Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

TONI AALTONEN

Ihmisen tunnistaminen vedenpohjasta käyttäen kuluttajaluokan kaikuluotaimen datalla opetettuja syviä neuroverkkoja

TEKNIIKAN YLEMPI TUTKINTO-OHJELMA
2022

TIIVISTELMÄ

Aaltonen Toni: Ihmisen tunnistaminen vedenpohjasta käyttäen kuluttajaluokan kaikuluotaimen datalla opetettuja syviä neuroverkkoja
Opinnäytetyö, ylempi AMK
Tutkinto-ohjelma Tekniikka
Joulukuu 2022
Sivumäärä: 80

Suomen sameisiin vesiin hukkuu ihmisiä joka vuosi. Osa uhreista olisi mahdollista pelastaa, jos heidät löydettäisiin nopeammin. Nykyisin etsintä tapahtuu käytännössä käsin pohjaa läpikäymällä.

Työn aiheena oli käydä läpi nykyistä vesipelastuksen tilaa ja prosessia sekä tutkia neuroverkkoihin perustuvaa tunnistusjärjestelmää, jolla on mahdollista nopeuttaa pelastettavan löytymistä vedenpohjasta. Työssä myös käytiin läpi tarvittava datankeräys, jonka pohjalta voitiin kerätä ja kehittää kansainvälisestikin merkittävä datasetti, joka sisältää yli 300 kuvaa ihmisestä vedenpohjassa.

Työn tavoitteena oli tutkia, voidaanko kuluttajaluokan viistokaikuluotaindatasta tunnistaa ihminen, ja yleistyivätkö tulokset täysin erilliseen, erilaiseen testidatasettiin. Tulosten perusteella voidaan sanoa, että kuluttajaluokan viistokaikuluotaimilla tunnistus voidaan tehdä. Ihmisen tunnistamisen tulokset saatiin yleistymään erilliseen testidatasettiin. Datasetin pienuuden takia ei voida varmistua, että malli vielä toimii kaikissa tilanteissa ja kaikilla kaikuluotaimilla.

Avainsanat: syväoppimen, neuroverkot, viistokaikuluotain

Abstrac

Aaltonen, Toni: Underwater human recognition using consumer grade sonar data trained deep neural networks
Master’s thesis
Master of Engineering
December 2022
Number of pages: 80

People drown in Finland's murky waters every year, some of the victims could have been saved if they had been found sooner, nowadays the search is practically done by going through the bottom by hand.

The topic of the work was to go through the current state and process of water rescue, to study a recognition system based on neural networks, which can be used to speed up the finding of a person to be rescued at the bottom of the water. The work also went through the necessary data collection, based on which a dataset containing more than 300 images of people at the bottom of the ocean could be developed.

The goal of the work was to investigate whether a person can be identified from the data of a consumer-grade sonar, and whether the results generalize to a separate, different dataset. Based on the results, it can be said that identification can be done with consumer class side scanning sonars, and the results were also generalized to separate test data, due to the small size of the dataset it cannot be confirmed that the model yet works in all situations and with all sonars.

Keywords: deep learning, neural networks, side scanning sonars,

ALKUSANAT

Erityisesti haluan kiittää Satakunnan pelastuslaitoksen Rauman toimipistettä datankeräyksen avustamisesta. Ilman laitoksen avokätistä apua ei olisi työn keskiössä olevaa dataa ollut mahdollista kerätä. Konkreettisen avun lisäksi sain paljon arvokkaita neuvoja ja tietoa aiheeseen liittyen. Haluan kiittää Ulla Tuomisen Säätiötä ja Satakunnan Korkean Teknologian Säätiötä datankeräykseen liittyvien laitteiden rahoittamisesta sekä Opetus- ja kulttuuriministeriötä AI Forum -hankkeen rahoittamisesta, sekä kaikkia muita erikseen mainitsemattomia tahoja ja henkilöitä, jotka ovat panoksellaan mahdollistaneet hankkeiden ja tämän työn toteutumisen ja etenemisen.

SISÄLLYS

1 JOHDANTO 7
2 AIHEEN ESITTELY 9
2.1 Vesisukelluspelastuksen prosessi 11
2.2 Kaikuluotaimet 11
2.3 Vedenalainen robotiikka 12
3 TYÖN TUTKIMUSKYSYMYKSET JA METODIT 13
4 TUTKIMUKSEN LUOTETTAVUUS & HAASTEET 18
4.1 Validiteetti 19
4.2 Reliabiliteetti 20
4.3 Eettiset kysymykset 21
5 TYÖN VIITEKEHYS 23
5.1 Ohjattu oppiminen 23
5.2 Neuroverkot 24
5.3 Aktivointifunktiot 25
5.4 Ylioppimisen välttäminen 26
5.5 Dropout 27
6 NEUROVERKKOARKKITEHTUURIN VALINTA 29
6.1 Työssä hyödynnetyt neuroverkot 30
6.2 Siirto-oppiminen 35
6.2.1 Verkkopohjainen siirto-oppiminen 35
6.2.2 Datapohjainen siirto-oppiminen 35
6.3 Optimaalinen virhearvo 36
7 DATAN HANKINTA KEHITTÄMISTYÖSSÄ 38
7.1 Datankeruu neuroverkon opetusta varten 38
7.2 Datan valmistelu neuroverkon opetukseen 40
7.3 Datan siirto-oppiminen 41
7.3.1 Domaindata 41
7.3.2 Lääketieteellinen kuvantamisdata 42
8 NEUROVERKKOMALLIN KEHITTÄMINEN 44
8.1 Ohjelmistot ja niiden versiot työssä 44
8.2 Ensimmäiset testit ja tulokset 45
8.3 Testit ja tulokset staattisella datasettillä 46
8.4 Datan Augmentointi datageneraattorilla 51
8.5 Testit ja tulokset eri oppimisnopeuksilla 55

8.6 Testit yksikanavaisella kuvalla 56
8.7 Verkkopohjainen siirto-oppiminen domaindatalla 59
8.8 Datapohjaisen siirto-oppiminen testi lääketieteellisellä datalla 61
9 TESTIEN JA TULOSTEN YHTEENVETO 69
9.1 Tulokset – tutkimuskysymys 1 69
9.2 Tulokset – tutkimuskysymys 2 71
9.3 Tulokset – tutkimuskysymys 3 71
10 JOHTOPÄÄTÖKSET JA POHDINTA 73
10.1 Datasetin suurentaminen 74
10.2 Hybridimallin kehitys 75
10.3 Synteettinen kaikuluotaindata 75
10.4 Ultraäänidatan tuloksien tarkempi tarkastelu 75
10.5 Metodin julkaisu kaikuluotaindatan käsittelystä ja analysoinnista 76
10.6 Koulutetun neuroverkon testaus pelastussukelluksessa 76

1 JOHDANTO

Vesionnettomuuden uhrin selviämisen kannalta on ensiarvoisen tärkeää, että paikannus tapahtuu nopeasti, ja elvytys voidaan aloittaa viipymättä. Toisaalta kylmässä vedessä uhri voidaan tietyissä olosuhteissa onnistua elvyttämään yli puolituntia veden alle painumisen jälkeen. Nykyisin etsintää jatketaan vähintään tunti hälytyksen vastaanottamisesta, tai ihmisen pinnan alle joutumisesta (Sisäasiainministeriö, 2007, s. 16).

Tässä työssä käydään läpi veden alle painuneen ihmisen pelastamisen nykyisiä pelastustoimen käytäntöjä, ja miten tätä prosessia voidaan kehittää teknologian avulla. Aiheen valitsin, koska sen kehitys on tärkeää, jotta voidaan pelastaa ihmishenkiä. Valintaan vaikutti myös se, että olen tehnyt aiheeseen liittyen jo tukimusta, julkaissut vertaisarvioituja kansainvälisiä jufo1 -luokiteltuja konferenssijulkaisuja, sekä tehnyt amk-opinnäytetyön aihetta sivuten. Tätä edeltävissä töissä neuroverkkopohjainen ihmisen tunnistus rajattiin pois työstä, koska tarkoitus oli kirjoittaa tämä työ näistä aiheista.

Maailmalla on jo vuosia ollut käytössä kaikuluotaimia uhrin etsinnän apuna, mutta näiden käyttö Suomessa on edelleen satunnaista, eikä niitä esimerkiksi ole lainkaan käytössä Satakunnan pelastuslaitoksen Rauman yksikössä, jonka pelastussukeltajia haastattelin työtä varten saadakseni paremman käsityksen sukelluspelastajan käytännön työstä ja siihen liittyvistä haasteista. Nämä haastattelut varmistavat sen, että aihe on tärkeä, ja ettei siihen liittyen ole kehitystä Suomessa tapahtunut.

Kun ensimmäisen kerran aloin selvittämään aihetta 2019, ei maailmallakaan ollut vielä kuin rajallinen määrä tieteellisiä julkaisuja syvien neuroverkkojen käytöstä kaikuluotainkuvan analyysissä, ja näistäkin käytännössä kaikki käyt-

tivät kalliita ammattitason kaikuluotaimia. Satakunnan pelastuslaitoksen Rauman yksikön avustamana myös toteutettiin datasetin keräys. Datasetti sisältää yli 300 viistokaikuluotainkuvaa ihmisestä vedenpohjassa, ja ne on kerätty edullisella kuluttajaluokan viistokaikuluotaimella. Tässä työssä käydään läpi tämän datasetin keräys ja prosessointi sekä toteutetaan neuroverkkopohjainen tunnistin, jolla voidaan nopeuttaa sekä automatisoida ihmisen etsintää.

Tähän työhön liittyen ja sen vaatimien laitteistojen hankintaa ja kehitystä on tukenut Satakunnan korkeanteknologian säätiö apurahalla ”Autonomous Underwater Vehicle AUV”, sekä Ulla Tuomisen säätiö hankkeella “Esiselvitys sukeltavien robottien tutkimuksesta”. Datan analysointia, neuroverkon opetustyötä, ja tämän työn kirjoittamista on tehty osana Opetus- ja kulttuuriministeriön AI Forum -hanketta.

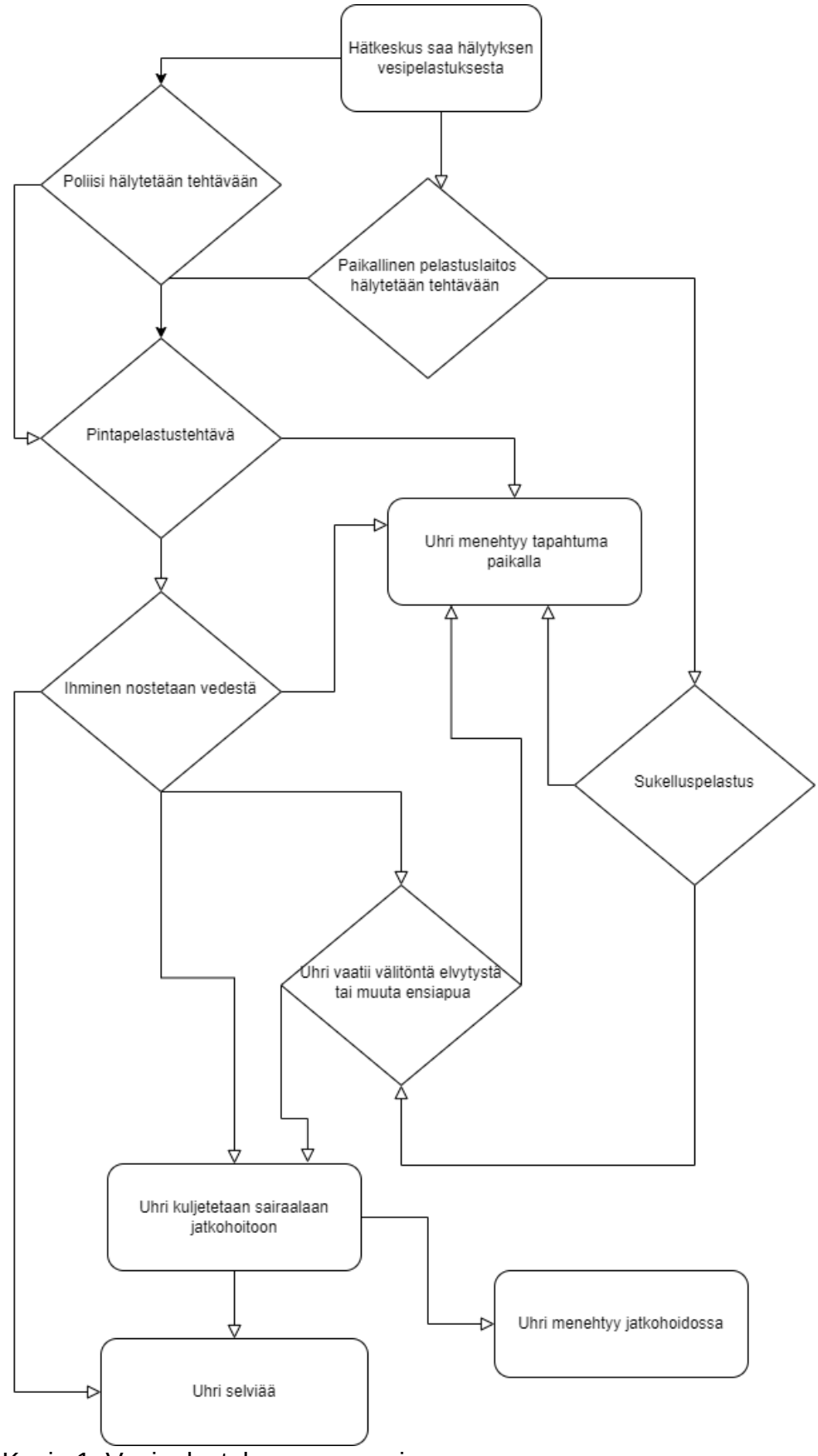
2 AIHEEN ESITTELY

Seuraavaksi esitetyt prosessit olen kehittänyt pelastussukeltajien kanssa käytyjen keskustelujen pohjalta, sekä alan kirjallisuutta tutkien. Vesipelastustehtävät jakautuvat pintapelastukseen ja vesisukelluspelastukseen. Pintapelastustehtäviä voivat myös poliisit suorittaa, mikäli he ehtivät paikalle nopeammin. (Pekari, 2016, s. 21.)

Vuosittain maailmassa hukkuu enemmän kuin 500 000 ihmistä, joista suuri osa lapsia ja nuoria aikuisia. Vesisukelluspelastuksessa uhrin löytymiseen vaikuttaa nykyisin suuresti tiedon tarkkuus siitä, missä kohdassa uhri painui pinnan alle. Mikäli paikka on hyvin tiedossa ja paikassa ei ole merkittäviä virtauksia, on pelastustehtävän onnistumisella paremmat mahdollisuudet. Mitä suuremmalta alueelta joudutaan etsintää tekemään, sitä pienemmät selviämismahdollisuudet ovat. (Szpilman ym., 2012.)

Satakunnan pelastuslaitoksella on myös kehitetty teleskooppivarren päässä olevan kameran käyttöä tehtävässä, mutta tälle on myös omat rajoitteensa, varsinkin Suomen sameavetisissä olosuhteissa. Tämä metodi toimiikin parhaiten tilanteessa, jossa uhri on painunut pinnan alle esimerkiksi venelaiturin läheisyydessä. (Puolitaival, 2015, s. 20.)

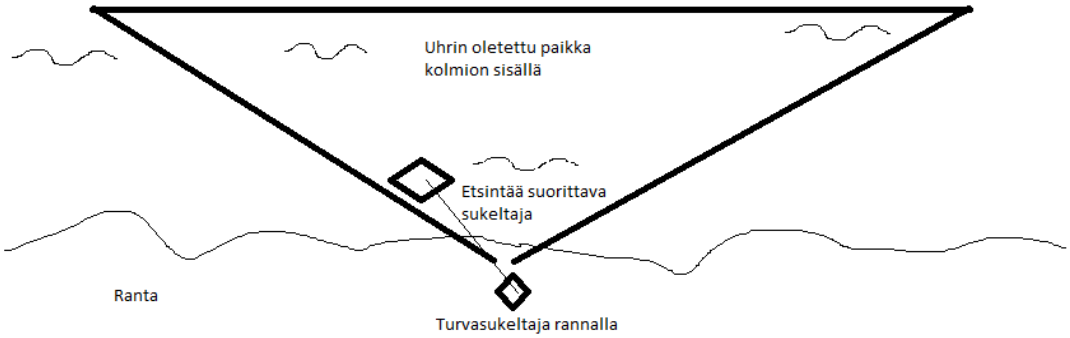
Seuraavassa kaaviossa on esitetty normaali vesipelastuksen prosessi (kuvio 1). Hälytyskeskus saa ilmoituksen onnettomuudesta, paikalle hälytetään pelastuslaitos, ja mahdollisesti myös poliisi. Pelastuslaitos hoitaa pelastuksen ensisijaisesti, ja vain koulutetut pelastussukeltajat voivat suorittaa pelastussukelluksen.



Kuvio 1. Vesipelastuksen prosessi

2.1 Vesisukelluspelastuksen prosessi

Vesisukelluksen prosessi on pysynyt lähes muuttumattomana läpi vuosikymmenten. Turvasukeltaja on rannalla ja hän käsittelee turvaköyttä, joka on kiinnitetty sukellusta tekevään sukeltajaan. Etsintää tekevä sukeltaja aloittaa sektorin laidasta läheltä rantaa, ja alkaa siirtyä pohjaa pitkin kohti sektorin toista laitaa pohjaa käsillä tunnustellen, kunnes turvasukeltaja havaitsee, että sektorin toinen laita on saavutettu. Tällöin hän antaa pelastusta tekevälle sukeltajalle merkin pysähtyä, ja antaa puoli metriä lisää turvaköyttä, jolloin sukeltaja siirtyy tämän verran lisää poispäin rannasta ja alkaa liikkua sivuttain kohti sektorin toista laitaa, taas pohjaa käsin tunnustellen. Tätä jatketaan, kunnes uhri löytyy, sektori on etsitty, jolloin voidaan kasvattaa etsintäaluetta, tai todetaan, että uhri on ollut pinnan alla vähintään tunnin, eikä selviämisestä käytännössä ole enää toivoa, ja etsintä voidaan lopettaa (kuvio 2).



Kuvio 2. Sukelluspelastuksen prosessi

2.2 Kaikuluotaimet

Kaikuluotaintekniikan nopea kehitys sekä kaikuluotainten hinnan aleneminen ovat mahdollistamassa niiden käytön pelastustehtävissä. Erityisesti kuluttajien veneilykäyttöön tarkoitettujen kaikuluotainten hintalaatusuhde on parantunut kovan kilpailun ja suurten volyymien ansiosta suurin harppauksin viime vuosina, ja lähes joka vuosi tulee uusia, entistä tarkempaa kuvaa tarjoavia malleja markkinoille. Haasteena näissä kaikuluotaimissa silti edelleen on resoluutio, ja

vain rajallisista, maksimissaan noin alle kuuden metrin syvyyksistä voidaan ihmisen kokoisia asioita kuvasta havainnoida. (Aaltonen, 2019, s. 12.)

2.3 Vedenalainen robotiikka

Koska edullisilla kaikuluotaimilla ei voida kohdetta löytää kovin pitkän matkan päästä, on yksi ratkaisu asentaa kaikuluotain vedenalaiseen robottiin. Näin voidaan aina olla pohjasta optimaalisella etäisyydellä.

Viime vuosina monia vedenalaisen robotiikan komponentteja on tullut mahdolliseksi joko hankkia edullisesti tai valmistaa itse, kuten 3D-tulostetut potkurijärjestelmät (Aaltonen, 2019, s. 11). Lisäksi ohjelmistopuolen avoimen lähdekoodin työkalut ovat parantuneet merkittävästi ja ”robot operating system” (ROS) -järjestelmän päälle on rakennettu kokonaisia avoimen lähdekoodin autonomisia vedenalaisia robotteja. (*Subjugator Auv Source Code*, n.d.)

Erityisesti pelastustoimintaan tarkoitettuja vedenalaisia laitteita ei ole vielä juuri kaupallisesti tarjolla, mutta niiden kehitystä on tehty yliopistoissa ja tutkimuslaitoksissa. Myös näiden laitteiden hinnan pitäisi olla kohtalaisen alhainen, jotta pelastuslaitosten olisi niitä mahdollista hankkia. (Shahria ym., 2019.)

Viistokaikuluotainkuvan tulkinta on usein melko hankalaa, varsinkin edullisten luotainten kuvan laatu on melko huonoa, ja varsinkin harjaantumattoman ihmisen on vaikea löytää kuvasta ihmistä. Tästä syystä on tärkeää, että tunnistusta voidaan tehdä automaattisesti. Myös langattoman autonomisen sukelluslaitteen kyseessä ollessa on haastavaa saada dataa siirrettyä riittävällä nopeudella edullisesti, jotta rannalla oleva ihminen voi tunnistuksen tehdä. Tähän on ratkaisuna konenäkö, ja tässä työssä kehitettävä neuroverkkopohjainen tunnistusratkaisu.

3 TYÖN TUTKIMUSKYSYMYKSET JA METODIT

Tieteellisessä työssä koko tutkimusprosessia ohjaavat tutkimusongelman määrittely ja rajaaminen, joten ongelman määrittely on hyvin tärkeää. Tämän opinnäytetyön tutkimusongelma on ihmisen tunnistaminen veden alta käyttäen kuluttajaluokan viistokaikuluotaindataa. Tunnistamiseen kehitetään erilaisia syviin neuroverkkoihin perustuvia tunnistimia, ja opetetaan kehitettyjä neuroverkkoja kaikuluotaindatalla, sekä testataan opetettujen verkkojen toimintaa. Työhön on otettu myös mukaan datankeräysvaihe, koska datankeräyksen luotettavuus vaikuttaa koko tutkimuksen luotettavuuteen ja toistettavuuteen. (Kananen, 2014, s. 32.)

Opinnäytetyön tutkimuskysymykset ovat seuraavat:

- Voiko kuluttajatason kaikuluotaimen datasta binaariluokittelijalla tunnistaa ihminen satunnaista valintaa tarkemmin.
- Voidaanko pienellä datasetillä opettaa verkko niin, että tunnistus on vielä mahdollista testidatalla, jonka distribuutio poikkeaa merkittävästi opetusdatasta.
- Voidaanko siirto-oppimista hyödyntämällä parantaa neuroverkon tarkkuutta.

Tutkimuksen metodologiat

Työssä ratkotaan käytännön työstä noussutta ongelmaa, ettei vesipelastus ole kehittynyt Suomessa juurikaan, vaikka teknologiat ovat kehittyneet. Työn tueksi ja tekemiseksi on hankittu systemaattisesti ja kriittisesti tietoa, osaamista sekä dataa. Näistä syistä on luontevaa, että opinnäytetyö kirjoitetaan kehittämistyön muotoon. (Ojasalo ym., 2014, s. 18.)

Koneoppiminen ja varsinkin syvien neuroverkkojen tutkimus on edelleen verrattain kehittyvä ala, eivätkä tutkimuksen parhaat toimintatavat ole kovin va-

kiintuneita, ja akateemista kirjallisuutta aiheesta on rajallisesti. Suuri osa parhaiden käytänteiden julkaisuista ovat yritysten julkaisemia ”white paper” -tyyppisiä julkaisuja sekä esimerkiksi blogikirjoituksia. (Serban ym., 2020, s. 2.)

Työssä luodaan konkreettinen koneoppimisen malli. Työn kaikissa vaiheissa tukeudutaan vahvasti tieteelliseen tutkimustietoon, sekä pyritään etsimään ja käyttämään parhaita mahdollisia käytänteitä. Lopputuloksena saadaan aikaan uudenlainen teoreettisesti perusteltu konstruktio eli ratkaisu, joka luo uutta tietoa tiedeyhteisölle, mutta myös pelastushenkilöstön käyttöön. Tämän lisäksi työ tuottaa toisen konkreettisen konstruktion, täysin uudenlaisen pelastustarkoituksiin kohdennetun datasetin muodossa. Tämän kaltaiseen työhön sopii hyvin hyödynnettäväksi konstruktiivisen tutkimuksen metodologia. (Ojasalo ym., 2014, s. 65.)

Ideaalitulos konstruktiivisessa tutkimusotteessa on tosielämän ongelman ratkaisu implementoidulla konstruktiolla. Ainut kontribuutio ei kuitenkaan ole konstruktio, vaan vähintään yhtä tärkeässä roolissa on itse ongelmaratkaisuprosessi ja sen tuottama kontribuutio niin käytännössä kuin teorian kannalta. Tämä on myös tämän työn tavoite. Vahva teoreettinen viitekehys ohjaa tutkimuskysymyksien viitoittamana kohti toimivaa konstruktiota.

Konstruktiivisen tutkimusotteen tyypilliseen tapaan empiirinen toiminta on voimakasta, tutkimus on luonteeltaan kokeellista, ja kehittyessään konstruktio testaa ja peilaa tieteelliseen viitekehykseen. Konstruktiivinen tutkimusote on filosofialtaan pragmaattista ja jokaisen testivaiheen jälkeen tehtävät analyysit kertovat siitä, mikä toimii ja mikä ei toimi sekä tuottavat yhtä lailla yhä jalostetumpaa konstruktiota kuin myös teoreettista ja tieteellistä kontribuutiota. Tehtäviä testejä ei niinkään suunnitella valmiiksi etukäteen, vaan annetaan osatulosten johdattaa kohti valmista konstruktiota. (Hookana-Turunen, 1999, s. 129–151.)

Kaikuluotainkuvien tutkimus on edelleen verrattain pieni ala, joten sen omia menetelmäjulkaisuja on vähän. Häiriö-/signaalisuhteeltaan voidaan ajatella,

että lääketieteen kuvantamisen data on lähellä viistokaikuluotaimen dataa, joten tästä syystä menetelmäoppaiden etsiminen on tässä mielessä perusteltua. Sääntely ja tulosten oikeellisuuden valvonta on lääketieteellisillä aloilla huippuluokkaa. Näistä syistä päädyin käyttämään soveltuvilta osin lääketieteellisen kuvantamisen tutkimuksiin tarkoitettua Checklist for Artificial Intelligence in Medical Imaging (CLAIM) -ohjetta (taulukko 1) (Mongan ym., 2020).

Taulukko 1. CLAIM muistilista, joka on kehitetty, varmistan tutkimuksen oikeellisuus ja toistettavuus. (Mongan ym., 2020)

Checklist for Artificial Intelligence in Medical Imaging (CLAIM)

Section/Topic	No.	Item
TITLE or ABSTRACT		
	1	Identification as a study of AI methodology, specifying the category of technology used (eg, deep learning)
ABSTRACT		
	2	Structured summary of study design, methods, results, and conclusions
INTRODUCTION		
	3	Scientific and clinical background, including the intended use and clinical role of the AI approach
	4	Study objectives and hypotheses
METHODS		
Study Design	5	Prospective or retrospective study
	6	Study goal, such as model creation, exploratory study, feasibility study, noninferiority trial
Data	7	Data sources
	8	Eligibility criteria: how, where, and when potentially eligible participants or studies were identified (eg, symptoms, results from previous tests, inclusion in registry, patient-care setting, location, dates)
	9	Data preprocessing steps
	10	Selection of data subsets, if applicable
	11	Definitions of data elements, with references to common data elements
	12	De-identification methods
	13	How missing data were handled
Ground Truth	14	Definition of ground truth reference standard, in sufficient detail to allow replication
	15	Rationale for choosing the reference standard (if alternatives exist)
	16	Source of ground truth annotations; qualifications and preparation of annotators
	17	Annotation tools
	18	Measurement of inter- and intrarater variability; methods to mitigate variability and/or resolve discrepancies
Data Partitions	19	Intended sample size and how it was determined
	20	How data were assigned to partitions; specify proportions
	21	Level at which partitions are disjoint (eg, image, study, patient, institution)
Model	22	Detailed description of model, including inputs, outputs, all intermediate layers and connections
	23	Software libraries, frameworks, and packages
	24	Initialization of model parameters (eg, randomization, transfer learning)
Training	25	Details of training approach, including data augmentation, hyperparameters, number of models trained
	26	Method of selecting the final model
	27	Ensembling techniques, if applicable
Evaluation	28	Metrics of model performance
	29	Statistical measures of significance and uncertainty (eg, confidence intervals)
	30	Robustness or sensitivity analysis
	31	Methods for explainability or interpretability (eg, saliency maps) and how they were validated
	32	Validation or testing on external data
RESULTS		
Data	33	Flow of participants or cases, using a diagram to indicate inclusion and exclusion
	34	Demographic and clinical characteristics of cases in each partition
Model performance	35	Performance metrics for optimal model(s) on all data partitions
	36	Estimates of diagnostic accuracy and their precision (such as 95% confidence intervals)
	37	Failure analysis of incorrectly classified cases
DISCUSSION		
	38	Study limitations, including potential bias, statistical uncertainty, and generalizability
	39	Implications for practice, including the intended use and/or clinical role
OTHER INFORMATION		
	40	Registration number and name of registry
	41	Where the full study protocol can be accessed
	42	Sources of funding and other support; role of funders

Täysin sellaisenaan CLAIM ei vastaa tämän tutkimuksen tarpeisiin, joten siihen tehdään seuraavia, tarvittavia muutoksia:

8 On muutettu niin, että selvitetään, millä kriteereillä kaikuluotainkuva on otettu datasettiin mukaan.

12 Kaikuluotainkuvasta henkilö ei ole identifioitavissa, kohta ei ole relevantti.

29 Datan määrä ei tule riittämään validiin analyysiin tulosten yleistyvyydestä, joten tätä kohtaa ei työssä toteuteta

34 Ei relevantti tässä kontekstissa.

40 Ei relevantti tässä kontekstissa.

41 Ei relevantti tässä kontekstissa.

Työssä tehtävät testit ja niissä käytettävät datankäsittelyn metodit tarkennetaan työn edetessä vaihe vaiheelta. Työn tekemistä tulee vahvasti ohjaamaan muokattu CLAIM-lista ja työn lopussa pohdintaosuudessa tarkastellaan listan toteutumista.

Varsinaista systemaattista kirjallisuuskatsausta ei työtä varten tehty, mutta työtä varten käytiin läpi merkittävä määrä tieteellisiä julkaisuja, blogitekstejä ja erilaisia ”white paper” -tyyppisiä julkaisuja. Lähteiksi työhön pyrittiin kuitenkin valikoimaan valtaosin vertaisarvioituja kansainvälisiä julkaisuja. Tilanteissa, joissa samasta aiheesta käytiin läpi monia julkaisuja, valikoitiin mahdollisuuksien mukaan ne, joita oli merkittävissä määrin referoitu muissa julkaisuissa, tai ne olivat kyseisen aiheen alkuperäisiä julkaisuja, jossa aihe on ensikertaa julkaistu.

Tietoa haettiin useiden eri tiedonhakuportaalien kautta eri hakutermein. Seuraavassa on esitetty pääasialliset tietolähteet:

- Ieeexplore
- Google Scholar
- Arxiv
- ResearchGate

Mahdollisuuksien mukaan pyrittiin tarkistamaan ainakin kymmenen ensimmäisen hakutuloksen abstraktit.

4 TUTKIMUKSEN LUOTETTAVUUS & HAASTEET

Viime vuosina samalla, kun koneoppimisen ja neuroverkkojen kirjastot ovat kehittyneet yhä helpommin käytettäviksi, on esiintynyt vaaraa siitä, että niitä käytetään ymmärtämättä koko prosessia, ja tuloksista tehdään tahattomasti tai pahimmillaan tahallisesti vääriä johtopäätöksiä. Tulokset esitetään usein epärealistisen hyvinä, eivätkä tulokset ole toistettavissa laboratorio-olosuhteiden ulkopuolella.

Viime aikoina paljon julkisuutta ovat saaneet metatutkimukset, jotka ovat osoittaneet tutkimuksellisia puutteita vertaisarvoiduissa tieteellisissä koneoppimisen julkaisuissa. Esimerkiksi Nature machine intelligence -julkaisusarjassa julkaistiin systemaattinen kirjallisuuskatsaus koneoppisen käytöstä röntgen- ja tietokonetomografiakuvien analysointiin ja potilaan diagnosointiin covid-19-tautitapauksissa. Kirjallisuuskatsauksessa käytiin läpi 2212 tutkimusta, joista 415 läpäisi alkukarsinnan. Tutkimuksen päätelmänä oli, ettei yksikään julkaisu täyttänyt kaikkia kolmea seuraavaa ehtoa:

1. Julkaisu määrittelee toistettavan metodin
2. Metodi toteuttaa parhaita käytänteitä koneoppimisen malleja kehitettäessä
3. Riittävä ulkopuolinen validointi, joka oikeuttaisi metodin laajemman käytön.

Tutkijat löysivät metodologisia ongelmia sekä puolueellisuuksia läpi tutkimusten, johtaen ylioptimistisiin tuloksiin. Samoin tutkimuksessa alleviivataan datasetin tärkeyttä, ja että sen käsittely on tehty oikein, ja siinä tehdyt toimet on myös dokumentoitu. (Roberts ym., 2021, s. 199, 214.)

Kuten esimerkistä voidaan havaita, on edelleen hyvin yleistä, jopa enemmän sääntö kuin poikkeus, että neuroverkkotutkimukset epäonnistuvat tuottamaan tieteellisesti toistettavia ja valideja tuloksia. Tämä ja muut kohtaamani esimerkit ovat vaikuttaneet vahvasti tämän työn tekemiseen ja siihen, että haluan do-

kumentoida hyvin tarkasti tavat, joilla dataa käsitellään, sekä millä parametreilla neuroverkkoja koulutetaan. Tässä osiossa ei käydä datankeruuta ja datasetin luontia läpi, vaan se avataan myöhemmin työssä.

Tässä työssä on tärkeää valita mitattavat arvot oikein, jotta tutkimuskysymyksiin vastataan mahdollisimman luotettavasti. Neuroverkkojen tarkkuutta voidaan mitata useilla eri mittareilla. Ensisijaiseksi mittariksi valittiin testisetin väärien tunnistusten määrä, jotta on yksi selkä numero, jolla verrataan ensisijaisesti onnistumista. Tämä ei kuitenkaan pelkästään anna välttämättä parasta mahdollista kuvaa tilanteesta, joten tämän lisäksi tarkastellaan myös validointidatan virheellisiä tunnistuksia, sekä vähäisissä määrin myös opetusdatan tunnistustarkkuutta, sillä malli, joka toimii näissä kaikissa tilanteissa, toimii myös parhaiten oikeassa maailmassa. Lisäksi tämä eri dataosuuksien tarkastelun tarve tulee myös muokatusta CLAIM-listasta.

4.1 Validiteetti

Neuroverkkojen validiteetin kannalta on tärkeää, että dataa käsitellään niin, että varmistetaan, että opetusdata, validointidata ja testidata pysyvät erillään. Tämä pitää varmistaa myös tehtäessä esimerkiksi datan argumentointia. Augmentoinnilla tarkoitetaan datan määrän lisäystä siihen tehtävillä muutoksilla. Augmentointia avataan myöhemmissä kappaleissa tarkemmin. Augmentointi pitää tehdä niin että argumentointi suoritetaan vasta sen jälkeen, kun datasetit on eriytetty. Muuten on olemassa riski, että hieman eri tavalla argumentoidut kuvat päätyvät kahteen eri osioon kuten opetus- ja validointidataan, tai pahimmassa tapauksessa kaikkiin kolmeen ja tämä opetusdatan vuotaminen testidataan aiheuttaa tuloksen vääristymää antaen epärealistisen hyviä tuloksia.

Datasetin jako opetus-, validointi- sekä testidatasetteihin voi aiheuttaa tulosten vääristymään. Kaikuluotainkuvissa on suurta vaihtelua, kun toiset kuvat ovat huomattavasti vaikeampia tunnistaa kuin toiset. Jopa niin, että osasta on ihmisen vaikea erottaa ihmistä esimerkiksi kivikasasta, kun taas toiset ovat erittäin

selkeitä havaita. Jos validointi- ja testisettiin tulee vähemmän vaikeasti tunnistettavia kuvia, voi se helposti aiheuttaa vääristymää saatujen tulosten ja yleisen tapauksen välille.

4.2 Reliabiliteetti

Käytettäessä neuroverkkoja reliabiliteetti voidaan jakaa kahteen eri osaan, mikä on datan reliabiliteetti ja toisaalta, mikä on neuroverkon koulutuksen toistettavuus. Näitä voidaan parantaa monilla eri tavoilla. Tässä työssä käytetyt metodit on kuvattu alla.

Testattaessa eri neuroverkkoarkkitehtuurien tarkkuuksia sekä haettaessa eri hyperparametreja voidaan poistaa datasetistä aiheutuva vaihtelu, kun datan argumentointi tehdään ennen opetuksia, ja käytetään yhtä argumentoitua datasettiä kaikissa kokeiluissa. Toisaalta argumentointiparametrit voidaan testata opettamalla neuroverkkoja vakioidulla joukolla arkkitehtuureja sekä hyperparametrejä. Argumentoinnissa käytetään myös monia satunnaisuutta sisältäviä muunnoksia dataan, joten myös tällä voi olla vaikutusta tuloksiin. Vaikka erot ovatkin vähäisiä, voidaan tätä satunnaisuutta poistaa tekemällä useampia argumentoituja datasettejä ja ajaa kaikki testit kaikilla eri dataseteillä. Huonona puolena joissain tapauksissa voidaan nähdä tallennetun datan määrä verrattuna siihen, että augmentoitu data luodaan tarvittaessa ja poistetaan ajon jälkeen. Toinen huono puoli on, että kaikki testit joudutaan ajamaan useampaan kertaan, joten tarvittavan laskentatehon määrä kasvaa suoraan suhteessa argumentoituihin datasetteihin. Tässä työssä kokeillaan kaikkia yllä esitettyjä vaihtoehtoja työn eri vaiheissa, kun yritetään selvittää eri oppimisvaikeuksia. Tärkeää on kuitenkin, että testidataan ei tehdä merkittäviä muutoksia, vaan se pidetään koko prosessin ajan mahdollisimman muuttumattomana, sekä säilytetään opetus- ja validointidatan jako samana koko ajan.

Datasetin pienuus aiheuttaa joitain haasteita datan reliabiliteettia eli luotettavuuden kannalta. Tätä on työssä koitettu kompensoida alla esitetyillä tavoilla. Niistä riippumatta, on asioita, jotka pitää huomioida, kun arvioidaan tulosten

yleistettävyyttä, kuten se, että datasetti ei sisällä yhtään kuvaa lapsesta tai nuoresta, joten tunnistuksen toimivuudesta lapsiin ja nuoriin voidaan antaa vain hypoteeseja. Kaikki datankeräys toteutettiin yhdellä ja samalla kaikuluotaimella, joten ei ole varmaa, miten eri kaikuluotaimen käyttö vaikuttaa tuloksiin.

Yllä mainituista syistä, ja koska tulokset eivät ensisijaisesti sellaisenaan mene tuotteeseen vaan tärkeintä on vastata tutkimuskysymyksiin mahdollisimman validisti ja luotettavasti. Tehdään datasetin jako opetus-, validointi- ja testisettiin niin, että ensimmäisenä datankeräyspäivänä kerätty data käytetään opetukseen ja validointiin niin, että 70 % datasta käytetään opetukseen ja 30 % validointiin. Toisena päivänä kerätty data käytetään testaukseen. Tämä poikkeaa yleisimmin käytetystä 70 %, 20%, 10% jaottelusta, ollen lähempänä 35%, 15%, 50%. Tämä tulee todennäköisesti heikentämään tarkkuutta, mutta lisää tulosten yleistettävyyttä.

Kuvien jako opetus- ja validointidataan tapahtuu satunnaisesti jaottelemalla pythonscriptin avulla, käyttäen satunnaisesti otettua siemenlukua. Tämä jako tullaan tekemään hyvin alkuvaiheessa pysyvästi, ja sen jälkeen kaikki testit ajetaan samalla jaolla.

4.3 Eettiset kysymykset

Syvien neuroverkkojen käyttöön liittyy monia eettisiä kysymyksiä ja niiden tuotoksia voidaan käyttää ja soveltaa moniin tarkoituksiin, joita alkuperäiset kehittäjät eivät välttämättä ole lainkaan ajatelleet. Kaikuluotainten käyttöön liittyvät vahvasti myös sotilaalliset ulottuvuudet. Monet armeijoihin liittyvät valtiolliset tahot rahoittavat vedenalaisten autonomisten laitteiden kehitystä. Yleistyessään kuvatun kaltaiset järjestelmät voivat kuitenkin potentiaalisesti pelastaa ihmishenkiä. Siinäkin tilanteessa, jossa uhria ei ajoissa ehditä pelastaa, on hukkuneen löytäminen kuitenkin omaisille tärkeää.

Erityisesti EU-alueella viime vuosina datan käyttöä on alettu sääntelemään enenevissä määrin. Kaikuluotainkuvat ovat kuitenkin niin heikkolaatuisia, ettei niistä ole mitenkään mahdollista tunnistaa yksittäisiä ihmisiä, joten tässä mielessä data ei ole läheskään niin henkilökohtaista kuin esimerkiksi röntgenkuvat ja vastaavat muut lääkinnällisen kuvantamisen datat, joista ihminen on mahdollista tunnistaa. Kun järjestelmä tulee käyttöön oikeaan tilanteeseen, esimerkiksi ruumin etsintään, pysyy kaikki data järjestelmässä, ja laskenta tapahtuu itse laitteessa. Laite on tarkoitus ottaa käyttöön pelastusalan ammattilaisilla, jotka ovat tottuneet toimimaan onnettomuuksien uhrien kanssa.

5 TYÖN VIITEKEHYS

Teoriaosuudessa käydään läpi tärkeimpiä tässä työssä käytettäviä tekniikoita, algoritmeja sekä arkkitehtuureja, luoden sitä teoriapohjaa, johon neuroverkkotunnistimen konstruktiivisen tutkimuksen toteutus ja testaus tukeutuu. Konenäkö ei ole kovin uusi asia automaattisen tietojenkäsittelyn historian aikajänteellä, mutta viimeisen kymmenen vuoden aikana se on kokenut täydellisen muutoksen. Noin kymmenen vuotta sitten Yann LeCu kumppaneineen yrittivät julkaista konenäön julkaisua yhdessä tärkeimmistä konferensseista, ja se hylättiin sillä perusteella, että se käytti neuroverkkoja. Vielä siihen aikaan oli tiedeyhteisössä hyvin vankka uskomus, että konenäön pitää perustua ihmisen opettamiin kuvan piirteisiin. Valtaosa kehityksestä suuntautui näihin niin kutsuttuihin asiantuntijajärjestelmiin (expert systems). Vain muutamassa vuodessa tämä näkemys on kääntynyt ja nykyään tutkimus keskittyy hyvin vahvasti syviin neuroverkkoihin, vaikkakin edelleen valtaosa teollisuuden sovelluksista tukeutuu perinteisen konenäön sovelluksiin. Neuroverkot ovat kehittyneet valtavin harppauksin, kehitystä on tapahtunut niin algoritmien kehittymisessä, ohjelmistotyökalujen tarjonnassa kuin rautatason kehityksessä. Tässä suurena muutoksena on ollut näytönohjainlaskennan eli GPU-laskennan käyttö neuroverkkojen opetuksessa. Yksi merkittävä tekijä on myös ollut suurien valmiiden datasettien julkaisu, ja niiden ympärille rakennetut kilpailut. (Almaghrabi ym., 2014; Brief History of Deep Learning from 1943-2019 [Timeline] - MLK - Machine Learning Knowledge, n.d.; Goodfellow Ian ym., 2016, s. 438–441; Krizhevsky ym., 2017, s. 1.)

5.1 Ohjattu oppiminen

Ohjatun oppimisen tärkein määritelmä on, että opetustilanteessa vastaukset ovat tiedossa. Tämä tarkoittaa esimerkiksi kuvien tapauksessa, jos niitä yritetään luokitella eri luokkiin, että nämä luokat ovat tiedossa. Toisin sanoen, on olemassa joku oppimisfunktio f(x), joka tuottaa tuloksen y. Opetuksessa käytetään oikeat vastauksen sisältäviä opetusesimerkkejä (x,y), jossa x on esimerkki, ja y on oikea vastaus. Koneoppimisen algoritmeja on paljon muitakin,

kuin ohjatun oppimisen algoritmit, mutta käytännössä suurin osa tämän hetken hyödyllisistä käyttökohteista käyttävät ohjatun oppimisen algoritmeja. Tämän työn koneoppimismallit koulutetaan ohjattua oppimista hyödyntäen. (Ng, 2018, s. 9.)

5.2 Neuroverkot

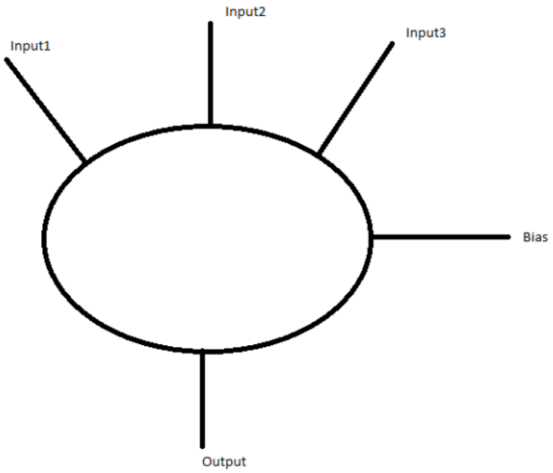
Neuroverkot ovat yksi koneoppimisen muodoista. Neuroverkolla koitetaan jossain määrin matkia, miten laskenta tapahtuu biologisissa järjestelmissä kuten aivoissa. Neuroverkko koostuu neuroneista, joka ovat neuroverkon pienin osa. Neuronilla voi olla n kpl sisääntuloja (input), sekä paino (weight) jokaista sisääntuloa kohti, useimmissa tapauksissa vakiotermi (bias), ja lähes poikkeuksetta yksi ulostulo, joka voi olla kytkettynä useisiin muihin neuroneihin. Neuronien ulostulo voidaan laskea kaavalla (1):

$$u_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{1}$$

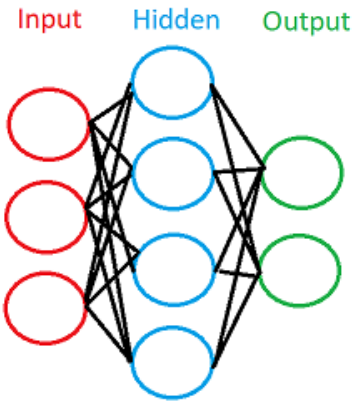
Kaavassa U_k on neuronin ulostulo, j on neuronin numero, W_{kj} neuronin paino, x_j neuronin sisääntulon arvo ja b_k on vakiotermi. Kuviossa 3 esimerkissä on yksi neuroni, jolloin ulostulo on ylemmän kaavan (kaava1) esimerkillä laskettavissa:

Output = input1*weight1+ input2*weight2+ input3*weight3+bias

Yksittäiset neuronit muodostavat kerroksen (layer), kuten kuviossa 4 voidaan nähdä. Eri kerrokset on merkitty eri väreillä. Tässä esimerkissä on kolmekerroksinen neuroverkko, jossa on yksi sisääntulokerros, yksi kätketty (hidden) kerros, sekä ulostulokerros.



Kuvio 3. Neuroni kolmella sisääntulolla



Kuvio 4. Neuroverkko

Kerrokset voivat olla hyvin erilaisia, kuvassa 3 näkyvät kerrokset ovat täysin kytkettyjä kerroksia (dense layer), mikä tarkoittaa, että jokaiseen kerroksen neuroniin ovat kytkeytyneet kaikki aikaisemman kerroksen neuronit, ja seuraavan kerroksen neuronit. (Alzubaidi ym., 2021, s. 17–20.)

5.3 Aktivointifunktiot

Aikaisemmasta neuronin kuvauksesta puuttui vielä useimmiten käytetty vaihe, aktivointifunktion (activation fuction) käyttö. Aktivointifunktion tarkoitus on useimmissa tapauksissa tuoda neuroverkkoon epälineaarisuutta. Syvästä neuroverkosta ei haluta lineaarista, koska monimutkainenkin lineaarinen kombinaatio on edelleen lineaarinen, eli toisin sanoen lineaarinen syvä neuroverkko voidaan redusoida yhteen kerroksen neuroverkkoon, joka saavuttaa samat tulokset. Tästä syystä epälineaarisuus on tärkeää, ja yksi sen lähteistä on aktivointifunktio. (Alzubaidi ym., 2021, s. 17–18.)

Aikaisemmin käytettiin monia erilaisia aktivointifunktioita, kunnes osoitettiin, hieman jopa yllättäen, että hyvin yksinkertainen funktio selviää monista tilanteista lähes yhtä hyvin tai paremmin kuin monimutkaisemmat, ja lisäksi on laskennallisesti hyvin kevyt. Tämän aktivointifunktion nimi on Relu (rectified linear unit). Nykyään voidaankin yksinkertaistettuna sääntönä sanoa, että mikäli ei ole mitään perusteltua syytä käyttää muuta aktivointifunktiota, niin käytetään

Relua. Yksinkertaisuudessaan Relun aktivointi on nolla negatiivisella alueella, ja lineaarinen nollan positiivisella puolella, sama esitettynä kaavassa2 alla:

$$y(x) = \begin{cases} x, & \text{kun } x \geq 0 \\ 0, & \text{kun } x < 0 \end{cases} \quad (2)$$

Kaavassa3 y on ulostulo, ja x aktivointi funktion sisääntulo, eli y arvolla x on x silloin kun x on nolla tai suurempi, ja muussa tapauksessa nolla. Kaikissa tämän työn neuroverkkoarkkitehtuureissa käytetään Relu-aktivointifunktiota. (Glorot ym., 2011, s. 4–8; Nair & Hinton, 2010, s. 7–8.)

Yksi huomattava poikkeus tilanteeseen, jossa usein halutaan käyttää jotain muuta aktivointifunktiota kuin Relua, on ulostulokerros. Ulostulokerroksessa käytetään laajasti eri funktioita, riippuen siitä millaista ongelmaa yritetään ratkaista. Yksi esimerkki tämänkaltaisesta funktiosta on Softmax. Mitä Softmax tekee ulostulolle, on, että se muuttaa kerroksen todennäköisyysjakaumaksi. Toisin sanoen kerrosten yhteenlaskettu aktivointi on aina 1, eli 100 %, ja tämä jakautuu eri ulostuloneuroneille sisääntulon ja verkon painojen mukaan. Alla on esitettynä Softmax-funktion kaava (kaava3).

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{c} e^{x_j}} \quad (3)$$

Kaavassa y_i on ulostuloarvot, eli laskettu normalisoitu todennäköisyysjakauma, x_i sisään tulevat neuronit, x_j on sisään tulevien neuronien summa. Tämän työn neuroverkoissa on ulostulokerroksen aktivointifunktiona Softmax. (Bridle, 1989.)

5.4 Ylioppimisen välttäminen

Ylioppiminen (over fitting) tarkoittaa tilannetta, jossa neuroverkko oppii liian hyvin ulkoa opetusdatan, muttei opi opetettavaa konseptia. Monet erilaiset syyt voivat johtaa ylioppimiseen kuten liian pieni datasetti, liian suuri neuroverkko, liian pitkällinen opettaminen. Datasetin kokoon ei usein voida vaikuttaa, ja siitä johtuvaa ylioppimista pitää usein vähentää muilla keinoin kuin datan määrää kasvattamalla. Tämä on tilanne myös tässä työssä, kun rajalliset resurssit

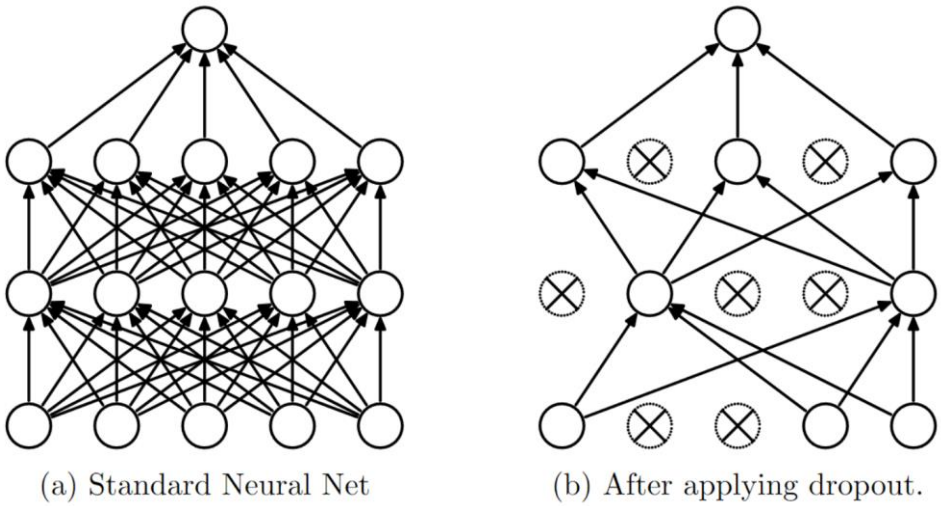
määrittivät datasetin kokoa. Myöhemmissä kappaleissa käydään läpi, miten neuroverkon koko vaikuttaa ylioppimiseen, sekä mitkä ovat siirto-oppimiseen liittyvä vaikutukset ylioppimiseen. Muita tapoja tarkastellaan seuraavaksi. (Brigato & Iocchi, 2020, s. 7–8; Szegedy ym., 2016, s. 1.)

Datan argumentoinnilla tarkoitetaan datan määrän synteettistä kasvattamista. Dataan voidaan tehdä esimerkiksi erilaisia käännöksiä. Tässä pitää huomioida, että useat käännökset pois lukien peilikuvat ja ylösalaisin kääntäminen sekä 90 asteen käännös neliökuvan tapauksessa, muuttavat jollain tavalla kuvan mittasuhteita, joko osa kuvasta menetetään, tai kuvaa on pienennettävä, ja tällöin kuvan reunoilla olevat pikselit pitää täyttää jollain tavalla. Usein tämä tehdään joko staattisilla arvoilla, tai kuvasta jotenkin generoimalla. Joissain tapauksissa myös tämän kaltaiset muunnokset saattavat johtaa vääriin tunnistuksiin, kuten jos opetetaan numeroita, niin sopivilla käännöksillä numerot kuusi ja yhdeksän ovat sama asia. Kuten tässä esimerkissä, aina on tärkeä ymmärtää data, ja mitkä muutokset ovat hyödyllisiä ja mitkä pahimmassa tapauksessa jopa haitallisia oppimiselle. Kuvan koon muutokset voivat olla myös itsetarkoitus, eikä pelkästään pakollinen paha, joka tulee kuvan käännöksistä. Kuvan kirkkautta voidaan myös muuttaa. Tässä on myös tärkeää ymmärtää datan luonne, ja eri värikanavien suhteet kyseisellä datalla, jotta augmentoitu data säilyy datan domainille tyypillisenä. Kuvaan voidaan myös lisätä kohinaa. Tässä yksi tyypillinen tapa on käyttää normaalijakautunutta kohinaa. Kuvien tapauksessa pitää päättää onko kohina saman arvoista kaikille kuvan kanaville, vai annetaanko arvot erikseen joka kanavalle. (Howard, 2013, s. 1–6; Krizhevsky ym., 2017, s. 4.)

5.5 Dropout

Modernien neuroverkkojen kyky oppia monimutkaisia riippuvuuksia on valtava, kun neuroverkossa on monia epälineaarisia kerroksia. Valtaosa näistä riippuvuuksista, joita verkko koulutuksessa oppii, on kuitenkin lähinnä opetusdatan kohinaa ja muita epäolennaisia piirteitä, jotka ovat joko turhia tai jopa haitallisia opetusdatan ulkopuolisille tunnistuksille jopa siinä tilanteessa, että

testidata on samasta lähteestä kuin opetusdata ja samalla jakaumalla. Tämän ongelma ratkaisemiseksi yksi käytetty metodi on dropout. Tässä satunnaisesti joka kerta kun verkko näkee dataa, jätetään joku prosentti neuroneista ja niiden yhteyksistä aktivoimatta kuten kuviossa 5. Tällä on kaksi etua; 1. mikään verkon neuroneista ei vahvistu korvattoman vahvaksi ja 2. opetusdatan kohi nan vaikutus vaimenee.



Kuvio 5. Dropoutin vaikutus neuroverkon toimintaan voidaan havaita, kun neuronien välisten yhteyksien määrä opetuskerralla laskee merkittävästi (Srivastava ym., 2014, s. 2)

Dropout tehdään pelkästään opetusvaiheessa. Validointi- ja testivaiheessa koko verkko on käytössä, mutta opetuksen kummassakin vaiheessa, kun tieto syötetään verkossa eteenpäin, ja takaisinkulkeutumisvaiheessa (Backpropagation), kun verkon uusia painoja lasketaan. Tässä työssä testataan Dropoutin vaikutusta verkon kykyyn oppia ja sen vaikutusta ylioppimiseen eri neuroverkkoarkkitehtuureilla. (Srivastava ym., 2014, s. 1–3.)

6 NEUROVERKKOARKKITEHTUURIN VALINTA

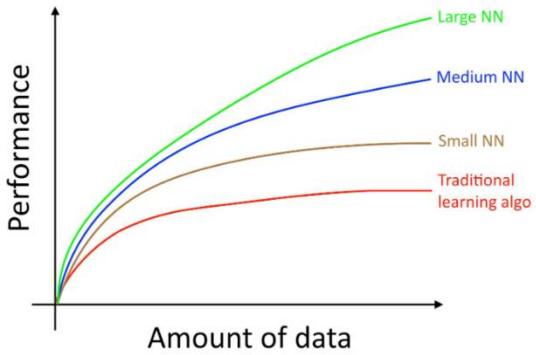
Nykyään yleisimmin robotiikassa objektien tunnistuksessa käytetään ”You only look once” (YOLO) algoritmia. Yolo ja vastaavat algoritmit vaativat kuitenkin valtavan määrän opetusdataa, ja esimerkiksi yleisesti käytetty, Microsoftin valmistama datasetti COCO, sisältää yli 2 500 000 kuvaa. COCO datasetin tekoon käytettiin yli 70 000 henkilötyötuntia, vaikka tämän kaltaisia normaaleja kamerakuvia on helposti saatavilla. (Grönman ym., 2021; Lin ym., 2014.)

Kaikuluotainkuvien datasettejä ei ole juuri ollut julkisesti saatavilla, ja niiden valmistus on erittäin aikaa vievää. Tässä työssä kerätyn alle 700 kuvan datasetin keräys, valmistus ja käsittely vei yli 200 henkilötyötuntia. Aivan viime aikoina on jotain kohtalaisen pieniä datasettejä ja niillä opetettuja pieniä verkkoja tullut saataville. Nämä ovat kooltaan tuhansia kuvia, joten tämänkaltaiset datasetit eivät riitä YOLO-tyyppisten algoritmien opetukseen. (Valdenegro-Toro ym., 2021.)

Datasetin koon rajoitukset johtivat siihen, että oli järkevämpää käyttää neuroverkkoa objektin tunnistukseen, niin että kuvan yli tehdään niin sanottua liukuvan ikkunan (sliding window) tunnistusta. Siinä pientä osaa kuvasta tunnistetaan kerrallaan ja tätä pientä ikkunaa siirretään kuvan yli. Näin saadaan tunnistuksen paikka kuvassa selville. Tämä soveltuu hyvin käytettäväksi viistokaikuluotainkuvaan, josta päivittyy vain yksi vaakataso kerrallaan. Näin ikkunaa pitää siirtää vain vaakatasossa. Tämä vaatii huomattavasti vähemmän dataa sekä toimii huomattavasti pienemmällä laskentateholla kuin YOLO, koska neuroverkkototeutus perustuu pohjimmiltaan yksinkertaisempaan luokittelijaan. (Redmon ym., 2016.)

Tässä työssä käytettävät neuroverkkoarkkitehtuurit on valittu sen perusteella, mitkä on todettu toimiksi muissa vastaavissa kohinaisien kuvien tunnistamistehtävissä. Myös käytettäviä hyperparametreja on etsitty kirjallisuudesta. Pienellä datasetillä neuroverkon koolla ei välttämättä ole niin suurta merkitystä

kuten kuviosta 6 voidaan havaita. (Brigato & Iocchi, 2020, s. 7; Ng, 2018, s. 11; Valdenegro-Toro ym., 2021.)



Kuvio 6. Neuroverkon koon vaikutus suorituskykyyn, neuroverkon suuremman koon hyödyt tulevat esiin vasta datamäärän kasvaessa (Ng, 2018, s. 11)

Havaittavia kategorioita on kuitenkin työssä käytetyn datan tapauksessa niin vähän ja datasetin koko niin pieni, että testeihin päätettiin ottaa myös huomattavasti pienempiä konvoluutioverkkoja vain muutamalla kerroksella, jotta voidaan testata toimivatko ne luotettavammin pienellä datasetillä.

6.1 Työssä hyödynnetyt neuroverkot

ResNet

Yleisesti mitä syvemmäksi neuroverkko tehdään, sitä rikkaampia piirteitä verkon on mahdollista oppia. Alkuun syvät neuroverkot kärsivät helposti häviävän gradientin ongelmasta, mikä johtaa vaikeuksiin verkon opetuksessa. Yksi ratkaisu tähän on ollut residuaaliset neuroverkot. Tallaisen neuroverkon kerrokset eivät ole yhdistettynä pelkästään edelliseen ja seuraavaan kerrokseen, vaan sisältävät myös oikopolun aikaisempaan kerrokseen. Alun perin residuaaliset neuroverkot otettiin käyttöön 2015, jolloin arkkitehtuuria käyttävä neuroverkko voitti COCO 2015 -kilpailun. ResNet on tähän työhön valituista neuroverkoista raskain ja siinä on eniten parametreja. (He ym., 2016.)

MobileNet

MobileNet-verkot on kehitetty laskentatehokkaiksi verkoiksi, tarkkuuden ollessa kuitenkin edelleen kohtalaisen hyvä. MobileNet-verkot kuuluvat konvoluutioverkkoihin, joissa tehtävien kertolaskujen määrää on vähennetty, jakamalla konvoluutio syvyyssuuntaiseen ja pisteittäiseen konvoluutioon. Tämä tietysti vähentää myös oppivien parametrien määrää, joten nopeuden hinta on useimmissa tapauksissa hieman laskenut tarkkuus. MobileNet on tässä työssä käytetyistä verkoista kolmanneksi raskain parametreilla mitattuna. (Howard & Wang, 2012.)

DenseNet

DenseNet-verkko yhdistää monia residuaalisen verkon ja MobileNetin ominaisuuksia. DenseNetissä on konvoluutiokerroksien joukkoja, joissa jokainen on kytketty jokaiseen kerrokseen, jossa on samankokoinen piirrekartta (feature-map). Tämän kaltainen verkko tarvitsee vähemmän parametrejä kuin perinteinen konvoluutioverkko, ja kerrosten väliset yhteydet madollistavat huomattavan syviä neuroverkkoja ilman katoavan gradientin ongelmaa. DenseNet on työssä käytetyistä neuroverkkoarkkitehtuureista toiseksi raskain parametreilla mitattuna. (Huang ym., 2017.)

Pienet perinteiset konvoluutioverkot

Kuten kuviosta 6 nähtiin, pienellä datamäärällä suuresta verkosta ei välttämättä ole juuri hyötyä. Tämän vuoksi tässä tutkimuksessa haluttiin kokeilla myös pienempiä neuroverkkoja. Toinen niistä on hyvin perinteinen konvoluutioverkko 3x3-kerneillä. Toisessa taas on hieman enemmän neuroneita ja sen kaksi ensimmäistä konvoluutiokerrosta on 5x5-konvoluutiota. Taulukoista 2 ja 3 voidaan nähdä näiden kahden tätä työtä varten kehitetyn neuroverkon arkkitehtuurit.

Taulukko 2. Työtä varten kehitetyn pienimmän neuroverkon, Konv3Verkon, arkkitehtuuri Tensorflow model summary -muodossa.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 118, 118, 32)      2432

 activation (Activation)     (None, 118, 118, 32)      0

 dropout (Dropout)           (None, 118, 118, 32)      0

 max_pooling2d (MaxPooling2D  (None, 59, 59, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 55, 55, 64)        51264

 activation_1 (Activation)   (None, 55, 55, 64)        0

 dropout_1 (Dropout)         (None, 55, 55, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 27, 27, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 25, 25, 128)       73856

 activation_2 (Activation)   (None, 25, 25, 128)       0

 dropout_2 (Dropout)         (None, 25, 25, 128)       0

 max_pooling2d_2 (MaxPooling  (None, 12, 12, 128)      0
 2D)

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 64)                1179712

 activation_3 (Activation)   (None, 64)                0

 dropout_3 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 2)                 130

 activation_4 (Activation)   (None, 2)                 0

=================================================================
Total params: 1,307,394
Trainable params: 1,307,394
Non-trainable params: 0
_________________________________________________________________
None
```

Taulukko 3. Työtä varten kehitetyn toiseksi pienimmän neuroverkon, Konv5Verkon, arkkitehtuuri Tensorflow model summary -muodossa.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 118, 118, 32)      2432

 activation (Activation)     (None, 118, 118, 32)      0

 dropout (Dropout)           (None, 118, 118, 32)      0

 max_pooling2d (MaxPooling2D  (None, 59, 59, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 55, 55, 64)        51264

 activation_1 (Activation)   (None, 55, 55, 64)        0

 dropout_1 (Dropout)         (None, 55, 55, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 27, 27, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 25, 25, 128)       73856

 activation_2 (Activation)   (None, 25, 25, 128)       0

 dropout_2 (Dropout)         (None, 25, 25, 128)       0

 max_pooling2d_2 (MaxPooling  (None, 12, 12, 128)      0
 2D)

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 64)                1179712

 activation_3 (Activation)   (None, 64)                0

 dropout_3 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 2)                 130

 activation_4 (Activation)   (None, 2)                 0

=================================================================
Total params: 1,307,394
Trainable params: 1,307,394
Non-trainable params: 0
_________________________________________________________________
None
```

Vaikka laajempi konvoluutiokerneli, 5x5, on laskennallisesti pienempää 3x3-kenneliä suurempi, ja tietyissä tilanteissa jopa moninkertainen, voi sen kyky oppia rikkaampia piirteitä kuvasta olla kuitenkin tuon laskentatehon tarpeen arvoinen. (Szegedy ym., 2016, s. 6.)

6.2 Siirto-oppiminen

Monet syvien neuroverkkojen käytön pullonkaulat johtuvat opetusdatan pienestä määrästä. Datan hankkiminen on yksinkertaisesti hankalaa tai kallista. Siirto-oppimisessa käytetään eri tapoja hyödyntää oppimista, mikä on saavutettu jossain toisessa domainissa tai eri datasetillä. Erilaiset siirto-oppimisen muodot ovat edelleen aktiivisen kehityksen kohde. Tässä työssä kerätty ja käytetty datasetti on niin pieni, että on tarpeen testata, voidaanko verkon oppimista parantaa erilaisten siirto-oppimistapojen avulla. (Tan ym., 2018.)

6.2.1 Verkkopohjainen siirto-oppiminen

Verkkopohjaisessa siirto-oppimisessa käytetään suoraan valmiiksi koulutettua neuroverkkoa, joka on opetettu jollain toisella datasetillä ja jonkun muun toimesta tai aikeisemmin johonkin muuhun tehtävään. Verkon neuronien painot voidaan osittain lukita tai uudelleen kouluttaa koko verkkoa domaindatalla. (Tan ym., 2018, s. 6–7.)

Tärkeää tämän tyyppisessä oppimisessa on etsiä domain, jossa opitut piirteet ovat saman tyyppisiä. Esimerkiksi kaikuluotaintapauksessa joko joku muu kaikuluotaimilla opetettu verkko tai esimerkiksi lääketieteellisen kuvantamisen tekniikat tuottavat usein saman tyylistä kohinaista kuvaa.

6.2.2 Datapohjainen siirto-oppiminen

Datapohjaisessa siirto-oppimisessa ei käytetä valmiiksi koulutettuja verkkoja, vaan otetaan pelkästään data, joka on kerätty johonkin toiseen tarkoitukseen. Datapohjainen siirto-oppiminen voidaan myös jakaa kahteen eri kategoriaan

riippuen siitä, pidetäänkö datasetit erillään vai yhdistetäänkö ne yhdeksi suuremmaksi datasetiksi. Mikäli datasetit pidetään erillään, tehdään koulutus ensin siirto-oppimista varten etsityllä datasetillä, ja sen jälkeen koulutetaan itse päädatalla. Mikäli datasetit yhdistetään, tehdään koulutus vain kerran. Tässä tapauksessa pitää myös harkita, muutetaanko myös testidatasettiä niin, että siihen lisätään myös siirto-oppimisen datasetin testidata, vai käytetäänkö siirtodataa pelkästään opetus- ja validointiseteissä. Tässä pitää huomioida myös, miten toimitaan kategorioiden määrän kanssa, eli mikä on neuroverkon viimeisen kerroksen neuronien määrä.

Tässä työssä testataan datapohjaista siirto-oppimista erillisellä datasetillä. Siirto-oppiminen tehdään niin, että koulutus tehdään ensin tällä erillisellä datalla, jonka jälkeen ulostuloneuronien määrä muutetaan kaikuluotaindatasetille sopivaksi.

6.3 Optimaalinen virhearvo

Suurimmassa osassa ongelmia ei voida saavuttaa 100 %:n oikeellisuutta. Usein ei ole edes selvää, miten lähelle tätä arvoa voidaan päästä. Yksi hyödyllinen mittari on optimaalisen virheen käsite, jos sellainen on saatavilla. Tämä tarkoittaa usein sitä tarkkuutta, jolla ihminen voi tehtävästä suoriutua. Esimerkiksi kaksi normaalikuuloista ja samaa kieltä äidinkielenään puhuvaa keskustelevat hiljaisessa ja rauhallisessa tilassa tutuista asioista. Tällöin voidaan odottaa, että optimaalinen virhearvo on pieni, toisin sanoen keskustelijat todennäköisesti kuulevat ja ymmärtävät toisiaan. Jos sama keskustelu siirretään tehdasympäristöön, jossa on paljon melua, ja stressiä aiheuttavia tapahtumia, ja ihmiset keskustelevat muulla kielellä kuin äidinkielellään, saman keskustelun ymmärtämisessä ihmisellä on paljon suurempi todennäköisyys tehdä virheitä. Tämän kaltaisissa tilanteissa optimaalisen virhearvon määrittää se, mistä pohjatotuus saadaan. Tällä tarkoitetaan sitä, mistä datasetin y-arvot kerätään ja tarkistetaan. Mikäli ne voidaan kerätä tai tarkastuttaa ennen tai jälkeen tilanteen alkuperäiseltä puhujalta, niin optimaalinen virhe on todennäköisesti pieni. Jos taas pitää rakentaa y-arvot kuulijan näkökulmasta, sisältää eri

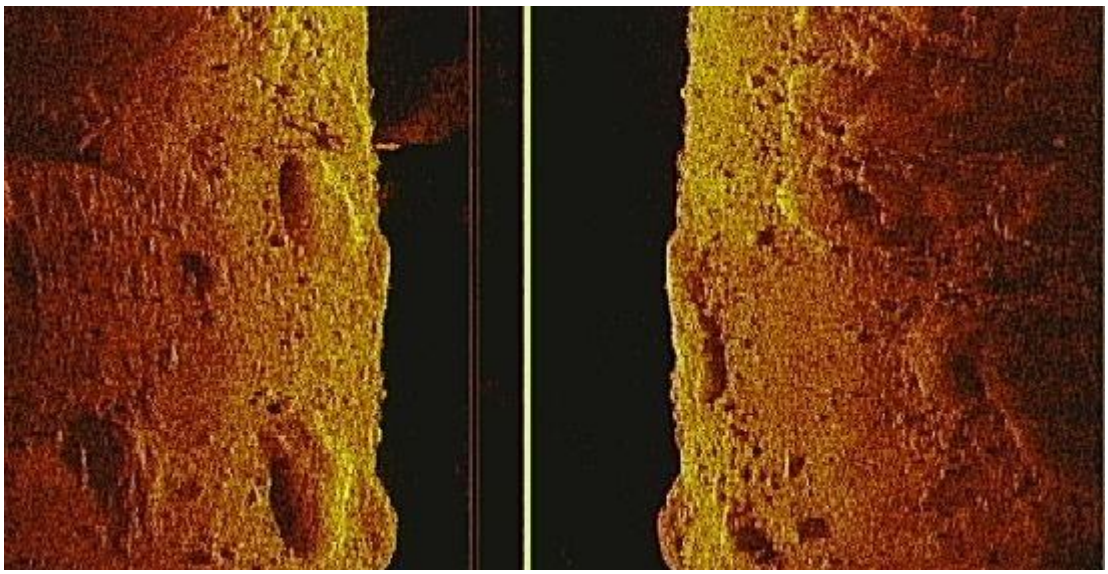
skenaarioissa y merkittävästi eri määrän virheitä. Tällöin myös optimaalinen virhearvo on eri. Ohjatussa oppimisessa on vaikea päästä parempiin tuloksiin, kuin opetusdatan tarkkuus. (Goodfellow ym., 2016, s. 417; Ng, 2018, s. 46–48.)

Optimaaliseen virheeseen vaikuttaa myös vahvasti se, onko opetus-, validointi- ja testidatassa sama jakauma ja löytyykö validointi- tai testidatasta sellaisia variaatioita, joita opetusdatasta ei löydy. Jossain määrin neuroverkoilla on mahdollista ymmärtää konseptia ja yleistää tunnistus myös näkemättömiin esimerkkeihin, mutta tälle on rajansa varsinkin pienellä datamäärällä ja suhteellisen pienillä neuroverkoilla. Useimmiten on suositeltavaa, että opetus-, validointi- ja testidata ovat samanlaisella jakaumalla, ja samankaltaista. Tässä on kuitenkin ongelmana, että tällöin on hankala tehdä päätelmiä siitä, miten hyvin luokittelija tulee toimimaan oikeassa ympäristössä, mikäli datasetti on suppea, eikä sisällä kaikkia oikeassa tilanteessa vastaantulevia muutoksia. Tämän työn optimaalista virhearvoa tarkastellaan seuraavassa kappaleessa datasetin keräyksen ja kehityksen yhteydessä. (Ng, 2018, s. 17, 71.)

7 DATAN HANKINTA KEHITTÄMISTYÖSSÄ

Kaikuluotainkuvan analysointi käyttäen perinteisiä konenäön menetelmiä on hankalaa, jollei jopa mahdotonta. Perinteiset konenäköjärjestelmät toimivat hyvin staattisissa olosuhteissa, jossa kohinaa on vähän, ja tunnistettavan kohteen muutokset ovat pieniä.

Kaikuluotainkuva on kuvattujen ominaisuuksien vastakohta. Kuva on matalaresoluutiosta ja ihmisen muoto hyvin vaihtelevaa, riippuen ihmisen koosta, vaatetuksesta sekä asennosta, joskin asento on tajuttomalla useimmiten raajat suorina pohjaa vasten, keskiruumis hieman pohjan yläpuolella, vatsa kohti pohjaa. Myös veden syvyys ja pohjan muoto ja materiaali vaikuttavat vahvasti kuvaan. Kuten kuvasta 1 voidaan nähdä, on uhrin löytäminen viistokaikuluotainkuvasta vaikeaa usein ainakin kokemattomalle ihmissilmälle.



Kuva 1. Viistokaikuluotainkuva kerätystä datasta, jossa pelastussukeltaja on merenpohjassa.

7.1 Datankeruu neuroverkon opetusta varten

Data kerättiin Satakunnan pelastuslaitoksen veneellä, johon kiinnitettiin Garmin 8400 Xsv -kaikuluotain sekä GT54UHD-TM-viistokaikuanturi (kuva2). Anturin syvyyttä voitiin säätää anturikiinniketolpan korkeutta säätämällä, jolloin

voitiin kerätä eri syvyyden dataa, myös samoista paikoista. Puomin ollessa lähellä maksimisyvyyttä aiheutti se veden vastuksesta anturiin huojuntaa, joka saattoi hieman laskea datan tarkkuutta. Tämä kuitenkin todennäköisesti simuloi hyvin nopeampia liikkeitä, joita huomattavasti pienemmässä vedenalaisessa robotissa on, kun se pyrkii pitämään suuntimaa ja nopeutta vakiona. Kaikuluotaimen kuva, sekä muu anturidata kerättiin ROSBAG-muodossa kannettavalle tietokoneelle, joka toimi ROS-masterina. Datasetin keräämiseen käytettiin noin 100 henkilötyötuntia, joista pelastussukeltajien osuus oli noin 30 tuntia.

Kuva 2. Datasetin keräys Rauman edustalla

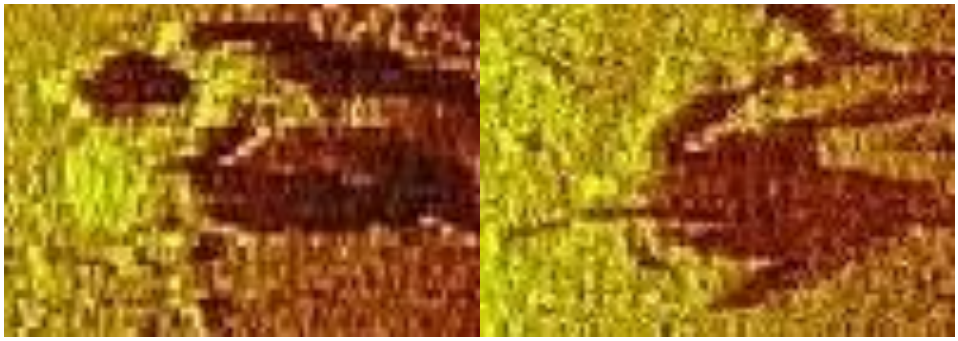
Kaikissa kuvissa sukeltajalla on märkäpuku sekä kasvomaski. Suurimmassa osassa kuvista sukeltajan happipullot ovat niin, että ne eivät näy kuvassa. Dataa kerättiin kahtena päivänä merellä Rauman edustalla. Ensimmäisen päivän data on Maanpäännokan alueelta ja toisen päivän data Syväraumanlahden alueelta. Molempien päivien datassa on kuvia eri kohdista pohjaa eri asennoissa kummastakin sukeltajasta. Sukeltajilla on huomattava pituusero. Ensimmäisen päivän kuvissa molemmilla sukeltajilla on räpylät. Toisen päivän kuvissa ei ole räpylöitä.

Datasettiin on jätetty myös kaikki huonolaatuisimmat kuvat, joista ihmisellä on suuria vaikeuksia tehdä tunnistusta. Ground Truth -tieto saatiin näistä kuvista, koska tiedettiin, että siinä ihminen on aikaleiman ja paikan perusteella. Muuten

osa olisi varmasti jäänyt tunnistamatta. Näin huonolaatuisten kuvien ottaminen mukaan datasettiin varmasti huonontaa tunnistustarkkuutta, mutta antaa realistisemman kuvan todellisesta tunnistusprosentista, kun käytetään edullisia kuluttajaluokan kaikuluotaimia.

7.2 Datan valmistelu neuroverkon opetukseen

Datan valmisteluvaiheessa BAG-tiedostot käytiin manuaalisesti läpi ja tarkastettiin ja samalla kirjattiin ylös aikaleimat kohdista, joissa näkyy ihminen. Tämän jälkeen valmistettiin python-ohjelma, joka käy BAG-tiedostot läpi ja tallentaa kuvat niistä aikaleiman kohdista, joissa on ihminen. Tämän lisäksi katsottiin data läpi vielä toisen kerran. Tällä kierroksella etsittiin kaikki ne aikaleimakohdat, joista löytyy jotain muita objekteja, joita voidaan käyttää neuroverkon opetuksessa negatiivisten tapausten opetukseen. Tallaisia kuvia olisi tietysti voinut taltioida datasta paljon enemmänkin, mutta positiivisten ja negatiivisten tapausten suhde haluttiin pitää kohtalaisen lähellä yhtä. Tämän jälkeen kuvat olivat muodossa, joka näkyy kuvassa 2. Tämän jälkeen halutun objektin kuva vielä leikattiin suuremmasta kuvasta. Lopullisten kuvien resoluutio on 100x70 pikseliä ja niissä on kolme värikanavaa. Edes erilleen leikatuista kuvista ei ole aina helppo havaita esimerkiksi kiven, ja ihmisen eroa (kuva3).



Kuva 3. Kasa kiviä merenpohjassa vasemmalla, sukeltaja merenpohjassa oikealla

Kerätyn kaikuluotaindatan luotettavuus on hyvä, ja sisäinen optimaalinen virhe on parhaan tiedon valossa nolla, mutta sen ulkoinen luotettavuus on vaikeampi määritellä. Tällä tarkoitetaan sitä, kuinka hyvin data edustaa kaikkia tapauksia,

joita voi esiintyä viistokaikuvissa, joissa on ihminen vedenpohjassa. Ainut käytännön tapa mitata tätä on käyttää neuroverkkojen koulutuksessa erillisenä datasettinä kahden eri päivän dataa, toista koulutukseen ja validointiin ja toisen päivän dataa testaamiseen. Mikäli opetusdatalla opetetun neuroverkon tulokset yleistyvät testidataan, voidaan sanoa, että datasetti on riittävän kattava yleistymään ainakin jossain määrin yleiseen tilanteeseen käytetyllä kaikuluotaimella.

Lopullinen ensimmäisenä päivänä taltiotu opetus-/validointidatasetti koostuu 205 kuvasta ihmisestä vedenpohjassa, sekä 249 kuvasta, jossa on joku muu objekti kuin ihminen. Tähän kategoriaan kuuluvat kuvat esimerkiksi kivistä, kasveista ja autonrenkaista. Tämän lisäksi toisella datankeräyskerralla taltioidusta datasetistä tehty testidatasetti sisältää yhteensä 125 kuvaa ihmisestä vedenpohjassa sekä 84 kuvaa, jossa ei ole ihmistä. Kaiken kaikkiaan kuvia on yhteensä 695, joista kuvia ihmisestä on 331 kappaletta, ja kuvia, jossa ei ole ihmistä on 364 kappaletta.

7.3 Datan siirto-oppiminen

Siirto-oppimisessa haluttiin testata sekä domaindataa että lääketieteellistä kuvantamisdataa. Domaindataa on hyvin niukasti saatavilla. Joillain yrityksillä on dataa, mutta sen kerääminen on ollut usein suuri investointi sekä kilpailuetu, joten sitä harvoin julkaistaan. Saatavilla oleva domaindata ei ole kerätty kuluttajaluokan kaikuluotaimilla, vaan huomattavasti kalliimmilla robotiikkaan tarkoitetuilla kaikuluotaimilla. (Jegorova, 2021, s. 1.)

7.3.1 Domaindata

Kääntöpöytä meren roskat -datasetti julkaistiin elokuussa 2021. Datasetti on taltioitu vesialtaassa, jossa on kääntöpöytä, jolle on asetettu vuorollaan yhteensä 18 eri esinettä 12 eri kategoriasta. Kääntöpöytää on käännetty jokaisella esineellä, minimistä maksimiin, ja näin taltioitu data eri kuvakulmista. Da-

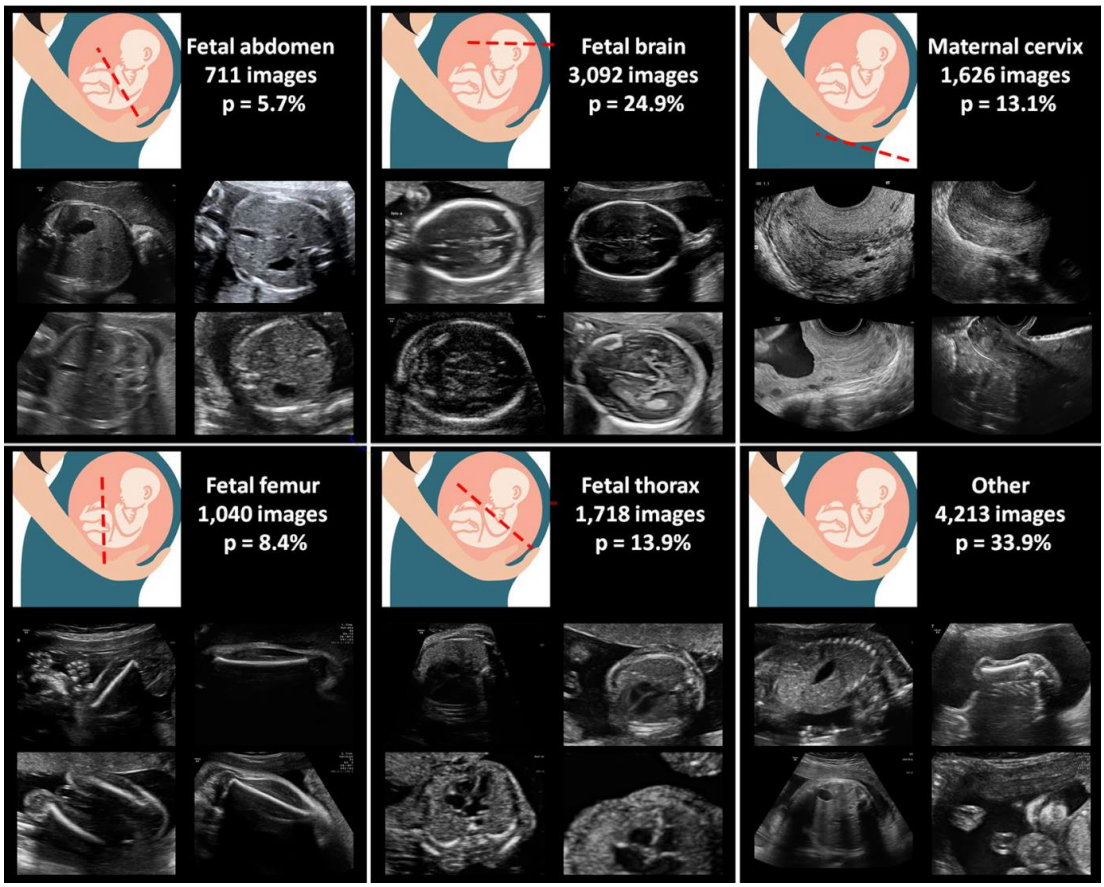
tan keräämiseen on käytetyttä ARIS Explorer 3000 -nimistä eteenpäin suuntaavaa kaikuluotainta 3 Mhz:n taajuudella. Datasetti on jaossa eri resoluutioisena. Tässä työssä käytetään parasta resoluutiota 96x96 pikseliä. Datasetti on jaettu HDF5-muodossa, ja sisältää 1505 kuvaa opetusdatasetissä, sekä 645 kuvaa testidatasetissä. (Valdenegro-Toro ym., n.d.)

Vesitankki meren roskat -datasetti julkaistiin kesäkuussa 2019. Data on 11 eri kategoriassa, joista viimeinen on tausta. Datan keräämiseen on käytetty ARIS Explorer 3000 -nimistä eteenpäin suuntaavaa kaikuluotainta. Datasetti on jaossa sekä kokonaisina kaikuluotainkuvina, että objektit eroteltuina eri resoluutioisina. Tässä työssä käytetään valmiiksi eroteltuja objekteja ja parasta resoluutiota 96x96 pikseliä. Datasetti on jaettu HDF5-muodossa, ja sisältää 1938 kuvaa opetusdatassa, 394 validointikuvaa sekä 395 testikuvaa. (Valdenegro-Toro ym., n.d.)

7.3.2 Lääketieteellinen kuvantamisdata

Sikiön ultraäänikuvantamisdata valikoitui käytettäväksi, koska dataa on hyvin saatavilla mutta myös siksi, että tämänkaltainen lääketieteellinen kuvantamisdata on hyvin samankaltaista kaikuluotainkuvan kanssa. Niissä molemmissa kohinan suhde dataan on hyvin suuri.

Yleiset äidin ja sikiön ultraäänikuvat -datasetti kerättiin ja julkaistiin osana Naturen Nature scientific reports -julkaisun artikkelia ‘Syvien konvoluutiohermoverkkojen arviointi tavallisten äidin ja sikiön ultraäänitasojen automaattista luokittelua varten’. Datasetissä on 12400 kuvaa yhteensä 1792 eri potilaasta, datasetti oli kirjoittajien mukaan, ainakin artikkelin julkaisun aikaan vuonna 2020, suurin julkaistu ultraäänikuvadatasetti. Kuvat on luokiteltu kuuteen eri luokkaan kuvan 4 mukaisesti; vatsa, aivot, reisiluu, rintakehä, äidin kohdunkaula sekä kuudes luokka, jossa ovat kaikki harvinaisemmat tasot.



Kuva 4. Yleiset äidin ja sikiön ultraäänikuvat -datasetin eri kategoriat. Kuvissa näkyy eri kuvantamiskohdat, ja niiden esimerkit (Burgos-Artizzu ym., 2020b)

Kuvia on kerätty eri sairaaloissa, eri henkilöiden toimesta, eri ultraäänilaitteilla. Kuvien resoluutioita ei ole normalisoitu datasetissä. Ennen kuvien käyttöä tässä työssä ne kaikki muokataan samaan resoluution kuin päädatasetti. (Burgos-Artizzu ym., 2020a; Burgos-Artizzu ym., 2020b.)

8 NEUROVERKKOMALLIN KEHITTÄMINEN

Tämä kehittämistyö on edennyt useissa eri vaiheissa. Dataa käsiteltiin eri tavoin ja erilaisia neuroverkkoja kokeiltiin. Eri vaiheissa kokeiltiin ja etsittiin parhaita parametreja sekä tapoja käyttää dataa ja augmentointeja. Samalla kokeiltiin erilaisia datan generointitapoja dynaamisilla ja staattisilla dataseteillä. Edellisten testivaiheiden tuloksista pääteltiin seuraavat askeleet kohti parempia tuloksia. Hyvin toimivien parametrien löydyttyä testattiin siirto-oppimisella, onko tällä saatavissa parempia tuloksia.

Kaikki testit on tehty Googlen kehittämän avoimen lähdekoodin Tensorflow-kirjastoilla. Osa testien laskennoista on ollut muutaman tunnin mittaisia, ja pisimmät on vieneet muutamia päiviä.

Kaikki kuvat on ennen neuroverkolle syöttämistä muutettu kahdeksan bittisistä merkittömistä kokonaisluvuista liukuluvuiksi nollan ja yhden väliin. Tämä normalisointi helpottaa neuroverkon opetusta. Erilaisten datan augmentointien jälkeen on varmistettu, että data on skaalattu takaisin nollan ja yhden väliin tai yli ja ali olevat arvot on leikattu.

8.1 Ohjelmistot ja niiden versiot työssä

Kaikki tämän tutkimuksen neuroverkkojen opetus ja testaus on tehty Linux Ubuntu 20.04 -ympäristössä ja Python 3.8 -versiolla. Opetukset on tehty Jupyter lab -ympäristössä käyttäen seuraavia python-moduuleja ja niiden versioita:

- tensorflow 2.9.0
- jupyterlab 3.5.0
- h5py 3.6.0
- keras 2.9.0
- Keras-Preprocessing 1.1.2
- matplotlib 3.6.2
- numpy 1.22.3
- Pillow 9.3.0

- pip 20.2.4
- scipy 1.9.3

Laskennassa on käytetty Nvidia-näytönohjaimia seuraavasti:

1. Ensimmäisen vaiheen testit on tehty GTX 1660 ti -näytönohjaimella, wsl-ympäristössä, windows 11 -isäntäjärjestelmällä.

2. Toisen vaiheen testit on tehty RTX 3080 -näytönohjaimella, wsl-ympäristössä, windows 11 -isäntäjärjestelmällä. cuda_version 11.2, cudnn_version 8.

3. Kolmannesta testistä eteenpäin testit on tehty docker-kontin sisällä Quattro A5000 -näytönohjaimilla, ubuntu 20.04 -isäntäjärjestelmässä. Näytönojainajurin versio oli 470.141.03 ja cuda-versio 11.2

8.2 Ensimmäiset testit ja tulokset

Ensimmäisessä vaiheessa tehtiin nopea testi, jossa datan muokkauksessa ja datan jaossa on käytetty kerasin omia funktiota. Tämän testin tarkoituksena oli ainoastaan testata koko tehtävän mahdollisuus, eli pystytäänkö näin pienellä datalla ylipäätään opettamaan edes validointidataan toimivaa neuroverkkoa.

Ensimmäiseksi testattavaksi neuroverkoksi valikoitui yksinkertainen konvoluutioneuroverkko kolmella konvoluutiokerroksella, max booling -kerroksilla, sekä lopussa kaksi täysin kytkettyä kerrosta. Aktivaatiofunktiona on RELU kaikissa muissa paitsi viimeisessä täysin kytketyssä kerroksessa, jossa on käytössä sigmoid-funktio.

Opetusvaiheessa opetusdatalle tehtiin erilaisia koonmuutoksia, kuvankäännöksiä, valoisuus- ja värimuutoksia, sekä väännös (shear) -muutoksia. Tällä pyrittiin lisäämään opetusdatan määrää ja näin myös vähentämään ylisovittumisen riskiä.

Tässä testissä käytettiin vain ensimmäisen datakeräyskerran dataa. Kuvat on jaettu opetus- ja validointidataan. Opetusdatassa on 164 kuvaa ihmisestä ja 205 kuvaa ei-ihmisestä ja validointidatassa 42 kuvaa ihmeestä ja 44 kuvaa ei-ihmisestä. Yhteensä tässä on siis 205 kuvaa ihmisestä ja 249 kuvaa ei-ihmisestä. Tämä ei tietysti ole optimaalinen tilanne vaan testidata pitäisi olla vielä kolmantena erillisenä datasettinä.

Tarkempi analyysi opetuksen jälkeisistä tuloksista käyttäen validointidataa kertoo, että 42 ihmisestä otetusta kuvasta 36 tunnistui oikein. Kuusi näistä tunnistui ei-ihmiseksi ja näistä kuudesta kaksi tunnistui juuri ja juuri väärään luokkaan. Kaksi oli melko väärässä, ja kaksi todella väärässä. Binaariluokittelijassa todella väärässä oleminen tarkoittaa sitä, että väärän luokan todennäköisyys on lähellä yhtä, ja oikean luokan todennäköisyys on lähellä nollaa. Juuri ja juuri väärässä olleessa taas molempien todennäköisyys on hyvin lähellä 50 prosenttia.

44 kuvasta, joissa ei ole ihmistä, 40 tunnistui oikein. Neljä tunnistui väärin ihmiseksi, näistä yski tunnistui juuri ja juuri väärään luokkaan, yksi melko väärään ja kaksi todella väärään luokkaan.

Alustavien testien perusteella voidaan sanoa että 10 tunnistusta 88 tehtiin väärin, eli tunnistustarkkuus on siis noin 88 %. Tämä oli jopa yllättävän hyvä tulos, ja selvästi riittävä siihen, että tarkempia testejä voidaan tällä datamäärällä lähteä tekemään.

8.3 Testit ja tulokset staattisella datasettillä

Ennen toiseen vaiheen alkua tehtiin lopullinen jako opetus- ja validointidatan välille. Jako tehtiin python random -moduulin avulla satunnaisella siemenellä. Toisessa vaiheessa otettiin myös käyttöön täysin erillinen testidata, jolloin voitiin alkaa testata mallien todellista suorituskykyä realistisessa tilanteessa. Testidata sisältää 209 kuvaa, joista 125 on kuvia ihmisestä ja 84 ovat kuvia, joissa

ei ole ihmistä. Kaikissa vaiheen neuroverkoissa oli käytössä keras-moduulin optimoija Adam käyttäen oppimisnopeus (learning_rate) -parametriä 0.001, joka on alkuperäisen kehittäjän suosittelema arvo. Neuroverkon painot tallennettiin testiä varten joka epookin välillä.

Toisen vaiheen testeihin otettiin mukaan vaihtelevasti eri kokoisia neuroverkkoja parametreilla laskettuna. Tämän vaiheen raskain neuroverkko on DenseNet, seuraavana on MobileNet, ja kevyimpinä on kaksi pientä perinteistä konvoluutioverkkoa. Kahden pienemmän verkon arkkitehtuurit näkyvät kuviossa 2 ja kuviossa 3. Näistä pienempää, 3x3 konvoluutiolla olevaa verkkoa, kutsutaan jatkossa nimellä konv3verkko, ja suurempaa, jossa on 5x5 konvoluutiot, kutsutaan jatkossa konv5verkko. Tässä testissä kaikissa muissa verkoissa paitsi DenseNet-verkossa on käytössä dropout-kerroksia. Näille testattiin arvoja 0, 0.5, 0.1, 0.2 ja 0.3. Kaikkia arkkitehtuureja testattiin viisi kertaa. DenseNet ja MobileNet otettiin valmiina toteutuksina tensorflown keras applications -luokasta parametreilla, jotka näkyvät alla olevasta taulukosta (taulukko4). konv3verkko ja konv5verkko määriteltiin tensorflow-kirjastolla.

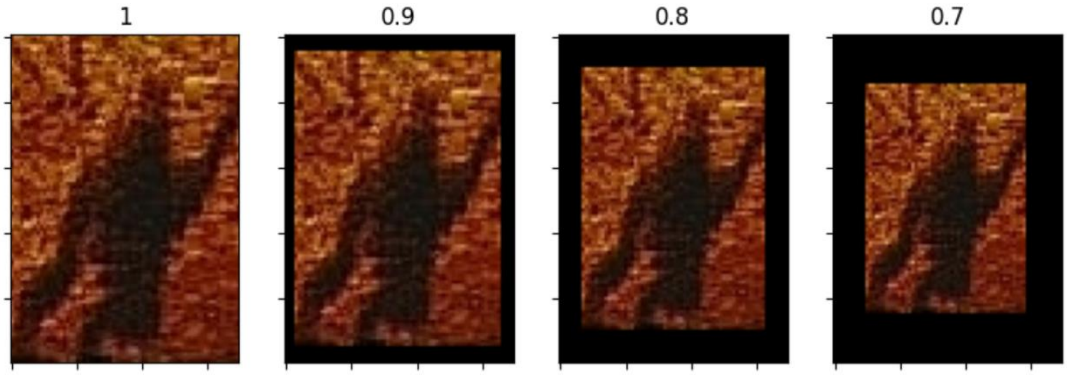
Taulukko 4. Käytettyjen neuroverkkojen parametrit

ResNet50	MobileNet	DenseNet121
include top=True,	input_shape= input_shape_s,	include_top=True,
weights=None,	alpha=1.0,	weights=None,
input_tensor=None,	depth_multiplier=1,	input_tensor=None,
input_shape=input_shape_s	dropout=drop_o,	input_shape= input_shape_s,
pooling=None,	include_top=True,	pooling=None,
classes=2,	weights=None,	classes=2,
classifier_activation="softmax"	input_tensor=None,	classifier_activation='softmax'

Toisessa vaiheessa testattiin staattista datasettiä. Ennen opetusta rakennettiin augmentoitu datasetti, jota käytettiin sellaisenaan kaikkien arkkitehtuurien opetuksessa. Datasetti tallennettiin hierarkkiseen datamuotoon .h5-tiedostoon käyttäen pythonin h5py-moduulia. Kuviin tehtiin erilaisia muunnoksia kuten

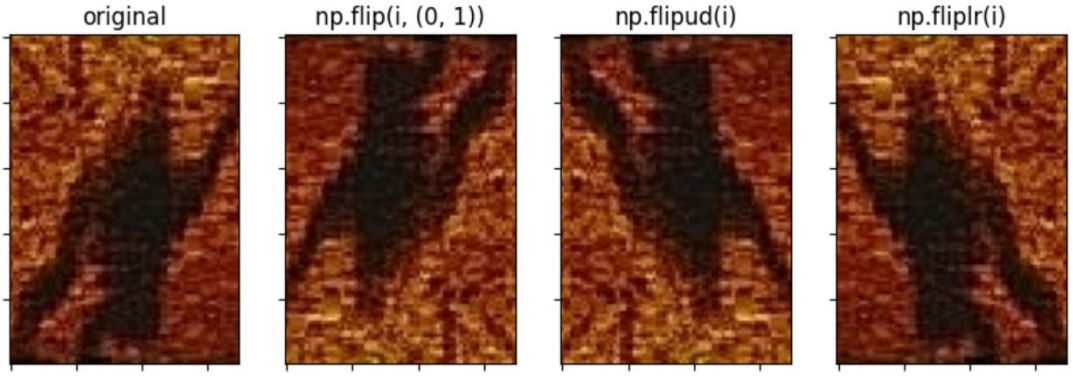
koonmuutoksia, käännöksiä, Gaussisen kohinan lisäystä, sekä kirkkauksien muutoksia.

Kuvan koonmuutokset toteutettiin niin, että jokaisesta kuvasta tehtiin neljä versiota. Kuvien koot suhteessa alkuperäiseen ovat 100 %, 90 % 80 % ja 70 %. Tässä kohtaa päädyttiin tekemään pelkkiä kuvanpienennöksiä, koska osa tärkeistä kuvan ominaisuuksista oli lähellä reunoja, eikä näitä haluttu menettää. Itse kuvanmuutokseen käytettiin scipy-työkalun ndimage-moduulin zoom-toimintoa. Jotta kuvat pysyisivät edelleen alkuperäisen kokoisina, täytettiin puuttuvat pikselit 0-arvoilla. Kuvasta 5 voidaan nähdä pienennöksen vaikutus kuviin.



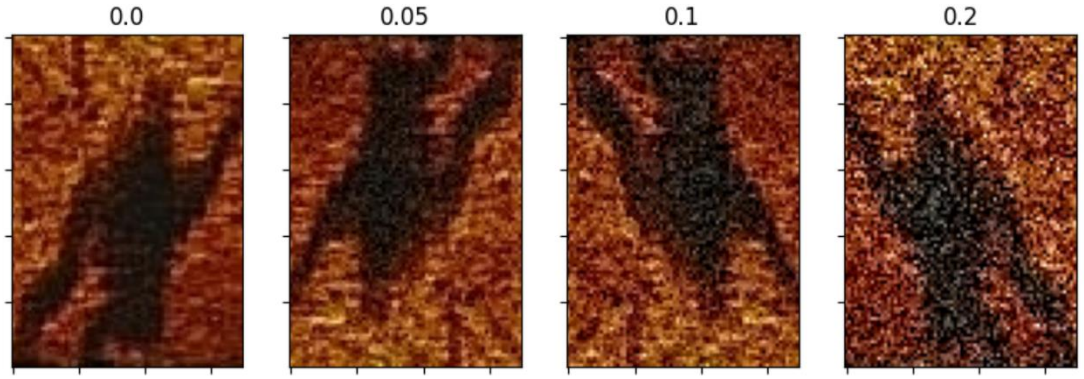
Kuva 5. Kuvapienennösten esimerkkikuvat

Kuvienkäännöksiä tuotettiin myös neljä versiota joka kuvasta; originaali, peilikuva, ylösalaisin, ja ylösalaisin peilikuva. Näihin päädyttiin, jotta kuvan kokoa tai suhteita ei tarvinnut muuttaa. Kuvamuutokset toteutettiin pythonin numpy-moduulista löytyvillä flip-, flipud-, fliplr-funktioilla. Alla esimerkkikuvat (kuva6).



Kuva 6. Kuvakäännösten esimerkkikuvat

Normaalijakauman mukaista kohinaa lisättiin kuviin neljä eri määrää, sigma-arvoilla 0.0, 0.05, 0.1 ja 0.2. Keskiarvo kaikissa tapauksissa oli 0. Kohina toteutettiin python numpyn random-moduulin normal-funktiolla. Jokaisen kuvan kanavaan kohinat tuotettiin erikseen, joten eri kanavien kohinat olivat erisuuruisilla satunnaisilla arvoilla. Kuvasta alla voidaan nähdä miten kohina vaikuttaa kuvaan (kuva 7).



Kuva 7. Esimerkkikuvat eri kohina-arvoilla

Kirkkauden muutokseen ei käytetty valmiita kirjastoja, vaan funktio sille valmistettiin työtä varten. Data on kolmekanavaista rgb-dataa, mutta kanavien suhde on erilainen kuin tavallisessa kameran kuvassa. Tämä pitää huomioida, kun eri kanavien kirkkauksia säätää. Kanavien tarkemmasta analyysistä voidaan havaita, että kanavien arvot eivät ole kovin lähellä toisiaan. Seuraavassa näkyy datasettien pikselikeskiarvot:

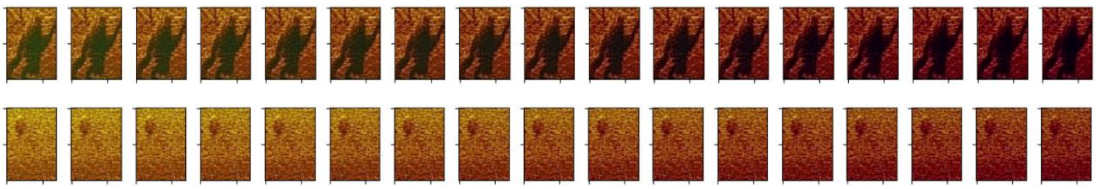
Koko train-datasetin pikselikeskiarvo
77.49494453781513
Koko train-datasetin punaisen kanavan pikselikeskiarvo
133.71902058823528
Koko train-datasetin sinisen kanavan pikselikeskiarvo
76.07553613445378
Koko train-datasetin vihreän kanavan pikselikeskiarvo
22.6902768907563

Testailujen jälkeen päädyttiin ratkaisuun, jossa jokaisesta kuvasta tehtiin 17 eri versiota, niin että:

Punaisen kanavan jokainen pikseli muutetaan arvoilla

[-40, -35, -30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25, 30, 35, 40]
Sinisen kanavan jokainen pikseli muutetaan arvoilla
[-20, -17, -15, -12, -10, -7, -5, -2, 0, 2, 5, 7, 10, 12, 15, 17, 20]
Vihreän kanavan pikselien arvoja ei muutettu.

Vaikutukset kuviin voidaan nähdä esimerkkikuvasta (kuva8). Ylemmän kuvan lähtöarvot ovat melko tummat, kun taas alemmassa lähtöarvot ovat melko kirkkaat, silti kumpikaan kuva ei näillä muutosarvoilla vielä mene täysin mustaksi tai ylivalotu.



Kuva 8. Esimerkkikuvat eri kirkkausmuutoksilla

Datasetti, joka sisältää vain yhden muutoksen, on vielä kohtalaisen pieni, mutta kumuloituvat muutokset kasvattavat datasetin kokoa nopeasti. Datasetin koon kasvaessa tämä alkoi käydä epäkäytännölliseksi, kun h5-tiedoston koko kasvoi yli 32 gigatavun. Tässä kohtaa kehitettiin oma datageneraattori, joka perii kerasin utils-moduulin Sequence-luokasta. Generaattori lataa satunnaisen 32 kuvan erän kerralla opetukseen. Näin RAM-muistin käyttö pysyy kohtuullisena.

Toisen vaiheen testitulokset olivat todella huonoja. Yksikään tallennetuista koulutetuista verkoista ei yltänyt yli 55 % tunnistustarkkuuteen. Rohkaisevaa näissä tuloksissa oli se, että on selvästi epätodennäköistä, että opetettu verkko sattumalta tunnistaa merkittävästi yli 50 % tarkkuudella, eli mikäli myöhemmissä testeissä saadaan parempia tuloksia, ne eivät todennäköisesti johdu sattumasta. Tässä testissä kuitenkin testattiin viisi opetuskierrosta neljällä arkkitehtuurilla ja näistä tallennettiin yhteensä 2000 neuroverkon painot, eikä niistä mikään tunnistanut merkittävästi satunnaista paremmin.

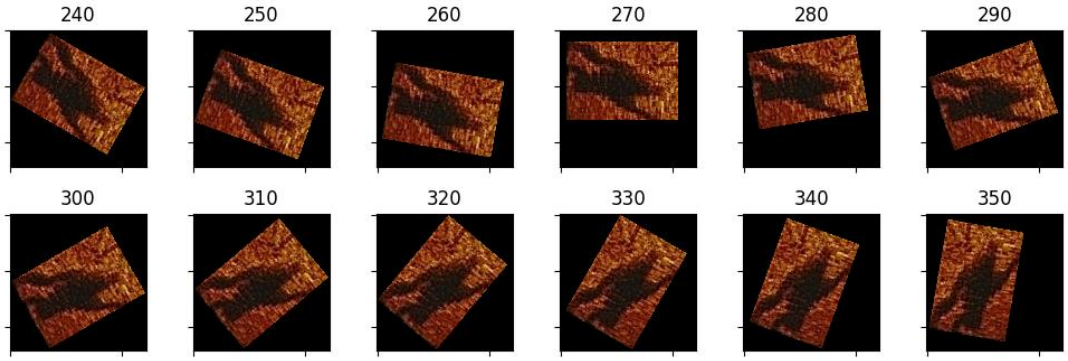
8.4 Datan Augmentointi datageneraattorilla

Kolmannessa testissä ajettiin yhteensä 20 opetuskierrosta neljällä eri arkkitehtuurilla ja viisi ajoa kullakin. Jokaisessa opetuskierroksessa on 1000 epookkia ja neuroverkon painot tallennetaan joka kymmenes kierros. Optimoijana on edelleen adam samoilla asetuksilla.

Kolmannen vaiheen testeihin otettiin mukaan DenseNet, MobileNet, sekä molemmat pienet konvoluutioverkot, konv3verkko ja konv5verkko. Datan argumentoinnin takia sisään tulevan datan dimensioita muutettiin muotoon 122x122x3. Tässä testissä dropout säädettiin nollaan.

Kolmatta testiä varten kehitettiin edelleen datageneraattoria. Aikaisemmin se vain jakoi opetusdatan satunnaisiin osiin ja latasi niitä tarpeen tullen tiedostosta. Uudessa versiossa datageneraattori myös hoitaa datan augmentoinnin, joka opetuserällä (batch).

Suurinta muutosta tavoitellaan kuvien käännöksillä. Siinä missä ennen käännöksiä tehtiin neljää erilaista, on nyt mahdollisten käännösten määrä tuhansia. Kuvaa käännetään ensin satunnainen kokonaisluku, joka on väillä 0 – 359. Tämän jälkeen syntynyt kuva sijoitetaan 122x122-kokoiselle kuvalle, koska itse kuvan mittasuhteet haluttiin säilyttää, eikä dataa haluttu hukata, joten kuvan kokoa suurennettiin. Kuva sijoitettiin suurempaan kuvakehykseen niin, että reunoille jäävä tila on satunnainen. Tämän toisen satunnaisuuden vaikutus näkyy hyvin esimerkiksi kuvan 9 käännösten 260 ja 270 kanssa, joissa itse käännöksen ero ei ole erityisen suuri, mutta sijoitus on sattunut toisessa lähes alalaitaan, ja toisessa lähes ylös.



Kuva 9. Esimerkkikuvat käännöksillä sijoitettuna 122 x 122 -kokoiseen kuvaan

Muun kuin opetusdatan muutoksia pyritään välttämään, mutta tässä tapauksessa muutokset olivat tarpeellisia, ja ne toteutettiin niin, etteivät kuvan mittasuhteet muutu. Validointidata ja testidata muutettiin niin, että alkuperäisen kuvan reunoille lisättiin tarvittava määrä mustia pikseleitä niin, että alkupäinen kuva on keskellä. Näin samaa verkkoa voidaan käyttää kaikissa tilanteissa.

Kolmannen vaiheen opetuskierroksista tallentui yhteensä 2000 verkon painot. Näistä tulokset ovat myös testidatalla huomattavasti aikaisempia parempia. 148 opetettua neuroverkkoa 2000 pääsi pienempään kuin 80 virheelliseen tunnistukseen. 84 opetetulla neuroverkolla tulos oli parempi kuin 70 virheellistä tunnistusta. Alle 60 virheen verkot on listattu alla taulukossa (taulukko5). Parhaat tulokset saavutettiin MobileNet-arkkitehtuurilla. Mikään muu arkkitehtuuri ei saavuttanut tätä tunnistustasoa. Tulos on hieman yllättävä, ja vaatii lisätutkimusta, sattuman mahdollisuutta pitää lähteä arvioimaan. Kuitenkin kaikista neljästä opetuskerrasta, eli eri dropout-arvoilla, on tuloksia tässä kohortissa, mikä vähentää sattuman mahdollisuutta. Eniten tuloksia tässä kohortissa on opetuksesta, jossa dropout on 0 %. Toisaalta paras tulos on dropout-todennäköisyydellä 10 %.

Taulukko 5. Kolmannen testin parhaat tulokset

Arkkitehtuuri	Epookki	Dropout prosent-teina	virheellisten ennusteiden määrä
MobileNet	220	10 %	42
MobileNet	210	0 %	47
MobileNet	170	10 %	49
MobileNet	310	20 %	50
MobileNet	820	5 %	51
MobileNet	580	10 %	52
MobileNet	520	0 %	52
MobileNet	300	10 %	54
MobileNet	320	0 %	55
MobileNet	190	0 %	55
MobileNet	980	5 %	55
MobileNet	50	0 %	55
MobileNet	110	0 %	57
MobileNet	70	0 %	58
MobileNet	220	0 %	58
MobileNet	300	20 %	58
MobileNet	620	0 %	59

Erikoista tuloksissa on se, että hyviä tuloksia on hyvin erilaisten epookkien välillä, kuten taulukosta 6 voidaan havaita. Eniten parhaan kohortin tuloksia on kuitenkin välillä 100 – 400 epookkia.

Taulukko 6. Parhaat tulokset epookeissa

Epookki	0 - 100	100 - 200	200 - 300	300 - 400	400 - 500	500 - 600	600 - 700	700 - 800	800 - 900	900 - 1000
Tallennetut neuro-verkot epookeilla	2	3	3	4	0	2	1	0	1	1

Yksi selittävä tekijä tälle voisi olla liian suuri oppimisnopeus. Toinen vaihtoehto on, että osa opetusdatan muutoksista huonontaa tarkkuutta. Yksi tapa varmistaa tulosten järkevyys on tarkastella tarkemmin parhaiden tulosten tarkkuutta validointi- ja augmentoimattoman opetusdatan kanssa.

Parhaan tuloksen saavuttanut MobileNet, 10 %:n dropoutilla 220 epookin kohdalla, teki 42 virheellistä tunnistusta testidatalla. Virheellisistä tunnistuksista 22 kuvaa oli sellaisia, joissa ihminen tunnistettiin joksikin muuksi, ja 20 sellaisia,

joissa joku muu kuin ihminen tunnistettiin ihmiseksi. Näistä 42 väärästä tunnistuksesta 13 oli alle 70 % varmuudella väärä, ja näistä kolme oli alle 55 %:n varma vastaus. Tässä ei vaikuta olevan mitään erikoista.

Tulokset tuntuivat kuitenkin sen verran heittelevän eri epookkien välillä, että tehtiin vielä yksi testi. Siinä testattiin parhaat tulokset, eli ne neuroverkot, joissa on 55 tai vähemmän virheellisiä tunnistuksia testisetillä. Testissä testattiin virheellisten tunnistusten määrä validointidatalla sekä augmentoimattomalla opetusdatalla. Näistä tuloksista saadaan paljon parempi kokonaiskuva. Tuloksista käy aika selvästi ilmi, että ne testitulokset, jotka ovat testidatalla tunnistuneet hyvin, ja joissa samalla myös epookkien määrä on todella pieni, saavuttavat huonoja tuloksia validointi- ja opetusdatalla, kuten taulukosta 7 voidaan havaita.

Taulukko 7. Tulokset eri dropout parametrin arvoista kaikilla dataseteillä.

Dropout	0	0,1	0,05	0,1	0,2	0,1	0,05	0	0	0	0	0,1
epookki	50	220	980	300	310	580	820	210	190	520	320	170
testivirheet	55	42	55	54	50	52	51	47	55	52	55	49
validointivirheet	52	53	17	29	23	25	30	45	33	29	22	35
opetusvirheet	94	118	7	37	47	26	24	94	52	15	33	66

Selvää hyötyä dropoutista ei kuitenkaan näytä olevan, ja monissa hyvissä tuloksissa dropout on 0 tai 0,05. Edelleen myös kaikilla dataseteillä hyviä tuloksia saavia tuloksia on pienillä epookkimäärillä, eli opetusnopeuden pienentämistä olisi hyvä kokeilla.

8.5 Testit ja tulokset eri oppimisnopeuksilla

Seuraavaksi testattiin viime testeistä esiin noussutta ajatusta opetusnopeuden laskemisesta. Tässä testissä käytettiin vain viime testin parasta MobileNet-arkkitehtuuria muuten täysin samoilla parametreilla, ainoastaan eri oppimisnopeuksilla. Jokainen opetus ajettiin viisi kertaa satunnaisuuden vähentämiseksi.

Ensimmäisessä testissä oppimisnopeudeksi valittiin 0.0001, yhtä kertaluokkaa pienempi kuin aikaisemmin. Opetus- ja validointivaiheessa oppiminen oli huomattavasti hitaampaa ja parhaat tulokset varsinkin opetusvaiheessa olivat lähellä epookkien loppua. Hieman yllättävää oli kuitenkin, miten huonosti opetus generalisoitui testidatalle. Ainoastaan yhdeksällä tallennetuista verkoista vääriä tunnistuksia oli alle 80. Kaikilla opetuskerroilla taso oli kuitenkin melko tasaista. Parhaaseen kohorttiin nousivat kaikki muut opetuskerrat kolmatta lukuun ottamatta. Tarkemmat luvut näkyvät taulukossa 8. Pahin ongelma tuloksissa on, että opetus- ja validointidata antaa näille verkoille todella huonoja tuloksia. Tästä voidaan päätellä, että ainakin epookkien määrää pitäisi kasvattaa huomattavasti, jos näin pienellä opetusnopeudella opetetaan. Tämä taas hidastaisi opetusta huomattavasti.

Taulukko 8. Oppimisnopeustestin kaikkien datasettien virheet oppimisnopeudella 0.0001

Ajokerta	2	1	1	4	5	1	1	2	1
epookki	500	560	520	70	380	550	630	120	490
testivirheet	76	72	80	79	77	80	79	77	80
validointivirheet	60	58	52	73	43	52	46	65	58
opetusvirheet	123	136	130	160	89	126	100	141	133

Tämän testin toisessa vaiheessa opetusnopeudeksi valittiin kahden aikaisemmin kokeillun puolivälistä arvo 0.0005. Tässä testissä tulokset olivat jo huomattavasti parempia kuin aikaisemmassa, joten voidaan päätellä että 0.0001 on ainakin tarpeettoman pieni oppimisnopeus tässä ongelmassa. Alle 80 virheelliseen tunnistukseen päästiin yhteensä 64 eri tallennetulla verkolla. Alle 70 virheelliseen tunnistukseenkin ylsi 21 eri verkkoa. Tämän testin parhaaseen luokkaan, 60 ja vähemmän virheellisiä tunnistuksia, ylsi kolme tallennettua

verkkoa. Nämä kaikki olivat eri ajokerroilla. Validointi- ja opetusdatan virheet tässä ovat jo paljon parempia, kuten taulukosta 9 voidaan nähdä.

Taulukko 9. kaikkien datasettien tunnistusvirheet opetusnopeudella 0.005

Ajokerta	2	3	5
epookki	610	880	590
testivirheet	57	59	56
validointivirheet	22	16	22
opetusvirheet	22	3	21

Tulokset ovat selvästi parempia opetusnopeudella 0.005 kuin nopeudella 0.0001, mutta edelleen huonompia kuin alkuperäisellä nopeudella 0.001. Voidaan siis todeta, että näillä parametreilla ja tällä datalla 0.001 tuntuu edelleen parhaalta opetusnopeudelta.

8.6 Testit yksikanavaisella kuvalla

Seuraavaksi tehtiin testejä yksikanavaisella kuvalla. Tässä testissä kokeiltiin kohtalaisen suurta muutosta. Tunnistusta tehtiin datalla, joka on muutettu yksikanavaiseksi harmaasävykuvaksi. Neuroverkon kokoon ja parametrien määrään on värikanavien määrällä melko vähän vaikutusta jo testatuilla arkkitehtuureilla, kuten taulukosta 10 voidaan huomata. Näin ollen verkon kykyyn muistaa ja hahmottaa monimutkaisia asioita tällä ei ole kovin suurta merkitystä. Tunnistustarkkuuksiin voi eri arkkitehtuureilla kuitenkin olla merkittäviäkin vaikutuksia, joten tässä kohtaa testit ajetaan taas kaikilla arkkitehtuureilla.

Taulukko 10. Parametrien määrät eri arkkitehtuureilla eri värikanavien määrien kanssa.

Arkkitehtuuri	Parametrit	sisäänt tulevan datan dimensiot (122, 122,1)	sisäänt tulevan datan dimensiot (122, 122,3)
DenseNet	Parametrit yhteensä	7,033,282	7,039,554
	Opetettavat parametrit	6,949,634	6,955,906
	staattiset parametrit	83,648	83,648
MobileNet	Parametrit yhteensä	3,230,338	3,230,914
	Opetettavat parametrit	3,208,450	3,209,026
	staattiset parametrit	21,888	21,888
ResNet50	Parametrit yhteensä	23,585,538	23,591,810
	Opetettavat parametrit	23,532,418	23,538,690

	staattiset parametrit	53,12	53,12
Konv5verkko	Parametrit yhteensä	1,305,794	1,307,394
	Opetettavat parametrit	1,305,794	1,307,394
	staattiset parametrit	0	0
Konv3verkko	Parametrit yhteensä	720,482	721,058
	Opetettavat parametrit	720,482	721,058
	staattiset parametrit	0	0

Yksikanavaisen datan testissä ovat mukana samat aikaisemmat DenseNet, MobileNet, ja kaksi perinteistä pientä konvoluutioverkkoa. Näiden lisäksi testiin otettiin mukaan vielä astetta suurempi ja syvempi verkko, ResNet50. Batch-koko tuplattiin 34:ään. Opetusnopeus on tässä testissä alkuperäinen ja kehittäjiensä suosittelema 0.001.

Pysyvä datasetti muokattiin yksikanavaiseksi niin, että opetus-, validointi- ja testidatan jako pysyi muuttumattomana. Opetusdata muutettiin yksikanavaiseksi neljällä eri tavalla; jokainen alkuperäisen kuvan kanava tallennettiin erilliseksi kuvaksi, jolloin saatiin kolme yksikanavaista kuvaa ja neljäntenä kuvana oli kanavien pikselikeskiarvo. Näin ollen opetusdatasetin koko nelinkertaistui. Validointi- ja testidatalle harmaasävykuva tehtiin niin, että uusi pikseli oli aina kanavien keskiarvo. Lisäksi opetusdatasta tallennettiin myös tämän kaltainen datasetti, jossa on vain keskiarvoistetut kuvat, tätä käytetään vain testivaiheessa. Tensorflow keras 2d-konvoluutiokerrokset odottavat dataa neliulotteisena, joten tästä syystä data tallennettiin muotoon (1360,100,70,1,) eli viimeinen akseli pidettiin mukana, vaikka sen koko on 1.

Opetuksen aikaiseen datageneraattoriin piti tehdä pieniä muutoksia, jotta se osasi käsitellä yksikanavaista dataa. Ainut datan ominaisuuksiin vaikuttava muutos oli kirkkauden satunnaisvaihtelun muuttaminen suhteelliseksi, kun aikaisemmin se oli absoluuttinen. Tämä tehtiin siitä syystä, että nyt kuvien keskiarvoiset pikseliarvot olivat melko erilaisia johtuen sinisen kanavan pienestä kirkkaudesta. Kirkkauden muutoksiksi valittiin -8 – 8 prosenttia.

Yksikanavaisen datan testitulokset ovat selvästi parhaat tähän mennessä. Suurin määrä verkkoja pääsi hyviin tuloksiin ja paras tulos myös saavutettiin.

Mikä tärkeää, kaikki hyvät tulokset saavuttaneet verkot pärjäsivät hyvin myös validointidatalla. Sen lisäksi testattiin vielä opetusdatalla tulokset ilman argumentointeja kuvilla, joissa kuva on värikanavien keskiarvo. Parhaaseen kategoriaan, testivirheitä 50 tai alle, päästiin yhteensä 13 tallennetulla neuroverkon painoilla. Näistä 12 oli DenseNet-verkkoja, ja yksi oli ResNet-verkko, kuten taulukosta 11 voidaan havaita. Parempaan kuin 60 tai vähemmän vääriä tunnistuksia, pääsi 46 verkkoa Tähän kohorttiin pääsivät kaikki muut paitsi toinen pienistä konvoluutioverkoista. Lisäksi näistä kaikissa muissa paitsi kolmessa validointidatan virheellisten tunnistusten määrä oli 35 – 24, kun kaikissa muissa se oli alle 20 virheellistä tunnistusta. Parempaan kuin 70 tai vähemmän vääriä tunnistuksia päästiin 108 opetetun verkon painoilla tässä testissä tallennetuista 400, ja kaikki arkkitehtuurit olivat tässä kohortissa jo edustettuina.

Taulukko 11. Testin parhaiten suoriutuneiden neuroverkkojen virheet

DenseNet	epokkien määrä	testivirheet	validointivirheet	Opetusvirheet
DenseNet	290	50	15	0
DenseNet	520	36	16	1
DenseNet	700	48	17	5
DenseNet	560	46	14	0
DenseNet	680	43	16	1
DenseNet	310	36	23	10
DenseNet	550	44	12	0
DenseNet	150	49	13	7
DenseNet	740	48	13	0
DenseNet	770	50	16	1
DenseNet	330	46	15	3
DenseNet	440	49	16	0
ResNet	670	50	22	8

Tarkempi analyysi parhaiten oppineesta verkosta DenseNet 520 -epookilla, jossa testivirheitä oli 36, paljastaa, että testivirheistä 25 oli tapauksia, joissa ihminen tunnistui muuksi kuin ihmiseksi, ja 11 tunnistusta, jossa jokin muu kuin ihminen tunnistui ihmiseksi. Taulukosta 12 voidaan vielä nähdä virheiden suuruus, eli oliko kuva tunnistunut täysin väärin, vai oliko neuroverkko tuloksesta epävarma. Parhaalla verkolla lähes kaikki virheet olivat varmoja tai lähes varmoja virheitä. Niiden kuvien määrät, joissa validointivirheitä oli 16 kappaletta, opetusdatasta virheitä oli vain 1.

Taulukko 12. Testin parhaan neuroverkon testin virheet jaoteltuna kategorian mukaan

tunnistustarkkuus	Kuva joka oli ihminen	Kuva joka oli jotain muuta kuin ihminen
40% tai alle		1
30% tai alle	2	1
alle 20%	23	9

8.7 Verkkopohjainen siirto-oppiminen domaindatalla

Tässä testissä kokeiltiin verkkopohjaista siirto-oppimista domaindatalla, eli neuroverkko on ensin opetettu käyttämällä ulkopuolista datasettiä “kääntö-pöytä meren roskat”. Opetetut verkot ladattiin github-palvelusta. Opetuksen ovat toteuttaneet datasetin alkuperäiset tekijät. Tässä testissä jokainen neuro-verkkoarkkitehtuuri opetettiin viisi kertaa satunnaisuuden vähentämiseksi. Jo-kaisessa opetuskerrassa oli 1000 epookkia ja verkon painot tallennettiin 10 epookin välein.

Tässä testissä käytettiin MobileNet-, DenseNet- ja ResNet20-arkkitehtuureja valmiilla painoilla. ResNet20 on matalampi versio ResNet-arkkitehtuurista, jota käytettiin aikaisemmissa testeissä. Alkuperäisessä datasetissä on 12 eri luok-kaa. Tästä syystä verkon viimeinen kerros poistettiin, ja uusi verkko tehtiin li-säämällä tähän poistettuun verkkoon viimeiseksi kerrokseksi täysin kytketty kerros kahdella neuronilla ja Softmax-aktivoinnilla.

Opetusdatan augmentointi tehtiin samalla tavalla kuin edellisessä testissä. Tä-män neuroverkon sisääntulokerros oli hieman pienempi, mitä edellisessä tes-tissä oli, ollen 96 x 96 pikseliä. Opetusvaiheen koon muutokset tehtiin pienen-tämällä augmentoidut 122 x 122 kuvat scipyn ndimagen zoom-funktiolla ko-koon 96 x 96 pikseliä.

Testivaiheessa kaikkien opetus-, validointi- ja testidatan kuvat muutettiin oike-aan kokoon käyttäen alkuperäisen datan 100 x 70 koon kuvia. Dimensioon,

jonka koko on 70, lisättiin numpyn pad-funktiolla 13 mustaa pikseliä molemmille reunoilla. Tämän jälkeen 100 x 96 kuva ajettiin scipyn ndimagen zoom-funktiolla lopulliseen 96 x 96 kokoon.

ResNet20-verkolla parempiin kuin 80 tai vähemmän vääriä tunnistuksia -tulokseen pääsi yhteensä 147 tallennettua neuroverkkoa. Kategoriaan, parempi kuin 70 tai vähemmän vääriä tunnistuksia, pääsi 90 tallennettua verkkoa. Kaikilla viidellä opetuskerralla päästiin tulokseen 60 tai vähemmän vääriä tunnistuksia. Yhteensä näitä tuloksia oli 48. Parhaaseen kategoriaan, eli 50 tai vähemmän vääriä tunnistuksia, pääsi 10 tallennettua neuroverkkoa. Parhaan kategorian tulokset kaikilla dataseteillä on esitetty taulukossa 13. Taulukosta voidaan havaita, että validointi- ja opetusvirheet ovat kohtalaisen pieniä.

Taulukko 13. ResNet20-verkon parhaat tulokset

opetuskerta	Epookki	Testivirheet	Validointivirheet	Opetusvirheet
4	270	41	24	25
4	440	41	36	24
1	380	42	26	16
5	290	44	24	28
4	960	46	26	13
1	550	47	27	19
4	820	47	24	17
5	890	50	23	15
5	600	50	32	40
1	360	50	24	30

DenseNet-verkolla parempiin kuin 80 tai vähemmän vääriä tunnistuksia pääsi yhteensä 74 tallennettua neuroverkkoa. Kategoriaan parempi kuin 70 tai vähemmän vääriä tunnistuksia pääsi 37 tallennettua verkkoa. Tähän kategoriaan päästiin kaikilla opetuskerroilla. Neljällä opetuskerralla viidestä päästiin tulokseen 60 tai vähemmän vääriä tuloksia, yhteensä näitä tuloksia oli 15. Parhaaseen kategoriaan, 50 tai vähemmän vääriä tunnistuksia, pääsi viisi tallennettua neuroverkkoa. Parhaan kategorian tulokset kaikilla opetuskerroilla on esitetty taulukossa 14. Parhaaseen luokkaan päästiin monella opetuskerralla, vaikka tuloksia oli vain neljä.

Taulukko 14. DenseNet-verkon parhaat tulokset

opetuskerta	Epookki	Testivirheet	Validointivirheet	Opetusvirheet
5	550	49	19	10
4	380	46	36	29
2	170	45	13	12
1	810	38	22	12
2	320	44	21	20

MobileNetverkolla parempiin kuin, 80 tai vähemmän vääriä tunnistuksia, pääsi yhteensä 137 tallennettua neuroverkkoa. Kategoriaan parempi kuin 70 tai vähemmän vääriä tunnistuksia pääsi 63 tallennettua verkkoa. Kaikilla viidellä opetuskerralla päästiin tulokseen 60 tai vähemmän vääriä tuloksia. Yhteensä näitä tuloksia oli 14. Parhaaseen kategoriaan, 50 tai vähemmän vääriä tunnistuksia, pääsi kolme tallennettua neuroverkkoa. Parhaan kategorian tulokset kaikilla opetuskerroilla on esitetty taulukossa 15. Tällä verkolla kaikki parhaat tulokset ovat välissä 300 - 500 epookkia.

Taulukko 15. MobileNet-verkon parhaat tulokset

opetuskerta	Epookki	Testivirheet	Validointivirheet	Opetusvirheet
1	340	46	31	25
2	470	46	29	37
2	330	48	17	7

Tässä testissä suuria eroja ei eri arkkitehtuurien välillä ollut. ResNet20 saavutti parhaat tulokset testidatalla, mutta jäi taas validointi- ja opetusdatalla hieman jälkeen. Suurin määrä hyviä tuloksia oli ResNet20-arkkitehtuurilla, joten kaiken kaikkiaan sen tulokset olivat parhaat, tosin pienellä marginaalilla, joka voi vielä olla satunnaisvaihtelua.

8.8 Datapohjaisen siirto-oppiminen testi lääketieteellisellä datalla

Tässä testissä verkko opetettiin eri datalla, ja tällä toisella datalla parhaiten oppineet verkot koulutettiin siirto-oppimisella kaikuluotaindatalla. Lääketieteellisessä ultraäänidatasetissä on huomattavasti enemmän kuvia kuin kaikuluotaindatasetissä, joten siitä syystä alkuopetuksessa epookkien määrää laskettiin monissa testeissä käytössä olleesta tuhannesta epookista 200 epookkiin.

Kaikki verkot koulutettiin alkukoulutuksessa kolme kertaa satunnaisuuden vähentämiseksi. Alkukoulutuksessa lääketieteellisellä datalla verkon painot tallennettiin vain siinä tapauksessa, jos validointivirhe epookin lopussa on pienempi kuin 0.7 ja pienempi kuin mikään edellinen validointivirhe.

Opetusten jälkeen jokaisesta arkkitehtuurista valittiin parhaiten suoriutunut tallennettu neuroverkko ja tämän jälkeen nämä parhaiten suoriutuneet verkot koulutettiin kaikuluotaindatalla. Tässä opetuksessa epookkien määrä on jälleen 1000, koska datasetti on huomattavasti pienempi. Kaikki verkot koulutetaan kolmeen kertaan kaikuluotainkuvalla.

Tähän testiin otettiin mukaan kaikki alkuperäisellä datalla käytössä olleet neuroverkkoarkkitehtuurit, ResNet, rensenet, MobileNet, ja kaksi pienempää perinteisempää konvoluutioverkkoa, Konv3verkko sekä Konv5verkko. Jokaisen neuroverkon ulostulokerroksen neuronien määrä piti ensimmäisessä testissä olla kuusi, koska tässä datassa on kuusi eri kategoriaa. Siirto-oppimisvaiheessa viimeinen kerros pitää taas vaihtaa kahteen neuroniin kuudesta. Muissa arkkitehtuureissa, paitsi MobileNeteissä, viimeinen täysin kytketty kerros vaihdettiin pienemmäksi. MobileNet-arkkitehtuurissa viimeisen dropout-kerroksen jälkeen poistettiin (None, 6, 1, 1) -muotoinen kaksiulotteinen konvoluutiokerros (Conv2D), muotoa (None, 6) oleva uudelleenmuokkauskerros (reshape), ja nämä korvattiin litistetyllä kerroksella (Flatten) muotoa (None, 1024) ja täysin kytketyllä kerroksella, jossa on kaksi neuronia. Kaikissa verkoissa viimeisessä täysin kytketyssä kerroksessa on softmax-aktivointi.
Lääketieteellinen ultraäänidatasetti ladattiin png-muotoisina kuvina. Tämän lisäksi mukana tuli csv-tiedosto, jossa oli tiedostonimen lisäksi tarkempia tietoja kuvista, kuten potilaan numero ja kategoria.

Kuvat jaettiin opetus-, validointi- ja testidataan niin, että yhdestä potilaasta otetut kuvat olivat aina vain yhdessä datassa mukana. Eli ensin potilaat jaettiin satunnaisesti pythonin random-moduulia hyväksi käyttäen kolmeen kategoriaan niin, että opetusdataan tuli 1380 potilasta, validointidataan 394 potilasta ja testidataan 197 potilasta. Tämä oli hyvin lähellä 70 %, 20 %, 10 % -jakoa,

vaikka eri potilaista on eri määrä kuvia. Arvio oli, lopullinen määrä olisi kuitenkin tarpeeksi lähellä tätä lukemaa.

Seuraavaksi käytiin pythonin csv-moduulin reader-funktiolla csv-tiedosto läpi iteroiden tiedoston jokainen rivi. Tämän jälkeen sijoitettiin potilaiden kuvien tiedostonimet ja kategoria kolmeen eri listaan riippuen siitä, mihin tämän potilaan numero oli arvottu.

Tämän jälkeen kuvat sijoitettiin numpy-matriiseihin, sekä kategoriatieto toisiin numpy-matriiseihin. Kuvien lukemiseen käytettiin Pillow-modulin Image-luokan open-metodia. Kuvat olivat myös vaihtelevasti harmaasävykuvia, kolmekanavakuvia ja nelikanavakuvia. Tässä vaiheessa kaikki kuvat muutettiin yksikanavaisiksi harmaasävykuviksi käyttäen Pillow-modulin ImageOps-luokan grayscale-metodia. Kuvat olivat myös eri kokoisia, joten niiden koot piti myös yhtenäistää. Kaikki kuvat muutettiin numpy resize -funktiolla samaan kokoon, jossa kaikuluotainkuvat tallennettiin. Ja numpyn newaxis -funktiolla lisättiin kolmas ulottuvuus, jotta kuvat saatiin lopulliseen muotoon 100 x 70 x 1 pikseliä. Ennen tallentamista kuvat vielä normalisoitiin 0 ja 1 väliin. Kuvat tallennettiin h5py-modulin File-luokalla .h5-tiedostoksi. Lopullisessa datasetissä opetuskuvia on 8950, validointikuvia 2425, ja testikuvia 1025. Prosentuaalisesti jako on noin 72 %, 20 %, 8 %. Tämän katsottiin olevan riittävän lähellä tavoiteltua jakoa, 70 %, 20 %, 10 %.

ResNet-verkon tulokset olivat testauksessa yhtä virhettä paremmat kuin DenseNetin, mutta opetusdatalla tulokset olivat hieman huonommat. Paras tulos saavutettiin ensimmäisellä opetuskerralla, 182 epookin kohdalla. Testivirheitä oli kahdeksan, validointivirheitä 29 ja opetusvirheitä 49. Kaikkien opetuskertojen parhaat tulokset nähdään taulukosta 14.

Taulukko 14. Datapohjaisen siirto-oppimisen ResNet-verkon parhaat tulokset eri opetuskerroilla ultraäänidatalla.

Opetuskerta	epookki	Testivirheet	validointivirheet	opetusvirheet
0	182	8	29	49
1	118	7	33	60
2	69	13	32	92

DenseNet-verkon tulos oli testin paras kaiken kaikkiaan. Testidatalla virheellisiä tunnistuksia on vain kahdeksan. Tässä on yksi enemmän kuin ResNet-verkon parhaassa tuloksessa, mutta validoinnissa 28 virhettä ja opetuksessa vain 26. Paras tulos saavutettiin ensimmäisellä opetuskerralla 145 epookin kohdalla. Taulukosta 15 voidaan nähdä kaikkien opetuskertojen parhaat tulokset. Varsinkin opetusvirheissä DenseNet oli muita parempi.

Taulukko 15. Datapohjaisen siirto-oppimisen DenseNet-verkon parhaat tulokset eri opetuskerroilla ultraäänidatalla

Opetuskerta	epookki	Testivirheet	validointivirheet	opetusvirheet
0	145	8	28	26
1	150	14	24	19
2	90	8	24	48

MobileNet-tulokset olivat myös hyviä. Paras tulos saavutettiin toisella opetuskerralla 151 epookin kohdalla. Testivirheitä oli 11, validointivirheitä 31 ja opetusvirheitä 47. Kaikkien opetuskertojen tulokset voidaan nähdä taulukosta 16.

Taulukko 16. Datapohjaisen siirto-oppimisen MobileNet-verkon parhaat tulokset eri opetuskerroilla ultraäänidatalla

Opetuskerta	epookki	Testivirheet	validointivirheet	opetusvirheet
0	129	12	40	67
1	151	11	31	47
2	159	14	30	58

Tässä tehtävässä pienempien perinteisten konvoluutioverkkojen koko alkoi todennäköisesti vaikuttaa tuloksiin, ja niillä tulokset olivat huonompia kuin isommilla verkoilla. Tätä tukee myös se, että paremmat tulokset olivat suuremmalla Konv5verkolla. Testivirheitä oli 15, validointivirheitä 30 ja opetusvirheitä 98. Parhaat tulokset saavutettiin ensimmäisellä opetuskerralla, epookilla 169. Kaikkien opetuskertojen tulokset nähdään taulukosta 17. Suurin ero suurempiin verkkoihin on validointi- ja opetusdatan virheissä.

Taulukko 17. Datapohjaisen siirto-oppimisen Konv5verkon parhaat tulokset eri opetuskerroilla ultraäänidatalla

Opetuskerta	epookki	Testivirheet	validointivirheet	opetusvirheet
0	169	15	30	98
1	110	22	63	220
2	169	20	55	192

Huonoiten tässä testissä suoriutui pienin Konv3verkko, joka sekin tosin tunnisti edelleen hyvin, vaikka tulokset jäivätkin isommista verkoista. Parhaan opetuskerran testivirheiden määrä oli 18, validointivirheiden määrä 43, ja opetusvirheitä oli 123. Taulukossa 18 nähdään kaikkien opetuskertojen tulokset. Varsinkin opetusvirheiden määrässä on muita tuloksia suurempaa hajontaa.

Taulukko 18. Datapohjaisen siirto-oppimisen Konv3verkon parhaat tulokset eri opetuskerroilla ultraäänidatalla

Opetuskerta	epookki	Testivirheet	validointivirheet	opetusvirheet
0	144	18	43	123
1	80	29	85	289
2	166	24	64	213

Tulokset olivat yllättävän hyviä. Kaiken kaikkiaan parhaan ResNet-verkon tulos oli 99,2 % todennäköisyys oikeaan tunnistukseen. Tämä on huomattavasti parempi kuin naturen scientific reports -julkaisun paras tulos 94.0 %, joka saavutettiin ResNeXt-101-arkkitehtuurin neuroverkolla. Todennäköisesti suurin selittävä tekijä on, että julkaisussa datan jaottelu on erilainen, ollen noin 40 % opetusdataa, 10 % validointidataa ja 50 % testidataa. Muita mahdollisia selittäviä tekijöitä voi olla heidän käyttämänsä pieni epookkien määrä, tai mahdollisesti riittämätön datan augmentointi. (Burgos-Artizzu ym., 2020; Coronado-Gutiérrez ym., 2020, s. 5.)

ResNet-arkkitehtuurilla parhaan tuloksen luokkaan, 50 tai vähemmän vääriä tunnistuksia, päästiin kaikilla kolmella opetuskerralla. 38 opetuskertaa pääsi tulokseen 60 tai vähemmän vääriä tunnistuksia. Tasan tai alle 70 virhettä -luokassa oli 70 opetettua verkkoa ja 80 tai vähemmän virheitä -tasoon päästiin 124 tallennetun neuroverkon kanssa. Parhaan luokan, 50 virhettä tai vähemmän, tulokset tarkemmin taulukossa 19.

Taulukko 19. Datapohjaisen siirto-oppimisen ResNet-verkon tulokset

opetuskerta	epookki	testivirheet	validointivirheet	Opetusvirheet
1	480	50	17	1
2	480	42	17	0
2	110	49	18	8
0	220	49	26	3
1	280	50	19	22
2	780	49	22	14
2	390	47	19	8
1	500	48	25	12
1	920	48	16	1
2	700	45	15	0

MobileNet-arkkitehtuurilla parhaaseen luokkaan, 50 virheellistä tunnistusta tai vähemmän, pääsi yhteensä 10 tallennettua neuroverkkoa. Tähän luokkaan ei kuitenkaan päästy kaikilla opetuskerroilla. Kaikki opetuskerrat pääsivät kuitenkin luokkaan 60 virheellistä tunnistusta tai alle. 70 tai alle väärää tunnistusta tulokseen pääsi 92 tallennettua neuroverkkoa. Viimeiseen luokkaan 80 tai vähemmän vääriä tunnistuksia, pääsi 150 tallennettua verkkoa. Taulukossa 20 on parhaan luokan, 50 tai alle, tunnistuksen virheet eriteltynä.

Taulukko 20. Datapohjaisen siirto-oppimisen MobileNet-verkon tulokset

opetuskerta	epookki	testivirheet	validointivirheet	Opetusvirheet
2	450	48	23	10
2	580	47	20	4
2	560	50	25	2
2	520	43	21	5
2	180	49	12	1
2	960	45	19	6
2	680	44	21	2
2	540	44	16	4
2	370	47	18	2
3	240	48	9	0

DenseNet-arkkitehtuurin tulokset olivat parhaita myös tässä testissä. Tässä testissä myös saavutettiin kakkien testien paras tulos. Ensimmäisen opetuskerran testitulos epookilla 710 saavutti tuloksen, jossa virheellisiä tunnistuksia oli vain 33 kappaletta. Kahdella opetuskerralla kolmesta saavutettiin tulos, joka on 40 testivirhettä tai alle. Kaikilla opetuskerroilla päästiin 50 tai alle testivir-

heen tulokseen. Tämän kaltaisia tuloksia oli yhteensä 17. Tulos, 60 tai vähemmän virheellisiä tunnistuksia, saavutettiin 51 tallennetun verkon tuloksilla. 70 väärää tunnistusta tai alle -tulos saavutettiin 109 tallennetun verkon kanssa. Luokkaan 80 virhettä tai alle pääsi 170 tallennettua verkkoa. Parhaan tuloksen saavuttanut verkko 33 testivirhe jakautui 18 virheelliseen tunnistukseen, jossa ihminen virheellisesti tunnistettiin joksikin muuksi kuin ihmiseksi. Näistä virheellisistä tunnistuksista vain yksi tunnistui juuri ja juuri virheellisesti, alle 5 % erolla. Kaikissa muissa tapauksissa neuroverkko oli hyvin varma virheestään, yli 30 % erolla. Tapauksia, joissa jokin muu kuin ihminen tunnistettiin ihmiseksi, oli 15 kappaletta. Näistä melko pienellä virheellä oli yksi tunnistus, vähemmän kuin 10 % erolla. Kohtalaisen suurella virheellä, 20 % tai alle, oli kaksi kappaletta. Suurimmassa osassa, eli 12 kappaleessa tunnistuksia virheen suuruus oli vähintään 30 %. Taulukossa 21 on tulokset parhaaseen luokkaan, 40 virheellistä tunnistusta tai vähemmän, päässeet neuroverkot.

Taulukko 21. Datapohjaisen siirto-oppimisen DenseNet-verkon tulokset

opetuskerta	epookki	testivirheet	validointivirheet	Opetusvirheet
1	710	33	23	8
2	530	40	11	3
1	650	35	21	3

Tulokset pienellä konvoluutioverkolla, konv3verkkolla, olivat tämän testin selvästi huonoimmat. Ainoastaan neljä tallennettua neuroverkkoa pääsi luokkaan 60 väärää tunnistusta tai vähemmän, ja näistä kaikki olivat yhden opetuskerran aikana. Seuraavassa, 70 virheellistä tunnistusta tai vähemmän -luokassa, oli 25 tallennettua neuroverkkoa. Tähän luokkaan päästiin kaikilla kolmella opetuskerralla. Luokkaan 80 virhettä tai enemmän pääsi 113 tallennettua neuroverkkoa. Parhaat tulokset, 60 virhettä tai vähemmän, voidaan nähdä taulukosta 22.

Taulukko 22. Datapohjaisen siirto-oppimisen pienimmän konvoluutioverkon, Konv3verkon, tulokset

opetuskerta	epookki	testivirheet	validointivirheet	Opetusvirheet
2	490	60	21	4
2	300	57	24	18
2	520	58	24	13
2	660	60	17	5

Tulokset Konv5verkkolla olivat Konv3verkkoa huomattavasti paremmat. 9 tallennettua verkkoa pääsi tulokseen 50 tai vähemmän vääriä tunnistuksia. Tähän luokkaan pääsi tuloksia kahdesta opetuskerrasta. Seuraavassa luokassa, 60 väärää tunnistusta tai alle, oli kaikkien opetuskertojen tuloksia, joita oli yhteensä 54. Luokkaan 70 tai vähemmän vääriä tunnistuksia pääsi 115 tallennettua verkkoa. Viimeiseen luokkaan pääsi jopa 197 tallennettua neuroverkkoa. Tämä oli tässä tarkkuusluokassa paras tulos. Taulukossa 23 voidaan nähdä tarkemmat tulokset kategoriassa 50 virhettä tai vähemmän.

Taulukko 23. Datapohjaisen siirto-oppimisen pienen konvoluutioverkon tulokset

opetuskerta	epookki	testivirheet	validointivirheet	Opetusvirheet
2	900	47	14	3
2	200	44	23	15
3	400	49	20	10
2	360	48	23	9
2	160	50	18	9
2	510	45	17	0
2	330	47	18	4
1	160	47	32	33
1	760	46	22	0

Tulokset kaikkiaan olivat hieman odotettua parempia, eivätkä parhaat tulokset jääneet MobileNetin ja ResNetin tuloksista jälkeen kuin muutamalla tunnistuksella.

9 TESTIEN JA TULOSTEN YHTEENVETO

Neuroverkon kehitys onnistui hyvin, ja tulokset olivat paljon odotettuja parempia. Kaikkiin tutkimuskysymyksiin saatiin vastaukset ja konkreettinen tuotos, eli koulutettu neuroverkko, jolla voidaan tunnistaa ihminen veden pohjasta, toteutettiin. Toisena konstruktiona tuotettiin maailmanlaajuisestikin hyvin ainutlaatuinen datasetti kuluttajaluokan kaikuluotainten käytöstä ihmisen tunnistamiseen.

Mielenkiintoinen havainto testeistä oli, että parhaat tulokset saavutettiin yksikanavaisella kuvalla. Erot eivät olleet erityisen suuria, mutta todennäköisesti tämänkaltaiset neuroverkot ovat myös paremmin yleistettävissä monille muille kaikuluotaimille, jotka joko ovat yksikanavaisia, tai niiden tallennetun datan värimaailma eroaa opetusdatan väritasapainosta.

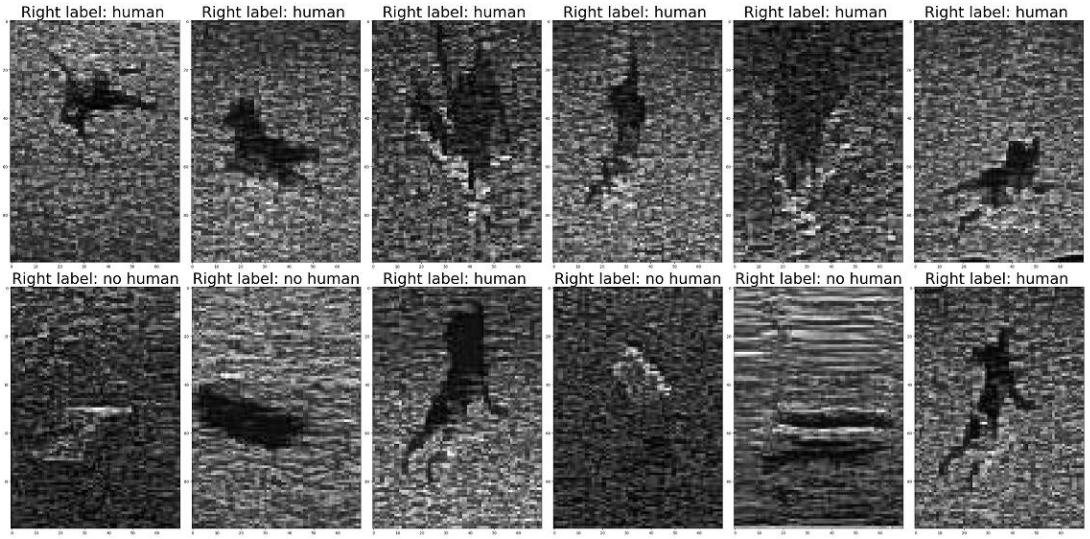
9.1 Tulokset – tutkimuskysymys 1

Ensimmäiseen ja tärkeimpään tutkimuskysymykseen – voiko kuluttajatason kaikuluotaimen datasta binaariluokittelijalla tunnistaa ihminen satunnaista valintaa tarkemmin – vastaus on kyllä. Ihmisen tunnistus on jopa huomattavasti satunnaisuutta tarkempi. Alun testeissä, joita ajettiin eri augmentoinneilla, olivat tulokset aina testidatalla huonoja, mikä antaa viitteitä siitä, etteivät hyvät tulokset myöskään ole satunnaisia. Kaikki loppupuolen testeissä hyvin toimineet verkot saivat myös hyviä tuloksia validointi- ja opetusdatalla testattaessa, mikä antaa myös uskoa siihen, että verkot ovat hyvin sisäistäneet ihmisen muodon.

Kaikkien testien tulosten synteesistä voidaan tutkia vielä, miten yhtenäisiä virheelliset tunnistukset ovat eri verkkojen ja opetuskertojen välillä. Kuuden parhaan tuloksen testisetillä antaneen verkon, eli kaikki tallennetut neuroverkot, joiden tuloksissa oli virheitä 40 tai vähemmän, virheet jakautuivat hieman eri kuville. Kaikissa kuudessa oli sama virhe 12 tapauksessa. Näistä kahdeksassa kappaleessa ihminen tunnistautui joksikin muuksi kuin ihmiseksi, ja 4 kertaa

jokin muu kuin ihminen tunnistautui ihmiseksi. Viidellä verkolla kuudesta oli 2 samaa virhettä. Neljällä verkolla kuudesta oli 14 sanaa virhettä. Kolmella verkolla oli yhteisiä virheitä 12. Kaksi neuroverkkoa kuudesta jakoi 10 virheellistä tunnistusta, ja 24 virhettä oli sellaisia, jotka tapahtuivat vain yhdellä tallennetuista neuroverkoista.

Näissä 12 kuvassa (kuvassa 10) tunnistus tapahtui väärin kaikilla kuudella parhaalla verkolla. Kahdeksassa kuvassa oli ihmisiä, jotka tunnistuivat väärin joksikin muuksi. Neljä kuvaa tunnistui virheellisesti ihmiseksi, vaikkei niissä ihmistä ollut. Kuvassa 10 voidaan nähdä, etteivät nämä neuroverkolle vaikeimmat kuvat ole kaikki ihmiselle kovin vaikeita. Mitään selkeää yhteistä tekijää ei virheellisille tunnistuksille löydy. Ihminen on eri asennoissa kuvaan nähden. Toisissa varjo on vahvemmin mukana, kun toisissa varjoa ei ole juuri lainkaan.



Kuva 10. Virheelliset tunnistukset, jotka toistuivat kaikilla kuudella parhaan tarkkuuden neuroverkolla

Kahden parhaan tuloksen, samalla opetuskerralla siirto-oppimisella opetetun verkon tapauksessa yhteisiä virheitä oli 26. Yhteensä eri virheitä näiden kahden verkon tunnistuksissa oli 42.

9.2 Tulokset – tutkimuskysymys 2

Seuraavaksi käsitellään toisen tutkimuskysymyksen – voidaanko pienellä datasetillä opettaa verkko niin, että tunnistus on vielä mahdollista testidatalla, jonka distribuutio poikkeaa merkittävästi opetusdatasta – tuloksia. Pienestä datasetistä riippumatta, todennäköisesti datamäärää suuresti kasvattavasta hyvin aggressiivisesta datan augmentoinnista johtuen, pienin Konv3verkko ei tunnistanut niin hyvin kuin suuremmat ResNet, DenseNet, MobileNet ja Konv5Verkko. Sen jälkeen, kun testeissä siirryttiin käyttämään yksikanavaista kuvaa, alkoi DenseNet-arkkitehtuuri voittamaan kaikki testit. Tosin erot olivat usein niin pieniä, että niiden selittäjänä voi olla myös osittain sattuma.

Parhaat koulutetut verkot pystyvät toimimaan 84,2 % tarkkuudella sellaiseen dataan, jota se ei ole koskaan nähnyt, ja joka on huomattavan erilaisilla parametreilla. Data oli tuotettu eri paikassa, eri kaikuluotaimen asetuksilla. Tämän lisäksi tunnistettavassa kohteessa olevan eron kanssa, kun opetusdatassa tunnistettavilla ihmisillä oli räpylät, ja testidatasetissä ei räpylöitä ollut. Joten vastaus tutkimuskysymykseen on myöntävä, pienikin datasetti riittää, kunhan datan augmentointi on riittävän aggressiivista, ja epookkien määrää kasvatetaan riittävästi.

9.3 Tulokset – tutkimuskysymys 3

Kolmannen tutkimuskysymyksen – voidaanko siirto-oppimista hyödyntämällä parantaa neuroverkon tarkkuutta – tulokset antavat viitteitä siirto-oppimisen hyödyistä tässäkin tilanteessa. Erot parhaaseen tulokseen ilman siirto-oppimista ja siirto-oppimisen kanssa ovat kuitenkin sen verran pieniä, ettei sattuman vaikutusta voida täysin poissulkea. Verkkopohjaisen siirto-oppimisen tulokset jäivät hieman jälkeen parhaista ei-siirto-oppimisen tuloksista. Verkkopohjaisen siirto-oppimisen paras tulos saatiin ensimmäisellä opetuskerralla epookin 810 kohdalla, jolloin testivirheitä oli 38, validointivirheitä 22 ja opetusvirheitä 12. Parhaaseen tulokseen ilman siirto-oppimista päästiin yksikanavaisella datalla DenseNet-verkolla, 520 epookin kohdalla, jolloin opetusvirheitä oli 36, validointivirheitä 16, sekä opetusvirheitä yksi kappale. Tässä saattaa myös

selittävänä tekijänä olla se, että verkon sisääntulodata oli resoluutioltaan pienempää. Tämä johtui jo opetetusta verkosta, jonka dimensiot olivat 96 x 96. Kun vastaavasti ilman siirto-oppimista sisään tuleva data on muodossa 122 x 122, ja on hyvin yleisesti tiedossa, että neuroverkot suoriutuvat suuremmilla kuvilla paremmin.

Kaikkien testien kaksi parasta tulosta saavutettiin siirto-oppimisen testissä. Nämä kaksi verkkoa olivat saman opetuskerran eri vaiheissa tallennettuja. Tässä ei voida kuitenkaan poissulkea, etteikö olisi ollut mahdollista päästä vielä parempiin tuloksiin ilman siirto-oppimista, lisäämällä epookkien määrä ja datan augmentointia. Erot parhaimpien verkkojen välillä olivat kuitenkin muutamien virheiden luokkaa. Tämän työn testien perusteella kuitenkin vaikuttaa siltä, että siirto-oppimisella voidaan saavuttaa parempia tuloksia.

10 JOHTOPÄÄTÖKSET JA POHDINTA

Työn tärkeimpänä ohjenuorana toimi konstruktiivisen tutkimusotteen empiirinen, kokeileva ja testaileva tutkimus, jonka tulosten ja teoreettisen viitekehyksen vuoropuhelu tuotti muokatun CLAIM-listan ohjaamana toimivat konstruktiot. Samalla saatiin paljon uutta tietoa ja oppia datankeräykseen, muokkaukseen sekä neuroverkkojen opetukseen. Työ täyttää kaikki muokattuun listaan mukaan otetut kohdat, joten tutkimus on toistettavissa ja samalla työssä on käytetty tunnettuja neuroverkkojen koulutuksen parhaita käytänteitä. Myös datankeräyksessä on yhtäläisesti noudatettu muokatun CLAIM-listan määrityksiä.

Työssä ensimmäisenä tuotettu konstruktio, ainutlaatuinen datasetti, onnistui hyvin. Riittävä määrä dataa pystyttiin keräämään neuroverkkojen opetusta varten. ROS-järjestelmä toimi erittäin hyvin datankeräyksessä. ROSBAG-muoto oli datalle hyvin toimiva muoto ja sen tarjoamat aikaleimat olivat hyvin hyödyllisiä varmistamaan kerätyn datan oikeellisuus. Lopullisen datasetin dataan valittu resoluutio aiheutti hieman hankaluuksia ja neliömuotoinen kuva olisi ollut monessa mielessä helpompi. Datankeräys käytetyillä menetelmillä oli kuitenkin hidasta. Datankeräyksen metodi oli hyvin validi ja luotettava. Datankeräys ja käytetyt laitteen on kuvattu tarkasti.

Pieni datamäärä aiheuttaa haasteita työn tuotoksen reaalimaailman suorituskyvyn määrittelylle. Tiedetään kuitenkin, että parhaiten oppineen verkon tunnistustarkkuus testidatalla on 84,2 %, validointidatalla 84,2 % ja opetusdatalla 97,6 %. Tässä hyvin rohkaisevaa on, että sekä testi- että validointidatan tulokset ovat hyvin samankaltaisia. Mielestäni tarkkaa validia tilastollista analyysiä tulosten yleistymisestä yleiseen tapaukseen ei pystytä tällä datalla tekemään, mutta ulkopuolisella datasetillä tunnistustarkkuus oli datamäärään nähden erittäin hyvä. Nämä tulokset ovat hyvin luotettavia.

Työhön valitut ulkopuoliset datasetit tukivat hyvin työn tekemistä ja niiden avulla pystyttiin hyvin testaamaan siirto-oppimisen eri metodeja. Käytetyn lääketieteellisen datan piirteet sopivat hyvin kaikuluotaindataan. Siirto-oppimisessa opetuksen virhe aleni merkittävästi nopeammin, kuin satunnaisesti alustetuilla verkoilla indikoiden, että kuvan aikaisemmin opitut ominaisuudet hyödyttivät siirto-oppimisprosessissa.

Kaiken kaikkiaan työ oli erittäin mielenkiintoinen ja opettava. Aiheen valinta oli mielestäni erittäin onnistunut, ja tulen jatkamaan mahdollisuuksien mukaan tutkimusta aiheen parissa. Työtä tehdessä on tullut hyvin selväksi, että pienilläkin dataseteillä tapahtuva opettaminen vie huomattavia laskentaresursseja. Tässä työssä tehtyjen laskentojen tekeminen hyvällä kannettavalla tietokoneella ja tehokkaalla erillisellä näytönohjaimella, olisi vienyt kuukausia.

Monessa mielessä pienillä datamäärillä toimiminen on vielä suuria datamääriä mielenkiintoisempaa niiden tuomien haasteiden vuoksi. Kirjallisuudessa on käsitelty jossain määrin vähemmän pienellä datamäärällä toimimista. On kuitenkin hyvin tyypillistä, että esimerkiksi lääketieteelliset datasetit ovat usein pieniä.

Työtä tehdessä esiin nousi merkittävä määrä kysymyksiä ja ideoita, jotka oli rajattava tämän työn ulkopuolelle, jotta työ tulee joskus valmiiksi. Seuraavassa pohditaan joitain esille nousseita jatkokehitystarpeita ja ideoita.

10.1 Datasetin suurentaminen

Tässä työssä käytetyn datasetin keräyksen jälkeen on markkinoille tullut kuluttajaluokan kaikuluotaimia, jotka pystyvät tuottamaan kuvaa myös paikallaan ollessaan. Luotain keilaa ympyrän muotoista aluetta viiva kerrallaan pyörien. Tämän kaltaisella anturilla molemmat vaiheet, datan keräys ja tunnistus, ovat helpompia. Dataa kerätessä voidaan pysyä paikallaan ja koko ajan saadaan uusia kuvia. Ihminen voi pohjassa vaihtaa asentoa pienin väliajoin ja myös hieman liikkua. Samoin tunnistus on huomattavan paljon tehokkaampaa, kun

tunnistusta ei tehdä vain kerran ohiajaessa vaan koko ajan, vaikka paikallaan pysyen. Näin tunnistusyrityksiä tulee myös ohiajaessa useita eri kulmista.

10.2 Hybridimallin kehitys

Tunnistuksessa perinteiset konenäön metodit eivät voita neuroverkkopohjaisia toteutuksia. Neuroverkkojen tunnistus saattaisi kuitenkin parantua, mikäli tunnistettavalle kuvalle tehdään ennen neuroverkolle siirtämistä muuttumattoman kernelin muutoksia, kuten kohinan vaimennusta, tai gradienttien laskentaa. Myös kokeilemisen arvoinen idea olisi testata tunnistusta kuvilla, joista on ensin reunantunnistuksella laskettu pelkät ääriviivat. Saman tyyppisiä menetelmiä voisi hyödyntää myös datan augmentointivaiheessa.

10.3 Synteettinen kaikuluotaindata

Datankeräys on hankalaa ja kallista. Yksi vaihtoehto tämän ongelman ratkaisuun on tuottaa dataa synteettisesti generoimalla. Tässä erityisesti kiinnostaa GAN-tyyppisten neuroverkkojen käyttö, jolloin voidaan esimerkiksi normaaleja kamerakuvia muuttaa, tyylin muutos neuroverkolla näyttämään kaikuluotainkuvilta. Jotain hyvin alustavia testejä tästä on kokeiltu, mutta vaatii vielä paljon enemmän kokeiluja, ennen kuin tuloksista voidaan sanoa mitään.

10.4 Ultraäänidatan tuloksien tarkempi tarkastelu

Työn odottamattomana sivutuloksena saatiin rohkaisevia tuloksia lääketieteellisellä datalla, joka ylitti alkuperäisten kirjoittajien tulokset. Tämä saattaa hyvin johtua kirjoittajien käyttämästä pienemmästä opetusdatan määrästä, kun suurempi osa on allokoitu testidatalle. Tässä työssä datan resoluutiota pienettiin, jotta neuroverkon dimensiot ovat yhteensopivat kaikuluotaindatan kanssa. Yleisesti tämän pitäisi laskea tunnistustarkkuutta. Yhtenä testaamisen arvoisena ideana on tarkistaa, pitääkö tämä yleinen käsitys paikkaansa tällä kohinaisella datalla. Joka tapauksessa alustavat tulokset ovat jatkoselvityksen arvoisia.

10.5 Metodin julkaisu kaikuluotaindatan käsittelystä ja analysoinnista

Työtä varten tehtyjen tietokantahakujen tuloksista huolimatta ei hyviä metodijulkaisuja kaikuluotaindatan käsittelyä varten löytynyt. Tiedeyhteisö varmasti hyötyisi standardisoidusta tavasta esittää datan keräys ja analysointimetodit, kuten lääketieteelliselle kuvantamisdatalle jo löytyy, esimerkiksi tässä työssä esitelty CLAIM-muistilista.

10.6 Koulutetun neuroverkon testaus pelastussukelluksessa

Työssä toteutettua tunnistinta olisi todella mielenkiintoista päästä testaamaan, joko veneeseen kiinnitetyn kaikuluotaimen kanssa, tai jopa sukelluslaitteeseen kiinnitettynä. Tässä pitäisi testata ja kehittää toimintaprosessia laitteiston ja sukelluspelastajien välille, ja myös testata konkreettisia etsintänopeuksien eroja perinteisen menetelmän ja kaikuluotainavusteisen etsinnän välillä.

LÄHTEET

Aaltonen, T. (2019). SUKELTAVAN ROBOTIN SUUNNITTELU JA TOTEUTUS Sähkö- ja automaatiotekniikan koulutusohjelma. [AMK-opinnäytetyö, Satakunnan ammattikorkeakoulu]

Almaghrabi, O. A., S., A. T., Albishri, H. M., & Moussa, T. A. A. (2014). ImageNet Large Scale Visual Recognition Challenge. Life Science Journal, 11(11), 764–772.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00444-8

Bridle, J. S. (1989). Training Stochastic Model Recognition Algorithms Training Stochastic Model Recognition Algorithms as Networks can lead to Maximum Mutual Information Estimation of Parameters. Advances in Neural Information Processing Systems 2

Brief History of Deep Learning from 1943-2019 [Timeline] - MLK - Machine Learning Knowledge. (n.d.). Noudettu 24. lokakuuta 2021, osoitteesta https://machinelearningknowledge.ai/brief-history-of-deep-learning/

Brigato, L., & Iocchi, L. (2020). A close look at deep learning with small data. Proceedings - International Conference on Pattern Recognition, 2490–2497. https://doi.org/10.1109/ICPR48806.2021.9412492

Burgos-Artizzu, X. P., Coronado-Gutierrez, D., Valenzuela-Alcaraz, B., Bonet-Carne, E., Eixarch, E., Crispi, F., & Gratacós, E. (2020a). FETAL_PLANES_DB: Common maternal-fetal ultrasound images. https://doi.org/10.5281/ZENODO.3904280

Burgos-Artizzu, X. P., Coronado-Gutiérrez, D., Valenzuela-Alcaraz, B., Bonet-Carne, E., Eixarch, E., Crispi, F., & Gratacós, E. (2020b). Evaluation of deep convolutional neural networks for automatic classification of common maternal fetal ultrasound planes. Scientific Reports, 10(1). https://doi.org/10.1038/s41598-020-67076-5

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. Journal of Machine Learning Research. 15.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. ISBN: 9780262035613

Grönman, J., Saarivirta, M., Aaltonen, T., & Kerminen, T. (2021). Review of Artificial Intelligence Applications in the ROS Ecosystem. 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), 1149–1153. https://doi.org/10.23919/MIPRO52101.2021.9596787

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hookana-Turunen, H. (1999). Tutkija, opettaja, akateeminen vaikuttaja ja käytännön toimija : professori Reino Majala 65 vuotta. [Turun kauppakorkeakoulu].

Howard, A. G. (2013). Some Improvements on Deep Convolutional Neural Network Based Image Classification. https://doi.org/https://doi.org/10.48550/arXiv.1312.5402

Howard, A. G., & Wang, W. (2012). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://arxiv.org/abs/1704.04861

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Jegorova, M. (2021). Generative Neural Data Synthesis for Autonomous Systems. July. https://doi.org/10.13140/RG.2.2.19876.60801

Kananen, J. (2014). Laadullinen tutkimus opinnäytetyönä : miten kirjoitan kvalitatiivisen opinnäytetyön vaihe vaiheelta. Jyväskylä : Jyväskylän ammattikorkeakoulu 2014.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8693 LNCS(PART 5), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Mongan, J., Moy, L., & Kahn, C. E. (2020). Checklist for Artificial Intelligence in Medical Imaging (CLAIM): A Guide for Authors and Reviewers. Radiology: Artificial Intelligence, 2(2), e200029. https://doi.org/10.1148/ryai.2020200029

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on International Conference on Machine Learning (s. 807–814)

Ng, A. (2018). machine-learning-yearning. http://www.mlyearning.org/

Ojasalo, K., Moilanen, T., & Ritalahti, J. (2014). Kehittämistyön menetelmät : uudenlaista osaamista liiketoimintaan. (3. uud. p.). Sanoma Pro.

Pekari, V. (2016). Tehtävä 483: ihmisen pelastaminen vedestä. [AMK-opinnäytetyö, Poliisiammattikorkeakoulu]

Puolitaival, M. (2015). Huomio kentän ideoille. Suomen Palomiesliitto SPAL Ry:N Jäsenlehti.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem, 779–788. https://doi.org/10.1109/CVPR.2016.91

Roberts, M., Driggs, D., Thorpe, M., Gilbey, J., Yeung, M., Ursprung, S., Aviles-Rivero, A. I., Etmann, C., McCague, C., Beer, L., Weir-McCall, J. R., Teng, Z., Gkrania-Klotsas, E., Ruggiero, A., Korhonen, A., Jefferson, E., Ako, E., Langs, G., Gozaliasl, G., … Schönlieb, C. B. (2021). Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. Nature Machine Intelligence, 3(3), 199–217. https://doi.org/10.1038/s42256-021-00307-0

Serban, A., Van Der Blom, K., Hoos, H., & Visser, J. (2020). Adoption and effects of software engineering best practices in machine learning. International Symposium on Empirical Software Engineering and Measurement. https://doi.org/10.1145/3382494.3410681

Shahria, T., Rabbi, S., Zaman, K. T., & Khan, M. M. (2019). Underwater Research and Rescue Robot. Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2019, March 2020, 1–5. https://doi.org/10.1109/ICECCT.2019.8869287

Sisäasiainministeriö. (2007). Pelastussukellusohje. http://urn.fi/URN:ISBN:978-952-491-281-5

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Teoksessa Journal of Machine Learning Research (Vsk. 15).

Subjugator auv source code. (ei pvm.). Noudettu 24. lokakuuta 2021, osoitteesta https://github.com/uf-mil/mil

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Szpilman, D., Bierens, J. L. M., Handley, A. J., & Orlowski, J. P. (2012). Current Concepts Drowning. https://www.nejm.org/doi/full/10.1056/NEJMra1013317

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. Lecture Notes in Computer Science (Including

Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11141 LNCS, 270–279. https://doi.org/10.1007/978-3-030-01424-7_27

Valdenegro-Toro, M., Preciado-Grijalva, A., & Wehbe, B. (n.d.). Releases · mvaldenegro/marine-debris-fls-datasets. Noudettu 12. kesäkuuta 2022, osoitteesta https://github.com/mvaldenegro/marine-debris-fls-datasets/releases/

Valdenegro-Toro, M., Preciado-Grijalva, A., & Wehbe, B. (2021). Pre-trained Models for Sonar Images. https://doi.org/10.23919/OCEANS44145.2021.9705825