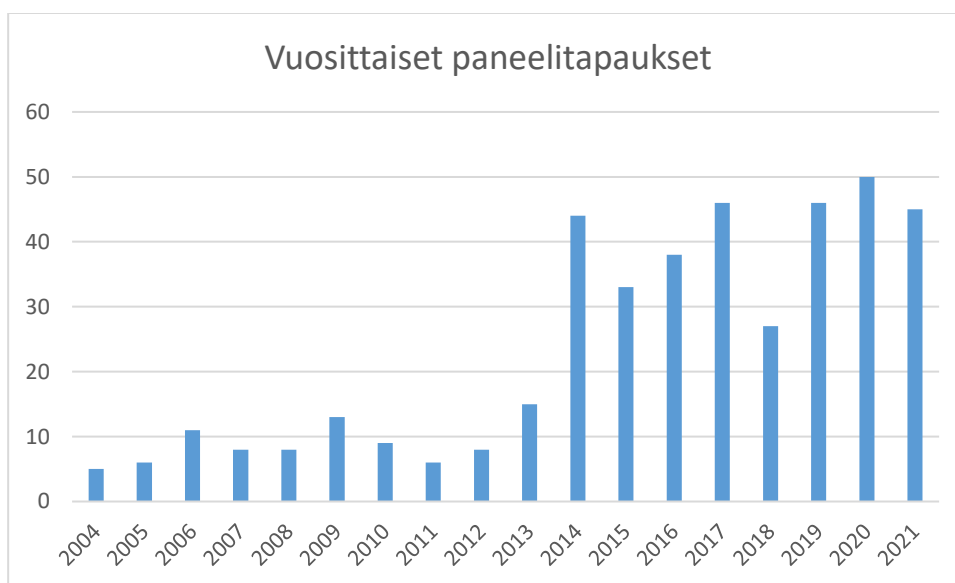
Datan epätasapainoisuus tulee ottaa huomioon, kun luodaan syväoppimisen malleja. Suurin osa syväoppimismenetelmistä olettaa datan olevan tasaisesti jakautunut. [Johnson and Khoshgoftaar 2019]

3.4 Aineiston luotettavuus

Kun henkilö ei ole tyytyväinen koiransa saamaan lonkkanivellausuntoon, hänellä on oikeus haastaa saatu lausunto. Tällöin kuvat lähtevät uudelleen arvioitavaksi lausuntopaneeliin. Riippumattomuuden vuoksi, kun Suomesta lähetetään tapaus uudelleen arvioitavaksi, paneeli koostuu kolmesta jäsenestä ja yksi jäsen on seuraavista maista: Tanska, Ruotsi ja Norja. Kukin paneelin jäsen antaa lonkka lausuntoja kotimaassaan. Kun paneelin jäsenet ovat antaneet omat toisistaan riippumattomat arvionsa uudelleen arvioitavalla tapaukselle, lasketaan näiden kolmen uuden lausunnon keskiarvo lopulliseksi tulokseksi. Alkuperäistä tulosta ei oteta paneelissa huomioon. Analysoimalla paneelin tuottamaa dataa voidaan arvioida lonkkanivellausuntojen luotettavuutta tai tarkkuutta kokonaisuudessaan.

3.4.1 Paneelidata

Paneelin tuottamaa dataa minulla on käytössä vuosilta 2004–2021. Näiden vuosien aikana paneelitapauksia oli kokonaisuudessaan 418, joka tarkoittaa uudelleen arvioitujen lonkkanivelten määrää olevan 816 (taulukko 6).



Taulukko 6. Vuositteiset lonkkanivelten paneelitapaukset.

Huomattavaa on, että lonkkanivelpaneelin ohjeisiin tuli muutos vuonna 2014, jota edeltävänä aikana paneelissa oli mukana Suomesta valittu neljäs jäsen. Muutoksen jälkeen paneelin koko muuttui nykyiseen kolmeen lausujaan. Tämän muutoksen jälkeen vuosittaiset paneelitapaukset kasvoivat merkittävästi.

Lisäksi aineistossa on tapauksia, joissa kuvat eivät ole käyneet paneelin kaikilla jäsenillä arvioitavana. Selvää syytä tähän en tiedä. Eräs mahdollisuus voisi olla, että mikäli 2/3 lausujista on antanut tapaukselle saman arvion ennen kuin kuvat on lähetetty kolmannelle lausujalle, tällöin on todettu,

ettei kolmannella lausunnolla ole merkitystä lopullisen tuloksen kannalta. Kokonaisuudessa tällaisia tapauksia on aineistossa 39 kappaletta.

Aineistossa on seuraavat sarakkeet: kuvausvuosi, alkuperäinen tulos, kunkin paneelin jäsenen antama tulos ja lopullinen paneelin keskiarvotulos. Tämä mahdollistaa laajan tulosten vertailun.

3.4.2 Analyysi

Oletus on, että pätevät viralliset eläinlääkärit antavat saman tuloksen samasta kuvasta, sillä he noudattavat samoja kriteerejä ja mittaavat samoja piirteitä. Nopeasti käy kuitenkin ilmi, ettei näin ole. Ainoastaan 34 % tapauksissa kaikki paneelin jäsenet ovat olleet samaa mieltä alkuperäisen lausunnon tuloksen kanssa, jolloin jäljelle jää 66 % lonkkanivelistä, joissa vähintään yksi lausuja on eri mieltä alkuperäisen tuloksen kanssa.

Kun paneeli ei ole yksimielinen lonkkanivelen laadusta, syntyy tällöin tapauksia, joissa lopullinen tulos eroaa alkuperäisestä tuloksesta. 31 % paneeliin menevistä lonkkanivelistä saivat paneelista alkuperäisestä eroavan tuloksen. Näissä tapauksissa, joissa lopullinen tulos vaihtui, 80 % saa alkuperäistä paremman tuloksen ja jäljelle jäävät 20 % alkuperäistä huonomman.

On myös hyvä selvittää, miten lähelle toisiaan tulokset osuvat, kun kyseessä on ordinaalinen asteikko. Kuten aiemmin mainitsin 284/836 eli noin 34 % tapauksissa ei ole eroa, kun kukin paneelin jäsen on antanut yksimielisesti saman tuloksen kuin alkuperäisenkin lausuja. Tapauksia, joissa eroa on yksi luokka eli esimerkiksi kolme lausujaa antaa tulokseksi B:n ja yksi antaa tulokseksi C:n. Tällaisia on 475/836 eli noin 57 %. Tapauksia, joissa eroa on kaksi luokkaa, on 72/836 eli noin 8.6 %. Lisäksi 3/836 tapauksista oli eroa jopa kolme luokkaa, jolloin näissä kaikissa yksi lausuja antoi lonkalle tulokseksi A ja toinen D ja loput jotain siltä väliltä.

Nämä tulokset värittävät epävarman kuvan lonkkalausuntojen tuloksista kokonaisuudessaan ja etenkin yksittäisen lausujan lausunnoista. Etenkin tapausten määrä, joissa eroa on kaksi tai useampi luokka, on erittäin huolestuttavaa, sillä näitä on lähes 1/10. Kahden luokan heitto on todella paljon lonkkalausunnossa, kun esimerkiksi ero B- ja D-lonkan välillä on suuri. B kuitenkin tarkoittaa lähes normaalia lonkkaa D:n ollessa keskivaikea LD. Tällaista eroa ei saisi olla. Tapaukset, jossa eroa on yksi luokka, ovat luonnollisia asteikon ollessa ordinaalinen, kuitenkin näidenkin osuus on liian suuri. Voisi kuitenkin olettaa rajatapausten olevan ylikorostettuina tässä paneeliaineistossa.

On myös huomattava, ettei paneelidata vastaa muuta käytössäni olevaa lonkkaniveldataa. Tämä johtuu pääasiassa kahdesta syystä. Ensimmäinen syy on edellä mainittu rajatapausten ylikorostuneisuus paneeliin menevissä tapauksissa. Toinen syy on A-lonkkien puuttuminen paneeliaineistossa, koska parasta tulosta harvemmin halutaan lähteä kumoamaan. Tämän vuoksi muut luokat ovat

ylikorostettuja, kun A-lonkkia on noin puolet kaikista tapauksista. A-lonkat saattavat olla helpoimmat tunnistaa, jolloin paneeli olisi useammin yksimielinen.

4 Syväoppiminen ja konenäkö

Konenäön tavoite on saada tietokone ymmärtämään digitaalisten kuvien sisältöä, mikä usein tarkoittaa ihmisen näkökyvyn imitoimista. Tämä mahdollistaisi erilaisten tehtävien automatisoinnin, jotka perustuvat ihmisen näkökykyyn. Konenäkö saattaa kuulostaa varsin yksinkertaiselta tehtävältä, sillä näkeminen on ihmisille triviaali tehtävä jo lapsenakin. Konenäkö on kuitenkin yksi haastavimmista tietojenkäsittelytieteen aloista. [viso.ai 2021] Vaikka konenäössä on viime vuosina tapahtunut suuriakin läpimurtoja, se on vielä kaukana lopullisesta tavoitteesta, geneerisestä ihmisen tavoin näkevistä järjestelmistä. Eräs syy tähän on puutteellinen ymmärrys biologisesta näkökyvystä – on vaikea imitoida jotain, mitä ei ymmärrä [Mihajlovic 2019].

Syväoppi on tekoälyn ja koneoppimisen osa-alue. Kehitykset syväopissa ovat mahdollistaneet nopean kehityksen konenäössä, sillä syväoppimismenetodit soveltuvat erittäin hyvin konenäön erilaisille tehtäville [Athanasios *et al.* 2018].

4.1 Konenäön tehtävät

Konenäköön kuuluu useita erilaisia tehtäviä, näistä yleisimmät ovat kuvan luokittelu, objektin tunnistus ja objektin segmentointi. Kuvan luokittelu on suoraviivainen tehtävä, jossa tavoitteena on luokitella kuva ennalta määritelyihin luokkiin, kuten koira tai kissa. Objektin tunnistustehtävissä tavoitteena on paikantaa ja luokitella kuvasta ennalta määrättyt objektit, kuten vaikkapa edellä mainitut koirat ja kissat. Objektin segmentointi on hyvin samankaltainen objektin tunnistuksen kanssa, mutta segmentoinnissa tavoitellaan tarkkaa rajausta, kun objektin tunnistuksessa riittää melko laaja rajalaukko. [Elgandy 2020 s.12]

Muita tyypillisiä tehtäviä konenäölle ovat muun muassa uusien kuvien luonti, kuvatekstin generointi ja kuvien suosittelujärjestelmät. [Elgandy 2020 s.14]

4.2 Konenäön sovelluskohteita

Konenäköä voidaan soveltaa erittäin laajasti useissa erilaisissa näköaistia vaativissa tehtävissä. Ajankohtainen esimerkki konenäköä hyödyntävästä sovelluskohteesta on autonomiset autot, joiden toiminta nojaa laajalti konenäköön, ympäristön hahmottamiseen ja tulkintaan [Mihajlovic 2019].

Terveystieteiden tutkimus on yksi lupaavista alueista, joissa konenäköä voisi hyödyntää. Terveystieteiden tutkimuksen tuottamasta datasta iso osa on kuvadataa. Tätä kuvadataa analysoimalla ja prosessoimalla, olisi mahdollista opettaa konenäköjärjestelmiä havaitsemaan muun muassa erilaisia syöpiä [Esteva *et al.* 2021]. Esimerkkinä on keuhkosyöpää tunnistava ja luokitteleva järjestelmä, jonka suorituskyky on verrattavissa kokeneisiin patologeihin [Yang *et al.* 2021].

Valvonta- ja turvallisuusjärjestelmät ovat yleinen konenäön hyödyntämiskohde. Toisin kuin ihmiset konenäköä hyödyntävä valvontajärjestelmä voi seurata samaan aikaan useaa valvontakameraa väsymättä koskaan etsien jatkuvasti normaalista poikkeavaa käyttäytymistä ja epänormaaleja objekteja. [Viernickel 2021]

4.3 Koneäköjärjestelmän vaiheet

Tyypillinen konenäköjärjestelmä koostuu seuraavista neljästä vaiheesta seuraavassa järjestyksessä.

1. Syöte
2. Esiprosessointi
3. Piirteiden erotus
4. Koneoppimisen malli

Riippuen käytettävistä tekniikoista piirteiden erotusvaihe saattaa jäädä kokonaan pois, sillä syväoppimisen mallit suorittavat piirteiden erotuksen automaattisesti. [Elgendy 2020 s.17]

4.3.1 Syöte

Koneäköjärjestelmän syöte on joko digitaalisia kuvia tai videoita. Videoita käsitellään vastaavaan tapaan kuin kuvia – video kuitenkin on vain sarja kuvia. Digitaaliset kuvat voidaan esittää funktiona, joka koostuu kolmesta muuttajasta (x , y , z). Muuttuja x kertoo kuvan leveyden, y korkeuden ja z ulottuvuuksien määrän. Tyypillisesti z -muuttujalla on kaksi mahdollista arvoa 1 ja 3. Mikäli ulottuvuuksia on 1, tällöin kuvaa tulkitaan mustavalkoisena tai harmaasävyisenä ja pikselien arvot viittaavat kyseisen pikselin kirkkauteen. Tyypillisesti tämä arvo on jotain 0 ja 255 välillä, nollan ollessa täysin musta ja 255 täysin valkoinen. Ulottuvuuksien määrän ollessa 3 tulkitaan kuvaa värillisenä, jolloin jokaisella pikselillä on kolme arvoa. Kukin arvo viittaa yhteen tyypillisesti RGB-värimallin värin voimakkuuteen, jotka ovat punainen, vihreä ja sininen. Kuvan tulkitseminen funktiona on erittäin hyödyllistä kuvan käsittelyn kannalta. Tällöin kuvalle on helppo suorittaa muunto-operaatioita, kuten kirkkauden muuttaminen ja kuvan kääntäminen. [Elgendy 2020 s.17-23]

4.3.2 Esikäsittely

Esikäsittely on tärkeä osa hyvin toimivaa konenäköjärjestelmää. Esikäsittelyn tarkoituksena on tehdä käytettävästä datasta mahdollisimman helppoa ja nopeaa analysoida, joka edistää koneoppimisen mallin suorituskykyä. [Elgendy 2020 s.23]

Eräs tyypillinen esikäsittelytekniikka konenäköjärjestelmille on syötekuvan muunto värillisestä mustavalkoiseksi. Värikuva koostuu kolmesta värikanavasta, jolloin sen prosessointi vaatii kolme kertaa enemmän resursseja verrattuna mustavalkoiseen kuvaan, joka koostuu vain yhdestä värikanavasta. Täten värilliset kuvat sisältävät enemmän informaatiota kuin mustavalkoiset kuvat. Värien tarjoama lisäinformaatio ei ole hyödyllistä kaikissa konenäön tehtävissä, jolloin syötekuvat kannattaa muuttaa

mustavalkoiseksi. Esimerkki tällaisesta tehtävästä on objektin tunnistus, jossa tunnistettavan objektin väri ei ole keskeisessä asemassa. [Elgendy 2020 s.23-24]

Koneoppimisen malleilla on erilaisia vaatimuksia syötedatan suhteen. Esimerkiksi konvoluutioneuroverkko vaatii kaikkien syötekuvien leveyden, korkeuden ja värikanavien määrän olevan samat kaikissa syötekuvissa, jolloin käytettävät kuvat tulee esikäsitellä samaan ennalta valittuun kokoon. [Elgendy 2020 s.24]

Esikäsitelyyn ei ole itsestäänselvää menetelmää, joka toimii jokaisessa tilanteessa. Vaikka on olemassa hyviä yleispäteviä ohjeita, niiden ei voida taata olevan optimaalisia jokaisessa tehtävässä. Esikäsitelyssä, kuten koneoppimisessa yleisestikin kannattaa kokeilla useita eri menetelmiä, jotta löytää parhaiten toimivat menetelmät erilaisiin tehtäviin. [Elgendy 2020 s.26-27]

Esikäsitelyn piiriin kuuluvat myös datan keinotekoinen lisääminen ja regularisaatio, joita avaan myöhemmin.

4.3.3 Piirteiden erotus

Koneoppimisessa piirre on jokin mittava ominaisuus. Esimerkiksi jos tehtävänä on luoda malli ennustamaan talon hintaa, tällöin hintaan vaikuttavia piirteitä voisivat olla esimerkiksi talon koko, huoneiden määrä ja rakennusvuosi. Oppimalla piirteiden vaikutuksen hintaan malli kykenee antamaan ennustuksen uudelle tapaukselle. [Elgendy 2020 s.27]

Konenäköjärjestelmä käyttää syöteenä kuvia. Tällöin piirteitä voivat olla vaikkapa haettavalle objektille tyypillinen väri ja muoto, kuten rengas, mikäli tehtävänä olisi esimerkiksi tunnistaa, onko kuvassa moottoripyörä vai kissa. Piirteiden erotus kuvadatasta ei perinteisesti ole ollut triviaali tehtävä. Kuitenkin kiitos syväoppimisen metodien piirteiden erotus tapahtuu automaattisesti, jolloin käyttäjän ei tarvitse sitä tehdä. [Elgendy 2020 s.32]

4.3.4 Koneoppimisen malli

Kun piirteiden erotus on tehty, on aika syöttää erotetut piirteet valitulle koneoppimisen mallille. Koneoppimisen malleja on kahdenlaisia, perinteisiä koneoppimisen malleja ja syväoppimisen malleja. Perinteisiä malleja ovat muun muassa päätöspuut, tukivektorikoneet, kun syväoppimisen malleihin kuuluvat muun muassa konvoluutioneuroverkko ja takaisinkytketty neuroverkko. Perinteisiä ja syväoppimisen malleja voidaan käyttää samoihin tehtäviin, mutta syväoppimisen mallit tekevät eroa perinteisiin malleihin etenkin, kun syöteenä toimii kuvadata. Syväoppimisen mallit myös hyötyvät

paremmin suuresta datamäärästä verrattuna perinteisiin koneoppimisen malleihin. [Elgendy 2020 s.34]

4.4 Konvoluutioneuroverkko

Konvoluutioneuroverkko (Convolutional neural network/CNN) on syväoppimisen algoritmi, jota tyypillisesti käytetään kuvadatan analysointiin.

4.4.1 Arkkitehtuuri

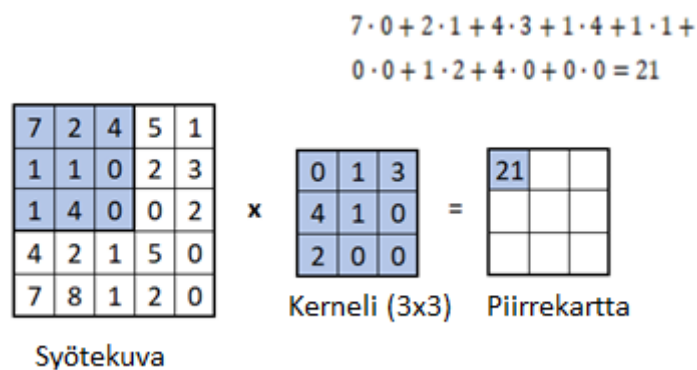
Konvoluutioneuroverkko koostuu useista eri kerroksista (layers), joiden kokonaisuus muodostaa verkon arkkitehtuurin. Arkkitehtuurilla on suuri vaikutus verkon suorituskykyyn. Erilaisia kerroksia on useita, joista yleisempiä ovat:

- Konvoluutio
- Pooli (Pooling)
- Täysin yhdistetty (Fully connected)

4.4.1.1 Konvoluutio

Konvoluutiokerrokset, joista konvoluutioneuroverkko on saanut nimensä, ovat konvoluutioneuroverkon arkkitehtuurin keskeisin osa. Kerros suorittaa useita konvoluutio-operaatioita syötedatan ja painoista koostuvan matriisin kanssa, jota kutsutaan nimellä suodin (filter) tai kerneli. Konvoluutiokerroksen toimintaa voidaan kuvailla piirteitä etsivänä ikkunana, jota systemaattisesti liikutetaan pitkin syötekuvaa yksi askel kerrallaan. Tyypillisesti ensimmäiset konvoluutiokerrokset oppivat tunnistamaan yksinkertaisia piirteitä kuten reunoja ja kulmia, kun syvemmät kerrokset monimutkaisempia piirteitä kuten vaikkapa silmän, renkaan tai kissan tassun. [Elgendy 2020 s.107-108]

Kuvassa 7 on esimerkki, kuinka kernelin ja syötekuvan samankokoisesta alueesta luodaan arvo piirrekarttaan. Tämä prosessi toistetaan, kunnes piirrekartta on täynnä liikuttamalla kerneliä systemaattisesti jokaisessa kuvan kohdassa, johon kerneli mahtuu.



Kuva 7. Piiirrekartta saa arvonsa syöte kuvan ja kernelin pistetulosta.

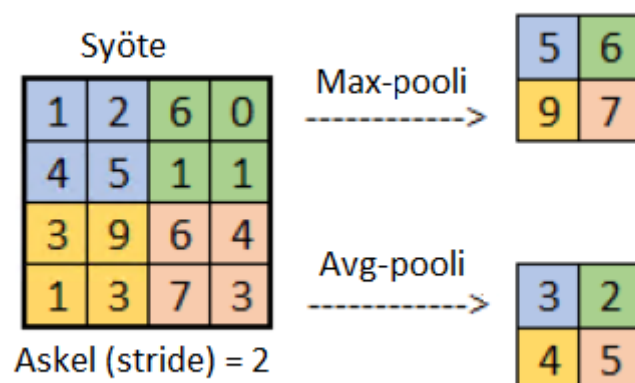
Yksi kerneli ei kuitenkaan ole tarpeeksi konvoluutioneuroverkon opetukseen. Yhden kernelin sijaan konvoluutiokerroksessa on useita kerneleitä, jotka rinnakkain tuottavat erilliset piirrekartat. Lopuksi nämä konvoluutiokerroksen tuottamat piirrekartat yhdistetään yhdeksi tulokseksi, joka lähetetään syötteenä eteenpäin verkon seuraavalla kerrokselle. [Elgendy 2020 s.110-111]

Piirrekartan kokoon voidaan vaikuttaa muutamilla parametreilla. Usein halutaan piirrekartan olevan samankokoinen kuin syötekuva. Tämän saavuttamiseksi käytetään niin sanottua zero-padding menetelmää, joka kasvattaa syötekuvan kokoa lisäämällä sille uuden ulkoreunan, jossa jokaisen pikselin arvo on 0. Mikäli halutaan pienentää piirrekartan kokoa, voi sen tehdä askel-parametria (stride) hyödyntäen. Yksinkertaisuudessaan askel määrittää, kuinka monta pikseliä liikutaan per askel. Mikäli askel on määritetty arvolla 2, tällöin kerneliä liikutetaan 2 pikseliä kerrallaan yhden sijasta. Kuvan 7 piirrekartta olisi 2x2 kokoinen 3x3 sijasta, mikäli askel olisi 2.

Jokaisen konvoluutiokerroksen jälkeen tulee aktivointikerros, jossa yhdistetylle piirrekartalle käytetään epälineaarista funktiota, joka tuottaa uuden piirrekartan. Erilaisia aktivointifunktioita on useita, mutta selkeästi yleisin on Rectified Linear Units (ReLU). [Rosebrock 2021]

4.4.1.2 Pooli (Pooling) kerros

Konvoluutiokerrosta usein seuraa poolikerros. Poolikerroksen tarkoitus on vähentää mallin parametrien määrää pienentämällä syötteen kokoa. Poolikerroksia on kahdentyypisiä, max ja avg. Max-pooli etsii suurimman arvon valitusta ikkunasta hyläten kaikki muut. Average-pooli puolestaan laskee ikkunan arvojen keskiarvon (kuva 8). [Elgendy 2020 s.115]



Kuva 8. Esimerkki max- ja avg-poolioperaatioista 4x4 syötteelle askeleen ollessa 2 ja pooli-ikkuna 2x2.

Poolikerroksen ikkunan koko on tyypillisesti 2×2 ja sen kanssa usein käytetään askelta arvolla 2. Tällöin tuloksena on niin sanottu ei-limittäinen pooli (non overlapping pooling), jossa jokaista data elementtiä tarkastetaan vain kerran. Poolikerros vähentää aggressiivisesti datan määrää, ja pienin mahdollinen pooli-ikkuna 2×2 ja askel 2 vähentää datan määrää 75 %. Kun datan määrää vähennetään kerros toisensa jälkeen 75 %, viimeisten kerrosten syötteen koko jää nopeasti liian pieneksi, joka vaikuttaa negatiivisesti mallin suorituskykyyn. [Rosebrock 2021]

4.4.1.3 Täysin yhdistetty (Fully connected) kerros

Täysin yhdistettyjä (fully connected) FC-kerroksia käytetään tyypillisesti verkon arkkitehtuurin viimeisinä kerroksina. Usein FC-kerroksia on kaksi tai kolme peräkkäin. FC-kerros suorittaa itse luokittelun edellisten kerrosten erottamista piirteistä. Luokittelutehtävissä FC-kerroksen jälkeen käytetään tyypillisesti softmax-aktiivointifunktiota, joka palauttaa kyseiselle tapaukselle todennäköisyyden kuulua johonkin ennalta määritettyyn luokkaan. [Rosebrock 2021]

4.4.1.4 Esimerkki arkkitehtuurista

Konvoluutioneuroverkon arkkitehtuuri voidaan visualisoida taulukon muodossa. Taulukossa 7 on AlexNet-arkkitehtuuri, joka mainitaan myöhemmin ImageNet luvussa 4.5. AlexNet käyttää syötteenä värillisiä kuvia, joiden resoluutio on 227×227 . Konvoluutio 1 ja max-pooli 1 pienentävät piirrekartan kokoa aggressiivisesti. Ensimmäisessä konvoluutiokerroksessa käytetään parametreinä kerneleiden määrää 96, kernelin kokoa 11×11 , askelta 4 ja ReLu aktiivointifunktiota. Ensimmäinen konvoluutiokerros tuottaa tuloksena $55 \times 55 \times 96$ kokoisen piirrekartan, jolle suoritetaan max-pooli 1 operaatio, jossa käytetään kernelin kokoa 3×3 ja askelta 2. Tämä palauttaa $27 \times 27 \times 96$ kokoisen piirrekartan seuraavalle konvoluutiokerrokselle 2. Konvoluutiokerroksen 2 tuloksena on $27 \times 27 \times 256$ kokoinen piirrekartta, jonka max-pooli 2 pienentää kokoon $13 \times 13 \times 256$. Seuraavaksi on vuorossa kolme peräkkäistä konvoluutiokerrosta 3–5, joiden tarkoitus on havaita monimutkaisia piirteitä. Konvoluutiokerrosten jälkeen suoritetaan viimeinen max-pooli 3 operaatio, joka palauttaa viimeisen $6 \times 6 \times 256$ kokoisen piirrekartan ennen FC-kerroksia. FC-kerrokset 1 ja 2 erottelevat lopulliset piirteet ReLu aktiivointifunktiota käyttäen lähettäen sen viimeiselle FC-kerrokselle 3, joka palauttaa Softmax aktiivointifunktion avulla mallin ennustamat todennäköisyydet sille, että syöte kuuluu kuhunkin ennalta määrättyyn 1000 eri luokkaan.

Kerros	Kerneleiden määrä	Kernelin koko	Askel	Padding	Piirrekartan koko	Aktivointifunktio
Syöte					227 x 227 x 3	
Konvoluutio 1	96	11 x 11	4		55 x 55 x 96	ReLu
Max-pooli 1		3 x 3	2		27 x 27 x 96	
Konvoluutio 2	256	5 x 5	1	2	27 x 27 x 256	ReLu
Max-pooli 2		3 x 3	2		13 x 13 x 256	
Konvoluutio 3	384	3 x 3	1	1	13 x 13 x 384	ReLu
Konvoluutio 4	384	3 x 3	1	1	13 x 13 x 384	ReLu
Konvoluutio 5	256	3 x 3	1	1	13x 13 x 256	ReLu
Max-pooli 3		3 x 3	2		6 x 6 x 256	
FC 1					4096	ReLu
FC 2					4096	ReLu
FC 3					1000	Softmax

Taulukko 7. AlexNet-arkkitehtuuri.

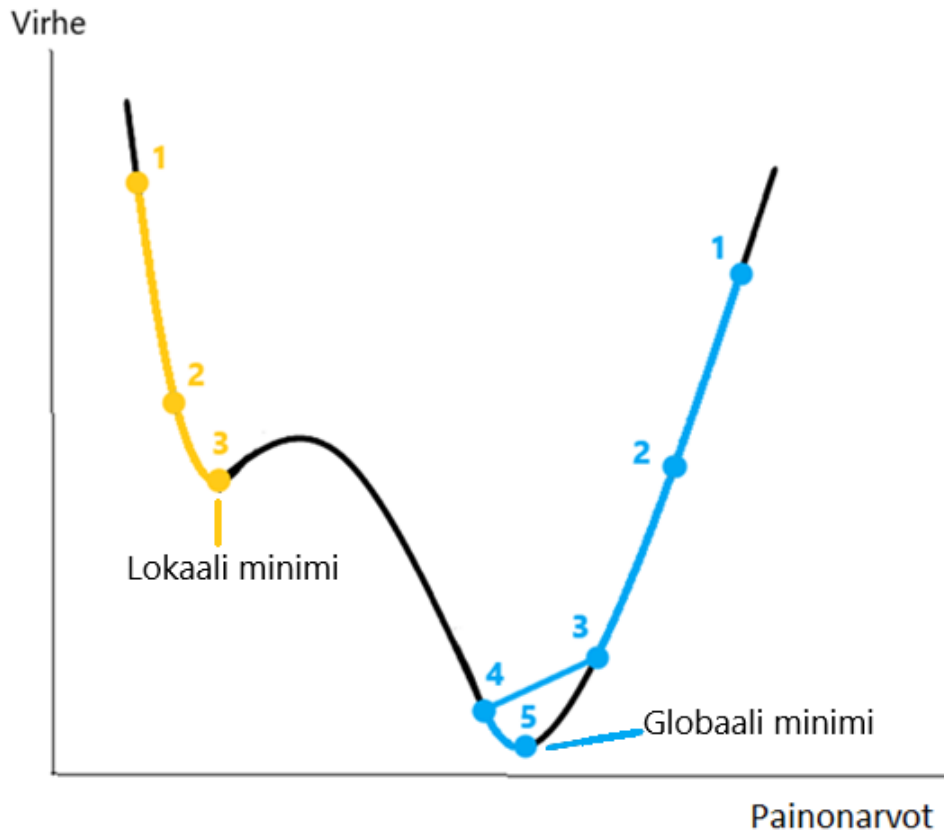
4.4.2 Takaisinsyöttävä käänös (Backward Propagation)

-nökseksi (backward propagation). Mallin tavoitteena on tuottaa tai oppia painojen arvot siten, että se tekee mahdollisimman vähän virheitä. [Towards Machine Learning 2018]

Opetusprosessin alussa mallin painojen arvoiksi asetetaan satunnaiset arvot. Tämän jälkeen syöte kulkee kerros toisensa jälkeen syvemmälle saavuttaen viimeisen kerroksen, joka tuottaa ennustetun luokan ja ennustamisvirheen (prediction error). Mikäli ei saavutettu haluttua tulosta, tämä ennustamisvirhe lähetetään taaksepäin edelliselle kerrokselle, jossa sen avulla muutetaan kerroksen painojen arvoja, joka puolestaan lähettää sen taaksepäin seuraavalle kerrokselle. Tämä prosessi jatkuu, kunnes jokainen kerros on käyty lävitse, jonka jälkeen uusien painojen arvojen avulla lasketaan syötteelle uusi ennustamisvirhe. Mikäli ei vielääkään saavutettu haluttua tulosta, lähtee ennustamisvirhe uudelle kierrokselle taaksepäin muuttaen ja tuottaen uudet arvot kerrosten painoille. Prosessi jatkuu, kunnes ennustamisvirhe saadaan tarpeeksi pieneksi, jonka jälkeen seuraava syöte jatkaa samoilla painojen arvoilla. Tavoitteena on löytää eli optimoida painojen arvot siten, että ne tuottavat mahdollisimman pienen ennustamisvirheen. Itse optimointi tapahtuu optimointialgoritmin avulla. [Towards Machine Learning 2018]

4.4.2.1 Optimointialgoritmit

Optimointialgoritmeja käytetään parhaiden painoarvojen löytämiseen, jotka tuottavat mahdollisimman pienen ennustamisvirheen eli globaalin minimin [Towards Machine Learning 2018] (kuva 9).



Kuva 9. Optimointialgoritmin liike kohti mahdollisimman pientä virhettä.

Kuva 9 hahmottaa, kuinka optimointialgoritmi liikkuu kohti globaalia minimiä. Pisteet numerolla 1 kuvaavat aloituspistettä, jotka ovat alustettu satunnaisilla painoarvoilla. Sininen saavuttaa oikein globaalin minimin, mutta keltainen ei. Keltainen jää jumiin lokaaliin minimiin. Eräs tapa välttää lokaaliin minimiin jumiin jäämistä on kasvattaa oppimisnopeusmuuttujaa (learning rate). Oppimisnopeus määrittää, kuinka paljon painoja muutetaan per askel täten määrittäen askeleen pituuden [Towards Machine Learning 2018]. Keltaisen tapauksessa korkeampi oppimisnopeus olisi mahdollistanut hypyn pienen nypyn yli, jolloin globaali minimi olisi saavutettu. Oppimisnopeuden tekemän hypyn näkee sinisen 3–4 askeleessa, jossa se hyppää globaalin minimin yli joutuen kääntymään takaisinpäin. Liian suurta oppimisnopeutta tulee myös välttää, sillä se nopeasti johtaa epäoptimaaliseen tulokseen.

Laskeutuva gradientti (gradient decent) on eräs yleisimmin käytetyistä optimointialgoritmeista. Kuva 9 vastaa laskeutuvaa gradienttia. Se on laskennallisesti tehokas ja helposti ymmärrettävissä. Laskeutuvan gradientin ongelma on sen alttius jäädä jumiin lokaaliin minimiin. Laskeutuvan gradientin algoritmista on olemassa muutamia muunnoksia, kuten stokastinen laskeutuva gradientti (Stochastic gradient decent (SGD)) ja erää käyttävä laskeutuva gradientti (Mini-batch gradient decent). Näiden molempien muunnosten ideana on päivittää painoarvoja useammin. Laskeutuva gradientti päivittää painoarvoja vain kerran kierroksen aikana. SGD päivittää painoarvoja jokaisen syötteen jälkeen, kun

mini-batch päivittää painoja jokaisen erän (batch) jälkeen. Nämä muunnokset eivät kuitenkaan ratkaise kaikkia laskeutuvan gradientin ongelmia, jonka vuoksi nekin kärsivät alttiudesta jäädä jumiin lokaaliin minimiin. [Doshi 2019]

Syväoppimista hyödyntävissä tehtävissä Adam on eräs tehokkaimmista optimointialgoritmeista. Adam ei kärsi samoista ongelmista kuin laskeutuva gradientti ja sen muunnokset. Tämän lisäksi Adam on laskennallisesti kohtalaisen nopea ja vähän muistia vaativa verrattuna moniin muihin vastaavan tehokkaisiin optimointialgoritmeihin kuten AdaDelta. [Arora 2021]

4.5 ImageNet

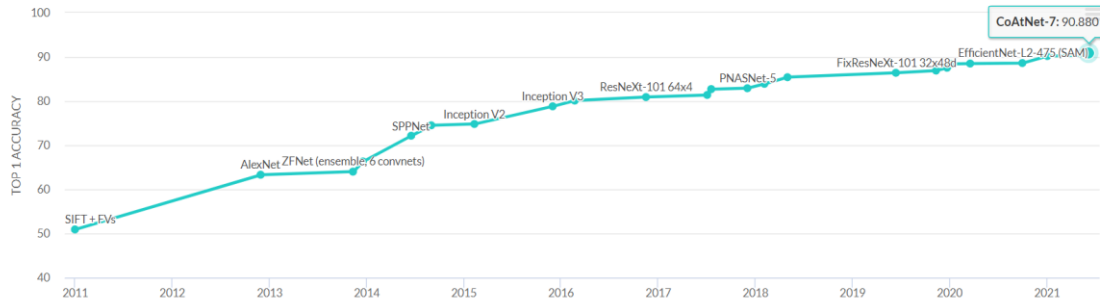
ImageNet kostuu suuresta määrästä manuaalisesti luokiteltua kuvadataa. Sen pohjimmainen tarkoitus on edistää konenäkö tutkimusta tarjoamalla suuri aineisto mallien opetukseen ja vertailuanalyysiin. Yhteensä ImageNet-aineisto pitää sisällään noin 14 miljoonaa manuaalisesti luokiteltua kuvaa ja noin 22 00 luokkaa. [Elgandy 2020 s.204] Aineistossa on kahdentyyppisiä luokittelutasoja, koko kuvan taso ja objektitaso. Koko kuvan taso koostuu binäärisestä luokittelusta, joka ilmaisee esimerkiksi: kuvassa on auto, kuvassa ei ole kissaa. Objektitason luokittelussa edellisen lisäksi luodaan rajalaatikat kullekin kuvassa olevalle luokalle, josta käy ilmi objektin sijainti kuvassa. [Russakovsky *et al.* 2014]

ImageNet-tulosten vertailuanalyysissä käytetään kolmea mittaria: Top 1 -tarkkuus, Top 5 -tarkkuus ja parametrien määrä. Top 1 -tarkkuutta voidaan pitää tärkeimpänä mittarina, koska se kertoo mallin tarkkuuden, kun sen ennuste rajoitetaan yhteen mallin mukaan todennäköisimpään luokkaan. Top 5 -tarkkuus puolestaan viittaa tilanteeseen, jossa oikea luokka löytyy viiden todennäköisimmän luokan joukosta. Parametrien määrä kertoo arkkitehtuurin laskennallisesta monimutkaisuudesta, mutta se itsessään ei kuitenkaan riitä koko kuvan luomiseen laskennallisesta monimutkaisuudesta.

Ensimmäinen merkittävä merkkipaalu ImageNet-kilpailussa tapahtui vuonna 2012, jolloin AlexNet voitti ylivoimaisesti ImageNet-kilpailun saavuttaen 63.3 % top 1 -tarkkuuden ja 84.6 % top 5 -tarkkuuden, kun kakkien muiden kilpailuun osallistuneiden mallien top 5 -tarkkuus jäi alle 75 %. [Elgandy 2020 s.211] AlexNet:in tuottamat tulokset herättivät suuren kiinnostuksen konenäkö tutkimukselle, ja sitä voidaankin pitää yhtenä merkittävimmistä edistysaskeleista koko konenäkö tutkimuksessa [Brownlee 2019a].

AlexNetin jälkeen top 1 ja top 5 -tarkkuudet ovat nousseet useaan otteeseen. Muutamia merkkipaaluja olivat vuonna 2015 VGG19-arkkitehtuurin 74.5 % top 1 -tarkkuus ja 92.0 % top 5 -tarkkuus. Vuonna 2016 Inception ResNet V2 toimi 80.1 % top 1 -tarkkuudella ja 95.1 % top 5 -tarkkuudella. Näiden suurten harppausten jälkeen kehitys ymmärrettävästi hidastui, muttei suinkaan lakannut. Korkeimman top 1 -tarkkuuden 90.9 % saavutti vuonna 2021 CoAtNet-7 (kuva 10). Top 5 -tarkkuus on jäänyt vähemmälle huomiolle, eikä muun muassa CoAtNet-7 ole julkaissut top 5 -tarkkuutta. Tällä hetkellä

parhaan top 5 -tarkkuuden saavutti Florence-CoSwin-H vuonna 2021 99 % tarkkuudella. Top 5 -kiinnostuksen puute saattaa johtua siitä, että sitä saatetaan pitää ratkaistuna, kun paras tarkkuus on ylittänyt jo 99 % rajan. [Papers with code 2021]



Kuva 10. ImageNet top 1 -tarkkuuden state-of-the-art-mallit.

4.6 Evaluointi

Koneoppimisen mallin evaluointi on tärkeä vaihe mitä tahansa koneoppimisen mallia hyödyntävää järjestelmää. Uskottava evaluointi vaatii ongelmaan sopivat evaluointimetriikat [Elgendy 2020 s.147] sekä datan säästämisen testaamiseen ja validointiin [Elgendy 2020 s.136].

4.6.1 Evaluaatiometriikat luokittelussa

Kun suoritetaan luokittelua, käytetään seuraavia arviointisuureita: oikea positiivinen (OP), oikea negatiivinen (ON), väärä positiivinen (VP) ja väärä negatiivinen (VN). Nämä suureet ovat helppoja hahmottaa binäärisen esimerkin kautta, jona tavoitteena on luokitella, onko ihminen terve vai sairas. Tällöin OP viittaa sairaisiin ihmisiin, jotka luokiteltiin oikein sairaaksi. ON terveisiin ihmisiin, jotka luokiteltiin oikein terveiksi. VP terveisiin, jotka luokiteltiin väärin sairaiksi. Ja VN sairaisiin, jotka luokiteltiin väärin terveiksi. Näiden neljän tapahtuman arvot taulukoidaan niin sanottuun sekaannusmatriisiin ja niitä käytetään luokittelun evaluaatiometriikoiden laskemiseen. Taulukossa 8 on esimerkki sekaannusmatriisista [Elgendy 2020 s.147-148]

		Ennustettu luokka	
		0	1
Oikea luokka	0	51 OP	9 VN
	1	5 VP	42 ON

Taulukko 8. Esimerkki sekaannusmatriisista

Yleisiä luokittelun evaluaatiometriikoita ovat tarkkuus, saanti, täsmällisyys ja F1-luku.

$$tarkkuus = \frac{OP + ON}{OP + ON + VP + VN}$$

$$saanti (recall) = \frac{OP}{OP + VN}$$

$$täsmällisyys (precision) = \frac{OP}{OP + VP}$$

$$F1-luku = \frac{2 * täsmällisyys * saanti}{täsmällisyys + saanti}$$

4.6.2 Epätasapainoinen data

Käytössä olevan datan luokkajakauman vinous on otettava huomioon, kun valitaan käytettävät luokittelun evaluaatiometriikat. Esimerkiksi tarkkuus arviointimetriikkana on hyvä ainoastaan silloin,

kun datan luokkajakauma on tasapainoinen. Vastaesimerkkinä on tilanne, jossa luokkaa 1 on aineistossa 100:1 luokkaan 2 verrattuna. Tällöin naiivi malli, joka ennustaa jokaisen tapauksen kuuluvan enemmistöluokkaan 1 saavuttaa 99 % tarkkuuden. Korkeasta tarkkuudesta huolimatta, malli ei kuitenkaan tarjoa hyödyllistä informaatiota. Tarkkuuden sijasta mielekkäämpiä metriikoita epätasapainoisen datan kanssa ovat saanti, täsmällisyys ja näillä laskettava F1-luku. [Elgendy 2020 s.147-149]

4.6.3 Moniluokkainen luokittelu

Huomattavaa on, että muun muassa F1-luku on luotu binääriseen luokitteluun ja mikäli sitä käytetään moniluokkaisessa luokittelutehtävässä, tulee sen laskukaavaa muuttaa. Tyypillinen lähestymistapa F1-luvun soveltamiseen moniluokkaisessa luokittelutehtävässä on käyttää yksi vastaan muut-menetelmää kunkin luokan oman F1-luvun laskemiseksi. Seuraavaksi tulee valita, kuinka yhdistää nämä erilliset luokkien F1-luvut yhdeksi arvoksi. Tähän on olemassa useampi erilainen vaihtoehto, mutta eräs yleinen vaihtoehto on painotettu keskiarvo kunkin luokan tapauksien määrien perusteella. [Baldung 2020]

4.6.4 Datan jakaminen osiin

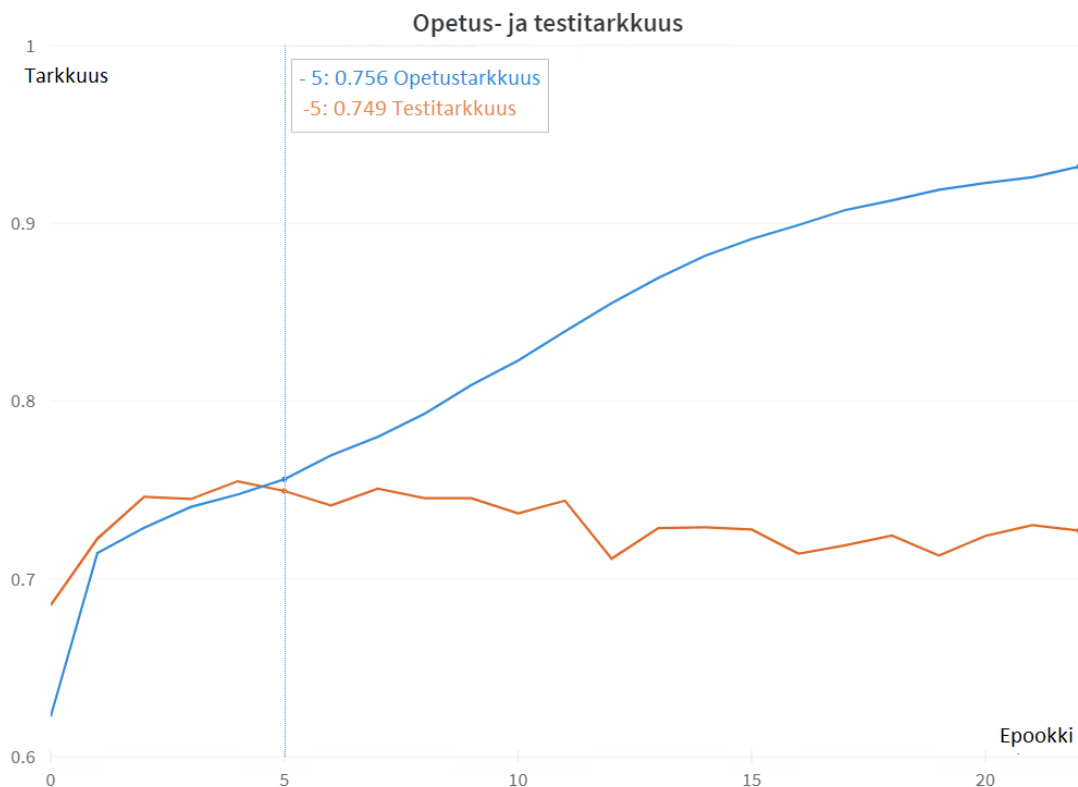
Datan säästämisen ideana on luoda erilliset datajoukot mallin opettamiseen, validointiin ja testaamiseen. Validointidatajoukkoa käytetään mallin opetuksen yhteydessä. Se avulla voidaan seurata mallin suorituskykyä opetuksen aikana, mikä mahdollistaa varhaisen lopetuksen hyödyntämisen ylisovittautumisen ehkäisemiseksi. Testidata mahdollistaa mallin suorituskyvyn arvioinnin opetuksen jälkeen uusilla tapauksilla, joita malli ei ole aikaisemmin nähnyt. Tämä siis simuloi tilannetta, jossa mallia käytettäisiin oikeasti. Ei ole yhtä oikeaa vastausta, miten suuriin osiin data tulisi jakaa. Opetukseen tulisi kuitenkin käyttää selkeästi eniten dataa. Esimerkki jaosta voisi olla 70/15/15, jolloin opetukseen menee 70 % ja validointiin ja testaamiseen 15 % datasta. [Elgendy 2020 s.151-152]

4.6.5 k-ristiinvalidointi (k-fold cross validation)

Etenkin silloin kun käytössä olevan datan määrä on suppea, kannattaa hyödyntää k-ristiinvalidointimenetelmää (k-fold cross validation). Sen ideana on jakaa opetusdata siten, että jokaista datatapausta käytetään opetuksessa sekä validoinnissa. Kun käytetään k-ristiinvalidointia ei tarvita erillistä validointidatajoukkoa. Menetelmä toteutetaan jakamalla opetusdata pienempiin osiin, tyypillisesti kymmeneen osaan (k=10). Tämän jälkeen mallia opetetaan 10 kertaa siten, että 9/10:aa datasta käytetään opetuksessa ja 1/10:aa validoinnissa. Prosessi toistetaan 10 kertaa, jolloin jokaista osaa on käytetty kerran validointidatana. Lopuksi mallin suorituskyky voidaan tiivistää muun muassa laskemalla evaluaatiometriikoiden keskiarvoja, vaihteluväliä ja ottamalla ylös suurin ja pienin tulos. [Brownlee 2018]

4.7 Ylisovittautuminen (overfitting)

Ylisovittautuminen (overfitting) viittaa tilanteeseen, jossa opetettu malli kykenee luokittelemaan opetusdataa erittäin tarkasti, mutta sillä on vaikeuksia uuden datan kanssa, kuten validointi tai testidata. Termi epookki (epoch) kertoo, kuinka monta kertaa malli on käynyt lävitse koko opetusdatan. Kuvassa 11 on esimerkki ylisovittautumisesta. Epookkiin viisi asti opetus- ja testidatan tarkkuus ovat lähellä toisiaan. Kuitenkin jo epookissa kuusi on jo huomattavissa melko selkeää ylisovittautumista, joka jatkaa voimistumistaan loppuun asti, jolloin tarkkuus opetusdatan kanssa kasvaa noin 92 % testi tarkkuuden tippuessa noin 72 %.



Kuva 11. EfcientNetB4-esimerkki ylisovittautumisesta.

Ylisovittautuminen on yleinen ja vaikea ongelma koneoppimisen mallien kanssa. Se on eräs yleisimmistä syistä mallin heikolle suorituskyvyille. Ylisovittautuminen perustuu ilmiöön, jossa malli oppii opetusdatan niin sanotusti ”ulkoa” eikä tällöin opi luokkien ominaispiirteitä. Tämä johtaa tilanteeseen, jossa mallilla on ongelmia luokitella uutta dataa, mikä on sen pohjimmainen tarkoitus. [Elgendy 2020 s.125]

4.7.1 Ylisovittautumisen ehkäisy

Ylisovittautumisen ehkäisyyn on useita eri keinoja. Vaikeutena on, ettei ole itsestäänselvää, miten hyvin mikäkin keino toimii missäkin tilanteessa. Tämän vuoksi on suositeltavaa kokeilla useita eri

menetelmiä ylisovittautumisen ehkäisyyn, jotta löytää parhaiten toimivat menetelmät kyseiseen tilanteeseen.

Tyypillisiä keinoja ylisovittautumisen ehkäisyyn syväoppimisen mallien kanssa ovat:

1. Varhainen lopetus (early stopping)
2. Opetusdatan määrän kasvatus
3. Keinotekoisen datan lisääminen (data augmentation)
4. Regularisaatio
5. Yhdistelmämetodit (Ensembled methods)

[IBM 2021]

4.7.1.1 Varhainen lopetus (early stopping)

Tehokas ja helppo keino eliminoida ylisovittautuminen on käyttää varhaisen lopetuksen (early stopping) menetelmää, jossa mallin opetus keskeytetään automaattisesti, kun vaikuttaa, että malli alkaa ylisovittua. Graafin 4 tilanteessa mallin opetus lopetettaisiin epookkiin 4 tai 5, jolloin validointitarkkuus oli korkeimmillaan noin 75 %. Varhainen lopetus itsessään kykenee estämään mallin ylisovittautumisen kokonaan. Sen merkittävänä riskinä kuitenkin on ylisovittautumisen vastakohta alisovittautuminen (underfitting), jolloin malli ei opi edes opetus dataa. Optimaalinen malli on sellainen, joka on ylisovittautumisen ja alisovittautumisen välillä. [IBM 2021]

4.7.1.2 Opetusdatan määrän kasvatus

Opetusdatan määrän kasvatuksen idea ylisovittautumisen ehkäisyyn on antaa mallille enemmän mahdollisuuksia löytää luokkien ominaispiirteet. Huomattavaa kuitenkin on, että uuden datan tulisi vastata vanhaa dataa. Mikäli uusi data poikkeaa vanhasta datasta, se voi johtaa entistä pahempaan ylisovittautumiseen. [IBM 2021]

4.7.1.3 Datan keinotekoinen lisääminen (data augmentation)

Datan määrän lisääminen ei aina ole mahdollista. Tällöin eräs vaihtoehto datan määrän kasvattamiseen on luoda uusia datatapauksia vanhoja muokkaamalla eli datan keinotekoinen lisääminen (data augmentation). Datan keinotekoinen lisääminen toimii erityisen hyvin ja intuitiivisesti kuvadatan kanssa. Kuva datalle on helppo suorittaa muunto-operaatioita, kuten kuvan peilaus, kierto, kirkkouden muuttaminen ja kohinan lisääminen. Näitä muunnettuja kuvia voidaan käyttää muun datan lisäksi, tai vaihtoehtoisesti suoritetaan satunnaisia muunto-operaatioita kaikille kuville, jolloin sama kuva on aina hieman erilainen jokaisessa erässä. [Elgendy 2020 s.180]

4.7.1.4 Regularisaatio

Ylisovitukselta kärsivä malli on liian monimutkainen ja sen painot ovat erikoistuneet opetusdataan. Neuroverkon monimutkaisuus syntyy kahdesta tekijästä, painojen määrästä ja niiden arvoista. Näiden tekijöiden kautta voidaan vaikuttaa mallin monimutkaisuuteen. Painojen määrään voidaan vaikuttaa muuttamalla mallin rakennetta ja painojen arvoihin parametrien avulla. [Brownlee 2019b]

Mallin monimutkaisuuden vähentäminen rakenteen kautta tarkoittaa tyypillisesti kerrosten systemaattista poistamista ja pienentämisestä. Tätä jatketaan, kunnes ylisovittautumista ei enää havaita. Rakenteellisten muutosten sijasta yleisempi tapa mallin monimutkaisuuden vähentämiseen on rajoittaa sen painojen arvoja. Tämä tarkoittaa painojen arvojen pitämistä pieninä. Pienet painojen arvot tekevät mallista yksinkertaisemmän, jolloin se ei ole niin altis syötedatan luonnollisille tilastollisille vaihteluille. Näitä menetelmiä, jotka rajoittavat painojen arvoja pysymään pieninä, kutsutaan regularisaatioksi. Termiä regularisaatio käytetään yleisesti myös metodille, joiden tarkoituksena on vähentää yleistämisvirhettä, muttei opetusvirhettä. Tämän vuoksi aikaisemmin mainittuja metodeja varhainen lopetus ja datan keinotekoinen lisääminen voidaan myös kutsua regularisaatiomenetelmiksi. [Brownlee 2019b]

Neuroverkkojen yleisin regularisaatio menetelmä on L2-regularisaatio (weight decay). Menetelmä kannustaa mallia pitämään painojen arvot pieninä. Tämä tapahtuu lisäämällä regularisaatiotermi mallin kustannusfunktioon (loss function). [Brownlee 2019b]

Toinen yleinen regularisaatiometodi on lisätä neuroverkon arkkitehtuuriin pudotuskerroksia (dropout layer). Yksinkertaisuudessaan pudotusmenetelmä deaktivoi osan kerroksen neuroneista, jolloin niitä ei oteta huomioon lainkaan kyseisellä kierroksella. Menetelmä saattaa aluksi vaikuttaa epäloogiselta, mutta se kannustaa jokaista neuronaa ottamaan enemmän vastuuta lopullisesta tuloksesta. Tällöin esitetään tilanne, jossa ainoastaan yksi dominoi muita. Pudotuskerros tyypillisesti alustetaan deaktivoidaan 30 % - 50 % kerroksen neuroneista. [Elgandy 2020 s.125-127]

5 Laitteisto, kuvien esikäsittely ja rajaukset

5.1 Laitteisto

Tutkimus suoritettiin kennelliiton tarjoamalla tietokoneella, jossa seuraava laitteisto:

Käyttöjärjestelmä: Windows 10

Prossessori: 11th Gen Intel i7-11700K

Hajasaantimuisti: 16GB

Näytönohjain: NVIDIA GeForce RTX 3070 8GB

Ohjelmointi kielenä käytin Python 3 ja sen seuraavia kirjastoja:

- Keras
- Tensorflow
- Pandas
- Weights and biases
- Jupyter lab
- Mathplotlib
- SKlearn
- LabelImg
- Skimage
- Numpy
- Pydicom

5.2 Tiedostomuoto

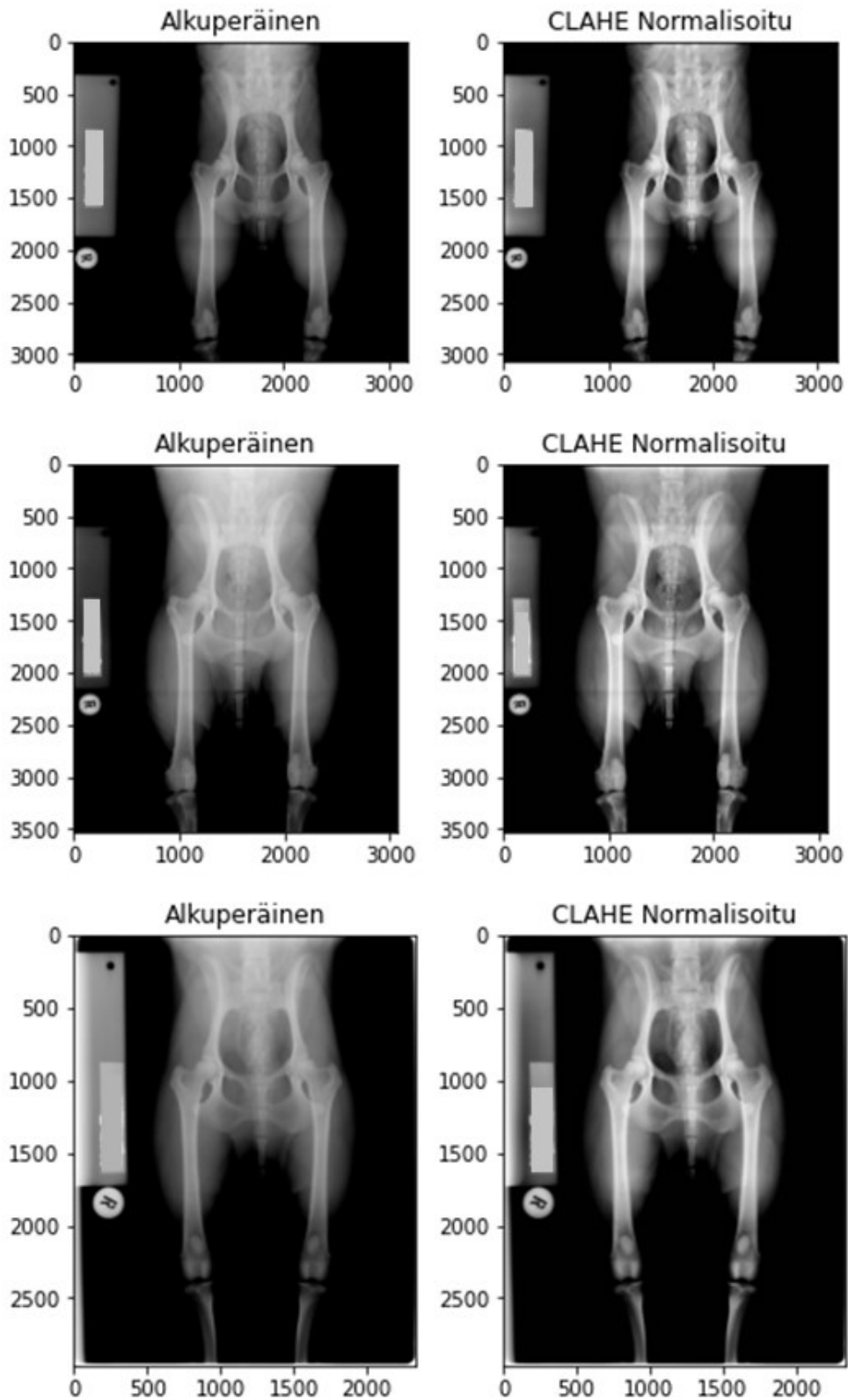
Alkuperäiset röntgenkuvat ovat DICOM-muodossa. Koneoppimisen kirjastot eivät tue laajasti DICOM-tiedostomuotoa, jonka vuoksi kuvien tiedostomuoto kannattaa vaihtaa paremmin tuettuun muotoon. Hyviä vaihtoehtoja lopulliselle kuvatiedostomuodolle on pari, JPG ja PNG. Lopulta valitsin PNG:n tiedostomuodoksi, sillä se on häviötön, eli kuva ei menetä informaatiota pakkauksessa.

5.3 Normalisointi

Lonkkanivelröntgenkuvia otetaan useiden eri valmistajien laitteilla ja erilaisilla asetuksilla. Tällöin otetuissa kuvissa on havaittavissa kuvauslaitteen kädenjälki, jolloin koneoppimisen mallit saattavat ylisovittautua laitteet tuottamaan pikselijakaumaan. [County 2020]

Röntgenkuville tyypillisiä normalisointitekniikoita ovat histogramminormalisointi ja CLAHE-normalisointi. Histogramminormalisoitu röntgenkuva näyttää usein tummemmalta, kun CLAHE-normalisointi tuottaa terävän kuvan, jossa etenkin luusto korostuu. Kun tarkoituksena on tarkastella koirien

lonkkaniveleä, on luuston korostus mielekäästä (kuva 12). Tämän vuoksi normalisoisin kaikki röntgenkuvat CLAHE-normalisoinnilla.



Kuva 12. Kuvia ennen ja jälkeen CLAHE-normalisoinnin.

5.4 AHE ja CLAHE

AHE *Adaptive Histogram Equalization* on digitaalisten kuvien prosessointitekniikka, jonka tavoitteena on parantaa kuvan kontrastia tehden kuvasta terävämmän. Tämä tapahtuu laskemalla useita pikseli-intensiteettihistogrammeja kuvan eri alueista. Näiden paikallisten histogrammien avulla uudelleen rakennetaan pikseleiden kirkkausarvot, joista syntyy alkuperäistä terävämpi kuva. [Rubini and Pavithra 2019]

AHEN yksi ongelma on se, että sillä on taipumus korostaa liikaa kuvassa esiintyvää kohinaa. CLAHE eli *Contrast Limited Adaptive Histogram Equalization* on AHE muunnos, jonka tarkoitus on vähentää AHEn ominaisuutta ylikorostaa kohinaa. CLAHE toimii muuten samanlailla kuin AHE, ainoana erona on, että se rajoittaa pikseleiden kontrastia viereisten pikseleiden kirkkauden perusteella, jolloin kuvan kohina ei ylikorostu. [Rubini and Pavithra 2019]

6 Lonkkanivelobjektin tunnistus

Lonkkanivelröntgenkuvissa on paljon tuloksen kannalta irrelevanttia aluetta. Jotta automaattinen lonkkanivellausuja saavuttaisi parhaan mahdollisen suorituskyvyn, lonkkanivelkuvat tulisi rajata siten, että ne sisältäisivät mahdollisimman vähän turhaa aluetta, ja täten pakotetaan malli tarkastelemaan lonkan laadun kannalta keskeisiä alueita. Tämän lisäksi yhdessä kuvassa on kahdet lonkat ja nämä tulisi erottaa toisistaan, jotta voidaan antaa molemmille itsenäiset tulokset.

Tehtävän suorittamiseksi loin objektintunnistumallin.

6.1 Data

Tällainen objektintunnistusalgoritmi käyttää oppimiseen kuvadataa ja kuviin liitettyjä annotointitiedostoja. Annotointitiedostosta käy ilmi kuvasta haettavien objektien sijainnit ja niiden luokat.

Annotointitiedostoja voi olla useaa eri muotoa. Itse valitsin XML-tiedostomuodon sen ollessa minulle ennestään tuttu. Nämä XML-tiedostot ovat muodoltaan seuraavanlaisia:

```
<annotation>
  <size>
    <width>3512</width>
    <height>4224</height>
    <depth>1</depth>
  </size>
  <object>
    <name>Lonkka</name>
    <bndbox>
      <xmin>996</xmin>
      <ymin>1605</ymin>
      <xmax>1496</xmax>
      <ymax>2118</ymax>
    </bndbox>
  </object>
  <object>
    <name>Lonkka</name>
    <bndbox>
      <xmin>2009</xmin>
      <ymin>1612</ymin>
      <xmax>2584</xmax>
      <ymax>2080</ymax>
    </bndbox>
  </object>
</annotation>
```

Kuva 13. Esimerkki XML-muotoisesta annotointi tiedostosta.

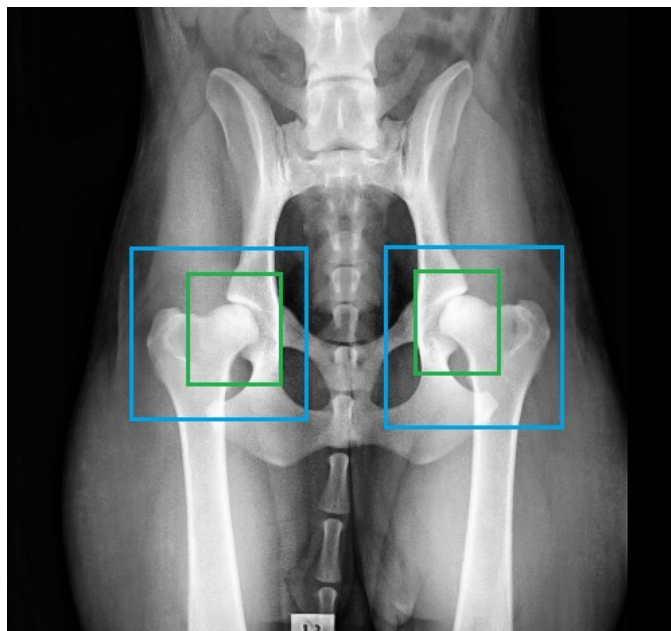
Kuvassa 13 elementti <size> pitää sisällään tiedot kuvan leveydestä, korkeudesta ja syvyydestä, eli onko kuva värillinen (3) vai mustavalkoinen (1). Elementti <object> kertoo objektin luokan, joka tässä tapauksessa on aina vakio ”Lonkka” sekä kyseisen objektin rajalaatikko elementin <bndbox>. Rajalaatikkoelementin alaelementit <xmin>, <ymin>, <xmax>, <ymax> ovat

kuvakoordinaattiarvoja, jotka kertovat kyseisen rajalaatikon vasemman yläkulman (x_{max} , y_{max}) ja oikean alakulman (x_{min} , y_{min}). Kun rajalaatikon vasen ylä ja oikea alakulma on tiedossa, voidaan niiden avulla luoda suorakulman muotoinen rajalaatikko.

Alkuun päästäkseni tuli minun luoda manuaalisesti dataa mallin opetukseen eli valita lonkkakuvia ja luoda niille annotointitiedostot. Tehtävän suorittamiseen hyödynsin LabelImg-työkalua [LabelImg 2022], jonka avulla annotointitiedostojen luomien on kohtalaisen nopeaa ja suoraviivaista. Myöhemmin lisäsin opetusdatan määrää syöttämällä lonkkanivelobjektin tunnistusalgoritmille uusia lonkkakuvia, joille järjestelmä automaattisesti loi annotointitiedostot. Tarkastettuani uusien annotointitiedostojen paikkansapitävyyden ja korjattuani niissä olleet virheet siirsin tapaukset opetusdataan.

6.2 Rajalaatikon oikeellisuus

Järjestelmän tavoitteena oli siis luoda rajalaatikko kuvassa olevien lonkkanivelten ympärille. Tämän rajalaatikon pitäisi pitää sisällään kaikki relevantti tieto lopullisen lonkkalausunnon antamiseksi. Näin jälkeempäin en ole varma tavoitteen toteutumisesta. Kukaan lausuntoja antava henkilö ei ole kommentoinut luomiani rajalaatikoita. On syytä epäillä, etteivät luomani rajalaatikat ole välttämättä optimaaliset. Tämä ilmenee, kun vertaa luomiani rajalaatikoita vastaavassa tutkimuksessa [McEvoy *et al.* 2021] käytettyihin rajalaatikoihin. Heidän kokeneet eläinlääkäriinsä loivat huomattavasti laajempia rajalaatikoita kuin minä kuvassa 14. Mikäli käy ilmi, että luomani rajalaatikat ovat liian pieniä, olisi järkevää opettaa järjestelmäni uudelleen parempien rajalaatikoiden kanssa. Huonot rajalaatikat todennäköisesti vaikuttaisivat negatiivisesti itse lonkkanivellausujan suorituskykyyn.



Kuva 14. Vihreät minun tekemäni rajalaatikat, sininen arvioni [McEvoy *et al.* 2021] tutkimuksessa käytetyistä rajalaatikoista.

6.3 Prosessi

Ensimmäinen vaihe prosessissa oli valita, mitä objektin tunnistusalgoritmia kannattaisi käyttää tässä tehtävässä. Selvittelyn jälkeen järkeviä vaihtoehtoja jäi kaksi: Mask R-CNN tai jokin YOLO-perheeseen kuuluva metodi. Aluksi käännyn YOLOv3:n puoleen, sillä se vaikutti olevan tehokkaampi kuin Mask R-CNN. En kuitenkaan saanut YOLOv3:ea toimimaan oikein, jonka vuoksi päädyin kokeilemaan Mask R-CNN -algoritmia. Mask R-CNN:n implementointi ei myöskään ollut erityisin suoraviivainen prosessi. Eniten hankaluutta aiheutti metodin riippuvuudet kirjastojen vanhoihin versioihin, jonka vuoksi jouduin tekemään useita versioiden takaisinkierroja.

Sain kuitenkin metodin loppujen lopuksi toimimaan. Seuraava vaihe oli datan valmistelu Mask R-CNN -mallin opetukseen. Alkuun valmistelin 200 kuvan aineiston, jossa oli yhteensä 400 rajalaatikkoa lonkkanivelille. Uskoin 200 kuvan aineiston olevan riittävä ensimmäisten testien tekemiseen, sillä tarkoitukseni oli hyödyntää siirto-oppia ja alustaa malli Mask R-CNN COCO-painoilla.

Opetin mallia alkuun viisi epookkia, jonka jälkeen testasin sen suorituskykyä syöttämällä testiksi 200 uutta kuvaa, joita malli ei ole aikaisemmin ”nähty”. Nämä tulokset olivat lupaavia ja malli loi suoraan oikeat rajalaatikot suurimpiin osiin kuvista. Samalla kuitenkin ilmeni muutamia ongelmia.

6.3.1 Ensimmäiset ongelmat

Ensimmäiseksi mallilla oli taipumus ”löytää” sama lonkkanivel moneen kertaan tehden yhdelle lonkkanivelelle useita osittain päällekkäisiä rajalaatikoita. Tutkiessani ongelmaa huomasin, että parhaalla lonkkanivelrajalaatikolla oli aina korkein todennäköisyys verrattuna muihin päällekkäisiin rajalaatikoihin. Ongelman korjaamiseksi muokkasin järjestelmään siten, että mikäli havaitaan toisiaan leikkaavia rajalaatikoita, tällöin jäljelle jätetään ainoastaan rajalaatikko, jolla on suurin todennäköisyys. Myöhemmin kuitenkin löysin Mask R-CNN-konfiguraatiosta asetuksen *DETECTION_MAX_INSTANCES*, jolla pystyy rajaamaan enimmäistunnistusten määrää. Asettamalla asetukselle arvoksi 2 rajoittaa mallia palauttamaan enintään 2 rajalaatikkoa korjaten alkuperäisen ongelman ja tehden muista muutoksista turhia.

Mallilla oli myös ongelmia tunnistaa huonoja D- ja E-lonkkaniveliä. Tämä oli mielenkiintoinen havainto, sillä se kertoo, että luokkien välillä on selkeitä havaittavia eroja, mikä antaa toivoa itse automaattiselle lonkkalausujalle. Ongelmia oli helppo korjata. Ensimmäisessä 200 kuvan opetusaineistossa ei ollut juurikaan näitä D- ja E-kuvia. Manuaalinen D- ja E-kuvien lisääminen korjasi ongelman.

Opetusaineistossa jokainen kuva oli suunnattu samansuuntaisesti. Testiaineistossa kuitenkin oli muutamia kuvia, jotka ovat ylösalaisin opetusaineiston kuviin verrattuna. Näistä eri tavoin suunnatuista kuvista malli ei kyennyt löytämään lonkkaniveliä erityisen hyvin. Lisäksi haluan muutenkin kaikkien kuvien olevan samoin päin suunnattu. Ongelman korjaamiseksi lisäsin järjestelmään seuraavan ominaisuuden. Mikäli malli ei löytänyt kahta lonkkaniveltä kuvasta, tällöin kuvaa käännetään 90 astetta, jonka jälkeen malli koittaa etsiä lonkkia uudestaan nyt käännetystä kuvasta. Kuvan kääntämistä ja

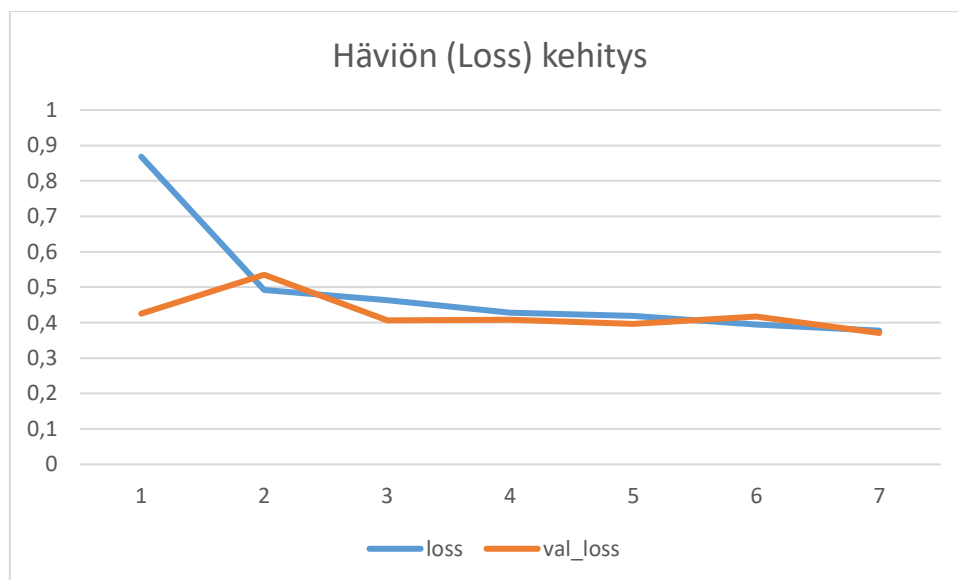
lonkkien uudelleen etsimistä toistetaan, kunnes malli löytää kahdet lonkat tai kuvaa on pyöritetty 270 astetta mitään löytämättä. Tästä operaatiosta lisäksi annotointitiedostoon elementin, joka kertoo, montako kertaa kuvaa on käännetty, jotta on löydetty kahdet lonkat. Tämä muutos vaikutti riittävältä ongelman ratkaisemiseksi. Lisää ongelmia kuitenkin ilmeni myöhemmissä testeissä.

6.3.2 Opetusaineiston kasvatus

Ensimmäisten ongelmien korjausten jälkeen vuorossa oli aika kasvattaa opetusaineiston kokoa. Korjatessani ongelmia lisäsin aineistoon yhteensä 100 kappaletta D- ja E-lonkkakuvia, joka nostivat opetusaineiston koon 300 kuvaan ja 600 lonkkaniveleen. Muuten kasvatin opetusaineiston kokoa syöttämällä mallille uusia testikuvia ja tarkastamalla mallin niille antamat rajalaatikat, ja mikäli rajalaatikko vaikutti hyvältä, lisäsin kuvan opetusaineistoon. Jos rajalaatikko oli huono tai kuvalle ei löydetty ollenkaan rajalaatikoita, korjasin ne ja lisäsin ne manuaalisesti. Manuaalista korjausta tarvitsi vain hyvin pieni määrä kuvia. Myöhemmin toistin tätä prosessia kuvien määrän lisäämiseen, jolloin lopullisen opetusaineiston koko oli yhteensä 1450 kuvaa.

6.3.3 Laajat testit

Lonkkanivelen hahmontunnistusalgoritmin tarkoitus tässä tutkimuksessa on siis tuottaa syötedataa lonkkanivelen automaattiselle lonkkalausuntojärjestelmälle. Kun lopullinen opetusaineisto lonkkanivelhahmontunnistuksen opetukseen oli koossa, oli aika luoda lopullinen malli. Tämän opetusdatana datana käytin 1216 ja validointidatana 243 rajalaatikoilla varustettua kuvaa.



Kuva 15. Opetus- ja validointihäviön kehitys.

Malli oppi nopeasti ja häviön kehitys vaimeni nopeasti ensimmäisen epookin jälkeen kuvan 15 mukaisesti. Opetin mallia seitsemän epookin ajan, joka kesti yhteensä noin 24 tuntia.

Kun lopullinen malli oli valmistunut, oli aika aloittaa mallin testaaminen ja syötedatan luominen automaattiselle lonkkanivellausujalle. Tämä tapahtui syöttämällä mallille kuvia, joita se ei ole aikaisemmin ”nähty”, jolloin mallin tehtävä on luoda rajalaatikot molemmille kuvassa oleville lonkkanivelille, joiden tiedot tallennetaan annotointitiedostoon. Näiden annotointitiedostojen avulla voidaan leikata alkuperäisestä lonkkakuvasta rajalaatikoiden merkkejä omiksi kuvikseen.

Ensimmäinen laaja testi piti sisällään noin 26 000 kuvaa eli noin 52 000 lonkkaniveltä. Tämän testin yhteydessä ilmeni viimeiset ongelmat. Ongelmana oli väärät positiiviset rajalaatikot kuvissa. Näitä oli kahdenlaisia. Ensimmäisenä olivat tapaukset, joissa malli löysi lonkkaniveliä kuvista, jotka eivät edes olleet lonkkakuvia. Toisena olivat lonkkanivelkuvat, joissa lonkkanivel tunnistettiin väärään kohtaan, tai lonkkanivel oli väärin suunnattu, usein ylösalaisin. Tutkiessani näitä vääriä positiivisia rajalaatikoita huomasin, että mallin antama todennäköisyys sille, että nämä rajalaatikot pitävät sisällään lonkkanivelen oli kaikissa selkeästi alle 90 % samalla, kun oikeiden positiivisten rajalaatikoiden todennäköisyys oli lähes aina yli 95 %. Päästäkseni eroon vääristä positiivisista rajalaatikoista, lisäsin seuraavan ehdon: Mikäli malli antaa rajalaatikon, jonka todennäköisyys on alle 90 %, tällöin voidaan olettaa rajalaatikon olevan väärä positiivinen, joka johtuu siitä, että kuva on väärin suunnattu tai se ei ole lonkkanivelkuva. Tällöin kuvaa käännetään 90 astetta, kunnes malli palauttaa kuvalle kaksi rajalaatikkoa, joiden molempien todennäköisyys on yli 90 % tai kuvaan on käännetty yhteensä 270 astetta eikä ehtoja täyttäviä rajalaatikoita ole syntynyt, jolloin kuva hylätään.

6.4 Tulokset

Kun lopulliset korjaukset on tehty, on aika tehdä viimeiset testit mallin suorituskyvyn evaluoimiseksi. Evaluointiin käytin seuraavia kolmea testiä:

1. Testataan mallin kyky kääntää kuva oikeaan suuntaan.
2. Testataan mallin kyky hylätä kuvat, jotka eivät ole lonkkanivelkuvia, kuten selkä ja kyynär.
3. Testata, kuinka malli toimii todellisen laajan aineiston kanssa.

6.4.1 Testi 1. Väärin suunnatut kuvat

Loin 216 lonkkakuvan aineiston, jossa jokainen kuva on suunnattu väärin eli alkuperäistä kuvaa käännetty 90, 180 tai 270 astetta. Tämän jälkeen syötin aineiston mallille ja tarkastin, onnistuuko malli kääntämään ja rajaamaan kaikki lonkat oikein.

Malli ei onnistunut luomaan rajalaatikoita kahdelle kuvalle, koska näissä todennäköisyydet jäivät alle 90 %. Kaikki muut 214 kuvaa käännettiin oikein ja malli palautti oikeat rajalaatikot. Parin kuvan hylkääminen ei ole erityisen vakavaa, tärkeämpää on, ettei syntynyt vääriä positiivisia.

6.4.2 Testi 2. Väärät tutkimukset

Loin 194 selkä- ja kyynärkuvan aineiston, jonka syötin mallille selvittääkseni kykeneekö se hylkäämään kaikki kuvat. Jokainen kuva hylättiin, kuten pitikin. Malli kuitenkin palautti kolmesta kuvasta yhden rajalaatikot, jotka ylittivät tuon 90 % todennäköisyyden.

Tämä testi kertoo myös mallin kyvystä tunnistaa lonkkakuva muista kuvista, minkä avulla voidaan seuloa väärin merkattuja kuvia lonkkakuva aineistosta.

6.4.3 Testi 3. Toiminta oikean aineiston kanssa

Tässä testissä oli kaikkiaan 26 095 kuvaa. Tätä aineistoa ei ole ennestään seulottu virheiden varalta, kuten väärin suunnatut kuvat ja ei lonkkakuvat.

25 878 kuvalle malli tunnistoi kahdet rajalaatikot ilman kuvan kääntöä, joiden todennäköisyys olla lonkkanivel on yli 90 %. Kuvan käännön jälkeen 135 kuvalle tunnistettiin rajalaatikot, joiden todennäköisyys olla lonkkanivel on yli 90 %. Jäljelle jäi 82 kuvaa, joista malli ei löytänyt lonkkaniveliä vaaditulla 90 % tarkkuudella. Eli hyväksytyjä kuvia, joissa kahdet yli 90 % tarkkuudella varustetut rajalaatikot oli yhteensä 26 013/26 095 eli noin 99.7 % ja hylättyjä kuvia 82/26 095 eli noin 0.3 %.

Kun selasin hyväksytyjä kuvia virheellisten rajalaatikoiden varalta, en havainnut ensimmäistäkään virhettä. Tämä ei tarkoita, etteikö niitä voisi olla ollenkaan olemassa, mutta voin kuitenkin varmuudella todeta oikeiden positiivisten rajalaatikoiden määrä olevan yli 99.5 %. Hylättyjä kuvia oli vähän, jolloin ne olivat nopea manuaalisesti tarkastaa. Näistä 82 kuvasta 44 hylättiin oikein, koska ne eivät olleet lonkkanivelkuvia. Jäljelle jäi 38 lonkkanivelkuvaa, joille syystä tai toisesta ei saatu luotua rajalaatikoita, joiden todennäköisyydet ovat yli 90 %. Useille kuville tässä joukossa löytyi melko selkeät syyt siihen, miksi malli ei kyennyt luomaan hyväksyttäviä rajalaatikoita. Tyypillinen syy oli se, että lonkkanivel oli osittain peittynyt muun muassa puolimerkin alle kuten kuvassa 16. Kokonaisuudessa tässä joukossa oli melko vähän lonkkanivelkuvia, joille ei ole selkeää syytä sille, miksi malli on hylännyt ne.



Kuva 16. Hylättyjä kuvia, joissa lonkkanivel on peittynyt tunniste- ja puolimerkin alle.

6.5 Pohdinta

Malli toimii erittäin hyvin. Tavoite minimoida virheelliset rajalaatikot onnistui erittäin hyvin, kun oikeita positiivisia rajalaatikoita on ainakin 99.5%. Uskon tämän osuuden olevan vieläkin korkeampi, koska olen selannut yli 10 000 kuvaa havaitsematta yhtäkään virheellistä rajalaatikkoa.

Lisäksi vielä väärin negatiivisten tapausten määrä oli hyvin pieni, jolloin ne olisi voinut vaikkapa manuaalisesti korjata. Tätä en kuitenkaan tehnyt, kun käytössä on kymmeniätuhansia kuvia, näin pienillä määrällä kuvia ei ole merkitystä. Tämä väärin negatiivisten tapausten vähyys yllätti ja olin varautunut huomattavasti isompiin määriin.

Valittu Mask R-CNN ei näin jälkeen päin ajatellen varmastikaan ollut paras vaihtoehto järjestelmän objektintunnistusmallille. Suurin ongelma oli sen hitaus, koska Mask R-CNN ei tue uusia RTX 30X0-grafiikkakortteja, jolloin malli käyttää ainoastaan prosessoria. Opetusdatan koon ollessa 1216 kuvaa kukin epookki kesti 3½ tuntia. Epookkien ja opetusdatan määrän kasvattaminen olisi nopeasti johtanut tilanteeseen, jossa mallin opetus olisi kestänyt useita päiviä. Olisi ollut mielenkiintoista selvittää, miten malli olisi oppinut, jos olisi opettanut sitä useamman epookin ja suuremman aineiston kanssa; olisiko tällöin viimeisetkin virheet karsiutuneet pois?

Rajalaatikoiden oikeellisuus kuitenkin jäi askarruttamaan. Nykytiedon valossa rajaisin lonkkanivelet laajemmin. Lopullinen automaattinen lausunnon antaja saattaisi toimia paremmin laajempien rajalaatikoiden kanssa. Tämä kannattaisi tutkia myöhemmin.

7 Automaattinen lonkkalausuja

Automaattisen lonkkalausuntojärjestelmän tavoite on tunnistaa koiran lonkkanivelen laatu FCI:n määrittämien ohjeiden mukaisesti. Mikäli automaattinen lonkkanivellausuja saavuttaa korkean suorituskyvyn lonkkanivelten laadun tunnistamisessa, olisi sillä potentiaalista kliiniseen käyttöön.

7.1 Data

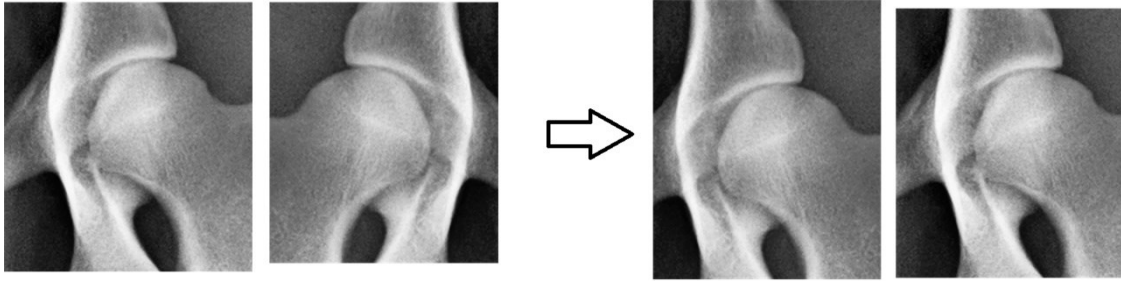
Automaattinen lonkkalausuja käyttää datana edellisen kappaleen lonkkanivelobjektin tunnistusjärjestelmän tuottamia rajalaatikoiden rajaamia kuvia. Datan määrä on kasvanut prosessin aikana useaan otteeseen ja lopullinen aineiston koko on 106 162 rajattua lonkkanivelkuvaa. Näistä 92 750 kuvaa käytetään opetus/validointidatana ja loppua 13 412 kuvaa testidatana. Opetus/validointidata koostuu röntgenkuvista vuosilta 2013–2020 ja testidata vuoden 2021 kuvista. Kuvien rodut on rajattu 20 yleisempään lonkkanivelkuvattuun rotuun, jotta prosessointiajat pysyisivät kohtuullisina.

Käytössä olevan datan luokkajakauma on vino, kuten taulukosta 9 voi havaita. Syväoppimisen mallit on luotu toimimaan etenkin tasapainoisen luokkajakauman kanssa, jolloin niiden suorituskyky tippuu vinon luokkajakauman kanssa. Todellisuudessa datan luokkajakauma on harvoin tasapainoista, jonka vuoksi on luotu useita erilaisia menetelmiä, joilla voidaan kamppailla vinon luokkajakauman aiheuttamia ongelmia vastaan. [Johnson and Khoshgoftaar 2019]

Data	A	B	C	D	E	Yht.
Opetus/validointi	45546	27584	14194	4552	874	92750
Testi	6630	3672	2452	506	152	13412

Taulukko 9. Opetus/validointi- ja testidatan määrä kussakin luokassa.

Prosessin alussa kaikki kuvat olivat alkuperäisessä asennossaan. Tämä muutettiin myöhemmin siten, että kaikki kuvat ovat käännetty samaan suuntaa, eli käytännössä toinen lonkkanivel on peilattu kuten kuvassa 17. Tämä tapahtui etsimällä kuvan lonkkanivel-rajalaatikoiden pienin x-arvo, jonka perusteella voidaan tunnistaa kuvan (ei koiran) vasen ja oikea lonkkanivel. Kun kuvan oikea lonkkanivel on tunnistettu, voidaan tälle suorittaa horisontaalinen kääntö, jolloin vasen ja oikea lonkkanivel osoittavat samaan suuntaan. Muutoksen avulla kuvista tuli yhdenmukaisimpia, joka paransi järjestelmän suorituskykyä merkittävästi.



Kuva 17. Lonkkanivelkuvien kääntö samaan suuntaan.

Lisäksi konvoluutioneuroverkko vaatii kaikkien syötekuvien olevan samankokoisia. Alkuperäisten leikattujen lonkkanivelkuvien resoluutio vaihteli paljon. Pienemmillään resoluutio oli 120×120 ja suurimmillaan yli 500×500 . Suurin osa kuitenkin oli 180×180 ja 250×250 välillä. Tämän vuoksi valitsin 200×200 olevan sopiva lopullinen resoluutio.

7.2 Luokittelu vai regressio?

Aineiston ollessa käyttövalmis seuraavaksi on vuorossa mallien luonti. Kysymyksenä kuitenkin oli, onko tämä tehtävä lähempänä regressiota vai luokittelua? Intuitiivisesti tehtävähän on selkeästi luokittelutehtävä, kun tavoitteena on ennustaa lonkkanivelten laatua viiteen eri luokkaa [A, B, C, D tai E]. Luokat ovat tässä tehtävässä ordinaalisia, jonka vuoksi luokittelu ei välttämättä ole mielekästä. Potentiaalinen ongelma syntyy siitä, ettei luokittelija välttämättä ota huomioon luokkien ordinaalisuutta. Tällöin voisi syntyä seuraava tilanne: Luokittelija antaa lonkalle vaikkapa 49 % todennäköisyyden kuuluvan A luokkaan ja 51 % todennäköisyys kuulua D luokkaan. Tässä tehtävässä tällainen tulos ei ole järkevä, sillä luokat A ja D ovat kaukana toisistaan eikä niiden välillä ole rajatapausta. Mikäli luokat olisivat A ja B, tällöin tulos olisi järkevä, koska tällöin on mahdollista, että kyseessä on rajatapaus. Regression etuna on, ettei edellinen tapaus olisi mahdollinen. Tehtävästä voidaan luoda regressiotehtävä muuntamalla luokkien diskreetit arvot jatkuviksi. Täten alkuun valitsin regression, ja mielivaltaisesti päätin asteikoksi [A = 5, B = 15, C = 25, D = 35, E = 45].

Käyttämieni konvoluutioneuroverkkomallien eräänä etuna on niiden helppous muuttaa regressiotehtävä luokittelutehtäväksi. Myöhemmin päätinkin testata, kuinka malli toimii, mikäli tehtävä muutetaan luokitteluksi. Pieneksi yllätyksekseni luokittelija vaikutti toimivan hyvin, ja luokittelija oppi datasta sen ordinaalisuuden. Tämän vuoksi päätin jatkaa järjestelmän kehitystä luokittelutehtävänä.

7.3 Mallit ja arkkitehtuuri

Tutkimus keskittyi syväoppimisen malleihin, etenkin konvoluutioneuroverkkoihin. Konvoluutioneuroverkko valikoitui sen hyvästä suorituskyvystä kuvadatan kanssa, joka on tämän tutkimuksen keskiössä. Kokeilin myös, kuinka tukivektorikone suoriutuu tehtävästä.

7.3.1 Mallien evaluointi

Koska aineiston luokkajakauma on vino, perinteinen tarkkuus ei ole mielekäs metriikka mallien suorituskyvyn evaluointiin. Tämän vuoksi käytän mallien evaluointiin F1-lukua, joka toimii hyvin vinon luokkajakauman kanssa.

7.3.2 Konvoluutioneuroverkon arkkitehtuuri

Mallin arkkitehtuurilla on erittäin suuri vaikutus mallin suorituskyvyn. Tämän vuoksi tutkimuksessa kului paljon aikaa erilaisten verkkoarkkitehtuurien kokeiluun, jotta löytäisin parhaiten toimivat tähän tehtävään.

Päästäkseni alkuun ja saadakseni ensimmäiset tulokset, joita lähteä vertailemaan, loin melko yksinkertaisen konvoluutioneuroverkon. Ensimmäisen mallin arkkitehtuuri oli rakennettu mielivaltaisesti ottamalla inspiraatiota VGG16-mallin arkkitehtuurista. Tarkoituksena tosiaan ei ollut luoda erityisen hyvää mallia, mutta pikemminkin ensimmäinen, jonka tulosta lähteä vertailemaan ja parantamaan.

7.4 Ylisovittautuminen

Ylisovittautuminen on sitkeä ongelma. Lähes jokainen kokeilemani malli kärsi ylisovittautumisesta enemmän tai vähemmän. Ylisovittautumisen ehkäisyksi ja vähentämiseksi kokeilin useita erilaisia menetelmiä. Ensimmäinen yritykseni oli yksinkertaisesti lisätä datan määrää, mutta tällä itsessään oli yllättävän pieni vaikutus ylisovittautumiseen. Kuitenkin kokeiluiden jälkeen jäljelle jäivät kolme erityyppistä menetelmää:

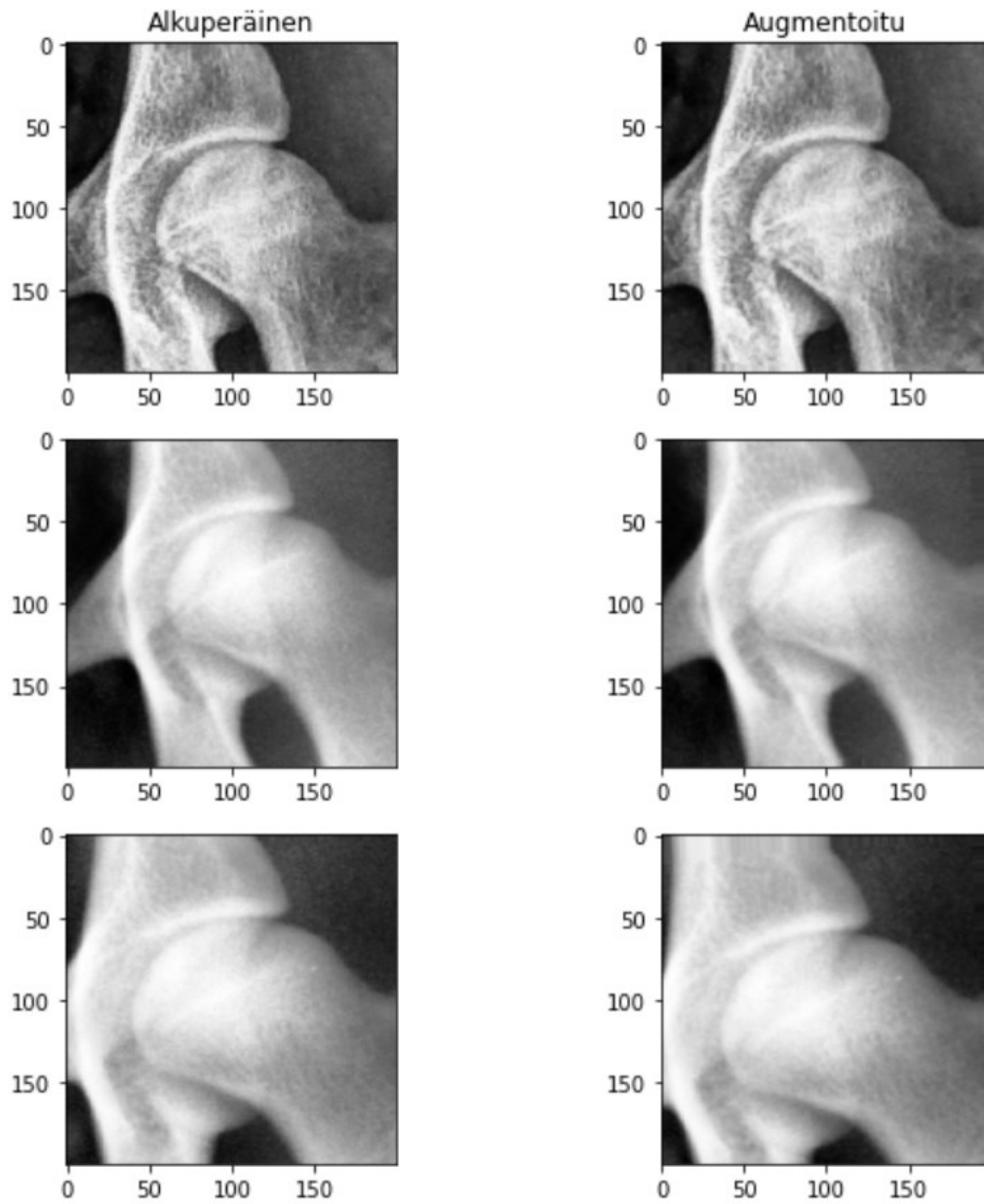
1. Varhainen lopetus
2. Regularisaatio
3. Datan keinotekoinen lisäys

Varhainen lopetus seuraa validoinnin F1-luvun kehitystä kuuden epookin ajan. Kun F1-luku ei ole parantunut kuuteen epookkiin, palautetaan parhaan epookin painot. Tämä on tehokas ja varma tapa varmistaa, ettei lopullinen malli kärsi ylisovittautumisesta.

Käyttöön jäi useita regularisaatiomenetelmiä. Näitä ovat muun muassa L2-regularisaatio arvolla 0.01, pudotuskerrokset aktivointitodennäköisyydellä 0.3–0.5 ja erän normalisointikerrokset.

Datan keinotekoiseen lisäykseen (data augmentation) jäi seuraavat menetelmät, pieni satunnainen 0–5 asteen kääntö oikeaan tai vasempaan suuntaan, kuvan liikuttaminen x-akselilla 0–10 pikseliä vasempaan tai oikeaan suuntaan, kuvan y-akselilla liikuttaminen 0–15 pikseliä ylös tai alas sekä

Gaussian kohinan lisääminen. Esimerkki datan keinotekoisen lisäyksen tekemistä muunnoksista on kuvassa 18.



Kuva 18. Alkuperäinen kuva ja kuva datan keinotekoisen lisäämisen suorittamien muunnosten jälkeen.

7.5 Tulokset

Malli	Testi F1	Aika s/epookki
Baseline	0.677	345
AlexNet	0.716	393
ResNet50V2	0.73	447
InceptionV3	0.736	377
VGG16	0.722	467
EfficientNetB2	0.746	903
EfficientNetB4	0.743	1370
ResNet50 + SVM	0.64	~

Taulukko 10. Mallin F1-luku ja yhteen epookkiin kulunut aika sekunteina.

Taulukon 10 tulokset on saatu opettamalla kutakin mallia k-ristiinvalidoimalla k:n arvolla 10. Ajan säästämisen vuoksi kaikkia malleja ei ole ajettu 10 kierrosta. Sen sijaan osa malleista on keskeytetty manuaalisesti aikaisintaan viiden kierroksen jälkeen, mikäli kierrosten validointitulokset eivät eroa toisistaan merkittävästi. Taulukon 10 testitulokset F1 on saatu valitsemalla kunkin mallin parhaan kierroksen painot, joita käyttämällä on luokiteltu testidatajoukosta.

Kaikki kokeilemani mallit ovat konvoluutioneuroverkkoja erilaisilla arkkitehtuureilla mukaan lukematta vaihtoehtoa ResNet50 + SVM. Tässä käytin ResNet50 konvoluutioneuroverkkoarkkitehtuuria piirteiden erotukseen ja tukivektorikonetta luokitteluun näiden piirteiden perusteella.

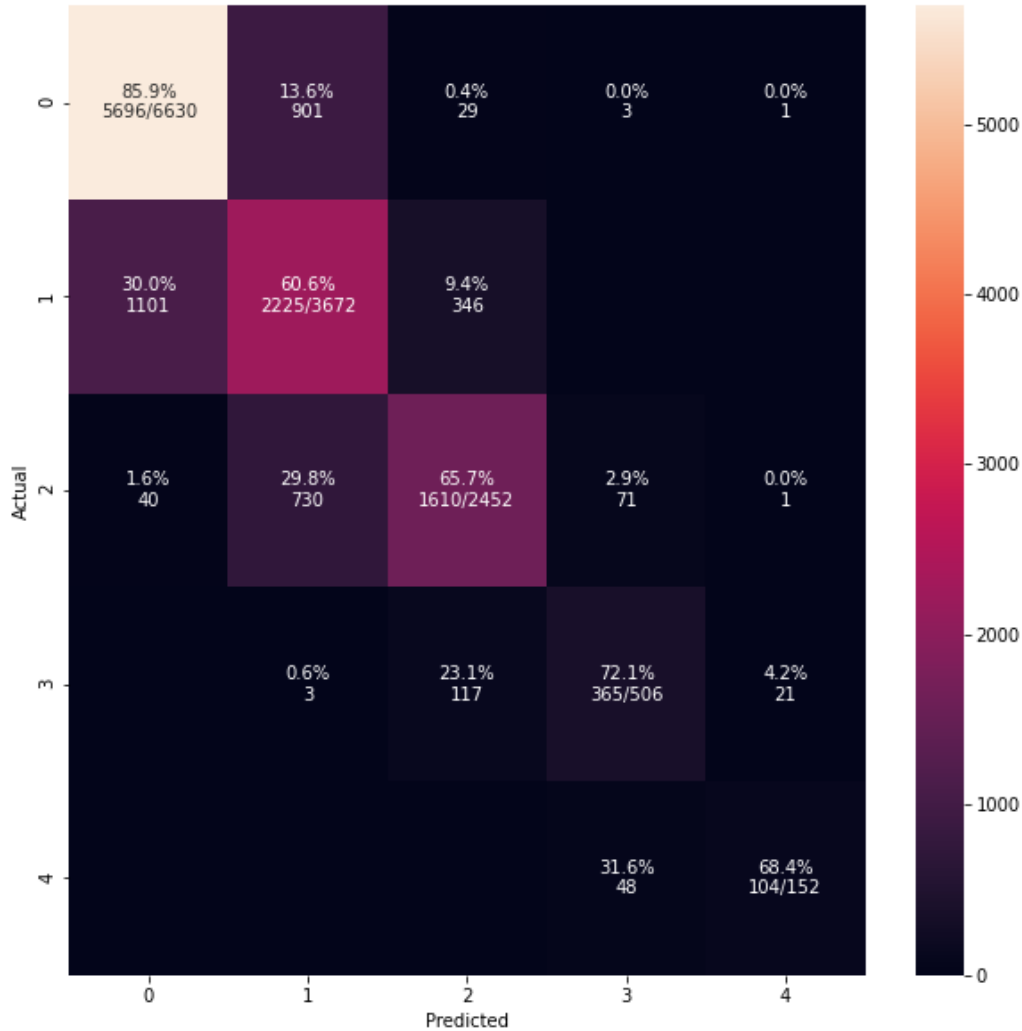
Taulukon 10 aika s/epookki ei ole täydellinen metriikka kuvaamaan mallin käyttämää kokonaisaikaa, sillä osassa malleista varhainen lopetus keskeytti opetuksen jo 10–20 epookin kohdalla, kun toisissa vasta 30–40 epookin välissä. Ei voida sanoa, että EfficientNetB2 käytti kaksi kertaa enemmän aikaa kuin ResNet50V2. Opetusaikoja olisi voinut merkittävästi pienentää, mikäli käytössä olisi ollut enemmän videomuistia (VRAM). Tämän vuoksi osassa malleista pystyi käyttämään enintään 12 kuvan erää kerrallaan. Lisäksi käytin kaikissa malleissa vakio-oppimisnopeutta 0.001 (learning rate).

EfficientNetB2 ja -B4 saavuttivat testeissä melko selkeästi korkeimmat F1-luvut muihin malleihin verrattuna, jonka vuoksi niiden voidaan todeta olevan parhaat mallit tehtävään. Ei kuitenkaan voida sanoa kumpi versio B2 vai B4 EfficientNetistä on parempi, koska niiden F1-luvut ovat hyvin lähellä toisiaan.

7.5.1 Luokkakohtaiset tarkkuudet

Mallin suorituskyvyn luokkakohtainen tarkastelu antaa tärkeää tietoa mallin toiminnasta. Sen avulla voidaan havaita, onko jokin luokka vaikeampi kuin muut luokitella. Lisäksi kun tässä tehtävässä luokiteltava luokka on ordinaalinen, on oleellista tarkastella virhetilanteessa, miten kauas ennustettu

luokka meni oikeasta luokasta. On huomattavasti suurempi virhe ennustaa A-lonkka D-lonkaksi kuin A-lonkka B-lonkaksi.

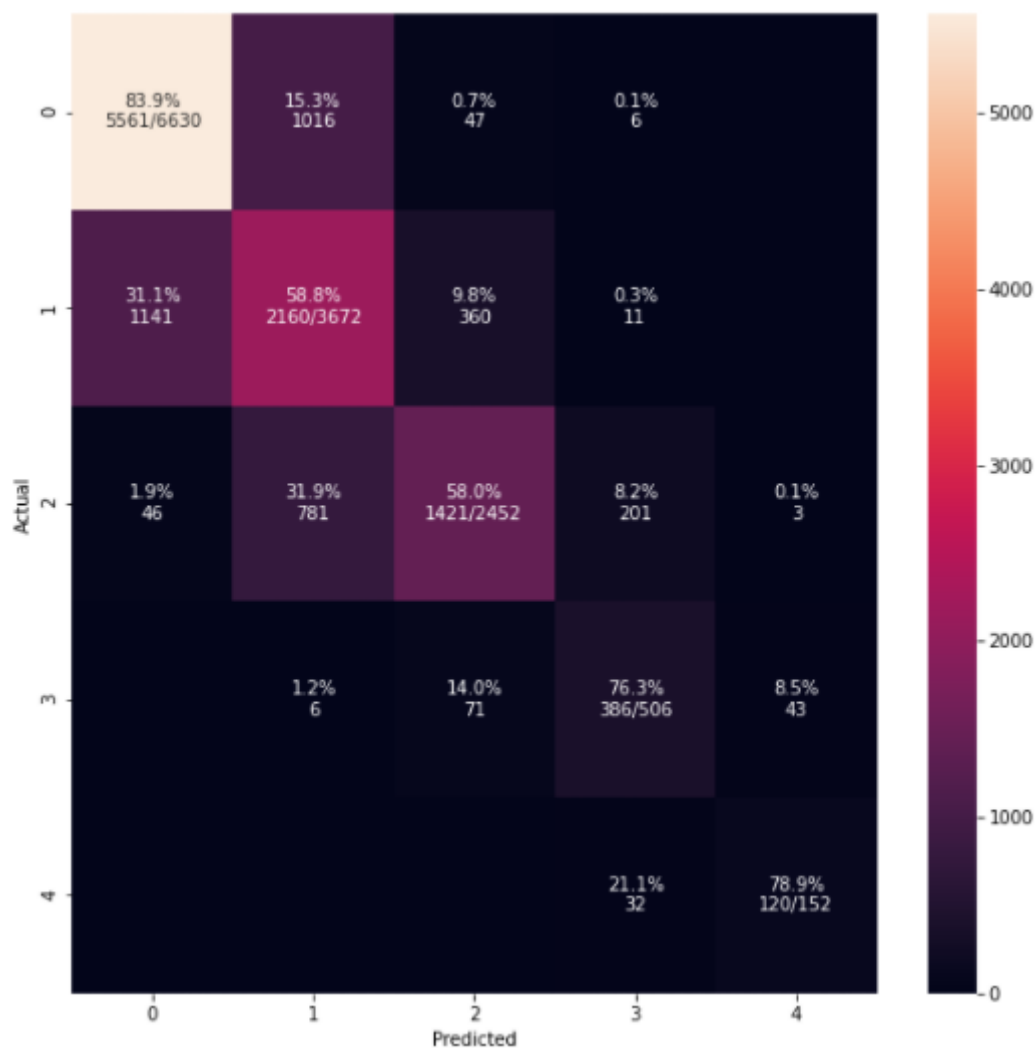


Taulukko 11. EffcientNetB2 sekaannusmatriisi. (0 = A, 1 = B, 2 = C, 3 = D ja 4 = E). Solun väri kertoo tapausten määrän solussa. Solu, jossa ei ole yhtään tapausta on musta ja solu, jossa eniten tapauksia on vaalea.

Taulukossa 11 on EffcientNetB2 luokittelijan testidatasta tuottama sekaannusmatriisi. Tarkastelemalla luokkakohtaisia tarkkuuksia voidaan havaita mallin toimivan selvästi parhaiten A-lonkkien kanssa ennustaen 85.9 % tapauksista suoraan oikein ja 13.6 % viereiseen B-luokkaan. Muiden luokkien kanssa on enemmän ongelmia. On mielenkiintoista havaita, että luokissa B, C, D, ja E mallilla on taipumus mieluummin ennustaa luokka yhden asteen todellista paremmaksi kuin asteen huonommaksi.

Kun tarkastellaan virhetilanteita taulukosta 11. Havaitaan, että malli kykenee luokittelemaan kaikki luokat erittäin tarkasti ± 1 tarkkuudella, eli ennustaa tapauksen kuuluvan joko oikeaan luokkaan tai yhteen naapuriluokista. Tällöin kaikki luokkakohtaiset tarkkuudet ovat 98 % ja 100 % välillä.

Erilaiset mallit tuottavat erilaisia sekaannusmatriiseja. Taulukon 11 sekaannusmatriisin malli on kokonaisuudessa paras, mutta ei välttämättä kaikissa luokissa.



Taulukko 12. EfficientNetB2, jossa käytettiin luokkainoita (class weights) luokkajakauman vinouden aiheuttamien ongelmien korjaamiseen.

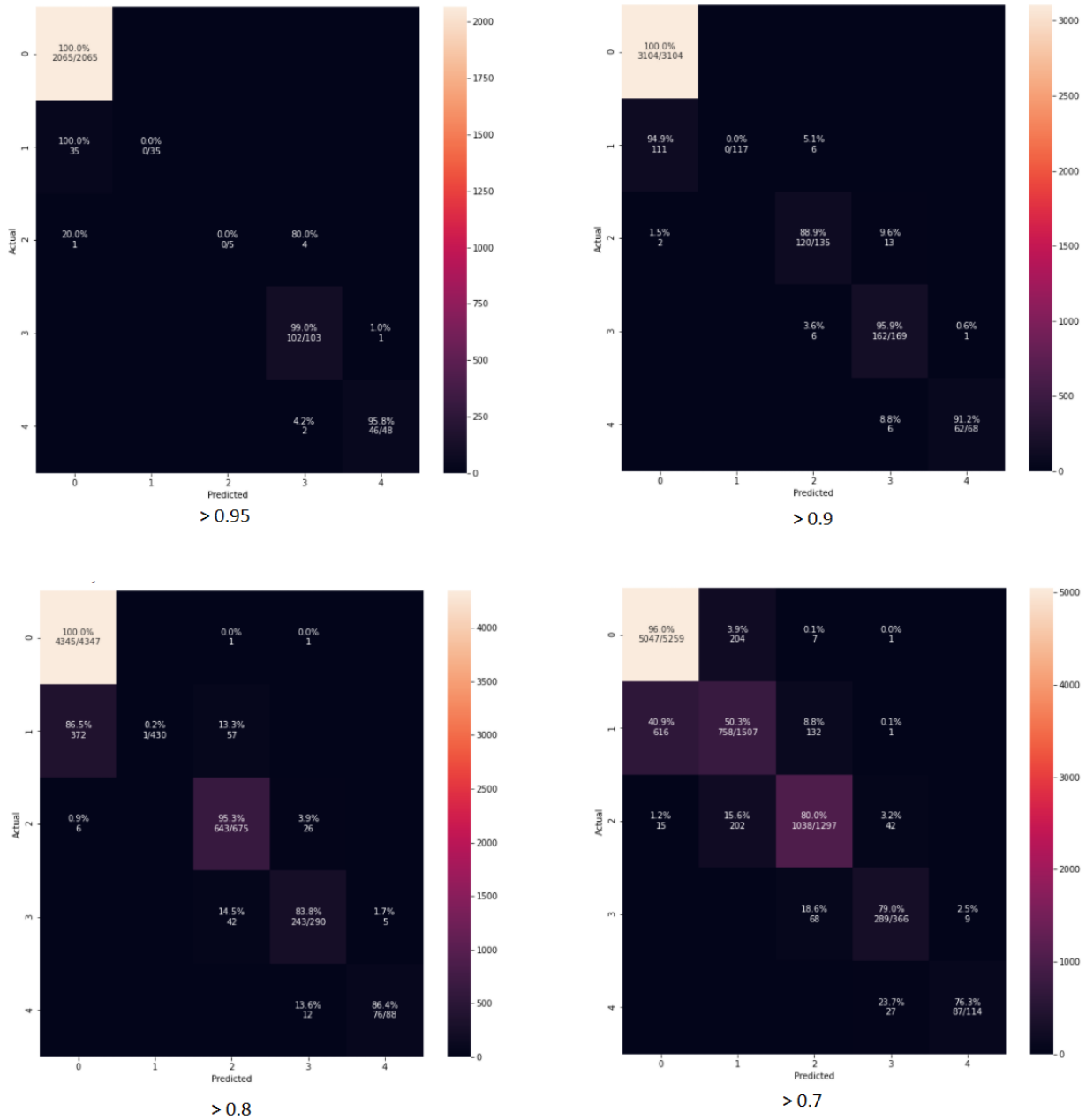
Esimerkiksi taulukon 12 mallin kyky luokitella E-luokkaa on huomattavasti parempi kuin taulukko 11 mallin. Taulukossa 11 malli luokittelee E-luokan 68.4 % tarkkuudella oikein, kun taulukon 12 malli 78.9 % tarkkuudella.

7.5.2 Ennustetun luokan todennäköisyys

Mallin kyky ilmaista varmuus ennustetuista luokista on hyödyllinen ominaisuus tässä tehtävässä. Koska tehtävässä on rajatapauksia, tulisi mallin kyötä ilmaisemaan nämä tapaukset. Ei voida olettaa mallin kykenevän luokitteluun kaikkia rajatapauksia oikein, kun ne vaikuttavat olevan usein vaikeita koulutetuille radiologeillekin.

Softmax-aktivointi mahdollistaa ennustuksien varmuuden tarkastelua. Tämän avulla malli antaa tuloksena, millä todennäköisyydellä tapaus kuuluu mihinkin luokkaan, esimerkiksi [A:0.758, B:0.239, C:0.002, D:0.0, E:0.0], joka tarkoittaa 75.8 % todennäköisyys, että tapaus kuuluu A-luokkaan, 23.9 % että B-luokkaan, 0.2 % että C-luokkaan ja 0 % C- ja D-luokkaan. Tässä esimerkissä malli palauttaisi tuloksena tapaukselle A-luokan. Asettamalla raja minimi-todennäköisyydelle, jolloin tapaus hyväksytään. Esimerkiksi todennäköisemmän luokan todennäköisyys pitää olla vähintään 90 % jotta tulos hyväksytään. Sen avulla voidaan tarkastella mallin suorituskykyä tapauksien kanssa, joissa malli on kohtalaisen varma ennustuksistaan.

Taulukosta 14 huomaa jälleen mallilla olevan paras suorituskyky A-luokan kanssa. Malli luokittelee kaikki A-lonkat 100 % tarkkuudella, kun minimi-todennäköisyys on yli 80 %. Tämä pitää sisällään 65 % aineiston A-lonkista. A-lonkkien luokittelutarkkuus pysyy korkealla 96 %, vaikka minimi-todennäköisyys lasketaan 70 %:iin. Tarkkuus A-luokan kanssa syntyy hieman B-luokan kustannuksella. Malli nimittäin ei kykene juuri lainkaan luokitteluun B-luokkaa. Malli luokittelee yli 80 % minimi-todennäköisyydellä B-luokan mieluummin A- tai C-luokaksi B-luokan sijasta. 3672 B-lonkasta ainoastaan yksi luokitellaan oikein B-luokaksi, 80 % minimi-todennäköisyydellä. B-luokkaa aletaan luokitteluun oikein vasta yli 70 % minimi-todennäköisyydellä, mutta tällöinkin tarkkuus on vain 50 % ja osuus 21 %. Mallin suorituskyky C-luokan kanssa on mielenkiintoinen. Vasta 80 % minimi-tarkkuudella malli alkaa luokitteluun hyvin C-luokkaa 95 % tarkkuudella ja 28 % osuudella. C-luokan tarkkuus kuitenkin laskee rajusti, kun minimi-tarkkuus lasketaan 70 %. D- ja E-luokan kanssa malli toimii odotettavasti. Tarkkuus laskee minimi-todennäköisyyden pienentyessä ja samalla osuus kasvaa.



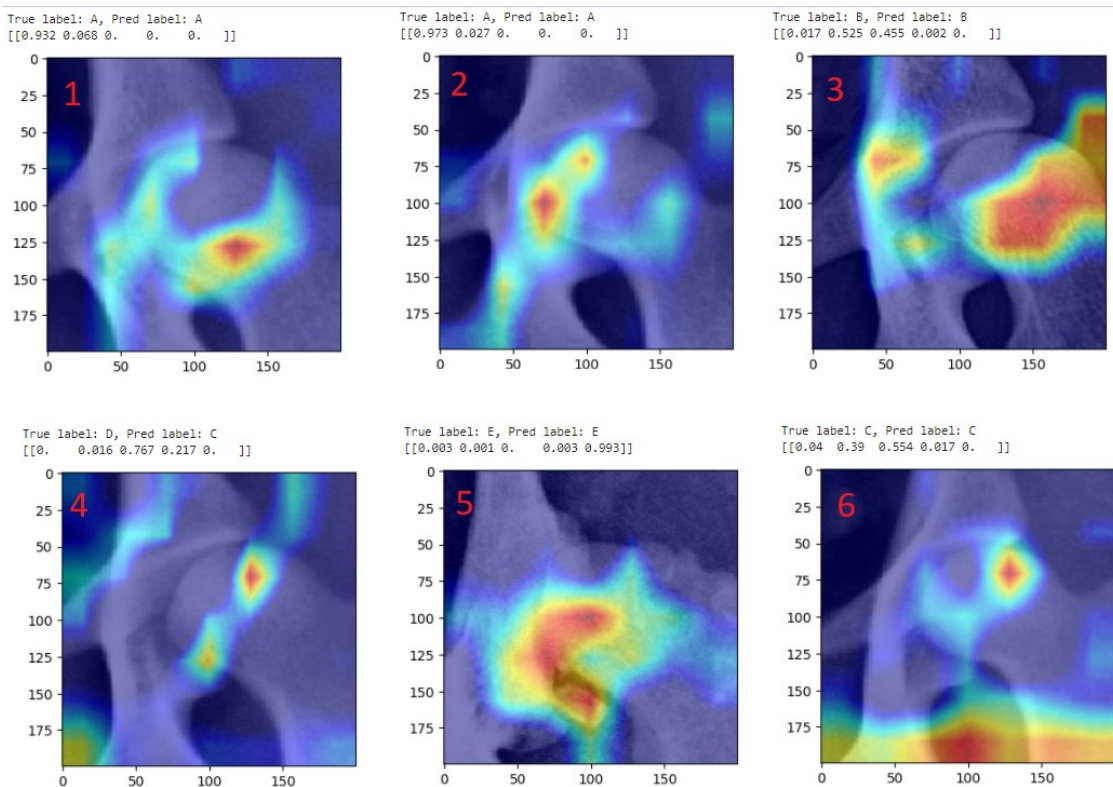
Taulukko 13. Sekaannusmatriiseja (EfficientNetB2, sama malli kuin taulukossa 11), kun asetetaan suurempi minimitodennäköisyys kuin 95 %, 90 %, 80 % ja 70 % ennustetulle luokalle.

	A, n=6630		B, n=3672		C, n=2452		D, n=506		E, n=152	
Min Tod	Oikein	Osuus	Oikein	Osuus	Oikein	Osuus	Oikein	Osuus	Oikein	Osuus
> 95 %	100%	31%	0%	1%	0%	0%	99%	20%	96%	31%
> 90 %	100%	47%	0%	3%	89%	5%	96%	33%	91%	41%
> 80 %	100%	65%	0%	12%	95%	28%	84%	57%	86%	58%
> 70 %	96%	76%	50%	21%	80%	53%	79%	72%	76%	75%

Taulukko 14. Taulukoituja arvoja taulukon 13 sekaannusmatriiseista, kun n kertoo kyseisen luokan kokonaismäärän. Oikein on oikein luokiteltujen tapausten määrä, kun ylitetään rivin minimitodennäköisyys ja osuus kertoo, kuinka monessa tapauksessa rivin minimitodennäköisyys ylitettiin.

7.5.3 Aktivointi

Konvoluutioneuroverkot eivät ole helposti tulkittavissa. Usein jää mysteeriksi, mitä kuvan alueita malli ”katsoo” antaessaan tuloksia. Eräs menetelmä, jolla voidaan hieman avata tätä mysteeriä, on luoda niin sanottu aktivointilämpökartta. Tämän avulla voidaan visualisoida mistä kuvan alueista malli on ”kiinnostunut”.

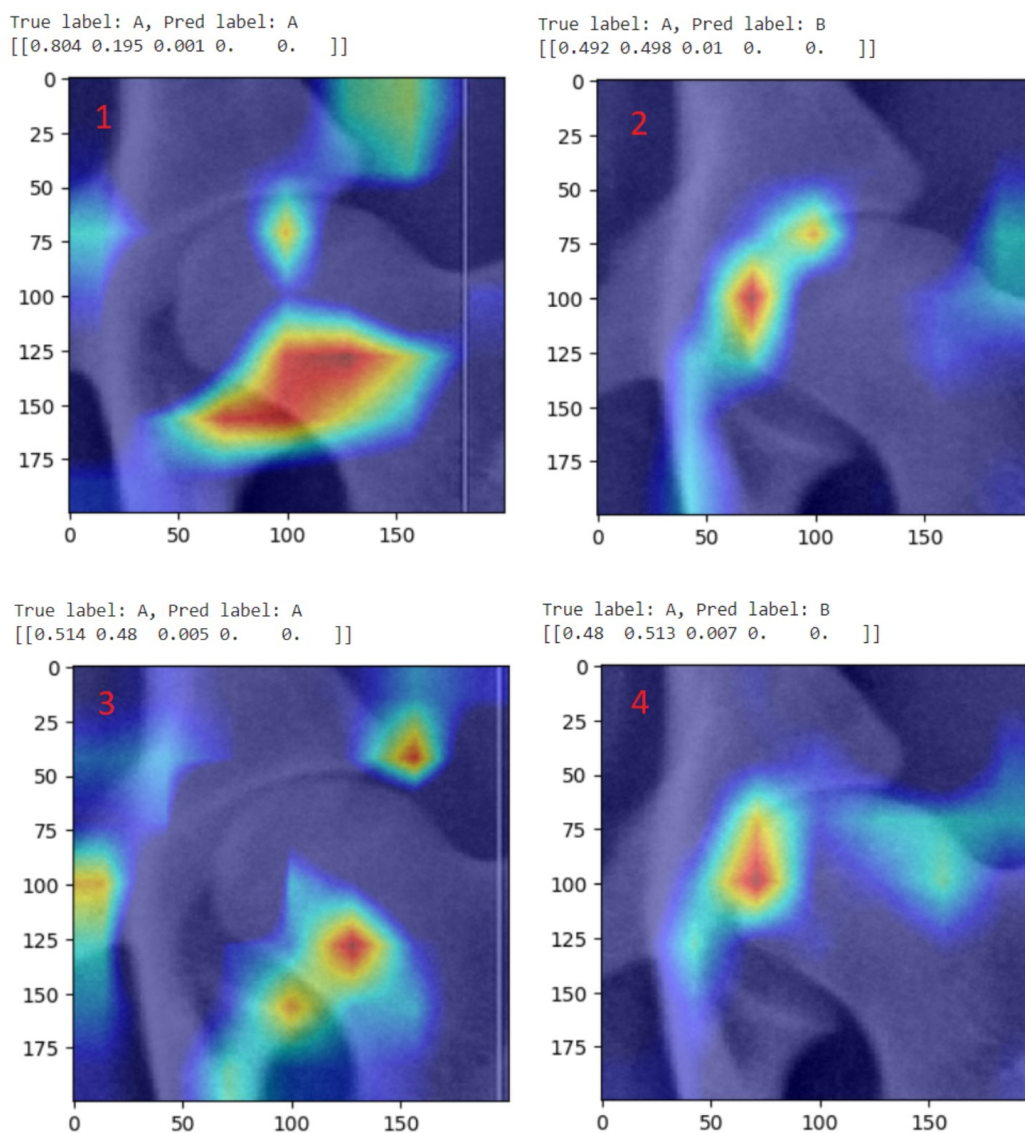


Kuva 19. EffcientNetB2:n tuottamia aktivointilämpökarttoja, tapauksen oikea luokka, ennustettu luokka ja softmaxin tuottama luokkakohtainen ennustus suluissa. Sulkujen ensimmäinen luku kertoo todennäköisyyden, että lonkka kuuluu A-luokkaa, toinen B-luokkaan, kolmas C-luokkaan, neljäs D-luokkaan ja viides E-luokkaan.

Kuvasta 19 huomaa aktivointien tapahtuvan ympäri kuvaa. Mitä punaisempi alue on, sitä tärkeämpänä malli on pitänyt kyseistä aluetta kyseisen ennustuksen tuottamisessa. Useassa tapauksessa aktivoinnit ovat jokseenkin ymmärrettävissä, kun malli on kiinnostunut etenkin lonkkanivelen ja -maljan alueesta. Kuvan 19 osa 6 on poikkeus tähän, sillä malli on erityisen ”kiinnostunut” kuvan alaosa, jossa ei pitäisi olla lonkkanivelen laatuun vaikuttavia tekijöitä. Järkevimät aktivoinnit vaikuttaisivat olevan kuvan 19 osissa, joissa ennustettu luokka sai korkean todennäköisyyden, eli 1, 2 ja 5. Näissä todennäköisemmän luokan todennäköisyydet olivat 93 %, 97 % ja 99 %.

7.5.4 Varianssi

Järjestelmän tulisi antaa sama tulos tutkimuksen kaikista kuvista olettaen, että kuvat ovat oikein kuvattu. Näin ei kuitenkaan vaikuta tapahtuvan.



Kuva 20. EfficientNetB2 Sama lonkka, eri kuva, vaihtelevia tuloksia, erilaisia aktivointilämpökarttoja.

Kuvassa 20 on eräs esimerkki tilanteesta, jossa malli on antanut eriäviä tuloksia samasta lonkasta. Osan 1 perusteella malli olisi antanut 80 % todennäköisyydellä tapaukselle tulokseksi A:n. Muissa osissa 2–4 malli on epävarma ja antaa vain noin 50 % todennäköisyyden korkeimmalle luokalle. Eri osat myös tuottavat hyvinkin erilaisia aktivointilämpökarttoja, vaikka ihmissilmään kuvat ovat lähes identtiset. Tämän vuoksi voidaan todeta mallin kärsivän korkeasta varianssista.

7.5.5 Vertailu toiseen tutkimukseen

Tuloksiani voidaan myös verrata vastikään julkaistuun tutkimukseen *Deep transfer learning can be used for the detection of hip joints in pelvis radiographs and the classification of their hip dysplasia status* [McEvoy et al. 2021], jossa suoritettiin vastaavaa lonkkakuvien binääristä luokittelua siirto-opin avulla. Jotta tuloksista saadaan vertailukelpoisia, tulee omat tulokseni muuttaa myös binäärisiksi. Tässä tapauksessa ennustettavat luokat ovat hyvä ja huono tarkoittaen A ja B ovat hyviä ja C, D ja E huonoja.

	A	B	C	D	E
[McEvoy et al. 2021]	92%	83%	31%	75%	92%
Esittelemäni konvoluutioneuroverkko	99%	91%	69%	99%	100%

Taulukko 15. Tulosten vertailu. [A, B] = LD negatiivinen, [C, D, E] = LD positiivinen. Esimerkiksi A yhtäkuin 92 % tarkoittaa että 92 % A-lonkista luokiteltiin oikein LD negatiiviseksi, loput 8 % väärin LD positiiviseksi.

Taulukon 15 tuloksista huomaa minun metodini toimivan huomattavasti paremmin jokaisessa luokassa. Yhteistä näille tuloksille on se, että C-luokka oli selkeästi vaikein luokitella oikein huonojen luokkaan [C, D, E]. Mielenkiintoista on huomata, että heidän metodinsa on merkittävästi huonompi kuin satunnainen arvaus kun kyseessä on C-luokka. Tuloksista huomaa myös A- ja E-luokkien olevan helpoimmat luokat luokitella. Tulos ei kuitenkaan ole yllättävä, kun A ja E ovat reunimmaisimmat luokat. Tämän lisäksi minun mallini pärjäsivät todella hyvin D-luokan kanssa saavuttaen 99 % tarkkuuden verrattuna heidän mallinsa 75 % tarkkuuteen.

7.5.5.1 Tutkimusten erot

Vaikka tutkittava kohde oli sama molemmissa tutkimuksissa, toteutuksessa oli merkittäviä eroja. Eräs suuri ero tutkimuksissa oli datan määrä. Käytin itse tutkimuksessani noin 110 000 lonkkanivelkuvaa sekä rajasin rodut 20 yleisimpään. Heillä puolestaan oli käytössä 23 518 lonkkanivelkuvaa 128 eri rodusta. Lisäksi molemmat tutkimukset kärsivät samasta datan luokkajakauman vinoumasta. Heidän tutkimuksissaan 87 % lonkista kuuluivat hyvään A- tai B-luokkaan, kun minun aineistossani vastaava A- ja B-luokan osuus oli 78 %. Heidän tutkimuksissaan aineiston koko korosti luokkajakauman

vinouman tuottamia ongelmia vielä entisestään. Heidän käyttämänsä kuvat olivat myös merkittävästi laajemmin rajattu verrattuna omiini (kuva 14). Tämän merkitystä suorituskykyyn on vaikea arvioida.

Heidän käyttämänsä malli oli YOLO v3-Tiny, joka hyödynsi siirto-oppimista (transfer learning) alustamalla COCO-datajoukon painot. Siirto-oppimisen ideana on hyödyntää toisessa tehtävässä opittuja painoarvoja ja alustaa malli näillä painoarvoilla satunnaisten arvojen sijasta [Elgandy 2020 s.240–241]. Valinta käyttää siirto-oppimista johtunee käytettävän datan suppeasta määrästä. Siirto-oppimisen hyödyt korostuvat etenkin, kun dataa on vähän. Heidän malliansa opetettiin 6000 epookkia ilman varhaista lopetusta tai suorituskyvyn monitorointia, kuten validointi. Täten mallin suorituskykyä ei ole yritetty optimoida. Tällainen menettely jättää paljon parannettavaa. Mallin suorituskyvyn optimointi ja sen evaluointi ovat kuitenkin oleellinen osa koneoppimisen tehtäviä. Lisäksi heidän tutkimuksessaan ei kerrota heidän mallinsa suorituskykyä opetusdatan kanssa, jonka kautta olisi voinut arvioida onko mallilla taipumusta ylisovittautumiseen. Uskonkin, ettei merkittävästi parempien tulosten saavuttamiseen olisi tarvittu paljota lisätyötä.

7.6 Pohdinta

Automaattisen lonkkanivellausujan tuottamat tulokset ovat lupaavia. Mallin kykyyn luokitella suuri määrä A-lonkkia oikein voi olla tyytyväinen. A-lonkkia on kuitenkin noin 50 % kaikista lonkista (ainakin tässä aineistossa). Vaikka mallia käytettäisiin vain A-lonkkien kanssa, kun minimi todennäköisyys on vähintään 80 %, kattaisi se jo noin 33 % kaikista tapauksista. Muiden luokkien kanssa joutuu tekemään vielä töitä, mikäli haluaa antaa järjestelmälle kyvyn luokitella niitä suurella tarkkuudella. Ongelmia on etenkin B-luokan kanssa, eikä malli kykene luokittelemaan niitä lainkaan riittävän suurella varmuudella.

Mallin arkkitehtuurin muutoksilla tuskin voidaan enää mallin suorituskykyä merkittävästi parantaa. Sen sijaan mallin suorituskykyä muiden luokkien kanssa voisi koittaa parantaa hyödyntämällä yhdistelmäoppimista (ensemble learning), eli ajaa syöte kuvan usean mallin lävitse, joiden antamista tuloksista otetaan vaikkapa painotettu keskiarvo lopulliseksi tulokseksi. Nämä yhdistelmäoppimisen tuottamat uudet mallit voisivat myös olla erikoistuneita joihinkin tiettyihin luokkiin, kuten tunnistamaan pelkästään A ja B toisistaan. Yhdistelmäoppiminen myös todennäköisesti vähentää varianssia, joka tällä hetkellä on suuri. Uskoisin yhdistelmäoppimisen olevan looginen seuraava askel järjestelmän suorituskyvyn parantamiseksi.

Mielestäni suurin este järjestelmän suorituskyvyn parantamiselle on lonkkalausuntoaineiston laadun tai tarkkuuden kyseenalaisuus. Paneeliaineiston analysointi ei anna kovinkaan vakuuttavaa kuvaa FCI-skeeman lonkkaniveldiagnoosien laadusta. Paneeli on yksimielinen ainoastaan 34 % paneeliin menevistä tapauksista ja 31 % tapauksissa alkuperäinen tulos muuttuu. Alkuperäiset diagnoosit eivät näin ollen vaikuta luotettavilta. Mikäli kaikki lonkkaniveltapaukset menisivät paneelin uudelleen arvioitavaksi, on todennäköistä, että merkittävä osuus tapauksista tulisi takaisin alkuperäisestä eriävän

tuloksen kanssa. Koneoppimisen malli voi olla vain niin hyvä kuin sen opetukseen käytetty data, jolloin huono tai epäluotettava data rajoittaa mallin suorituskykyä. Olisi mielenkiintoista selvittää, miten malli toimisi, jos mallin datana käytettäisiin OFA:n tuottamaa dataa, joka vaikuttaisi olevan merkittävästi luotettavampaa. Kaikissa OFA-lausunnoissa on yhdistettynä kolmen radiologin lausunto, jolloin virheiden määrän voisi uskoa olevan pienempi. Lisäksi he ovat yksimielisiä 73.5 % tapauksista, eli kukin lausuja antaa riippumattomasti yhden saman tuloksen seitsemästä eri vaihtoehdosta.

Muutenkin paneelidatan analysoinnista saamieni tulosten perusteella olisi järkevää tutkia laajemmin lausuntojen luettavuutta. Minun on vaikea uskoa, että nykyinen tilanne on tarpeeksi hyvä, kun uudelleen arvioitavista tapauksista 1/3 tapauksessa voidaan todeta alkuperäisen tuloksen olevan väärin. Suurimmassa osassa tapauksissa lausujien diagnooseissa on enemmän tai vähemmän vaihtelevuutta.

Mikäli automaattista lonkkanivellausujaa aiotaan jatkokehittää, olisi järkevää lisätä käytettävään dataan kaikki koirarodut. Tällä hetkellä malli on opetettu käyttäen ainoastaan yleisintä 20 rotua, joista on eniten lonkkanivellausuntoja. Lisäksi käytössä on ainoastaan tapaukset, joissa vasen ja oikea lonkka on saanut saman tuloksen. Mikäli opetusdataan lisätään kaikki rodut ja luodaan järjestelmä, mikä mahdollistaa niidenkin kuvien mukaan ottamisen, jossa vasemman ja oikean puolen tuloksen eroavat toisistaan, saataisiin datan määrää lisättyä noin 113 000 lonkkanivelestä 250 000 lonkkaniveleen. Näin suurella datan määrällä lisäyksellä voisi myös olla positiivinen vaikutus mallin suorituskykyyn.

Ehdottamani lonkkanivellausuja suorittaa automaattisesti piirteiden erotuksen. Eräs vaihtoehtoinen tapa voisi olla sellainen, jossa opetetaan malli tai malleja erottamaan samat piirteet, joita radiologitkin katsovat. Tämä voisi tapahtua BVA-skeeman tapaan, jossa lonkkanivelestä pisteytetään 9 ominaisuutta 0–6 asteikolla. Täten sen sijaan että malli opetetaan antamaan lopullinen lonkkanivellausunto, malli antaisi pisteet jollekin ominaisuudelle. Tällainen lähestymistapa vaatisi uutta dataa, jossa jokainen ominaisuus on pisteytetty. Tähän soveltuvan datan luominen tyhjästä on kumminkin työlästä ja todennäköisesti epärealistista, koska dataa tarvittaisiin paljon ja sen luomiseen vaaditaan radiologin ammattitaitoa. Mikäli käytössä olisi BVA:n lonkkanivelaineisto, jossa oletettavasti olisi tieto siitä, mistä lopulliset pisteet koostuivat, uskoisin tällaisen lähestymistavan olevan kokeiluain arvoista.

Kaikesta huolimatta näen järjestelmän käyttöönotolle paljon potentiaalia. Näiden tulosten perusteella lonkkanivellausuntojen täydellinen automatisointi on epärealistinen tavoite, mutta osittainen automatisointi erittäin realistinen. Järjestelmää voisi muun muassa käyttää siten, että kuvaava eläinlääkäri antaa oman arvionsa lonkkanivelten laadusta, jonka jälkeen kuvat ajetaan järjestelmän läpi, joka palauttaa oman arvionsa. Mikäli kuvaavan eläinlääkäriin ja järjestelmän antamat arviot lonkkien laadusta täsmäisivät (voitaisiin rajoittaa vain A-lonkkiin) ja järjestelmä on luottavainen omaan arvioonsa,

voidaan tätä tulosta käyttää suoraan lopullisena tuloksena lähettämättä sitä eteenpäin lausuvalla eläinlääkärille. Tilanteissa, jossa kuvaava eläinlääkäri on mallin tuloksen kanssa eri mieltä, tai malli on epävarma, jolloin kyseessä on todennäköisesti rajatapaus, lähetettäisiin kuvat nykyiseen tapaan lausuvan eläinlääkärin arvioitavaksi.

Järjestelmällä tulisi myös olla ominaisuus arvioida, täyttääkö arvioitava kuva lonkkanivelkuvalle asetetut vaatimukset. Suurin huoli tässä on, mikäli järjestelmä ei pysty tähän, voisi tuloksia manipuloida asettamalla koiran jalat kulmiin, jotka saavat lonkat näyttämään paremmilta kuin ne todellisuudessa ovat. Lisäksi järjestelmän varianssia tulisi vähentää, jotta samasta lonkasta tulisi aina sama tulos eikä sitä voisi manipuloida syöttämällä useita kuvia samasta lonkkanivelestä ja odottaa, että saavutetaan mieleinen tulos.

Lisäksi mikäli löydetään hyvin toimiva järjestelmä lonkkanivelten luokitteluun, voitaisiin vastaavaa käyttää myös koiran kyynärnivelten kasvuhäiriön luokitteluun. Koiran kyynärnivelten kasvuhäiriötä diagnosoidaan vastaavaan tapaan kuin lonkkanivelten kasvuhäiriötä. Kyynärnivelten kasvuhäiriön diagnosointiin käytetään 4-asteista asteikkoa 0–4, kun 0 normaali ja 4 voimakkaat muutokset. Kyynärnivelten kasvuhäiriön luokittelu saattaa olla helpompi tehtävä kuin lonkkanivelten häiriönluokittelu, sillä siinä on yksi luokka vähemmän. Dataakin on jo valmiiksi saatavilla kyynärnivelten kasvuhäiriöstä, röntgenkuvia ja niihin liittyviä tietoja.

8 Yhteenveto

Viime vuosien edistysaskeleen ja lupaavat tulokset konenäkö tutkimuksessa ovat inspiroineet konenäköjärjestelmien laajaa testaamista useissa eri toimintaympäristöissä. Tutkielmani tavoite oli luoda ja testata, voisiko konenäköä hyödyntää koirien virallisten lonkkalausuntojen tekemisessä. Tulokseni osoittivat sen olevan mahdollista. Vaikkei luomani malli kykene kaikkia lonkkia tarkasti luokittelemaan, toimii se erittäin hyvin A-lonkkien kanssa, joka on selkeästi yleisin luokka. Kehittämällä järjestelmää entisestään voitaisiin merkittävästi parantaa sen suorituskykyä muidenkin luokkien kanssa.

Suurimpana esteenä järjestelmän suorituskyvylle on lonkkanivelaineiston laadun tai tarkkuuden kyseenalaisuus. Paneeliaineiston perusteella yksittäinen lonkkalausunto ei ole luotettava. Laajempi tutkimus Suomen lonkkanivellausuntojen laatuun ja tarkkuuteen on tarpeen.

9 Viiteluettelo

AKC. 2021. Hip dysplasia in dogs. <https://www.akc.org/expert-advice/health/hip-dysplasia-in-dogs/>. Katsottu 10.12.2021. pp.178-194.

Esteva Andre *et al.* 2021. Deep learning-enabled medical computer vision. *npj Digit. Med.* Volume 4. Number 5. pp.1-9.

Avi Arora. 2021. Complete glossary of Keras optimizers and when to use them (with code). <https://analyticsarora.com/complete-glossary-of-keras-optimizers-and-when-to-use-them-with-code/#What-Optimizer-to-Use>. Katsottu 10.12.2021.

AVCS. Canine hip dysplasia. <https://www.acvs.org/small-animal/canine-hip-dysplasia>. Katsottu 10.12.2021.

Baeldung. 2020. F-1 score for multi-class classification. <https://www.baeldung.com/cs/multi-class-f1-score>. Katsottu 10.12.2021.

Jason Brownlee. 2018. A Gentle Introduction to k-fold cross-validation. <https://machinelearningmastery.com/k-fold-cross-validation/>. Katsottu 10.12.2021.

Jason Brownlee. 2019a. A gentle Introduction to the ImageNet challenge (ILSVRC). <https://machinelearningmastery.com/introduction-to-the-imagenet-large-scale-visual-recognition-challenge-ilsvrc/>. Katsottu 10.12.2021.

Jason Brownlee. 2019b. How to avoid overfitting in deep learning neural networks. <https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/>. Katsottu 10.12.2021.

Antoine Buetti-Dinh, Vanni Galli, Sören Bellenberg, Olga Ilie, Malte Herold, Stephan Christel, Mariia Boretska, Igor V. Pivkin, Paul Wilmes, Wolfgang Sand, Mario Vera, Mark Dopson. 2019. Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition. *Biotechnology Reports*. Volume 22. Pages 5.

S. Burti, V. Longhin Osti, A. Zotti, T. Banzato. 2020. Use of deep learning to detect cardiomegaly on thoracic radiographs in dogs. *The Veterinary Journal*. Volume 262. Pages 5.

Viernickel Clemens. 2021. Computer vision startups enhancing security & surveillance. <https://www.siasearch.io/blog/computer-vision-startups-security-surveillance>. Katsottu 10.12.2021.

Vilnius County. 2020. Popular X-ray image normalization techniques. <https://www.kaggle.com/raddar/popular-x-ray-image-normalization-techniques>. Katsottu 10.12.2021.

Ruth Dennis. 2012. Interpretation and use of BVA/KC hip scores in dogs. *InPractice*. Volume 34. Number 4.

- Sanket Doshi. 2019. Various optimization algorithms for training neural network. <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>. Katsottu 10.12.2021.
- Mohamed Elgendy. 2020. *Deep learning for vision systems*. Manning.
- Evidensia. 2019. Koiran lonkkavika eli lonkkadysplasia. <https://evidensia.fi/hoitovinkit/koiran-lonkkavika-eli-lonkkadysplasia/>. Katsottu 10.12.2021
- McEvoy Fintan J, Proschowsky Helle F, Müller Anna V, Moorman Lilah, Bender-Koch Johan, Svalastoga Eiliv L, Frellsen Jes, Nielsen Dorte H. 2021. Deep transfer learning can be used for the detection of hip joints in pelvis radiographs and the classification of their hip dysplasia status. *Vet Radiol Ultrasound*. Volume 62. Number 5. pp.378-393
- Mark Flückiger. Scoring radiographs for canine hip dysplasia - the big three organisations in the world. https://www.dysplasie-schweiz.unibe.ch/unibe/portal/fak_vetmedizin/micro_dysplasie/content/e332465/e406506/e406508/Scoring_radiographs_ger.pdf. Katsottu 29.12.2021
- IBM. 2021. Overfitting. <https://www.ibm.com/cloud/learn/overfitting>. Katsottu 10.12.2021.
- Johnson, J.M., Khoshgoftaar, T.M. 2019. Survey on deep learning with class imbalance. *J Big Data*. Volume 6. Number 27. pp.1-54.
- Kenneliitto. 2013. Lonkkanivelen kasvuhäiriö. <https://www.kenneliitto.fi/kasvatus-ja-terveys/koiran-terveys/perinnolliset-sairaudet-ja-koiran-hyvinvointi/lonkkanivelen-kasvuhairio>. Katsottu 10.12.2021.
- LabelImg. 2022. <https://github.com/tzutalin/labelImg>. Katsottu 2.1.2022.
- Xuechen Li, Linlin Shen, Xinpeng Xie, Shiyun Huang, Zhien Xie, Xian Hong, Juan Yu. 2020. Multi-resolution convolutional networks for chest X-ray radiograph based lung nodule detection. *Artificial Intelligence in Medicine*. Volume 103.
- Ilija Mihajlovic. 2019. Everything you ever wanted to know about computer vision. <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-is-so-awesome-e8a58dfb641e>. Katsottu 10.12.2021.
- MVS. Surgery for hip dysplasia in dogs. <https://www.memphisveterinaryspecialists.com/site/blog-cordova/2020/08/17/surgery-for-hip-dysplasia-in-dogs>. Katsottu 10.12.2021.
- Katariina Mäki. 15.9.2021. Viesti koirien jalostusasiatuntijalta.
- OFA. Hip grade details. <https://www.ofa.org/diseases/hip-dysplasia/grades>. Katsottu 10.12.2021.
- OFA. Breed statistics. <https://www.ofa.org/diseases/breed-statistics>. Katsottu 9.12.2021
- Papers with code. Image classification on ImageNet. <https://paperswithcode.com/sota/image-classification-on-imagenet>. Katsottu 10.12.2021.

Adrian Rosebrock. 2021. Convolutional neural networks (CNNs) and layer types. <https://www.pyimage-search.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/>. Katsottu 10.12.2021.

C. Rubini, N. Pavithra. 2019. Contrast Enhancement of MRI Images using AHE and CLAHE Techniques. *International Journal of Innovative Technology and Exploring Engineering*. Volume 9. Number 2. pp.2442-2445.

Olga Russakovsky *et al.* 2014. ImageNet large scale visual recognition challenge. *International journal of computer vision*. Volume 115. pp.211-252.

Ohlerth Stefanie, Geiser Britta, Flückiger Mark, Geissbühler Urs. 2019. Prevalence of canine hip dysplasia in Switzerland between 1995 and 2016—a retrospective study in 5 common large breeds. *Frontiers in Veterinary Science*. Volume 6. pp.378.

Towards Machine Learning. 2018. Deep learning series, P1: basics of neural networks and understanding gradient descent. <https://towardsml.com/2018/08/28/deep-learning-series-p1/>. Katsottu 10.12.2021.

Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis. 2018. Deep learning for computer vision: a brief review. *Computational Intelligence and Neuroscience*. Volume 2018. Pages 10.

viso.ai. 2021. What is computer vision? The complete technology guide for 2021. <https://viso.ai/computer-vision/what-is-computer-vision/>. Katsottu 10.12.2021.

Yang, H *et al.* 2021. Deep learning-based six-type classifier for lung cancer and mimics from histopathological whole slide images: a retrospective study. *BMC Med*. Volume 19. Number 80. pp.1-14.