Department of Computer Science

Machine Learning for Healthcare

Joel Jaskari

Aalto University

DOCTORAL THESES

Aalto University publication series
DOCTORAL THESES 161/2022

Machine Learning for Healthcare

Joel Jaskari

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall Circular Raw Materials Hub, C100 Aluminium (Vuorimiehentie 2, Espoo) of the school on 23 November 2022 at 12.00.

Aalto University
School of Science
Department of Computer Science

Supervising professor
Professor Arno Solin, Aalto University, Finland

Thesis advisors
Professor Simo Särkkä, Aalto University, Finland
Professor Kimmo Kaski, Aalto University, Finland

Preliminary examiners
Professor Matthew Blaschko, KU Leuven, Belgium
Professor Antti Airola, University of Turku, Finland

Opponent
Professor Chris Holmes, University of Oxford, United Kingdom

Aalto University publication series
DOCTORAL THESES 161/2022

ISBN 978-952-64-1008-1 (printed)
ISBN 978-952-64-1009-8 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-64-1009-8

Unigrafia Oy
Helsinki 2022

Finland

NORDIC SWAN ECOLABEL
Printed matter
4041-0619

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Joel Jaskari

Name of the doctoral thesis
Machine Learning for Healthcare

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL THESES 161/2022

Field of research Computer Science

Manuscript submitted 9 May 2022 **Date of the defence** 23 November 2022

Permission for public defence granted (date) 17 August 2022 **Language** English

☐ **Monograph** ☒ **Article thesis** ☐ **Essay thesis**

Abstract

Machine learning has been recently proposed for various medical applications. Especially the deep neural network based approach has been found to achieve state-of-the-art performance in various classification tasks. However, many of these studies use simplified classification systems, for example, the referable/non-referable system in the case of diabetic retinopathy classification. Moreover, the studies that have used clinical classification systems have not considered the uncertainty of the classifiers, which is of paramount interest in the medical field. In addition, extensive analysis of automatic segmentation algorithms that includes comparison to the interobserver variability of multiple radiologists' segmentations has not yet been performed for some challenging tasks, such as the automatic segmentation of the mandibular canals. The machine learning algorithms should also be able to be trained on local hospital data, which can pose issues relating to the amount of available training data.

This thesis considers machine learning for various tasks in healthcare using Finnish hospital data. Deep convolutional neural networks (CNNs) are utilized for diabetic retinopathy and macular edema classification based on clinical severity scales. In addition, approximate Bayesian deep learning approaches are systematically studied for uncertainty-aware diabetic retinopathy classification of clinical data. A connection is derived between the referral of uncertain classifications and reject option classification, and it is used to develop a novel uncertainty measure. A CNN approach will also be introduced for the segmentation of the mandibular canal in cone beam computed tomography volumes. The approach is then compared to the interobserver variability of multiple radiologists' segmentations of the canal. Lastly, this thesis will examine multiple machine learning approaches for very low birth weight neonate mortality and morbidity prediction.

The results suggest that even a relatively small set of Finnish hospital data can be utilized to train deep learning classifiers for diabetic retinopathy and macular edema classification with clinical classification systems. It also turns out that approximate Bayesian neural networks and the derived novel uncertainty measure can be used to accurately estimate the uncertainty in clinical diabetic retinopathy classification. The deep learning approach is shown to set a new state-of-the-art for the mandibular canal segmentation task and it is also found to localize the canals with lower variability than the interobserver variability of four radiologists. A random forest classifier turned out to outperform other methods in neonatal mortality and morbidity prediction.

Keywords machine learning, deep learning, approximate Bayesian deep learning, healthcare

ISBN (printed) 978-952-64-1008-1 **ISBN (pdf)** 978-952-64-1009-8

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2022

Pages 174 **urn** http://urn.fi/URN:ISBN:978-952-64-1009-8

Tekijä
Joel Jaskari

Väitöskirjan nimi
Koneoppiminen terveydenhuollossa

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL THESES 161/2022

Tutkimusala Tietotekniikka

Käsikirjoituksen pvm 09.05.2022 **Väitöspäivä** 23.11.2022

Väittelyluvan myöntämispäivä 17.08.2022 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Koneoppimista on lähiaikoina ehdotettu moniin lääketieteellisiin tehtäviin. Erityisesti syvillä neuroverkoilla on saavutettu erinomaisia tuloksia luokittelutehtävissä. Monet tutkimukset ovat kuitenkin käyttäneet yksinkertaistettuja luokittelujärjestelmiä, kuten lähetteen vaativuuden ennustaminen diabeettisen retinopatian tapauksessa. Kliinisiä vakavuusasteikkoja käyttävät tutkimukset eivät ole arvioineet luokittimien epävarmuutta, joka on tärkää tietää niiden soveltamiseen lääketieteen saralla. Automaattisten segmentaatioalgoritmien kattavaa analyysiä ja vertailua useamman radiologin väliseen vaihteluun ei ole lisäksi tehty joillekkin vaativille tehtäville, kuten mandibulaarikanavan automaattiselle segmentaatiolle. Koneoppimisalgoritmeja tulisi pystyä kouluttamaan sairaalaympäristön omalla aineistolla, joka saattaa olla haastavaa aineiston määrän suhteen.

Tämä väitöstyö käsittelee koneoppimisen soveltamista moniin lääketieteellisiin tehtäviin suomalaisten sairaala-aineistojen avulla. Syviä konvoluutioneuroverkkoja sovelletaan diabeettisen retinopatian ja makulaturvotuksen luokitteluun kliinisesti käytetyillä vakavuusasteikoilla. Myös bayesilaisten neuroverkkojen approksimaatioita tutkitaan epävarmuuden huomioivaan diabeettisen retinopatian luokitteluun kliinisellä aineistolla. Työssä luodaan yhteys epävarmuuspohjaisen luokittelujen hylkäämisen ja hylkäysvaihtoehtoluokituksen välille, jonka avulla kehitetään uusi epävarmuusmittari. Konvoluutioneuroverkkoa sovelletaan myös mandibulaarikanavan segmentaatioon kartiokeilatietokonetomografiakuvista ja sitä verrataan useamman radiologin väliseen vaihteluun tässä tehtävässä. Lopuksi tässä työssä tarkastellaan useita koneoppimismenetelmiä pikkukeskosten kuolleisuuden ja sairastuvuuden ennustamiseen, ja tutkitaan mitkä piirteet ovat tärkeitä tähän tehtävään.

Tulosten perusteella voidaan arvioida, että jopa suhteellisen pientä suomalaista sairaala-aineistoa voidaan käyttää diabeettisen retinopatian ja makulaturvotuksen luokitteluun kliinisesti käytetyillä vakavuusasteikoilla. Bayesilaisten neuroverkkojen approksimaatiot pystyvät myös hyödyntämään kehitettyä epävarmuusmittaria tarkkaan epävarmuuden arviointiin diabeettisen retinopatian luokittelussa kliinisellä vakavuusasteikolla. Syvän konvoluutioneuroverkon näytetään saavuttavan parempia tuloksia kuin aikaisemmat lähestymistavat mandibulaarikanavan segmentaatiossa. Sen näytetään myös paikallistavan mandibulaarikanavan pienemmällä vaihtelulla kuin useamman radiologin välinen vaihtelu. Satunnaismetsä -luokitin saavutti parempia tuloksia kuin muut menetelmät pikkukeskosten kuolleisuuden ja sairastuvuuden ennustamisessa.

Avainsanat koneoppiminen, syväoppiminen, bayesilaisten neuroverkkojen approksimaatiot, terveydenhuolto

ISBN (painettu) 978-952-64-1008-1 **ISBN (pdf)** 978-952-64-1009-8

ISSN (painettu) 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2022

Sivumäärä 174 **urn** http://urn.fi/URN:ISBN:978-952-64-1009-8

Preface

The research presented in this doctoral thesis started in 2018 and was completed in 2022. It was carried out at both the Department of Computer Science, Aalto University and the Department of Electrical Engineering and Automation, Aalto University. My research was funded by the Department of Electrical Engineering and Automation, Aalto University, the Department of Computer Science, Aalto University, Business Finland, the Academy of Finland, Nokia Solutions and Networks Oy, and the Department of Ophthalmology, Helsinki University Hospital and University of Helsinki with the highly appreciated grants from Evald and Hilda Nissi Foundation, Vaasa, Finland, and Glaucoma Research Foundation LUX, Helsinki, Finland.

The journey that has lead to this thesis began in 2017, when I had observed a summer job advertisement for a position that included deep learning and medical image analysis. I was (and still am) extremely fascinated about machine learning and interested in medical imaging, and so I decided to apply for the position. I want to thank Prof. Kimmo Kaski for choosing me for the position, and thus setting my path to becoming a researcher. I was fortunate to have the opportunity to continue to a master's thesis position under his supervision, and then to continue to my doctoral studies under the supervision of Prof. Simo Särkkä and Prof. Arno Solin with Prof. Kimmo Kaski as my instructor. I am thankful for the doctoral study opportunity, and for the guidance and support of Prof. Simo Särkkä, Prof. Arno Solin, and Prof. Kimmo Kaski during my studies. I want to also thank Prof. Leo Kärkkäinen for the opportunities in research and in the applications of pedagogy, as well as for the countless interesting discussions of the possibilities of deep learning and beyond. I want to collectively express my deepest gratitude to the four professors for all the knowledge they have imparted to me, and for aiding me in finding the joy of discovering the unknown.

I want to thank my colleague and a fellow doctoral student M.Sc. Jaakko Sahlsten for the past five years of friendship and research together. Brainstorming ideas, planning research avenues, and debugging code together

has certainly had a factor of ≥ 2 positive effect on the quality and quantity of our research together, as well as on the overall doctoral study experience of mine. I want to thank all the medical professionals who I have had the pleasure of collaborating with. I am grateful to D.Med.Sc. Kustaa Hietala and Title of Docent Paula Summanen for their expertise in eye diseases that has been instrumental for my research and also for my understanding of diabetic retinopathy. I also greatly appreciate D.D.S. Jorma Järnstedt and D.D.S. Helena Mehtonen for their expertise in dentomaxillofacial radiology and computed tomography in medicine. I want to thank Prof. Sture Andersson and D.Med.Sc. Markus Leskinen for their expertise in neonatology, and I also appreciate their prior knowledge of machine learning that has been extremely beneficial for our collaboration.

I want to thank all the people I have co-authored with for having the pleasure and the privilege. I greatly appreciate Prof. Theodoros Damoulas and Prof. Jeremias Knoblauch for their expertise in Bayesian deep learning and for their valuable insights in robust variational inference. I want to thank the members of the Sensor informatics and medical technology group for all the collaborations, interesting discussions, and, of course, for our weekly Journal Club meetings, which have always had exciting presentations and kept me in the loop with the latest developments in Bayesian filtering and smoothing. I am grateful for all the people I have shared the office with over the years for the pleasant working environment I have had. I want to thank Aalto Science-IT for their immense expertise in both the hardware and the software required for scientific computing.

I want to express my deepest gratitude to the Department of Ophthalmology, Helsinki University Hospital and University of Helsinki, the Department of Radiology, Tampere University Hospital, the Department of Ophthalmology, Central Finland Central Hospital, and the Children's Hospital, Helsinki University Hospital and University of Helsinki, as their collaboration is what has enabled there to be *machine learning for healthcare*, which I believe will be of great benefit to both the medical professionals and the patients in the future.

Last but not least, I want to thank my friends and family for their love and encouragement, and for their endless support for me in my endeavors.

Espoo, October 24, 2022,

Joel Jaskari

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Jaakko Sahlsten, Joel Jaskari, Jyri Kivinen, Lauri Turunen, Esa Jaanio, Kustaa Hietala, Kimmo Kaski. Deep Learning Fundus Image Analysis for Diabetic Retinopathy and Macular Edema Grading. *Scientific Reports*, Volume 9, Article number: 10750, July 2019.

II Joel Jaskari, Jaakko Sahlsten, Theodoros Damoulas, Jeremias Knoblauch, Simo Särkkä, Leo Kärkkäinen, Kustaa Hietala, Kimmo Kaski. Uncertainty-aware Deep Learning Methods for Robust Diabetic Retinopathy Classification. *IEEE Access*, Volume 10, Pages 76669-76681, July 2022.

III Joel Jaskari, Jaakko Sahlsten, Jorma Järnstedt, Helena Mehtonen, Kalle Karhu, Osku Sundqvist, Ari Hietanen, Vesa Varjonen, Vesa Mattila, Kimmo Kaski. Deep Learning Method for Mandibular Canal Segmentation in Dental Cone Beam Computed Tomography Volumes. *Scientific Reports*, Volume 10, Article number: 5842, April 2020.

IV Jorma Järnstedt, Jaakko Sahlsten, Joel Jaskari, Kimmo Kaski, Helena Mehtonen, Ziyuan Lin, Ari Hietanen, Osku Sundqvist, Vesa Varjonen, Vesa Mattila, Sangsom Prapayasotok, Sakarat Nalampang. Comparison of Deep Learning Segmentation and Multigrader-annotated CBCT Mandibular Canals. *Submitted to Scientific Reports and Accepted in September*, arXiv preprint arXiv:2205.13874, April 2022.

V Joel Jaskari, Janne Myllärinen, Markus Leskinen, Ali Bahrami Rad, Jaakko Hollmén, Sture Andersson, Simo Särkkä. Machine Learning Methods for Neonatal Mortality and Morbidity Classification. *IEEE Access*, Volume 8, Pages 123347-123358, June 2020.

Author's Contributions

Publication I: "Deep Learning Fundus Image Analysis for Diabetic Retinopathy and Macular Edema Grading"

Kaski and Hietala came up with the original idea. Jaskari, Sahlsten, and Kivinen designed the experiments, wrote the code, and analyzed the results. Turunen and Jaanio provided the data. All authors participated in writing the manuscript.

Publication II: "Uncertainty-aware Deep Learning Methods for Robust Diabetic Retinopathy Classification"

Jaskari, Sahlsten, Damoulas, Knoblauch, and Kaski came up with the original idea. Jaskari and Sahlsten designed the experiments, wrote the code and analyzed the results with joint effort. Jaskari came up with the idea of QWK-Risk with the help of Särkkä and derived the measure. All authors participated in writing the manuscript.

Publication III: "Deep Learning Method for Mandibular Canal Segmentation in Dental Cone Beam Computed Tomography Volumes"

The original idea was a joint effort of all authors. Jaskari and Sahlsten designed the experiments, wrote the code and analyzed the results. Jänstedt and Mehtonen provided the data. All authors participated in writing the manuscript.

Publication IV: "Comparison of Deep Learning Segmentation and Multigrader-annotated CBCT Mandibular Canals"

Jaskari, Jänstedt, Sahlsten, and Kaski came up with the original idea. Sahlsten and Järnstedt came up with most of the experiments and analyzed results. Jaskari came up with the rest of the experiments and assisted on the analysis of results. Sahlsten wrote most of the code. Lin came up with the new post processing algorithm. Jänstedt and Mehtonen provided the data. Jaskari, Jänstedt, Sahlsten, and Kaski wrote most of the manuscript and other authors gave useful comments and revised it.

Publication V: "Machine Learning Methods for Neonatal Mortality and Morbidity Classification"

Särkkä, Hollmén, and Andersson came up with the original idea. Jaskari came up with the idea of majority class under-sampling. Bahrami Rad came up with the idea of F1-score for classifier selection. Jaskari and Myllärinen wrote the code for machine learning methods. Jaskari analyzed the results and wrote the majority of the manuscript. Leskinen and Andersson wrote the medical parts of the manuscript. Other authors revised and gave useful comments about it.

Abbreviations

ASSD	Average symmetric surface distance
AUROC	Area under the receiver operating characteristic curve
BNN	Bayesian neural network
BPD	Bronchopulmonary dysplasia
CART	Classification and regression trees
CBCT	Cone beam computed tomography
CNN	Convolutional neural network
CT	Computed tomography
DNN	Deep neural network
ERM	Empirical risk minimization
ETDRS	Early Treatment Diabetic Retinopathy Study
FN	False negatives
FP	False positives
GDPR	General Data Protection Regulation
GP	Gaussian process
GVI	Generalized variational inference
HMC	Hamiltonian Monte Carlo
IAN	Inferior alveolar nerve
IQR	Interquartile range
KL	Kullback-Leibler

KNN	k-nearest neighbor
LDA	Linear discriminant analysis
LR	Logistic regression
MAP	Maximum a posteriori
MC	Monte Carlo
MCD	Mean curve distance
MDCT	Multi-detector computed tomography
MFVI	Mean field variational inference
ML	Maximum likelihood
MLE	Maximum likelihood estimate
NEC	Necrotizing enterocolitis
PIMEC	Proposed international clinical diabetic macular edema scale
PIRC	Proposed international clinical diabetic retinopathy scale
QDA	Quadratic discriminant analysis
QRDR	Ungradable/non-referable/referable diabetic retinopathy scale
QWK	Quadratic weighted Cohen's kappa
RDME	Referable diabetic macular edema
RDR	Referable diabetic retinopathy
ROP	Retinopathy of prematurity
RF	Random forest
SGD	Stochastic gradient descent
SMCD	Symmetric mean curve distance
STD	Standard deviation
TN	True negatives
TP	True positives
VLBW	Very low birth weight

Symbols

$a, \boldsymbol{a}, A$	A scalar, a vector, and a matrix, respectively
$x, \boldsymbol{x}$	Input variables
$y, \boldsymbol{y}$	Target variables
Bern	Bernoulli distribution
$\mathcal{N}$	Normal distribution
$\sim$	Distributed as
$\odot$	Hadamard product
$\|\cdot\|_p$	p-norm
$D_{KL}[\cdot\|\cdot]$	Kullback-Leibler divergence
$\mathcal{I}[\cdot]$	Indicator function
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^d$	The set of real vectors of dimension d
$\mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$	The set of real multidimensional arrays of dimension $d_1 \times d_2 \times \cdots \times d_N$

1. Introduction

By the year 2022, machine learning has been utilized to obtain impressive results in a wide range of problems, and it has become popular to the general public as a form of artificial intelligence. Furthermore, machine learning has already become a component of many everyday systems, such as search engines, social media applications, online video services, and even some cars. The rise in the popularity of machine learning has been largely driven by deep neural networks (DNNs) that have achieved state-of-the-art performance in, for example, large scale image classification [1, 2], image segmentation [3], image synthesis [4], and natural language processing [5]. The artificial neural networks were already proposed in 1957 [6] and periods of interest have emerged over the years, such as with the self-organizing map [7] and Oja's learning rule in the 1980s, the Helmholtz machine [8] in the mid 1990s, and the restricted Boltzmann machine [9] in the mid 2000s. The current re-emergence and widespread popularity of DNNs is largely due to the advances in high-performance computing power by graphical processing units (GPUs) and large sets of annotated data that have facilitated the training of these networks [10].

Machine learning has also been proposed for various medical tasks, ranging from diabetic retinopathy detection [11] and bone age estimation [12] to cell segmentation [13] and low-dose computed tomography denoising [14]. Despite the large quantity of medically oriented studies and that machine learning is being used in practice in many other areas, machine learning has yet to become a widely used practical tool in the medical domain. This is likely because the evaluation of algorithms differ between the technically and clinically oriented research, for example in the use of retrospective and prospective studies, that hinders the credibility of the claims made in machine learning oriented studies [15].

Another, still open question, is how to design more robust DNNs, in terms of the out-of-distribution performance and uncertainty quantification. It has been observed that modern DNNs are overconfident in the predictions they make, which means that in classification tasks they always place a high probability for one of the possible classes, even when the class is

incorrect [16]. In the medical domain, it is critical to know if the prediction of the system can be trusted or not, such that in the latter case the decision needs to be left for an expert.

This thesis considers machine learning for healthcare with a multidisciplinary view on the subject that encompasses both the technical and the medical research interests. The focus of this thesis is in the application of machine learning methods for various medical data modalities and prediction tasks with clinically used classification systems, as well as in a more clinically oriented validation of the human mandibular canal segmentation task, which is a type of medical segmentation task. In addition, uncertainty in deep learning for medical domain is also covered for a clinical multi-class classification scheme of diabetic retinopathy and a clinically oriented uncertainty measure is derived for this task. The contributions of this thesis are as follows.

- Development of a convolutional neural network approach for diabetic retinopathy and macular edema classification using clinical severity scales and a relatively small retinal image dataset.
- Comprehensive analysis of approximate Bayesian deep learning methods for uncertainty-aware diabetic retinopathy classification using clinical hospital data and a clinical severity scale.
- Derivation of a connection between the uncertainty-based referral process and reject option classification, and development of a novel uncertainty measure for clinical diabetic retinopathy classification.
- Development and interobserver variability analysis of a fully convolutional neural network for state-of-the-art mandibular canal segmentation.
- Systematic analysis of machine learning classifiers for neonatal mortality and morbidity prediction, and feature importance analysis for the tasks.

The structure of this thesis is as follows. First, Chapter 2 presents background on machine learning methods, models, and evaluation. Second, Chapter 3 describes deep learning and approximate Bayesian deep learning for diabetic retinopathy and macular edema classification based on Publication I and Publication II. Third, in Chapter 4, a deep learning approach for mandibular canal segmentation is presented and its performance is compared to the interobserver variability in the task, corresponding to Publication III and Publication IV. Then, machine learning classification of neonatal mortality and morbidity is covered in Chapter 5, with a focus on Publication V. Finally, Chapter 6 summarizes the Publications, and Chapter 7 concludes this thesis with a discussion.

2. Machine Learning

Machine learning is a term used for algorithms that can learn from data [17], as opposed to handcrafted algorithms that are designed to do a certain task. The motivation behind the paradigm of learning an algorithm can be that the problem is too complex to be solved by handcrafted algorithms [18] or simply that it is not known how to do certain tasks [19]. In addition, machine learning can be used to automate the generation of algorithms for analysis of large quantities of data and to provide insights to it [20].

In this thesis, x and y will denote the so-called "input" and "target" variables, respectively. Other terms used for these variables in literature are "independent variables" for x and "dependent variables" for y [21]. Typically the x is some measurement that can be easily obtained, for example a retinal color image. The y is often a label of category or some continuous assessment of x that can be harder to obtain, for example the diabetic retinopathy severity score based on a retinal image.

The notation used for the inputs and targets is as follows. Scalar inputs and targets are written as x, y and vectors or multidimensional arrays as $\boldsymbol{x}, \boldsymbol{y}$. When the input is a scalar it will be $x \in \mathbb{R}$, in the case of a vector $\boldsymbol{x} \in \mathbb{R}^d$, in the case of an image $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$, and in the case of a volume $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W \times D}$, where C denotes the number of channels, and H, W, and D denote the spatial height, width, and depth, respectively. When the realizations of the inputs and targets are referred to as parts of a dataset, then x_n and y_n refer to the n:th input and target of the dataset, respectively. The function defined by the machine learning model will be denoted as f. When it is important to include the parameters $\boldsymbol{\theta}$ of the model in the context, the notation will be $f_{\boldsymbol{\theta}}$.

2.1 Learning Problems

Supervised learning assumes that there exists some unknown function f^* that maps the inputs to the targets [20]:

$$\boldsymbol{y} = f^*(\boldsymbol{x}), \tag{2.1}$$

and that it is possible to train a machine learning model to approximate this function, i.e. to predict the value of the target variable given the input variable [17, 18, 21]. This can be written as $\boldsymbol{y} \approx \hat{\boldsymbol{y}} = f(\boldsymbol{x})$, where the $\hat{\boldsymbol{y}}$ is the approximation made by the model f. In order to train the model, a dataset of paired examples $\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$ is required. The term "supervised" represents that the target variable is given for each input variable, analogous to external feedback of what the correct answer is for a given example [17]. The final objective of supervised learning is to obtain a model that can predict the target for new inputs, i.e. data that is not a part of the training dataset. The performance of the model for the unseen data is called *generalization* [20] that will be covered later in this thesis.

Unsupervised learning considers a variety of tasks for learning properties of the input variables without the target variables. This usually means that the only available data to the model is $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. Classic examples of unsupervised learning are clustering [21], density estimation, and component analysis [18]. Modern examples include more general tasks, such as synthesis or generation of $\boldsymbol{x}$ [17] without an explicit probability density or mass function, for example using generative adversarial networks [22], learning to predict the values of missing parts of $\boldsymbol{x}$, for example the next token in a sequence [5] or missing pixel values in an image [23], and removal of added noise [24]. The two latter examples are also known as *self-supervised learning*, a term that some researchers prefer over the "unsupervised learning" [25], that represents that the target variables were created from the input variables.

The third type of the usual characterizations of learning is *reinforcement learning*, which deals with learning actions in an environment to maximize rewards [18]. This type of learning is often used to train agents that can play games, such as Go [26], and thus is less related to the subject of this thesis. There also exist variations of the supervised and unsupervised learning, such as *semi-supervised learning* that considers learning with partially annotated datasets [27].

2.2 Classification and Image Segmentation

This thesis considers the supervised learning problem and specifically the supervised learning tasks of *classification* and *image segmentation*. In

classification, the target variable belongs to a set discrete categories $\mathcal{C}$ [18]. The set of categories could be $\mathcal{C} = \{healthy, sick\}$ or $\mathcal{C} = \{cat, dog, \ldots, fish\}$, where the former case is said to be a "binary classification task", since there are two classes, and the latter is referred to as a "multiclass classification task", as there are multiple classes [18, 20]. For computer processing, the categories are encoded in a numerical manner by mapping them to numbers [17], similar to a dictionary in Python programming language for example. The following convention is used in this thesis: for binary classification the set of the classes is $\mathcal{C} = \{0, 1\}$, and for multiclass classification with K classes it is $\mathcal{C} = \{0, 1, \ldots, K-1\}$. Even though the classes are encoded with integers, this thesis considers only the *nominal classification* setting that has no ordering among the classes. The converse case is called *ordinal regression* [28].

This thesis considers classification with probabilistic modeling. This approach is beneficial when the data is noisy or when it can be considered as such given a lack of knowledge about the true data generating process [29]. The probabilistic approach models the conditional distribution of the target given the input $p(y \mid x) = f(x)$, instead of directly modeling $y \approx f(x)$, and thus can encode the uncertainty with the conditional probabilities [10]. The predicted label can be selected to minimize the probability of misclassification by [18]:

$$\hat{y} = \arg\max_c p(y = c \mid x). \tag{2.2}$$

In image segmentation, also called "semantic segmentation", each pixel (voxel) of an image (volume) is classified to belong to a certain class [30]. This means that in image (volumetric) segmentation the target is an image (volume) and the pixel (voxel) values of it correspond to what class that pixel (voxel) belongs to. Thus the target has typically the same spatial shape as the input and is spatially aligned with it. As an example, the input could be an X-ray image $\boldsymbol{x} \in \mathbb{R}^{1 \times H \times W}$ and a possible target segmentation with two classes $\boldsymbol{y} \in \{0, 1\}^{H \times W}$ contains 1 on pixels where the $\boldsymbol{x}$ contains a tumour and 0 elsewhere. The probabilistic approach is simple to extend to the segmentation task, by jointly modeling all the elements of the targets. In this thesis, these elements are assumed to be conditionally independent given the input. The predicted label for a target index i, j, k is then given by, here for a volumetric target:

$$\hat{\boldsymbol{y}}_{i,j,k} = \arg\max_c p(\boldsymbol{y}_{i,j,k} = c \mid \boldsymbol{x}). \tag{2.3}$$

Remark 2.1. The conditional independence of the elements of the targets, i.e. the pixel or voxel labels, given the input is a strong assumption, however, it is a common simplification used in segmentation and also other image modeling tasks. For example, the sum of the element-wise negative log-likelihood loss of a segmentation target [3] or a decoded image [31] is

computing $-\sum_{i=1}^{H}\sum_{j=1}^{W}\log p(\boldsymbol{y}_{i,j} \mid \boldsymbol{x}) = -\log\prod_{i=1}^{H}\prod_{j=1}^{W} p(\boldsymbol{y}_{i,j} \mid \boldsymbol{x})$, where the target $\boldsymbol{y}$ has two spatial dimensions in this example. Thus, it can be interpreted that we assume that $\prod_{i=1}^{H}\prod_{j=1}^{W} p(\boldsymbol{y}_{i,j} \mid \boldsymbol{x}) = p(\boldsymbol{y} \mid \boldsymbol{x})$, which means that the conditional distribution of the target elements is fully factorized, i.e. they are independent given the input.

2.3 Statistical Inference and Parameter Estimation

Most of the models considered in this thesis are parameterized such that the conditional distribution of the targets given the inputs can be written as:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}). \tag{2.4}$$

The training of the machine learning model then amounts to estimating the parameters $\boldsymbol{\theta}$. This section presents the *maximum likelihood* estimation, *maximum a posteriori* estimation, and *empirical risk minimization*.

Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$, the maximum likelihood (ML) estimates the parameters of the model by [17]:

$$\begin{aligned}\boldsymbol{\theta}_{MLE} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N \mid \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N, \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}),\end{aligned} \tag{2.5}$$

where the $\boldsymbol{\theta}_{MLE}$ is the maximum likelihood estimate (MLE) of the parameters and the factoring of the likelihood is by the assumption that all the examples are independent. The MLE is thus the parameter estimate that maximizes the probability of the targets given the inputs [10]. The ML estimation is equivalent to minimizing the negative log-likelihood (NLL) [10], that has a computationally convenient sum of log terms:

$$\boldsymbol{\theta}_{MLE} = \arg\min_{\boldsymbol{\theta}} -\sum_{i=1}^{N} \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}). \tag{2.6}$$

The MLE is a point estimate of the parameters, which means that the ML estimation yields a single estimate for each of the parameters. On the other hand, *Bayesian inference* [32] can be used to compute the Bayesian posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ of the parameters. It can be used to specify prior knowledge in the distribution of $\boldsymbol{\theta}$ in terms of a *prior distribution* $p(\boldsymbol{\theta})$ [33]. The data is then used to update the prior to obtain the best current estimate of the distribution of the parameters. The update rule is called *Bayes' theorem* (also called Bayes' rule) and is presented in Theorem 2.1.

Theorem 2.1. *Bayes' theorem: The posterior distribution of $\boldsymbol{\theta}$ given a likelihood function $p(\mathcal{D} \mid \boldsymbol{\theta})$, a prior $p(\boldsymbol{\theta})$, and evidence $p(\mathcal{D})$ is given by:*

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}.$$

In machine learning, the prediction task is usually of more interest than the analysis of the posterior distribution [20]. When the posterior distribution has been computed, the *posterior predictive distribution* can be used to predict the target $\boldsymbol{y}_*$ for a new input $\boldsymbol{x}_*$ by marginalization over the parameters [32]:

$$p(\boldsymbol{y}_* \mid \boldsymbol{x}_*, \mathcal{D}) = \int p(\boldsymbol{y}_* \mid \boldsymbol{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta}. \tag{2.7}$$

The integral on the right-hand side can be interpreted as a weighted average of the conditional probability over the posterior distribution [20]. It thus combines predictions obtained by different parameter configurations and includes the uncertainty in them by multiplying with the posterior density of the configuration.

Even though the Bayes' theorem provides a principled approach to learning the posterior distribution, it is often intractable to compute for more complex models, such as neural networks [17]. However, the parameter value that maximizes the posterior distribution, called maximum a posteriori (MAP) estimate [33], can be obtained efficiently. The MAP estimate is obtained by a similar optimization problem as the MLE:

$$\begin{aligned} \boldsymbol{\theta}_{MAP} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}) \\ &= \arg\max_{\boldsymbol{\theta}} \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} \\ &= \arg\min_{\boldsymbol{\theta}} -\sum_{i=1}^{N} \log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}). \end{aligned} \tag{2.8}$$

There are also posterior distribution approximation methods that yield approximate distribution of the parameters, instead of single point estimates. These methods will be introduced later in the context of deep neural networks.

Besides the probabilistic approach, there is an optimization-centric view to machine learning, called *empirical risk minimization* (ERM) [19]. The ERM is derived from the risk of a classifier [34] and can be used with any *loss function* [10]:

$$\boldsymbol{\theta}_{ERM} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} l(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i), \tag{2.9}$$

where l is the loss function.

The ERM can be seen to correspond to the ML estimation when:

$$l(f(\boldsymbol{x}_i), \boldsymbol{y}_i) = -\log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}), \tag{2.10}$$

and to the MAP estimation when:

$$l(f(\boldsymbol{x}_i), \boldsymbol{y}_i) = -\log p(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}) - \frac{1}{N} \log p(\boldsymbol{\theta}). \tag{2.11}$$

Indeed, the optimization problems can be written in a similar manner by defining a *batch loss*:

$$\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}, l) = \sum_{i=1}^{N} l(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i), \tag{2.12}$$

where the loss function l defines what approach is used, and the optimal parameter values are determined by:

$$\boldsymbol{\theta}_{opt} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}, \boldsymbol{\theta}, l). \tag{2.13}$$

2.4 Evaluation Measures

The performance of the classification and segmentation models is often evaluated using multiple different measures to analyze the performance more comprehensively or in a more human interpretable manner. It is also common that the loss value is not interesting itself, but the loss acts as a surrogate for some other measure that cannot be easily optimized, for example the negative log-likelihood of a Bernoulli or categorical distribution acts as a surrogate loss for the classification error [20].

This section will introduce the evaluation measures used in the Publications. Publication I, Publication II, and Publication V consider medical classification tasks, for which important measures are *sensitivity*, *specificity*, *precision*, *F1-score*, *area under the receiver operating characteristic curve*, and *quadratic weighted Cohen's kappa*. Publication III and Publication IV consider mandibular canal segmentation, for which the (*Søresen*) *Dice coefficient*, *mean curve distance*, and *average symmetric surface distance* were used.

In binary classification, one of the classes is often called the "positive" and the other the "negative" class. In this thesis, the positive detection is always used for the detection of mortality or disease that is denoted as the class 1, and the negative detection is used for the normal condition that is denoted as the class 0. The true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) describe how the positives and negatives were detected correctly or incorrectly. These are computed as [20]:

$$TP = \sum_{i} \mathcal{I}[\hat{y}_i = 1, y_i = 1], \tag{2.14}$$

$$TN = \sum_{i} \mathcal{I}[\hat{y}_i = 0, y_i = 0], \tag{2.15}$$

$$FP = \sum_{i} \mathcal{I}[\hat{y}_i = 1, y_i = 0], \tag{2.16}$$

$$FN = \sum_i \mathcal{I}[\hat{y}_i = 0, y_i = 1], \tag{2.17}$$

where $\mathcal{I}$ is the indicator function and $\hat{y}$ is the predicted class label that has the highest conditional probability, which was given in Equation (2.2). Sensitivity and specificity are defined using these quantities as:

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \tag{2.18}$$

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{2.19}$$

Sensitivity thus measures the proportion of the positive cases that the model detected correctly and specificity the proportion of the negative cases that the model detected correctly. Precision is given by [20]:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{2.20}$$

It measures the proportion of correct positive detections to all the positive detections.

The F1-score is computed as the harmonic mean of sensitivity and precision [20]:

$$\text{F1-score} = 2\,\frac{\text{Sensitivity} \cdot \text{Precision}}{\text{Sensitivity} + \text{Precision}}. \tag{2.21}$$

It can be used when the performance of the model for the positive cases is important, because it does not take the true negatives into account [35]. When the dataset has a minority positive class distribution, the F1-score is beneficial over accuracy, i.e. the proportion of all correct detections, as a model that detects very few positive cases and almost always predicts the negative class can reach a high accuracy, but a very low F1-score.

In the binary case, the selection of the class label that has the highest conditional probability can also be written as:

$$\hat{y} = \begin{cases} 1, \text{ if } p(y = 1 \mid \boldsymbol{x}) \geq \tau, \\ 0, \text{ else,} \end{cases} \tag{2.22}$$

where τ is a threshold parameter, also called an *operating point* [36], with a value of 0.5. If sensitivity and specificity are computed as a function of τ, the resulting curve is the *receiver operating characteristic* (ROC) curve [20]. The area under the ROC curve is the *area under the receiver operating characteristic curve* (AUROC). The AUROC value is in the interval of $[0, 1]$, and it can be interpreted as an estimate of the probability that a classifier is able to correctly identify from two samples which is a positive and which is a negative case [37]. Thus, AUROC values close to 1.0 indicate high performance and values close to 0.5 that the performance is close to random guessing.

Cohen's kappa is a measure of agreement between two graders for classification [38]. The weighted Cohen's kappa introduces weights for different disagreement scenarios [39]. It requires a disagreement weight matrix W, the confusion matrix C of a classifier, and the expected agreement matrix E computed from the confusion matrix. For a classification task with K classes, these matrices are all $K \times K$ matrices. The confusion matrix is defined as:

$$C_{i,j} = \sum_{n=1}^{N} \mathcal{I}[\hat{y}_n = i, y_n = j]. \tag{2.23}$$

The element i, j of the matrix thus contains the number of examples that had the target j but were classified as i. The expected agreement matrix E is defined using the confusion matrix as:

$$E_{i,j} = \frac{1}{N} \sum_{a=1}^{K} C_{i,a} \sum_{b=1}^{K} C_{b,j}. \tag{2.24}$$

The weighted Cohen's kappa is then defined as [39]:

$$\kappa = 1 - \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} W_{i,j} C_{i,j}}{\sum_{i=1}^{K} \sum_{j=1}^{K} W_{i,j} E_{i,j}}. \tag{2.25}$$

The quadratic weighted Cohen's kappa (QWK) weights the disagreements based on the square of the difference of the class labels:

$$\kappa_{QW} = 1 - \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} (i-j)^2 C_{i,j}}{\sum_{i=1}^{K} \sum_{j=1}^{K} (i-j)^2 E_{i,j}}. \tag{2.26}$$

In segmentation, the Dice coefficient can be used to measure the quality of a segmentation with two classes. The Dice coefficient is another name for the F1-score, however, this thesis makes a distinction between these, such that the F1-score is calculated over a dataset of classification targets and the corresponding predictions, whereas the Dice coefficient is calculated for a single segmentation target and prediction. Indeed, in Publication III, the segmentation performance is evaluated as the mean Dice coefficient that is calculated as the average of the Dice coefficients obtained for all the segmentations. Publication III and Publication IV concentrate on volumetric segmentation of the mandibular canal, which is a task with two segmentation targets: the mandibular canal (class 1) and non-mandibular canal (class 0) regions. Thus, this thesis considers volumetric predictions and targets with two possible classes. Let $\boldsymbol{y}, \hat{\boldsymbol{y}} \in \{0, 1\}^{H \times W \times D}$, where $\hat{\boldsymbol{y}}$ is computed with Equation (2.3). The Dice coefficient is then computed as [40]:

$$\text{Dice}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = 2 \frac{|\boldsymbol{y} \cap \hat{\boldsymbol{y}}|}{|\boldsymbol{y}| + |\hat{\boldsymbol{y}}|} = 2 \frac{\sum_{i,j,k} \boldsymbol{y}_{i,j,k} \, \hat{\boldsymbol{y}}_{i,j,k}}{\sum_{i,j,k} \boldsymbol{y}_{i,j,k} + \sum_{i,j,k} \hat{\boldsymbol{y}}_{i,j,k}}, \tag{2.27}$$

where $|\cdot|$ is the cardinality of a set.

In Publication III and Publication IV, the mandibular canal localization accuracy is calculated using the mean curve distance (MCD). It measures the average distance from one curve to another. The curves are assumed to be collections of coordinates in three dimensions. The "point to curve" distance is defined for a point $\boldsymbol{x}$ and a curve defined by the set S by:

$$d(\boldsymbol{x}, S) = \min_{\boldsymbol{s} \in S} ||\boldsymbol{x} - \boldsymbol{s}||_2. \tag{2.28}$$

For two curves A and B: $A = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N\}$ and $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_M\}$, the MCD is then computed as [41]:

$$MCD(A, B) = \frac{1}{|A|} \sum_{\boldsymbol{a} \in A} d(\boldsymbol{a}, B). \tag{2.29}$$

It should be noted that the MCD is not a symmetric measure in the arguments, as changing the order of the arguments can produce a different MCD value. In Publication IV, a symmetric version of the MCD is proposed, which will be described later.

The average symmetric surface distance (ASSD) computes the average distance between two sets of segmentation surface voxels in a symmetric manner [42]. This resembles the MCD but is symmetric in the arguments:

$$ASSD(A, B) = \frac{1}{|A| + |B|} \left(\sum_{i=1}^{N} \min_{\boldsymbol{b} \in B} ||\boldsymbol{a}_i - \boldsymbol{b}||_2 + \sum_{j=1}^{M} \min_{\boldsymbol{a} \in A} ||\boldsymbol{b}_j - \boldsymbol{a}||_2 \right), \tag{2.30}$$

where the A and B are now the sets of surface voxel coordinates.

2.5 Generalization in Machine Learning

The performance of a machine learning model on data that has not been used to train it is called *generalization* [17], and good generalization is the goal of most machine learning algorithms [18]. Indeed, perfect performance can be easily obtained for the training data, by simply memorizing all the inputs and the corresponding targets, and then always returning the target for a corresponding query train input, for example using a 1-nearest neighbor model [20]. This type of phenomenon is called *overfitting*, which means that the machine learning algorithm performs well on the training data but does not generalize to unseen data [17]. Thus, to estimate generalization, the performance needs to be evaluated on data that has not been used in the training.

The set of data that is used for training is called the *training set* and the data used for estimating generalization is called the *test set* [17]. A common approach is to divide all the available data to training and test sets, such that no data exists simultaneously in both of the sets. For uncurated data,

such as clinical hospital datasets, there can also exist near duplicate data, for example multiple retinal images for a single patient, that can lead to overestimation of the performance. In the Publications I–V, the data has been divided to the training and test sets such that no patient exists simultaneously in both of the sets. The test data is left untouched until the machine learning model can be deemed to be complete in every aspect [20].

It is also desirable to estimate the generalization performance of a model while training it. To achieve this, a portion of the training set can be selected as the so-called *validation set* that is not used to directly optimize the parameters, but is used to estimate the generalization performance during the training [17]. It is also often used to select the so-called *hyperparameters* that are parameters of the model that cannot be directly estimated with MLE, MAP, or ERM approaches. Instead, the hyperparameters can be estimated by training the model with different hyperparameter values and selecting the best hyperparameter values based on the validation set performance [17]. Even though the validation set is not used to directly optimize the parameters, the model can indirectly overfit to it when it is used for the hyperparameter selection. Thus, the results evaluated on the validation set can be over-optimistic, and it is why the test set is needed to estimate the true generalization performance [18].

If there is only a small amount of data available, the so-called *K-fold cross validation* can be used [18]. In K-fold cross validation, the dataset is divided into K (integer ≥ 2) folds. The folds are then iterated, by using one fold as the test set and the rest of the folds as the training set. In the end, the test results of each fold are averaged to estimate the generalization performance. When the hyperparameters need to be estimated, the so-called *nested K-fold cross validation* can be used to avoid overoptimistic results due to fitting the hyperparameters with the full set [43]. It does not use the test folds for the selection of the hyperparameters, but rather performs a cross validation loop for each of the inner K-fold training sets. After the hyperparameters have been optimized, the folds are used as in the K-fold cross validation.

2.6 Machine Learning Models

In the Publications I–IV, deep convolutional neural networks were used for classification and segmentation tasks. In Publication V, the logistic regression, linear and quadratic discriminant analysis, k-nearest neighbor, support vector machine, Gaussian process, and random forest classifiers were used as the machine learning models. This section will first briefly describe the machine learning models used in Publication V and then introduce deep neural networks in a more detailed manner. Interested reader can refer to Hastie et al. [21], Rasmussen and Williams [44], and Murphy

[10] for a comprehensive review of the methods used in Publication V.

Logistic Regression Classifier

Logistic regression is a linear classifier, which means that the *decision boundary* is (or boundaries are) linear. For K possible target classes, sometimes called multiclass logistic regression, i.e. $y \in \mathcal{C} = \{0, 1, \ldots, K-1\}$, the input $\boldsymbol{x}$ is mapped to the conditional probabilities of each class with a linear transform and an activation function that ensures that the probabilities sum up to one [18]. The linear transform is determined by a weight matrix $W \in \mathbb{R}^{d \times K}$ and a bias vector $\boldsymbol{b} \in \mathbb{R}^K$, which are the parameters of the model, and the activation function is the so-called *softmax* function. The conditional probability of a class i is then given by:

$$p(y = i \mid \boldsymbol{x}, W, \boldsymbol{b}) = \frac{\exp(W_i^T \boldsymbol{x} + \boldsymbol{b}_i)}{\sum_{j=0}^{K-1} \exp(W_j^T \boldsymbol{x} + \boldsymbol{b}_j)}, \tag{2.31}$$

where W_i^T denotes the i:th row of the transposed matrix.

Remark 2.2. The multiclass logistic regression presented in Equation (2.31) is actually overparameterized, as the parameters of one class, for example a, can be ignored and the conditional probability of that class can be computed as $1 - \sum_{j \in \mathcal{C} \setminus \{a\}} p(y = j \mid \boldsymbol{x}, W, \boldsymbol{b})$. However, the overparameterized version is closely related to how it is usually implemented as a layer of neural networks, and thus used in this thesis. The $K - 1$ parameterized version is presented in Hastie et al. [21, p. 119], for example.

For binary logistic regression, where $y \in \{0, 1\}$, it can be seen that Equation (2.31) becomes for class 1:

$$\begin{aligned} p(y = 1 \mid \boldsymbol{x}, W, \boldsymbol{b}) &= \frac{\exp(W_1^T \boldsymbol{x} + \boldsymbol{b}_1)}{\sum_{j=0}^{1} \exp(W_i^T \boldsymbol{x} + \boldsymbol{b}_j)} \\ &= \frac{\exp(W_1^T \boldsymbol{x} + \boldsymbol{b}_1)}{\exp(W_1^T \boldsymbol{x} + \boldsymbol{b}_1) + \exp(W_0^T \boldsymbol{x} + \boldsymbol{b}_0)} \\ &= \frac{1}{1 + \exp(-((W_1 - W_0)^T \boldsymbol{x} + (\boldsymbol{b}_1 - \boldsymbol{b}_0)))} \\ &= \frac{1}{1 + \exp(-(\boldsymbol{w}^T \boldsymbol{x} + b))}. \end{aligned} \tag{2.32}$$

In the binary case, the model can be defined with just one weight vector and bias, because the probability of the second class can be computed by $1 - p(y = 1 \mid \boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b})$. The function $1/(1 + \exp(-(\cdot)))$ is called the *logistic sigmoid* function [18].

The logistic regression parameters are typically learned by ML or MAP estimation. As there is no analytical solution for the parameters, iterative gradient based methods need to be used, such as the Newton-Raphson

algorithm, to optimize the parameter values [21]. Detailed example of how the logistic regression model is trained using Newton-Raphson can be found in Bishop [18, p. 208]. A general overview of gradient based optimization is presented later.

Discriminant Analysis Classifiers

The linear discriminant analysis (LDA) is another linear classifier. The quadratic discriminant analysis (QDA) is a quadratic classifier, which means that the decision boundary is a quadratic function. LDA and QDA are also *generative classifiers*, i.e. they model the joint distribution $p(y, \boldsymbol{x})$. Both of these models assume that the conditional distribution of the input given the target $p(\boldsymbol{x} \mid y)$ is a Gaussian, however, the LDA additionally assumes that the covariance matrix is shared among the conditional densities [21].

For a classification task with K classes, Theorem 2.1 can be used to derive:

$$\begin{aligned} p(y = i \mid \boldsymbol{x}) &= \frac{p(\boldsymbol{x} \mid y = i)p(y = i)}{p(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x} \mid y = i)p(y = i)}{\sum_{j=0}^{K-1} p(\boldsymbol{x} \mid y = j)p(y = j)}. \end{aligned} \tag{2.33}$$

Equation (2.33) is then for the LDA:

$$p(y = i \mid \boldsymbol{x}) = \frac{\frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)\right)p(y = i)}{\sum_{j=0}^{K-1} \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_j)\right)p(y = j)}, \tag{2.34}$$

and for the QDA:

$$p(y = i \mid \boldsymbol{x}) = \frac{\frac{1}{|2\pi\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)\right)p(y = i)}{\sum_{j=0}^{K-1} \frac{1}{|2\pi\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_j)\right)p(y = j)}, \tag{2.35}$$

where the parameters are $\boldsymbol{\mu}_k$, Σ, and Σ_k [21]. The maximum likelihood estimation can be used to find analytical solutions for the parameters, presented in, for example, Hastie et al. [21, pp. 109–110].

k-Nearest Neighbor Classifier

The k-nearest neighbor (KNN) classifier is a conceptually simple nonparametric method. The training data is used to classify new query points based on how close they are to the training data. For a metric $d(\cdot, \cdot)$, such

as the Euclidean distance, and the number of neighbors $k \in \mathbb{Z}_+$ (a positive integer), chosen beforehand, the KNN algorithm finds the k closest training data inputs and selects the predicted target as the one that is the most frequent among the closest inputs [21]. The classification rule can be written for a query point $\boldsymbol{x}_*$ as:

$$\begin{aligned} S &= \underset{s \in \binom{\{1,\dots,N\}}{k}}{\arg\min} \sum_{i \in s} d(\boldsymbol{x}_*, \boldsymbol{x}_i), \\ \hat{y}_* &= \underset{c}{\arg\max} \sum_{j \in S} \mathcal{I}[y_j = c]. \end{aligned} \tag{2.36}$$

It is also possible to estimate the posterior probability of y_* by

$$p(y_* = i \mid x_*, \mathcal{D}, k) = \frac{k_i}{k}, \tag{2.37}$$

where k_i is the number of nearest neighbors belonging to class i. A detailed derivation of the result is presented in Bishop [18, p. 125].

Support Vector Machine Classifier

The support vector machine (SVM) classifier is a type of maximum margin classifier [18] that in the linear case resembles the logistic regression classifier. The linear SVM model for binary classification computes $\boldsymbol{w}^T\boldsymbol{x}+b$, and when the classes are encoded as $y \in \{-1, 1\}$, the classification is performed by $\hat{y} = sign(\boldsymbol{w}^T\boldsymbol{x}+b)$ [21]. It can be seen that, similar to logistic regression, the model defines a hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$, but instead of using the logistic sigmoid function to estimate the probability, the SVM simply uses the sign operator to predict the class label, and thus does not provide any probabilities [21].

The training of SVMs is motivated by finding the hyperplane that has the largest distance to the closest training input, called the *margin*, while separating the two classes to the respective sides of the hyperplane [34]. The maximum margin principle can then be used to find the optimal parameters [34]:

$$(\boldsymbol{w}, b) = \underset{\boldsymbol{w},b}{\arg\max} \min_{i \in \{1..,N\}} \frac{y_i(\boldsymbol{w}^T\boldsymbol{x} + b)}{||\boldsymbol{w}||_2}. \tag{2.38}$$

In the end, the closest inputs define the hyperplane and are called the *support vectors* [34]. Equation (2.38) can be represented as a quadratic programming problem [34] and it can also be relaxed to allow for some misclassifications with the use of the so-called slack variables [21]. Furthermore, the SVM model can be extended to define a nonlinear decision boundary with the use of a kernel function [18]. The kernel SVM is described in depth in Hastie et al. [21, pp. 423–429].

Gaussian Process Classifier

Gaussian processes (GPs) [44] are stochastic processes that can be used to define probability distributions over functions. If a random function $f(\boldsymbol{x})$ is a GP, then it is completely characterized by the *mean function* $m(\boldsymbol{x})$ and the *covariance function* $K(\boldsymbol{x}, \boldsymbol{x}')$, and additionally, if the function is evaluated for a finite collection of inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the joint distribution of the function values is a multivariate Gaussian distribution [44]:

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')), \tag{2.39}$$

$$\begin{aligned} &p(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N) \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \\ &\mathcal{N}\left(\begin{bmatrix} m(\boldsymbol{x}_1) \\ \vdots \\ m(\boldsymbol{x}_N) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \ldots & K(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_N, \boldsymbol{x}_1) & \ldots & K(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix} \right). \end{aligned} \tag{2.40}$$

The mean function is often set to zero, and thus the parameters $\boldsymbol{\theta}$ of the GP model are located in the kernel function [44].

In classification, the random function is regarded as a *latent function* that is not observed, and the target variables are thought to be distributed conditional to it $p(y \mid \boldsymbol{f})$ [44]. This can be modelled by applying the logistic sigmoid function on the latent function. The latent function can be marginalized out to obtain the marginal likelihood of the targets [44]:

$$p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \int p(y \mid \boldsymbol{f}) p(\boldsymbol{f} \mid \boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{f}, \tag{2.41}$$

which in turn can be used with gradient based optimization methods to learn the parameters $\boldsymbol{\theta}$ [44]. For a query point $\boldsymbol{x}_*$, the model can be used to predict the label by:

$$p(y_* \mid \boldsymbol{x}_*, \mathcal{D}, \boldsymbol{\theta}) = \int p(y_* \mid \boldsymbol{f}_*) p(\boldsymbol{f}_* \mid \boldsymbol{x}_*, \mathcal{D}, \boldsymbol{\theta}) d\boldsymbol{f}_*. \tag{2.42}$$

For exact details of training and implementation of GPs for classification, the reader is referred to Rasmussen and Williams [44, ch. 3 & 5].

Random Forest Classifier

Random forest is a type of ensemble model that consists of decision trees [21]. To understand the random forest classifier, the classification and regression trees (CART) approach is described in brief first. In CART, a decision tree is grown to iteratively split the input space to two regions along a coordinate axis [20]. In classification, this split is determined in a way to separate the classes in the input space, for example by selecting the

split location to minimize the classification error. The splitting process is terminated when the improvement in performance does not exceed some predetermined threshold [20]. The final tree then performs a classification by:

$$p(y = i \mid \boldsymbol{x}, \boldsymbol{\theta}) = \sum_{r \in R} w_r^{(i)} \mathcal{I}[\boldsymbol{x} \in S_r], \tag{2.43}$$

where the R is the set of regions, $w_r^{(i)}$ is the average target class i within the region (computed using the training set), S_r is the region in the input space, and $\boldsymbol{\theta}$ includes the parameters of the splits [20]. For example, if the input has two dimensions $\boldsymbol{x} = [x_1, x_2]$, the first split might happen on x_2 on threshold θ_1, and then the input space is divided to two regions $R_1 = \{\boldsymbol{x} \mid x_2 > \theta_1\}$ and $R_2 = \{\boldsymbol{x} \mid x_2 \leq \theta_1\}$. For these two regions new splitting rules are then determined, for example: $R_3 = \{\boldsymbol{x} \mid x_1 > \theta_2, \boldsymbol{x} \in R_1\}$, $R_4 = \{\boldsymbol{x} \mid x_1 \leq \theta_2, \boldsymbol{x} \in R_1\}$, $R_5 = \{\boldsymbol{x} \mid x_1 > \theta_3, \boldsymbol{x} \in R_2\}$, and $R_6 = \{\boldsymbol{x} \mid x_1 \leq \theta_3, \boldsymbol{x} \in R_2\}$. The final terminal regions, i.e. those regions that are not split anymore, are called the "leaves" of the tree [21], and correspond to the regions in the set R in Equation (2.43).

The random forest algorithm consists of training multiple decision trees, each with a bootstrap resample of the training data. Additionally, when determining a splitting rule, the splitting variable is selected from a random sample of all variables, in contrast to the CART method that selects the splitting variable from all the variables [21]. This stochasticity allows to reduce the correlation between the individual trees to decrease the *variance* of the random forest estimator as a whole [20].

Deep Neural Networks

Feedforward neural networks are a class of models that typically consist of a chain of linear and nonlinear functions [10]. The term "feedforward" refers to that there are no feedback connections in the computational graph [17]. This thesis considers only the feedforward type neural networks, and thus "neural networks" will refer specifically to the "feedforward neural networks".

Deep neural networks (DNNs) are a composition of L functions $f_1, \ldots, f_L$ that are also called *layers* [17]. The composition of more functions can be thought to increase the depth of the composite function, i.e. the neural network, that inspires the "deep" in the name "deep neural networks" [17]. There is no unanimous definition of how many layers are required for a neural network to become a DNN, and thus this thesis calls neural networks with more than two layers "deep neural networks", as opposed to the more traditional neural networks with two layers. For a *fully-connected* DNN, also called the multilayer perceptron (MLP), the layers are often defined as [10]:

$$\boldsymbol{h}_i = f_i(\boldsymbol{h}_{i-1}) = \rho(W_i^T \boldsymbol{h}_{i-1} + \boldsymbol{b}_i),\ i \in \{1, \ldots, L-1\}, \tag{2.44}$$

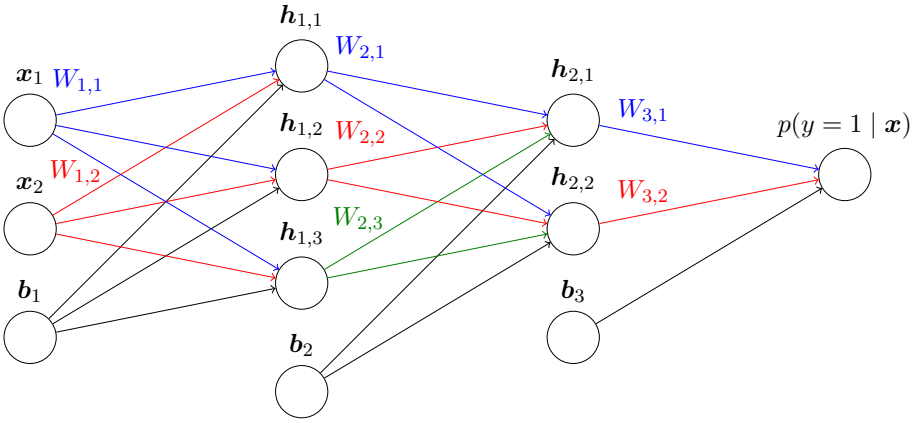

Figure 2.1. A simple deep neural network classifier for binary classification. The subscript notation $\boldsymbol{x}_i$ denotes the element i of the input vector, $\boldsymbol{h}_{i,j}$ denotes the i:th hidden layer neuron j, $W_{i,j}$ the i:th layer weight matrix row j, and $\boldsymbol{b}_i$ the bias vector of the layer i. The colored lines highlight the weights that are from the same row of the weight matrix, and the black lines indicate that the connection is simply a summation of the element of the bias vector corresponding to the neuron it connects to.

where $\boldsymbol{h_0} = \boldsymbol{x}$, the weight matrix W_i and the bias vector $\boldsymbol{b}_i$ are learned parameters associated with the layer i, and ρ is a nonlinear function, also called an *activation function*. In classification, the final layer f_L is often defined as a logistic regression classifier, as is done in Publication I and Publication II, such that for binary classification:

$$p(y = 1 \mid \boldsymbol{x}, \boldsymbol{\theta}) = \sigma(\boldsymbol{w}_L^T \boldsymbol{h}_{L-1} + b_L), \tag{2.45}$$

where σ denotes the logistic sigmoid activation and $\boldsymbol{\theta}$ all the parameters of the DNN. For multiclass classification with K classes, the softmax activation function is used:

$$p(y = c \mid \boldsymbol{x}, \boldsymbol{\theta}) = \frac{\exp(W_c^T \boldsymbol{h}_{L-1} + \boldsymbol{b}_c)}{\sum_{j=0}^{K-1} \exp(W_j^T \boldsymbol{h}_{L-1} + \boldsymbol{b}_j)}. \tag{2.46}$$

The functions $f_i,\ i = 1, \ldots, L-1$ of a DNN are called the *hidden layers* and the final layer f_L the *output layer* [17]. The elements of the vectors $\boldsymbol{h}_i$ are called neurons [17]. The name "fully-connected" refers to that each neuron of one layer is connected to every neuron of the next layer with the weight matrix. A visual illustration of a small fully-connected DNN is presented in Figure 2.1.

The flexibility of neural networks comes from the nonlinear activation functions that enable them to approximate nonlinear functions [17, 18]. Without the activations, a neural network would simply compute a linear transform, as is seen by removing the activation from Equation (2.44). The activation functions used in the DNNs of this thesis are the *rectified linear unit* (ReLU) and *Leaky ReLU*. The ReLU is defined as:

$$\rho_{ReLU}(x) = \max(0, x), \tag{2.47}$$

and the Leaky ReLU as:

$$\rho_{LeakyReLU}(x, \alpha) = \max(0, x) + \alpha \min(0, x), \tag{2.48}$$

where the α is usually a small value that is not learned [17]. The rectifier-based activations are popular because they preserve the derivatives on the positive domain of the input, and thus make gradient-based optimization easier [17]. Neural networks have been shown (given certain conditions) to be *universal approximators* [18] that is often used to justify that these networks can learn any function, including that which maps the inputs to the targets or to the true conditional distribution of them [17].

Convolutional Neural Networks

If there is prior knowledge that the target variable is invariant or equivariant to translation of the input, the *convolutional neural network* (CNN) architecture can be beneficial [17]. The CNN is a type of DNN that introduces *weight sharing* in a specific pattern, such that the weight matrix in Equation (2.44) is applied on small patches, i.e. spatial windows, of the input, instead of on the entire input like with the fully-connected DNNs [45]. The spatial shape of the window is called the "kernel size". This type of computation is efficiently implemented with the convolution operation, which gives the CNNs their name [17]. The i:th convolutional layer computes the following "2D convolution" for an image or multichannel 2D hidden layer input:

$$\boldsymbol{h}_{i,o,a,b} = \rho(\sum_{z} \sum_{k} \sum_{l} \boldsymbol{h}_{i-1,z,a+k,b+l} W_{i,o,z,k,l} + \boldsymbol{b}_{i,o}), \tag{2.49}$$

and the following "3D convolution" for a volume or multichannel 3D hidden layer input:

$$\boldsymbol{h}_{i,o,a,b,c} = \rho(\sum_{z} \sum_{k} \sum_{l} \sum_{m} \boldsymbol{h}_{i-1,z,a+k,b+l,c+m} W_{i,o,z,k,l,m} + \boldsymbol{b}_{i,o}), \tag{2.50}$$

where a, b, and c denote the output spatial height, width, and depth index, respectively, o denotes the output channel index, and z denotes the input channel index [17]. The outputs of the convolutional layers are called *feature maps* [17].

Remark 2.3. The summations in Equation (2.49) and Equation (2.50) actually compute "cross-correlation", however, in deep learning literature this computation is still called "convolution" [10, 17], and thus this thesis will also use the "convolution" naming convention. The discrete convolution would reverse the order of the elements of the weight that is not done in cross-correlation. As each element of the weights of a CNN are being learned, the same weights are learned regardless of the reversing operation [17].

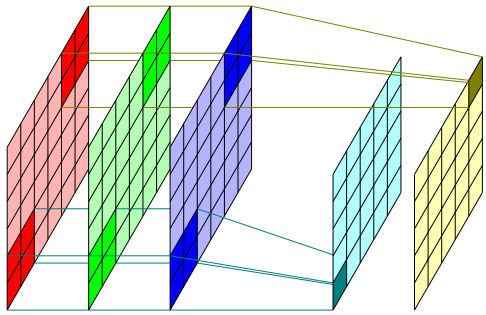

Figure 2.2. Illustration of a 2D convolutional layer with a kernel size 2×2. The red, green, and blue channels are the input channels, and the cyan and yellow channels are the output channels. The cyan lines show the connectivity pattern from the input channels to the cyan output channel bottom left corner. Additionally, the yellow lines show the connectivity pattern to the yellow channel top right corner. The line colors also represent that unique weights are used to compute the two output channels.

The weight W_i is a 4D array, also called a *tensor*, in Equation (2.49) and a 5D array in Equation (2.50). It connects each element of the input over the spatial window to a single element of a channel in the next hidden layer. The operations in Equation (2.49) and Equation (2.50) are repeated for all the spatial locations and output channels. The output $\boldsymbol{h}_i$ is then a multichannel image or volume, and the number of channels it has is determined by the first dimension of the weight tensor W_i. The bias $\boldsymbol{b}_i$ is a vector, such that a unique bias is applied to each output channel separately [10]. Visual illustration of a convolutional layer is presented in Figure 2.2.

The convolutional layers can be implemented with modifications to Equation (2.49) and Equation (2.50). One important variant of the convolution is the *strided convolution* that has a step-size larger than 1. For a stride s in both the height and width dimensions, the 2D convolution becomes:

$$\boldsymbol{h}_{i,o,a,b} = \rho(\sum_z \sum_k \sum_l \boldsymbol{h}_{i-1,z,sa+k,sb+l} W_{i,o,z,k,l} + \boldsymbol{b}_{i,o}), \tag{2.51}$$

that can be seen to skip some spatial locations, which means it downsamples the input [10]. It is simple to generalize for the 3D case, by considering the stride in also the third spatial dimension. There are also other variants, such as dilated convolution [10] and tiled convolution [17], however, these are out of scope of this thesis.

Another specification is called *padding*. It controls how the convolution is handled on the edges of the input. In some cases, the kernel does not tile the input completely, for example when the input spatial shape is smaller than the kernel size or when using a stride larger than one. There are two common principles of how this is handled. The first approach is to not compute the convolution on the border if the weight does not tile the input completely, which is called the *valid convolution* [20]. The

other approach is to use padding that concatenates some values on the borders to let the convolutional weight slide over the borders of the original input [17]. A special case of padding is the *same convolution* that pads the input such that the output has the same spatial shape as the input when stride $s = 1$ is used [10]. A popular type of padding is *zero padding* that simply concatenates zeros to the borders [17].

The convolutional layers presented so far are equivariant to translation (aside from "sub-stride" translations) meaning that for a CNN $f(\cdot)$ with purely convolutional layers and a translation operator T: $f(T\boldsymbol{x}) = Tf(\boldsymbol{x})$. For segmentation, this property is desired, as the target is expected to be spatially aligned with the input, and thus, the output of the model should translate with the input. However, for image classification, the output should usually be translation *invariant*, i.e. $f(T\boldsymbol{x}) = f(\boldsymbol{x})$. Translation invariance can be introduced by so-called *pooling* operations, the *max pooling* and the *average pooling* being two popular ones [17]. These operations are applied on small spatial windows, similar to the convolution operation, however, they are typically applied on each feature channel independently and do not include learned components. The max pooling computes the maximum and the average pooling the mean of the values inside a spatial window, which are invariant to small spatial translations [17]. It is also common to use so-called *global pooling*, for example the global average pooling that computes the average of each feature channel, which can be used to produce a vector from the feature maps [10]. This vector can then be used as an input to a fully-connected network or logistic regression to perform classification.

The convolution with stride $s = 1$ preserves the spatial shape of an input feature map, given a suitable padding, and a stride $s \geq 2$ will downsample it. There is also *fractionally strided convolution* that is more commonly known as the *transposed convolution* [46]. The transposed convolution can increase the spatial shape of the input [10], which is especially useful for autoencoder-like CNN architectures. The transposed convolution can be thought of as constructing a Toeplitz matrix that is equivalent with a certain convolution operation, and then multiplying the input with the transpose of the matrix [10].

Training Deep Neural Networks

The parameters of DNNs, both the fully-connected and the convolutional neural networks, can be estimated by the ML, MAP, or ERM approaches, introduced in Section 2.3. As DNNs are nonlinear, the optimization problems have no analytical solutions, and thus, iterative gradient-based optimization methods need to be used to solve them [18]. The optimization of the parameters of a DNN starts with an initial estimate of them $\boldsymbol{\theta}_0$, which is usually drawn randomly from some heuristically selected distribution [17].

Then, the parameters are iteratively updated to obtain $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, $\boldsymbol{\theta}_3$, and so on, to reach estimates of the parameters that result in a lower loss. The gradient descent methods choose the update based on the gradient [10]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t), \tag{2.52}$$

where the DNNs parameters at current iteration t are $\boldsymbol{\theta}_t$, ∇ is the gradient operator, $\mathcal{L}(\boldsymbol{\theta}_t)$ is shorthand for $\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}_t, l)$, and α is the step-size, also called the *learning rate*. It can be shown that the update direction based on the negative gradient never increases the first order Taylor series approximation of the loss:

$$\mathcal{L}(\boldsymbol{\theta}^* + \boldsymbol{\Delta}) \approx \mathcal{L}(\boldsymbol{\theta}^*) + \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^*)^T \boldsymbol{\Delta}, \tag{2.53}$$

where Δ is the update direction. Indeed, for a sufficiently small learning rate the loss always decreases or stays the same if a local or global minimum is found [17].

To compute the gradients of the loss with respect to the parameters of a DNN $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$, the chain rule of calculus can be utilized [17]. Let $\boldsymbol{h}_l^{(i)}$ be the l:th hidden layer vector computed for the i:th training example. Then the gradient of the batch loss with respect to the parameters of the j:th layer $\boldsymbol{\phi}$ is:

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}_t) &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\phi}} l(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_i), \boldsymbol{y}_i) \\
&= \sum_{i=1}^{N} (\nabla_{\boldsymbol{\phi}} \boldsymbol{h}_{j+1}^{(i)}) (\nabla_{\boldsymbol{h}_{j+1}} \boldsymbol{h}_{j+2}^{(i)}) \\
&\quad \cdots \\
&\quad (\nabla_{\boldsymbol{h}_{L-1}} \boldsymbol{h}_L^{(i)}) (\nabla_{\boldsymbol{h}_L} l(\boldsymbol{h}_L^{(i)}, \boldsymbol{y}_i)).
\end{aligned} \tag{2.54}$$

For neural networks, efficient computation of the gradient is achieved with the *reverse mode differentiation* that computes the gradient recursively starting the multiplications from the end of the right hand side term in Equation (2.54), which effectively propagates the error signal backwards through the network [10, 18]. This algorithm is known as the *back-propagation* algorithm [17, 18, 21].

The batch loss is computed on all of the N training data examples. Thus, for the computation of the gradients, there needs to be N passes through a DNN to compute the loss values, and N back-propagation passes to compute the gradients. In addition, all the intermediate values, i.e. the outputs of all linear and nonlinear operations, need to be stored in memory to compute the gradients. Thus, for large datasets, the number of computations and required memory can become prohibitive. A common approach is to utilize a stochastic estimate of the gradient of the batch loss,

by sampling a *mini-batch*, i.e. a small subset of M examples, and updating the parameters based on the gradient computed on the mini-batch. This is called *stochastic gradient descent* (SGD) [17]. The stochastic estimate of the gradient is given by [10]:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t) \approx g_{\boldsymbol{\theta}_t} = \frac{N}{M} \sum_{i=1}^{M} \nabla_{\boldsymbol{\theta}} l(f_{\boldsymbol{\theta}_t}(\hat{\boldsymbol{x}}_i), \hat{\boldsymbol{y}}_i), \tag{2.55}$$

where $(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i) \sim \mathcal{D}$, and it is used to update the parameters like in Equation (2.52):

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha g_{\boldsymbol{\theta}_t}. \tag{2.56}$$

A typical strategy is to sample the examples $\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i$ without replacement [20], and once all the data in the training set has been sampled once, it is said that the model has been trained for an *epoch* [17]. Multiple epochs are typically required, but this depends on the size of the training set [17].

The SGD algorithm can prove to be slow to converge, and thus, modifications have been proposed to enhance the convergence. In the Publications I–IV, the SGD with momentum or the Adam algorithm [47] were utilized. The SGD with momentum accumulates updates with an exponential decay term $\beta \in (0, 1)$ and computes updates as follows [17]:

$$\begin{aligned} \boldsymbol{m}_t &= \beta \boldsymbol{m}_{t-1} - \alpha g_{\boldsymbol{\theta}_t}, \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \boldsymbol{m}_t. \end{aligned} \tag{2.57}$$

The momentum can decrease the stochasticity of the updates with the moving average [17].

The Adam algorithm [47] is a type of diagonal preconditioned SGD variant [10]. It uses estimates of the first and second order moments of the gradient for this preconditioner [47]. The algorithm is given by, using the efficient and bias corrected version presented in Kingma and Ba [47]:

$$\begin{aligned} \boldsymbol{m}_t &= \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1) g_{\boldsymbol{\theta}_t}, \\ \boldsymbol{v}_t &= \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2) g_{\boldsymbol{\theta}_t}^2, \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha \frac{\sqrt{1 - \beta_2^t}\, \boldsymbol{m}_t}{(1 - \beta_1^t)(\sqrt{\boldsymbol{v}_t} + \epsilon_t)}, \end{aligned} \tag{2.58}$$

where the power of two in the second line and the division in the last line are element-wise. β_1 and β_2 are the momentum terms for the first and second noncentral moments, and $\epsilon_t = \sqrt{1 - \beta_2^t}\epsilon$, where ϵ is a small stabilization factor, such as 10^{-8} [47].

The iterative methods update the parameters based on the gradients computed on the training set, and usually the parameters of the DNN that resulted in the best validation performance, according to the loss, AUROC,

or some other measure, are selected as the parameters of the trained model. As the validation set performance is an estimate of the true generalization performance, this procedure should yield the parameter values that result in the DNN with the best generalization. Furthermore, DNNs can start to overfit while training, which means that the validation performance of the latest iteration might be lower than that of an earlier iteration in the training. A practical method is the so-called *early stopping* algorithm that continuously monitors the validation performance and stops the training if the validation performance has not improved during a certain amount of iterations [17]. The number of iterations of non-increasing performance is often called the *patience*.

The speed of convergence of the training procedure can be sometimes improved with *transfer learning*. Instead of initializing the weights using a random draw from some distribution, in transfer learning they are initialized by pretraining a DNN with another dataset. It has been observed that CNNs learn similar convolutional weights when trained on different datasets and tasks [48], and that the features which a CNN learns on large natural image databases are useful for other prediction tasks as well [49]. Transfer learning has also been observed to improve the performance on the primary task [48], however, Raghu et al. [50] observed that for medical images, such as retinal images, the main benefit is the speed of convergence and they did not observe better performance in comparison to randomly initialized parameters.

To mitigate overfitting, there also exist other regularization strategies than parameter regularization. *Data augmentation* is one such strategy [17]. It creates synthetic examples from the training data that increase the effective number of training examples. It is often used for CNNs, as images can be easily augmented to create realistic synthetic examples. For example, flipping an image along the vertical axis creates a mirrored version that together with the original data already doubles the number of the training examples. Modern deep learning libraries enable efficient augmentation while training, such that different augmentations, for example flipping, rotations, and color-space pertubations, can be randomly applied to the training examples [51]. On the other hand, augmentations need to be carefully selected in order to not break the input-target relationship [17]. For example, augmenting an image with random blurring removes small details, such as very small bleeds in retinal images, and thus it can remove the signs of a disease. For segmentation, the augmentations that alter the spatial information, such as flipping and rotation, should be applied to both the inputs and targets, with the same configuration, to preserve the spatial correspondence.

2.7 Bayesian Deep Learning

The ML, MAP, and ERM approaches of training DNNs yield a point estimate $\hat{\boldsymbol{\theta}}$ of the parameters. The prediction for unseen inputs $\boldsymbol{x}_*$ is then performed as $p(\boldsymbol{y}_* \mid \boldsymbol{x}_*, \hat{\boldsymbol{\theta}}) = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_*)$. However, this approach does not quantify any uncertainty in the parameter configuration $\hat{\boldsymbol{\theta}}$ [52].

Bayesian neural networks (BNNs) [52, 53] consider the Bayesian approach to infer the posterior density of DNNs parameters, given in Theorem 2.1. The posterior density can then be used to obtain the posterior predictive distribution $p(\boldsymbol{y}_* \mid \boldsymbol{x}_*)$ for an unseen input, as presented in Equation (2.7), that marginalizes the parameters out. The Bayesian posterior is proportional to the product of the data likelihood and the prior, and the prior should reflect some domain knowledge about the parameters [54]. The problem for DNNs is that the parameters are typically not interpretable [52], and thus the priors are often selected based on computational convenience only, and thus, are not well specified [54]. However, simple Gaussian priors have been observed to lead to competitive posteriors [55], and combined with the inductive biases, such as the convolutional architecture, even networks that are sampled from a Gaussian prior have surprising discriminative performance [53].

Unfortunately, there typically exists no closed-form expression for the BNN posterior distribution [55]. Furthermore, the golden standard approach of directly sampling from the posterior with Hamiltonian Monte Carlo (HMC) [56] is very expensive, both computationally and memory-wise, for modern DNNs. Indeed, in Izmailov et al. [55], HMC was used for sampling from the posterior of a BNN with a modern DNN architecture, however, 512 TPUv3 compute devices [57] were required even for the small datasets used in the study. The high computational demand of HMC has inspired research to approximations of the BNN posteriors.

This thesis considers the BNN approximation with *deep ensembles*, *Monte Carlo dropout*, *mean field variational inference*, and *generalized variational inference*. There are many other approximations, such as stochastic gradient HMC [58] and Stochastic Weight Averaging-Gaussian [59], however, the methods presented in this section are selected based on those used in Publication II. For the remainder of this section, let the true posterior distribution of the neural network, a fully-connected or CNN type, be $p(\boldsymbol{\theta} \mid \mathcal{D})$ and let the approximate posterior distribution be denoted as $q(\boldsymbol{\theta})$. The approximate posterior predictive distribution substitutes the $p(\boldsymbol{\theta} \mid \mathcal{D})$ with $q(\boldsymbol{\theta})$ in Equation (2.7), and the integral can be numerically computed with Monte Carlo integration:

$$p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \mathcal{D}) \approx \frac{1}{M} \sum_{i=1}^{M} p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \boldsymbol{\theta}^{(i)}), \tag{2.59}$$

where $\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta})$ and M is the number of Monte Carlo samples.

Deep ensembles [60] consist of a set of DNNs. The individual networks are trained with different settings, for example using a different random seed while training each network, with the aim that the resulting networks can have non-identical parameters. The ensemble prediction is computed as the average of the predictions of the individual ensemble members. The deep ensemble approximate posterior distribution can then be thought to be a discrete uniform distribution over the set of parameters of the ensemble members. Even though the approach is conceptually simple, it has been observed that it approximates the HMC solution with higher fidelity than some other approximations [55].

Monte Carlo dropout (MC dropout) [61] applies the dropout [62] regularization method during test-time to sample a set of neural networks that have some parameters masked. The standard dropout method randomly masks some neurons of a neural network to zero while training, such that the connectivity pattern between consecutive layers is randomly altered. This has been observed to have a regularizing effect, however, in Gal and Ghahramani [61], it was also discovered to have a connection to variational inference considering a certain Gaussian process prior. The random masking of neurons can be seen as masking the rows (or the columns) of the weights and elements of the biases that gives arise to the dropout variational distribution of the parameters. The weights W_l and biases $\boldsymbol{b}_l$ of a layer can be sampled from the dropout variational distribution $q(\boldsymbol{\theta})^{dropout}$ by [61]:

$$\begin{aligned} W_l &= \hat{W}_l \ \mathrm{diag}(\boldsymbol{z}), \\ \boldsymbol{b}_l &= \mathrm{diag}(\boldsymbol{z}) \ \hat{\boldsymbol{b}}_l, \\ \boldsymbol{z} &\sim \mathrm{Bern}(\boldsymbol{p}_{drop}), \end{aligned} \tag{2.60}$$

where $\hat{W}_l$ and $\hat{\boldsymbol{b}}_l$ are the variational parameters, $\mathrm{Bern}(\boldsymbol{p}_{drop})$ denotes a Bernoulli distribution over binary vectors with the probability of each element being one given by the corresponding element of the dropout probability vector $\boldsymbol{p}_{drop}$. However, the approach in Equation (2.60) is rarely used, as the random masking of neurons instead of the parameters is more efficient and available in modern deep learning libraries, such as Pytorch [51]. MC dropout can be also extended to the convolutional layers by dropping channels of the feature maps.

Mean field variational inference (MFVI) [63] assumes that the posterior distribution of the parameters factorizes, typically in the extreme such that all the parameters are independent [64, 65], and it seeks a variational approximate distribution that minimizes the following Kullback-Leibler (KL) divergence [66]:

$$D_{KL}[q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathcal{D})] = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathcal{D})} d\boldsymbol{\theta}. \tag{2.61}$$

Equation (2.61) can be shown to result in the negative evidence lower bound by:

$$\begin{aligned} D_{KL}[q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta} \mid \mathcal{D})] &= \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})p(\mathcal{D})}{p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})} d\boldsymbol{\theta} \\ &= \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \log p(\mathcal{D} \mid \boldsymbol{\theta}) d\boldsymbol{\theta} \\ &+ \log p(\mathcal{D}) \\ &\leq D_{KL}[q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta})] - \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathcal{D} \mid \boldsymbol{\theta})]. \end{aligned} \tag{2.62}$$

The optimal MFVI approximate posterior for a classification task is then [66]:

$$q(\boldsymbol{\theta})^{MFVI} = \arg\min_{q \in Q} D_{KL}[q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta})] - \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(y \mid \boldsymbol{x}, \boldsymbol{\theta})]. \tag{2.63}$$

For a parametric approximate posterior $q_{\boldsymbol{\gamma}}(\boldsymbol{\theta})$, this amounts to finding the parameters $\boldsymbol{\gamma}$ that minimize the upper bound in Equation (2.62) [66]. It is common to define the approximate posterior $q_{\boldsymbol{\gamma}}(\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$ as diagonal multivariate Gaussian distributions [64, 65], and then the parameters $\boldsymbol{\gamma}$ of the approximate posterior are the mean and variance of the Gaussian for all the parameters $\boldsymbol{\theta}$. To allow for gradient-based learning of the approximate distribution, the so-called *reparametrization trick* [31] can be used for the Gaussians. It decomposes the samples of $\boldsymbol{\theta}$ from a diagonal multivariate Gaussian distibution $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$ as:

$$\begin{aligned} \boldsymbol{\theta} &= \boldsymbol{\mu} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}, \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, I), \end{aligned} \tag{2.64}$$

where $\odot$ is the Hadamard product.

In Farquhar et al. [65], it was observed that the MFVI approach with multivariate Gaussians has numerical issues related to the high norm of the samples. As the multivariate Gaussian distribution has the so-called "soap-bubble" phenomenon in high dimensions, which means that the probability mass is concentrated on a thin sphere and the samples have a high norm on expectation, the high norm of the samples can produce numerical issues while training. In Farquhar et al. [65], the *Radial MFVI* was proposed that considers a special type of approximate posterior distribution, the samples of which have the same expected norm regardless of the dimension. It can thus avert the issues related to the norm of the samples. The Radial posterior distribution does not have a closed form

expression, but it can be efficiently sampled from by:

$$\begin{aligned} \boldsymbol{\theta} &= \boldsymbol{\mu} + \hat{\boldsymbol{\epsilon}} \odot \boldsymbol{\sigma}, \\ \hat{\boldsymbol{\epsilon}} &= r\frac{\boldsymbol{\epsilon}}{||\boldsymbol{\epsilon}||_2}, \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, I), \\ r &\sim \mathcal{N}(0, 1). \end{aligned} \tag{2.65}$$

The authors also presented how the KL divergence can be evaluated up to a constant, and thus, the Radial posterior is simple to use in place of the multivariate Gaussian in Equation (2.62) with Monte Carlo integration.

Generalized variational inference (GVI) [54] is a novel approach for robust variational inference. The GVI arises from the so-called *Rule of Three* that views statistical inference as an optimization problem of a loss function and a divergence measure given a space of feasible solutions. The divergence can be selected to introduce robustness to prior misspecification, and thus, can mitigate issues related to priors selected solely on the computational aspects. The GVI approach arises when the space of feasible solutions is chosen as some subset of the space of all probability measures, and the loss function and divergence measure are chosen freely. The GVI posterior is the solution to:

$$q^{GVI}(\boldsymbol{\theta}) = \underset{q \in Q}{\arg\min}\, \mathbb{E}_{q(\boldsymbol{\theta})}[l(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))] + D[q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta})], \tag{2.66}$$

where l is a loss function, f is a DNN, D is any divergence, and $p \in Q \subset \mathcal{P}(\boldsymbol{\theta})$. It can be seen that it recovers the MFVI approach when the loss function is the negative log-likelihood and the divergence is the KL divergence. In Publication II, the Rényi's α-Divergence was used as a robust divergence. The "rescaled" variant of it is calculated as [54]:

$$D^{\alpha}_{AR}[q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta})] = \frac{1}{\alpha(1-\alpha)} \log \int q(\boldsymbol{\theta})^{\alpha} p(\boldsymbol{\theta})^{1-\alpha} d\boldsymbol{\theta}. \tag{2.67}$$

2.8 Machine Learning for Medical Data

Medical data poses many practical challenges to machine learning research. It is often so that the medical data has high *class imbalance*, which means that there are significantly fewer examples of some classes than of the others [67]. Indeed, for example in Publication V, only 3.2% of the patients were diagnosed with necrotizing enterocolitis, and in Publication II, the clinical hospital dataset had less than 1% cases with proliferative diabetic retinopathy. The collection of more data to accommodate the imbalance is difficult in the medical domain, as a disease or other condition of interest

might be inherently rare in the population and the data can only be collected when a patient visits a healthcare provider.

The availability and accessibility of medical data also differs from other types of data. Publicly available datasets exists in the internet, for example in the form of data for competitions hosted by Kaggle, MICCAI, and PhysioNet, however, these datasets have often restrictions to their use, for example permission for research purposes only. Furthermore, there might exist no publicly available data for certain problems, which means that research is impossible without access to private data. Private data, such as hospital data records, are also subject to regional and local regulations that can demand that the data is collected, stored, and accessed in a certain manner. Thus, if an entity, such as a healthcare provider or device manufacturer, wants to develop a machine learning algorithm for some prediction task, the data needs to already exist or it needs to be collected, there must be permissions for the access and use of the data for the purposes of the research or development, and all the regional regulations need to be accounted for.

The notable regulation for medical data analysis in the European Union region is the *EU General Data Protection Regulation* (GDPR) [68]. Many medical classification tasks require data that is so-called *personal data* under the regulation. This is because data that is related to the physical or mental health status or medical history of a patient is considered to be personal data. However, classifying a physiological or mental medical condition requires this type of data for training. Data collected by healthcare providers is also not exempt of the personal data status. The GDPR sets guidelines when personal data is permitted to be processed. Important notion is the *controller* that is an entity responsible for the data and for compliance of all actions with the regulation. For machine learning research, the relevant permission for data processing is the "data controller's legitimate interests", which can consist of scientific research or product development if they do not infringe the patients rights and do not compromise privacy. Indeed, GDPR promotes pseudonymization of personal data, which means that the data cannot be identified to a specific patient in a straightforward manner. Furthermore, if the data is completely anonymized, i.e. it cannot be identified to a specific patient, it is no longer considered to be personal data by GDPR.

Other regulations and laws need to be taken account for as well. Relevant for the Publications I–V are the Finnish *Medical Research Act* [69] and the *Secondary Use of Health and Social Data Act* [70]. If the data collection requires intervention in the integrity of a person, for example taking a blood sample, and the modeling task is related to health, then the research is considered to be *medical research* by the Medical Research Act. Medical research requires informed consent of all the patients and it requires an ethical committee approval. Retrospective *register data* that is collected for

other purposes can be used for research without considering the research to be medical research, as the Secondary Use of Health and Social Data Act considers the research to be *secondary use of data*. The Secondary Use of Health and Social Data Act also requires the physical computational machine and the software environment to be secure, such that the private information cannot be compromised. Thus, it is often so that the machine learning model has to be selected with the computational limitations of the secure machine in mind. Further institutional regulations need to be also adhered to and these can depend on the healthcare district.

3. Deep Learning for Diabetic Retinopathy and Macular Edema Classification (Publications I & II)

In 2019, it was estimated that there were 463 million diabetics globally, and that the number of diabetics grows to 578 million by 2030 [71]. High blood glucose concentration, associated with both the type 1 and type 2 diabetes, can cause complications and damage to the small blood vessels of the human body, including those in the human eye [72]. *Diabetic retinopathy* is the term used for the different complications of the retina that are caused by the high glucose concentration [73]. Diabetic retinopathy affects a third of the diabetic population and is the leading cause of blindness in the working-aged population [74]. It can deteriorate sight and lead to blindness if it is untreated [73]. Thus, it is important to detect diabetic retinopathy early, such that the condition can be monitored and treated if necessary. Diabetic retinopathy can be detected by ophthalmological means by eye examination and retinal fundus imaging, and national screening programmes exist in some countries, however, the screening is currently done manually by medical experts [73]. To reduce the global burden on manual screening programmes, automatic detection and classification of diabetic retinopathy is of paramount interest.

There are many classification systems for the severity of diabetic retinopathy. A comprehensive severity scale was introduced in the Early Treatment Diabetic Retinopathy Study (ETDRS) [75]. However, this scale is too fine-grained for clinical use, for which the 5-class proposed international clinical diabetic retinopathy scale (PIRC) was introduced in Wilkinson et al. [76]. The PIRC scale is used for example in Finland [77], and some benchmark diabetic retinopathy datasets have also been annoted using it [78, 79]. The five severity classes of PIRC are the *no diabetic retinopathy* (class 0), *mild diabetic retinopathy* (class 1), *moderate diabetic retinopathy* (class 2), *severe diabetic retinopathy* (class 3), and *proliferative diabetic retinopathy* (class 4). The classes are progressively more severe with the class 0 having no signs of diabetic retinopathy, classes 1–3 having increasing number and severity of abnormalities, and finally the class 4 consisting of proliferative diabetic retinopathy. In the proliferative diabetic retinopathy, the retina is suffering from a low blood supply, which causes new small blood vessels to

proliferate on the retina [73]. In the center of the retina resides the macula of the eye that provides the most accurate vision. Diabetic retinopathy that manifests on this area is especially dangerous for vision, and it is often classified separately as *diabetic macular edema* [73, 77]. One severity scale for it is the proposed international clinical diabetic macular edema scale (PIMEC), also introduced in Wilkinson et al. [76]. The PIMEC has 4 classes: *no diabetic macular edema* (class 0), *mild diabetic macular edema* (class 1), *moderate diabetic macular edema* (class 2), and *severe diabetic macular edema* (class 3).

In early studies of machine learning for diabetic retinopathy classification, a simplified binary *referable*/*non-referable diabetic retinopathy* (RDR) classification system was used [11, 80]. This scale is derived from the PIRC system by defining the referable diabetic retinopathy as moderate or worse (class ≥ 2) and non-referable diabetic retinopathy as no or mild diabetic retinopathy (class ≤ 1). This system was also examined in Publication I and Publication II.

It has been observed that modern DNN classifiers suffer from *poor calibration* [16]. It means that the DNN always places a high probability for one of the classes, even when the class is wrong, and thus the output "probability" values that the network infers cannot be used to quantify uncertainty. In diabetic retinopathy, it is critical to correctly quantify the uncertainty in the predictions, such that if the model is uncertain for some cases, they can be referred to a medical expert. One promising avenue for "uncertainty-aware" deep learning is the approximate Bayesian deep learning that is studied for diabetic retinopathy classification tasks on clinical data in Publication II.

3.1 Prior Work

Early studies for automatic detection of diabetic retinopathy from retinal images utilized very specialized algorithms [81, 82] or combination of hand-crafted features and machine learning algorithms [83, 84]. The datasets had also been of modest size, as one of the largest had 66 680 images from 16 670 patients [82]. Deep learning has been widely applied for diabetic retinopathy classification since 2016 [85], which has been facilitated by publicly available datasets [86]. This thesis considers the studies by Gulshan et al. [11], Ting et al. [80], and Krause et al. [87] due to their influentiality, originality in the sense of the used data, and as these works were considered to be the state-of-the-art in Publication I. Deep learning has also been applied to a wide range of other diabetic retinopathy related tasks, such as lesion segmentation [88, 89] and detection [90], and prediction of the progression of the disease [91], however, these tasks are out of the scope of this thesis.

In 2016, Gulshan et al. [11] utilized deep learning for the RDR task with high sensitivity, specificity, and AUROC. The deep learning approach was an ensemble of 10 ImageNet [1] pretrained Inception-v3 CNNs [92] that utilized retinal images of resolution 299×299. They used a very large dataset in training that originated from EyePACS in the United States and from three eye hospitals in India. The training set consisted of 128 175 images from 69 573 patients. For testing the model, they used a dataset that originated from the EyePACS, which had 9963 images from 8906 patients, and also the publicly available Messidor-2 dataset that had 1748 images [93, 94].

In 2017, Ting et al. [80] used deep learning for RDR, vision-threatening diabetic retinopathy, possible glaucoma, and age-related macular degeneration classification. Vision-threating diabetic retinopathy was defined as severe or worse (PIRC class ≥ 3) diabetic retinopathy. They used ensembles of two VGG-like [95] CNNs and retinal images of resolution 512×512. Their training set and one of the test sets originated from Singapore. The training set had 76 370 images from 13 099 patients and the Singapore originating test set had 71 896 images from 14 880 patients. They also utilized an out-of-distribution heterogeneous test set that had images from Guangdong, Singapore, and Beijing, as well as from the African American Eye Disease Study and clinics from Ireland, Mexico, and Hong Kong. However, for possible glaucoma and age-related macular degeneration, they used slightly different sets, as the out-of-distribution Singapore originating data was included in the training set and the evaluation was only performed on the within-distribution test set. They observed similar results as in Gulshan et al. [11], i.e. high sensitivity, specificity, and AUROC.

In 2018, Krause et al. [87] presented a continuation study of Gulshan et al. [11]. They studied deep learning classification of RDR, referable diabetic macular edema (RDME), and also the clinical 5-class PIRC grade. The deep learning model was a more recent Inception-type CNN called Inception-v4 [96] and 10 ensemble members were used. They also increased the retinal image resolution to 799×799. The training set was extremely large, as it consisted of 1 665 151 images from 238 610 patients. It originated from the EyePACS and it included the training and validation set of Gulshan et al. [11]. In addition, they examined the evaluation of the algorithm when the grades were produced under an "adjudication process", where three retinal specialists graded the images independently, and for cases with conflicting grades they decided the grade together. On the test set of 1958 images, they observed that the model had similar performance in the PIRC task as individual opthalmologists and retinal specialists, and also showed that their modifications outperformed the models in Gulshan et al. [11].

Approximate Bayesian deep learning has been used for diabetic retinopathy classification in literature, however, only for binary classification tasks.

In 2017, Leibig et al. [97] first proposed these methods for diabetic retinopathy classification. In the work, MC dropout method was used for RDR and any diabetic retinopathy detection. It was compared to a standard neural network and a Gaussian process classifier that was fit to the last layer of a standard neural network. All the networks had VGG-like [95] CNN architectures. The models were trained on an EyePACS originating Kaggle contest dataset (EyePACS-K) [98] and the out-of-distribution performance was evaluated on the Messidor set [93]. The EyePACS-K training set consisted of 35 126 images and the test set of 53 576 images. They observed that the MC dropout CNN could estimate the uncertainty better than the standard CNN and the Gaussian process method for both the classification tasks and datasets.

In Filos et al. [64], more BNN approximations were studied for the RDR classification system. The selected methods were MC dropout, MFVI, deep ensembles, and ensemble MC dropout. The work also used a VGG-like [95] CNN architecture. The EyePACS-K set was used as the training set and the APTOS [79] set was used as an out-of-distribution test set. All the approximate Bayesian methods outperformed the standard neural network and it was observed that the ensemble MC dropout performed the best out of them in all of the tasks.

In Farquhar et al. [65], the Radial MFVI posterior was proposed to remedy the issues with the multivariate Gaussian posterior. One of the experiments in the study was RDR classification on the EyePACS-K set. The experiment used a VGG-16 [95] type CNN architecture with global average and global max pooling after the convolutional layers. It was found that the Radial MFVI approach outperformed the MC dropout and MFVI with the Gaussian posterior, and an ensemble of Radial MFVI networks outperformed the MC dropout ensemble in the quality of the uncertainty estimates. They also studied the calibration of MFVI, Radial MFVI, MC dropout, and deep ensembles, and found that the Radial MFVI had the best calibration.

Finally, a Bayesian diabetic retinopathy classification benchmark was introduced in Band et al. [99]. The study describes RDR classification in two settings: when the test data suffers from diabetic retinopathy severity distribution shift, and when there is distribution shift by country. The former was investigated using the EyePACS-K set and the latter using the APTOS dataset as a country shifted test set. They conducted experiments on these tasks with a standard neural network with l_2 regularization (i.e. a MAP optimization task), MFVI, Radial MFVI, function space variational inference, MC dropout, and Rank-1 parameterized MFVI. They also presented experiments with ensembling for each of the methods. In contrast to previous studies, they used the ResNet-50 [2] CNN architecture. For the retinopathy severity shift task, they observed that the approximate BNNs outperformed the MAP neural network. In the case of distribution shift

by country, they observed that no method consistently outperformed the others. For the within-distribution tests, the results suggest that even the MAP network can provide useful uncertainty estimates and the ensemble MC dropout performed the best.

There have also been studies that try to estimate the uncertainty in diabetic retinopathy classification using other methods than approximate BNNs. In Ayhan and Berens [100] and Ayhan et al. [101], "test-time augmentation" was used to estimate the predictive distribution. It works by augmenting the test images with random transforms, such as random rotation or color-space transforms, and then averaging the DNN outputs of multiple augmented versions of an image. In Ayhan and Berens [100], this is said to model the heteroscedastic aleatoric uncertainty, i.e. inherent data noise that differs for each example. Ayhan et al. [101] also studied softmax temperature scaling that aims to calibrate a trained network by scaling the pre-softmax output of the network. In Araújo et al. [102], a standard neural network was trained in an ordinal regression setting where the output of the network was one dimensional and considered to be distributed as a Gaussian conditional to the PIRC label. The means of the Gaussians were set as the numeric PIRC label and the standard deviation was inferred by the network for each image independently. The probabilities of the PIRC labels were then computed similarly as with the LDA classifier, shown in Equation (2.34), and the inferred standard deviation was used to measure the uncertainty. As Publication II considers approximate Bayesian deep learning for uncertainty-aware diabetic retinopathy classification, the test-time augmentation, softmax temperature scaling, and the ordinal regression approaches are out of the scope of this thesis.

3.2 Deep Learning Diabetic Retinopathy and Macular Edema Grading on a Finnish Dataset (Publication I)

In Publication I, deep learning was studied for the RDR and PIRC diabetic retinopathy classification systems, and for the RDME and PIMEC diabetic macular edema classification systems. The referable diabetic macular edema was defined as any class of PIMEC other than no macular edema (PIMEC > 0) and non-referable diabetic macular edema as no macular edema (PIMEC $= 0$) for the RDME system. The clinical PIMEC scale had not been studied before using deep learning to predict the label from retinal fundus images. An additional diabetic retinopathy classification system was created, which consisted of three mutually exclusive labels: non-referable diabetic retinopathy, referable diabetic retinopathy, and ungradable image quality (QRDR). In contrast to previous studies that had used only one image resolution or only considered one classification task while varying the image resolution, Publication I considered image

sizes of 256×256, 299×299, 512×512, 1024×1024, and 2095×2095 for all the tasks. Similar to Gulshan et al. [11], the model was an Inception-v3 CNN with ImageNet pretrained parameters, however, due to the large image sizes, ensembling was used only for the 512×512 resolution that was selected as it was used in Ting et al. [80].

The Finnish dataset was provided by Digifundus Ltd and it consisted of 41 122 retinal images from 14 624 patients. The retinal images were selected for the study with a preference for the severe cases to alleviate the class imbalance problem present in the population distribution. These images were captured with a Canon CR2 retinal camera while the patients' pupils were dilated with tropicamide eyedrops. The dataset was randomly divided into train, validation, and test sets, separately for each task, with 70%, 10%, and 20% proportions, respectively. The division was performed for all the tasks independently to achieve near identical grade distributions among the training, validation, and test sets. The random dataset division algorithm was implemented such that the patients did not overlap between the sets. The number of images and class distributions of all sets are presented in Table 3.1. The Messidor-1 dataset [93] was used as an out-of-distribution test set for the RDR and RDME tasks and it consisted of 1200 images.

Results

For the binary RDR and RDME tasks, higher image resolutions provided better AUROC, however, the increase in performance started to saturate on resolution 512×512. For the RDME task, the ensemble model with 512×512 could be utilized to outperform even the highest resolution single model performance. For the PIRC and QRDR tasks, the highest QWK was achieved at 1024×1024 and for the PIMEC task using the highest resolution. Similar to the RDME task, PIMEC also improved beyond any single model performance when the ensembling approach was used.

Comparisons of the best results obtained in Publication I to those of the previous publications are presented in Tables 3.2 and 3.3. Similar to Gulshan et al. [11], the operating point was selected for sensitivity and specificity analysis such that the models had 90% sensitivity on the validation set. It can be seen that the models presented in Publication I have comparable or better performance to those presented in other studies. Indeed, the RDR model outperforms Ting et al. [80] and Krause et al. [87], and has only 0.04 lower AUROC in comparison to Gulshan et al. [11]. For PIRC, the best model of Publication I outperforms that of Krause et al. [87] in the QWK measure. Additionally, the Messidor AUROC values are fairly high, however, it can be observed that for RDME the operating point does not generalize well to the set, as the 90.0% sensitivity point corresponds to 57.5% sensitivity point in Messidor-1. The lower sensitivity

Table 3.1. Statistics of the Finnish diabetic retinopathy dataset used in Publication I. Modified from Publication I.

Task		Training Set	Validation Set	Test Set
RDR	Images Total	24 806 (100%)	3706 (100%)	7118 (100%)
	Images Class 0	13 895 (56.0%)	2079 (56.1%)	4031 (56.6%)
	Images Class 1	10 911 (44.0%)	1627 (43.9%)	3087 (43.4%)
PIRC	Images Total	24 941 (100%)	3560 (100%)	7129 (100%)
	Images Class 0	11 160 (44.7%)	1573 (44.2%)	3229 (45.3%)
	Images Class 1	2793 (11.2%)	408 (11.5%)	842 (11.8%)
	Images Class 2	9221 (37.0%)	1312 (36.9%)	2597 (36.4%)
	Images Class 3	1480 (5.9%)	225 (6.3%)	382 (5.4%)
	Images Class 4	287 (1.2%)	42 (1.2%)	79 (1.1%)
RDME	Images Total	24 651 (100%)	3675 (100%)	7304 (100%)
	Images Class 0	20 819 (84.5%)	3113 (84.7%)	6162 (84.4%)
	Images Class 1	3832 (15.5%)	562 (15.3%)	1142 (15.6%)
PIMEC	Images Total	24 791 (100%)	3535 (100%)	7304 (100%)
	Images Class 0	20 958 (84.5%)	2974 (84.1%)	6162 (84.4%)
	Images Class 1	1531 (6.2%)	237 (6.7%)	465 (6.4%)
	Images Class 2	1566 (6.3%)	222 (6.2%)	438 (6.0%)
	Images Class 3	736 (3.0%)	102 (2.9%)	239 (3.3%)
QRDR	Images Total	28 787 (100%)	4109 (100%)	8226 (100%)
	Images Class 0	3827 (13.3%)	533 (13.0%)	1132 (13.8%)
	Images Class 1	14 005 (48.7%)	1991 (48.5%)	4009 (48.7%)
	Images Class 2	10 955 (38.1%)	1585 (38.6%)	3085 (37.5%)

is likely due to the differing annotation used for the Messidor set, as the risk of macular edema label of Messidor is 1 for images with any exudates, whereas the PIMEC label of mild diabetic macular edema (class 1) requires the exudates to be within a certain distance from the center of the retina. Thus some Messidor images with label 1 would be considered as class 0 in the PIMEC scale, which would translate to an RDME class 0. This would explain why the sensitivity decreases, as there are false negative cases by the Messidor labels. However, there is also distribution shift caused by the different imaging device, study population, and some Messidor images were imaged without pupil dilation, which can all modify the appearance of the Messidor images in comparison to the Finnish dataset.

Table 3.2. Comparison of the RDR and RDME results to previous publications. Sensitivity and specificity computed at 90% sensitivity operating point. Publication I results are for the configuration with the best AUROC. "OOD" denotes out-of-distribution, "Sens." sensitivity, and "Spec." specificity.
* Operating point selected with a different criterion.

Task	Publication	Dataset	AUROC	Sens.	Spec.
RDR	Gulshan et al. [11]	Internal	**0.991**	0.903	0.981
		Messidor-2	0.990	0.870	0.985
	Ting et al. [80]	Internal	0.936	0.905	0.916
		OOD Best	0.983	0.989	0.922
		OOD Worst	0.899	0.971	0.820
	Krause et al. [87]	Internal	0.986	0.971*	0.923*
	Publication I	Internal	0.987	0.896	0.974
		Messidor	0.967	0.859	0.971
RDME	Gulshan et al. [11]	Internal	-	0.908	0.987
		Messidor-2	-	0.904	0.988
	Krause et al. [87]	Internal	-	0.949*	0.944*
	Publication I	Internal	**0.989**	0.947	0.954
		Messidor-1	0.953	0.575	0.995

Table 3.3. Comparison of the PIRC results and illustration of the novel results with PIMEC and QRDR systems.

Task	Publication	Dataset	QWK
PIRC	Krause et al. [87]	Internal	0.840
	Publication I	Internal	**0.915**
PIMEC	Publication I	Internal	0.871
QRDR	Publication I	Internal	0.938

3.3 Uncertainty-aware Deep Learning Methods for Diabetic Retinopathy Classification of Clinical Data (Publication II)

In Publication II, BNN approximations were utilized for diabetic retinopathy classification to leverage the uncertainty information they provide. In the literature, the uncertainty has been often quantifed with the entropy of the approximate posterior predictive distribution [64, 103], mutual information between the prediction and the approximate posterior [65, 103], or the standard deviation of the posterior predictive distribution [97]. The uncertainty value is then compared to a threshold that determines if the classification is accepted, or if the retinal image is referred to an expert. In practice, this type of "referral process" is simulated by computing the

uncertainty values for each example in the test set, ordering the predictions based on the uncertainty values, and then, for a given threshold of uncertainty, leaving the most uncertain predictions out while evaluating the performance on the remaining examples.

Publication II showed that the referral of uncertain examples is actually a type of *reject option classification* [18]. In reject option classification, the typical approach is to use a minimal risk function [104] to quantify the risk in a classification:

$$r(\boldsymbol{x}) = \min_{\hat{y}} \sum_{i=1}^{C} p(y = i \mid \boldsymbol{x}) l(\hat{y}, i), \tag{3.1}$$

where $l(\hat{y}, i)$ is interpreted as a loss function for a correct label i and a label prediction $\hat{y}$. The rejection then occurs if the risk exceeds some threshold τ, i.e. $r(\boldsymbol{x}) > \tau$. The classic minimum risk for the zero-one type loss is the risk $r_{0/1}(\boldsymbol{x}) = 1 - \max_{\hat{y}} p(y = \hat{y} \mid \boldsymbol{x}))$ [105], which is equivalent to examining if the largest conditional probability is lower than some threshold [18].

In Publication II, the *expected conditional risk* was used instead of the minimum risk. The expected risk of a classifier is given as the expected value of the loss over the input and target variables [106]:

$$I(f) = \sum_{i=1}^{C} \int p(y = i, \boldsymbol{x}) l(f(\boldsymbol{x}), i) d\boldsymbol{x}. \tag{3.2}$$

The point-wise risk in $\boldsymbol{x}$ can be derived by leaving the integration over the inputs out:

$$R(\boldsymbol{x}) = \sum_{i=1}^{C} p(y = i \mid \boldsymbol{x}) l(f(\boldsymbol{x}), i). \tag{3.3}$$

When the classifier is trained to model the conditional distribution $f(\boldsymbol{x}) = p(y \mid \boldsymbol{x})$, the risk becomes:

$$R(\boldsymbol{x}) = \sum_{i=1}^{C} p(y = i \mid \boldsymbol{x}) l(p(y \mid \boldsymbol{x}), i). \tag{3.4}$$

The risk $R(\boldsymbol{x})$ is called the expected (in targets) conditional (on an input) risk to not confuse it with the expected risk of a classifier. It is straightforward to show that the expected conditional risk corresponds to the entropy of the posterior predictive distribution if the loss is the negative log-likelihood of categorical labels:

$$\begin{aligned} R_{NLL}(\boldsymbol{x}) &= -\sum_{i=1}^{C} p(y = i \mid \boldsymbol{x}) \log p(y = i \mid \boldsymbol{x}) \\ &= \mathbb{H}[p(y = i \mid \boldsymbol{x})]. \end{aligned} \tag{3.5}$$

This correspondence reveals that the entropy-based referral of uncertain examples can be viewed as reject option classification when the expected conditional risk is used with the negative log-likelihood loss.

In Publication II, the expected conditional risk approach was used to develop a new uncertainty measure for PIRC classification based on the negative QWK as a loss. The negative QWK loss was defined as:

$$l_{QWK}(p(y \mid \boldsymbol{x}), i) = -\sum_{j=1}^{M} p(y = j \mid \boldsymbol{x}) \kappa_{QW}(C + S_{j,i}), \tag{3.6}$$

which is the negative expected QWK value for a predictive distribution $p(y = j \mid \boldsymbol{x})$. As the quadratic weighted Cohen's kappa will be zero for any single-entry matrix with element 1, it was evaluated using the confusion matrix of the validation set C as $k_{QW}(C + S_{j,i})$, where the $S_{j,i}$ is the single-entry matrix corresponding to a target i and prediction j. The expected conditional risk with the negative QWK loss was called the ***QWK-Risk***, computed by:

$$R_{QWK}(\boldsymbol{x}) = -\sum_{i=1}^{M} p(y = i \mid \boldsymbol{x}) \sum_{j=1}^{M} p(y = j \mid \boldsymbol{x}) \kappa_{QW}(C + S_{j,i}). \tag{3.7}$$

The QWK-Risk can be interpreted as the expected negative QWK value for an input example $\boldsymbol{x}$, similar to entropy being the expected negative log-likelihood.

Results

In Publication II, nine approximate BNNs were examined for RDR and PIRC classification using three benchmark datasets and an uncurated Finnish hospital dataset. The benchmark datasets were the EyePACS-K [98], Messidor-2 [93, 94], and the APTOS [79]. The Finnish dataset was collected in clinical work in the Central Finland Central Healthcare district. This dataset was called the KSSHP set. The EyePACS-K and KSSHP sets had sufficient images for training, whereas the Messidor-2 and APTOS sets were smaller, and thus, they were used exclusively for testing the out-of-distribution performance. The number of images and class distributions of the EyePACS-K and KSSHP sets are shown in Table 3.4 and of the APTOS and Messidor-2 sets in Table 3.5.

The set of approximate BNNs selected were deep ensembles, MC dropout, MFVI, GVI, and Radial MFVI. Ensembling was also additionally combined with MC dropout, MFVI, GVI, and Radial MFVI resulting in nine total BNN approximations. The baseline was a MAP network trained with dropout and l_2 weight regularization. The CNN architecture was selected as a VGG-like [95] network, similar to most of the previous works. The uncertainty was evaluated with the entropy of the posterior predictive distribution, and additionally for the PIRC task, with the QWK-Risk uncertainty measure. The utility of the uncertainty information was evaluated with the referral process by referring no data (0%), 30% of the

Table 3.4. Statistics of the EyePACS-K and KSSHP datasets.

EyePACS-K				
Task		Training Set	Validation Set	Test Set
RDR	Images Total	35 125 (100.0%)	10 906 (100.0%)	42 669 (100.0%)
	Images Class 0	28 252 (80.4%)	8850 (81.1%)	34 445 (80.7%)
	Images Class 1	6873 (10.5%)	2056 (18.9%)	8224 (19.3%)
PIRC	Images Total	35 125 (100.0%)	10 906 (100.0%)	42 669 (100.0%)
	Images Class 0	25 809 (73.4%)	8130 (74.5%)	31 403 (73.6%)
	Images Class 1	2443 (7.0%)	720 (6.6%)	3042 (7.1%)
	Images Class 2	5292 (15.1%)	1579 (14.5%)	6281 (14.7%)
	Images Class 3	873 (2.5%)	237 (2.2%)	977 (2.3%)
	Images Class 4	708 (2.0%)	240 (2.2%)	966 (2.3%)
KSSHP				
Task		Training Set	Validation Set	Test Set
RDR	Images Total	39 482 (100.0%)	5652 (100.0%)	11 285 (100.0%)
	Images Class 0	35 333 (89.5%)	5055 (89.4%)	10 094 (89.4%)
	Images Class 1	4149 (10.5%)	597 (10.6%)	1191 (10.6%)
PIRC	Images Total	39 482 (100.0%)	5652 (100.0%)	11 285 (100.0%)
	Images Class 0	27 086 (68.6%)	3857 (68.2%)	7723 (68.4%)
	Images Class 1	8434 (21.4%)	1224 (21.7%)	2431 (21.5%)
	Images Class 2	3350 (8.5%)	482 (8.5%)	930 (8.2%)
	Images Class 3	471 (1.2%)	80 (1.4%)	177 (1.6%)
	Images Class 4	141 (0.4%)	9 (0.2%)	24 (0.2%)

data, and 50% of the data. These proportions of referred data were called the 0%, 30%, and 50% referral levels. AUROC was used as the performance measure for the RDR task and QWK for the PIRC task.

For the RDR task, the findings were similar to those of Leibig et al. [97] and Band et al. [99]. When the models were trained on the EyePACS-K set, the Messidor-2 generalization of the uncertainty estimates was good, whereas for the APTOS set, the uncertainty estimates were detrimental to the referral process, as was also observed in Band et al. [99]. The EyePACS-K trained models benefitted from uncertainty for the clinical KSSHP set, however, not to the same extent as for the Messidor-2 set. When trained using the KSSHP set, the quality of the within-distribution uncertainty estimates was not as good as for the EyePACS-K trained models. The uncertainty estimates generalized to the EyePACS-K set, aside from those produced by the MAP, deep ensemble, and Radial MFVI ensemble, and they also generalized to the Messidor-2 set. Similar degradation of performance of the KSSHP trained models was observed for the APTOS set as with the EyePACS-K trained models.

Table 3.5. Statistics of the APTOS and Messidor-2 sets. Adapted from Publication II.

Task		APTOS	Messidor-2
RDR	Images Total	3662 (100.0%)	1744 (100.0%)
	Images Class 0	2175 (59.4%)	1279 (73.3%)
	Images Class 1	1487 (40.6%)	465 (26.7%)
PIRC	Images Total	3662 (100.0%)	1744 (100.0%)
	Images Class 0	1805 (49.3%)	1017 (58.3%)
	Images Class 1	370 (10.1%)	270 (15.5%)
	Images Class 2	999 (27.3%)	347 (19.9%)
	Images Class 3	193 (5.3%)	75 (4.3%)
	Images Class 4	295 (8.1%)	35 (2%)

In the PIRC task, the entropy of the posterior predictive distribution was first used as the measure of uncertainty. The EyePACS-K trained models benefitted mostly up to 30% referral level on the within-distribution test set, and no model improved consistently when referring data in the KSSHP out-of-distribution test. However, the referral process did improve the performance on the Messidor-2 and APTOS sets. When trained on the KSSHP set, no model consistently improved on the within-distribution test set and only the Radial MFVI, MFVI ensemble, and Radial MFVI ensemble improved consistently on the EyePACS-K set. All the models benefitted from uncertainty on the Messidor-2 set, however, only the ensembles benefitted consistently on the APTOS set.

When using the QWK-Risk as the uncertainty measure, both the EyePACS and KSSHP trained models improved on the respective within-distribution test sets when referring data. In addition, the EyePACS-K and KSSHP trained models benefitted from the uncertainty when using the other set as an out-of-distribution test set. It turned out that the EyePACS-K trained models benefitted from uncertainty on the Messidor-2 set, but not on the APTOS. Additionally, the performance of the KSSHP trained models degraded on both the Messidor-2 and the APTOS sets in contrast to the entropy-based uncertainty experiments. Results for the clinical KSSHP set, using both the entropy and QWK-Risk uncertainty measures, are presented in Table 3.6.

Table 3.6. PIRC results in the quadratic weighted Cohen's kappa for the clinical KSSHP set. "Ref. X%" denotes the referral level X%. Bold font denotes the best result within the referral level for an uncertainty measure. Results are given as the mean ± standard deviation of 100 bootstrap resamples of the KSSHP test set. Adapted from Publication II.

Uncertainty	Approach	Trained on EyePACS-K			Trained on KSSHP		
		Ref. 50%	Ref. 30%	Ref. 0%	Ref. 50%	Ref. 30%	Ref. 0%
Entropy	MAP	47.6 ± 3.1	58.8 ± 2.0	67.5 ± 0.9	70.9 ± 1.6	76.0 ± 1.0	81.0 ± 0.6
	MC dropout	32.6 ± 4.8	47.2 ± 2.7	67.7 ± 0.8	72.1 ± 1.4	75.3 ± 1.1	80.3 ± 0.6
	MFVI	40.5 ± 3.2	45.3 ± 2.4	63.8 ± 0.9	**81.4 ± 1.1**	**81.6 ± 0.8**	79.9 ± 0.6
	GVI	61.1 ± 1.8	65.6 ± 1.2	62.8 ± 0.8	74.8 ± 1.7	76.1 ± 1.1	80.0 ± 0.6
	Radial MFVI	49.9 ± 2.9	59.6 ± 2.1	65.4 ± 0.9	77.6 ± 1.6	78.5 ± 1.3	79.9 ± 0.6
	Deep ensemble	**62.3 ± 2.0**	**73.1 ± 1.1**	**69.6 ± 0.8**	78.4 ± 1.3	78.8 ± 1.0	**81.1 ± 0.5**
	MC dropout ensemble	43.7 ± 3.9	56.0 ± 2.0	68.4 ± 0.9	71.3 ± 1.8	75.0 ± 1.0	80.6 ± 0.6
	MFVI ensemble	24.4 ± 5.3	39.4 ± 2.9	66.1 ± 0.9	74.7 ± 1.5	77.7 ± 1.1	80.5 ± 0.6
	GVI ensemble	40.0 ± 3.4	55.0 ± 1.9	66.8 ± 0.9	66.6 ± 2.0	76.4 ± 1.1	80.9 ± 0.5
	Radial MFVI ensemble	33.4 ± 3.5	56.2 ± 2.2	66.2 ± 1.0	73.2 ± 1.6	77.8 ± 1.1	80.2 ± 0.6
QWK-Risk	MAP	81.4 ± 0.9	78.6 ± 0.9	67.5 ± 0.9	89.4 ± 0.5	87.9 ± 0.5	81.0 ± 0.6
	MC dropout	79.6 ± 1.0	77.5 ± 0.9	67.7 ± 0.8	88.6 ± 0.5	87.1 ± 0.5	80.3 ± 0.6
	MFVI	76.1 ± 1.0	74.5 ± 0.9	63.8 ± 0.9	87.3 ± 0.5	85.9 ± 0.5	79.9 ± 0.6
	GVI	75.8 ± 0.8	73.6 ± 0.8	62.8 ± 0.8	87.9 ± 0.4	86.5 ± 0.4	80.0 ± 0.6
	Radial MFVI	78.7 ± 0.9	76.6 ± 0.8	65.4 ± 0.9	87.8 ± 0.5	86.5 ± 0.5	79.9 ± 0.6
	Deep ensemble	**81.6 ± 0.8**	**80.2 ± 0.8**	**69.6 ± 0.8**	89.5 ± 0.4	88.1 ± 0.4	**81.1 ± 0.5**
	MC dropout ensemble	81.1 ± 0.9	78.9 ± 0.9	68.4 ± 0.9	89.4 ± 0.4	87.9 ± 0.5	80.6 ± 0.6
	MFVI ensemble	80.0 ± 0.9	77.9 ± 0.9	66.1 ± 0.9	89.7 ± 0.4	87.7 ± 0.4	80.5 ± 0.6
	GVI ensemble	79.3 ± 0.9	76.7 ± 0.9	66.8 ± 0.9	**89.9 ± 0.4**	**88.4 ± 0.4**	80.9 ± 0.5
	Radial MFVI ensemble	79.4 ± 1.0	77.1 ± 0.9	66.2 ± 1.0	89.1 ± 0.5	87.4 ± 0.5	80.2 ± 0.6

4. Deep Learning for Mandibular Canal Segmentation in CBCT Images (Publications III & IV)

The human jawbone, also called the mandible, has two nerve canals that pass through the bone under the teeth. These canals are called the mandibular canals and both of them contain an inferior alveolar nerve (IAN), an artery, and a vein. Different computed tomography (CT) techniques are used in dentomaxillofacial surgical planning, such as in dental implantology, where the canals need to be accurately located to avoid damage to the IAN and other structures. This is currently performed manually by 3D imaging tools, such as Planmeca Romexis® software. Automatic segmentation and localization of the mandibular canals would allow to reduce the manual burden of the task.

The conventional CT approach is the fan-beam X-ray source with an array of detectors that scans 2D axial slices of the patient anatomy, i.e. it samples slices of the plane that divides the body in the upper and lower parts [107]. These slices are then concatenated to produce a volumetric image, also called a scan, of the imaged area. Multi-detector variants (MDCT) can record multiple slices at the same time, such that the total imaging time can be reduced [107]. Cone beam CT (CBCT) is a more recent technique that has become popular in dentomaxillofacial radiology. It uses a cone shaped beam instead of a fan shaped beam [108]. It subjects the patient to less radiation and has comparable scanning time to MDCT [107], however, because of the lower radiation dosage, it has more noise and soft tissue resolution is decreased [108].

4.1 Prior Work

Mandibular canal segmentation using conventional CT images has been explored before, for example using a combination of voxel value thresholding, edge detection, and line tracking [109], and a combination of voxel value thresholding and active appearance models [110]. However, it has been observed that these algorithms might not generalize to the CBCT data [111]. Before 2020, there were five notable works that explored

mandibular canal segmentation from CBCT volumes. In Kainmueller et al. [112], a statistical shape model was used to extract a bone surface model of the mandible and a Dijkstra's algorithm-based path search was then used to find the canal tunnel inside the bone. In Kroon [111], Lukas-Kanade tracking, two registration methods, statistical shape model, and active appearance models were examined. In Moris et al. [113], a combination of thresholding and specialized hand-crafted algorithms were proposed. In Abdolali and Zoroofi [114], active appearance models and registration were used in conjunction, and in Abdolali et al. [42], a statistical shape model was used to model the mandible and the fast marching algorithm to find the canal inside it.

Deep learning was proposed for mandibular canal and teeth segmentation from orthopantomograms, i.e. panoramic-like 2D images that traces the center line of the mandible, in Vinayahalingam et al. [115]. More recently, Cha et al. [116] investigated maxillary sinus, maxilla, mandibular canal, and mandible segmentation from the orthopantomograms. These segmentation tasks differ from volumetric mandibular canal segmentation in that they cannot be used to obtain an accurate 3D model of the canals due to the orthopantomograms being projections to a 2D image.

In early 2020, Kwak et al. [117] used deep learning for mandibular canal segmentation from volumetric CBCT images. The work was published three days earlier than Publication III and proposed deep learning for the task concurrently. The work considered three network architectures: a 2D "SegNet", 2D U-Net, and 3D U-Net. The 2D variants were examined using two parameter initialization methods: random initialization and pretraining with natural images. The 2D variants were used to segment a single slice of a volume at a time, however, the 2D U-Net was also studied when four adjacent slices were used as different input channels. The CBCT volumes used in the work were preprocessed by a mandible extraction algorithm that combined voxel thresholding and heuristics to determine the mandible mask. If the extraction algorithm failed, it was tuned by hand to find the mandible. The CBCT volumes were then masked such that only the region inside the mandible was preserved. The work did not use established measures of mandibular canal segmentation performance, such as the Dice coefficient or mean curve distance, and thus the results of the study cannot be compared to those of Publication III. More recently, Kurt Bayrakdar et al. [118] also studied volumetric segmentation from CBCT images, however, the study did not use conventional performance measures either.

4.2 Deep Learning Segmentation Approach (Publication III)

In Publication III, the dataset consisted of 637 CBCT volumes from 594 patients from the Cranio and Dentomaxillofacial Radiology Department of The University Hospital of Tampere, Finland. The CBCTs were imaged using Soredex Scanora 3Dx and three Planmeca ProMax devices (Promax 3D, ProMax 3D Max, and ProMax 3D Mid). There were multiple resolutions among the volumes. 492 volumes had spatial resolution of 0.2 mm and 141 had spatial resolution of 0.4 mm. There were also resolutions 0.1 mm, 0.15 mm, 0.3 mm, and 0.6 mm, however, there was only one volume of each of these resolutions. The dataset was divided to training, validation, and test sets such that the patients with multiple volumes were included only in the training set. Also, the 0.2 mm and 0.4 mm volumes were divided among the sets in similar proportions. All the volumes with rare resolutions were placed in the training set. After the division, the training, validation, and test sets contained 457, 52, and 128 volumes, respectively.

The dataset had a number of heterogeneities present in some of the patients' CBCT volumes. Heterogeneities caused by the imaging procedure were differences in images based on the imaging device, artefacts caused by the motion of the patient, and variability in the pose of the head. The heterogeneities that were related to the health or operative status of the patients included metallic artefacts, osteoporosis, benign or malignant tumours, difficult anatomy or bone structure, and post bisagittal osteoma operation. In addition, some of the volumes were from cadavers. The heterogeneities were not mutually exclusive and it was possible for multiple heterogeneities to be present in a single volume.

All the volumes were annotated using Planmeca Romexis® software, which has a built-in mandibular canal annotation tool, by two medical professionals. The tool requires the user to specify spline control points on the mandibular canal path, which are then used with spline interpolation to create an approximate canal curve. The curve was then expanded to a 1.5 mm diameter tube along the curve path to approximate the canal surrounding the nerve. As the mandibular canal is not a constant diameter tube, these annotations can be thought to be a form of noisy labels. 15 volumes from the test set were also annotated in voxel-level detail using Mimics inPrint 3.0 software (Materialise, Leuven, Belgium). These high quality annotations were used to evaluate the segmentation performance with greater accuracy, whereas the 128 spline interpolation-based coarse annotations were utilized to examine the performance of the model with respect to the visual quality of the canals in the CBCT volumes.

The CNN architecture was a 3D modification to the U-Net architecture [13]. It is a type of fully-convolutional neural network with an encoder/decoder structure with skip connections between them. The output of the network was a volume with the same spatial shape as the input, and

by using a sigmoid activation these output voxel values were constrained to be within $[0, 1]$. The network was trained to minimize the so-called "Dice loss" [119] that is a continuous relaxation of the Dice coefficient defined as:

$$\mathcal{L}_{Dice}(f(\boldsymbol{x}), \boldsymbol{y}) = -2\frac{\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{d=1}^{D} f(\boldsymbol{x})_{h,w,d}\, \boldsymbol{y}_{h,w,d}}{\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{d=1}^{D} f(\boldsymbol{x})_{h,w,d} + \boldsymbol{y}_{h,w,d}}, \tag{4.1}$$

where f represents the 3D U-Net with the sigmoid output activation. It can be seen as a more ERM type objective due to the lack of a probabilistic interpretation. The Dice loss was selected as it performs well under class imbalance, which was very severe in the data with approximately 0.01% mandibular canal voxels, whereas the negative log-likelihood and weighted variants of it did not produce competitive Dice coefficient values.

Two data preprocessing steps were performed to reduce the memory footprint of the algorithm. First, each volume was resized to a 0.4 mm isotropic voxel resolution that served to standardize the data and to reduce the size of the largest volumes. Second, patches of shape $32 \times 32 \times 32$ were sampled randomly from the full CBCT volumes during training. The patch sampling was performed only around the mandibular canal, based on the annotations, to further reduce the class imbalance. During inference, the volumes were processed by dividing the entire volume to patches that were processed independently. To produce the segmentation map, the output of the model was thresholded, such that output voxels with values ≥ 0.5 were treated as positive, i.e. mandibular canal, and otherwise negative, i.e. not mandibular canal.

As the model was trained with patches from the neighborhood of the canal, there were false positive canal detections in the regions not present in the training data, such as in the nasal cavity and neck. These were typically small in size, and thus a post processing algorithm was developed to extract the most likely mandibular canal segmentations from the output of the model. The algorithm performed connected component analysis that first computed the connected components and then selected the two largest components as the mandibular canals. The mandibular canal curve, used to compute MCD, was extracted by a skeletonization algorithm.

Results

A comparison of the results of Publication III and those of Kainmueller et al. [112] and Abdolali et al. [42] are presented in Table 4.1. It can be seen that the deep learning approach outperforms the previous work in MCD and ASSD, presented in Equation (2.29) and Equation (2.30), respectively, and is also more stable, which can be seen as a lower standard deviation of the results. Indeed, the deep learning approach set a new state-of-the-art for mandibular canal segmentation. In the analysis of the performance conditional to the subjective visibility of the canal, it was found that when

Table 4.1. Mandibular canal segmentation results. Results are given as the mean ± standard deviation. Publication III results are computed on the high quality voxel-wise annotations. For MCD and ASSD lower is better and for Dice coefficient higher is better.

	Right Canal			Left Canal		
Method	MCD	ASSD	Dice	MCD	ASSD	Dice
Kainmueller et al. [112]	1.0 ± 0.6	-	-	1.2 ± 0.9	-	-
Abdolali et al. [42]	0.82 ± 0.25	0.84 ± 0.18	-	0.92 ± 0.15	0.79 ± 0.22	-
Publication III	**0.50 ± 0.19**	**0.45 ± 0.11**	0.58 ± 0.09	**0.61 ± 0.16**	**0.45 ± 0.12**	0.57 ± 0.08

the canal was marked as clearly visible, the model had similar performance as for the high quality annotations. However, for those canals that were difficult to distinguish, the model performance degraded significantly.

4.3 Multi-grader and Deep Learning Observer Variability in Mandibular Canal Segmentation Task (Publication IV)

In Publication III, the main objective was to introduce a deep learning approach for mandibular canal segmentation and to compare the approach to previously proposed methods. However, as the path of the mandibular canal can be ambiguous depending on the visibility of the canal, it is possible that different annotators have differing opinions on the path of the canal. Thus, it is hard to determine how well the CNN performs based on a single annotation of the canal. In Publication IV, a comprehensive validation of the deep learning method was performed with the main focus on comparing the canal segmentations produced by the CNN to the interobserver variability between multiple human annotators. In addition, the dataset in Publication IV was much larger and consisted of CBCT volumes imaged with five different scanners, and the patient population consisted of both Finnish and Thai patients.

The dataset consisted of 1132 CBCT images from 1103 patients. There were 649 images from Planmeca Promax 3D Max/Mid, 125 from Planmeca Viso G7, 124 from Soredex Scanora 3Dx, 120 from NSTDA DentiScan, and 114 from NewTom GiANO HR devices. The dataset had similar heterogeneneities as the dataset used in Publication III, as described in Section 4.2, however, the dataset had more devices and there was additional variability introduced by the inclusion of geographically and ethnically separate clinical data from Thailand. The data was divided to

validation and test sets, by selecting 20 images from each device for the validation set and 30 for the test set. The rest of the data was used as the training set. The test data was annotated by four radiologists, referred to as Expert1, Expert2, Expert3, and Expert4. Expert3 and Expert4 also annotated the training and validation data, however, they did not annotate the same data.

The deep learning model had the same architecture as in Publication III. The segmentation post processing script was further developed by concatenation of close components that were found using connected component analysis, and then a score for the anatomical plausibility of each component was computed. The score was used to filter out some components, and then the two most symmetric components were selected as the two canals.

The interobserver variability analysis was performed in three different ways. In the first experiment, the interobserver variability was evaluated with all pair-wise comparisons between the radiologists, and the human-model variability was evaluated by comparing the model to the individual radiologists. In the second experiment, the largest variability between the model and any radiologist was compared to the largest variability between the radiologists. In the third experiment, the four radiologists' segmentations were combined by majority voting to create a reference consensus segmentation. The radiologists and the model were then compared to it to evaluate the objective performance of them. A senior radiologist also examined the radiologists' annotations and the segmentations of the model to assess qualitatively the differences between the human and machine made errors.

For evaluation, the MCD, presented in Equation (2.29), and the symmetric mean curve distance, proposed in Publication IV, were used. As the MCD is not symmetric with respect to its arguments, meaning that $MCD(T, P)$ can have a different value than $MCD(P, T)$, it is hard to use for the summarization of the performance. For this reason, Publication IV proposed the *symmetric mean curve distance* (SMCD) for summarizing the distance between two curves T and P. It is computed as:

$$SMCD(T, P) = \frac{MCD(T, P) + MCD(P, T)}{2}. \tag{4.2}$$

It can be seen to be similar to the ASSD, presented in Equation (2.30), however, ASSD normalizes the sum of the distances by the number of total points on the two curves. The issue with this normalization scheme is that if one of the curves is much longer, it will dominate the ASSD value. In contrast, the SMCD is the average of the MCDs computed both ways, which ensures that even when one of the curves is much shorter, it will have the same influence on the final score as the other.

Table 4.2. Reference consensus segmentation results in millimetres. IQR stands for the interquartile range and STD for the standard deviation. Adapted from Publication IV.

Evaluator	Median (IQR) SMCD	Mean ± STD SMCD
Expert1	0.62 (0.23)	0.68 ± 0.38
Expert2	0.55 (0.22)	0.62 ± 0.39
Expert3	0.47 (0.14)	0.52 ± 0.38
Expert4	0.42 (0.14)	0.47 ± 0.40
Model	**0.39 (0.11)**	**0.46 ± 0.39**

Results

The median of all pair-wise MCD comparisons revealed that the lowest variability between any pairs was with the model and Expert4, and the highest between Expert1 and Expert3. Overall, the variability between the model and any radiologist was in the range 0.45–0.69 mm, whereas the interobserver variability was between 0.48–0.70 mm. In the largest variability analysis, the model had 0.74 (0.28) mm median (IQR) and 0.81 ± 0.41 mm mean ± standard deviation (STD) SMCD, compared to the interobserver variability with 0.77 (0.25) mm median (IQR) and 0.84 ± 0.28 mm mean ± STD.

For the reference consensus segmentation results, presented in Table 4.2, the model had the best performance in the median and mean of SMCD, followed by Expert4, Expert3, Expert2, and Expert1, in a decreasing order by performance. There were two canals with large errors by all experts and the model, which were identified manually to have been caused by disagreements between the experts. The disagreements in the canal path resulted in no consensus segmentation for the path of the canal on significant portions that resulted in high SMCD.

The qualitative analysis revealed that there were differences between the errors of the radiologists' annotations and the CNNs segmentations. The model produced too short canals for three out of the 300 (1%) test volumes. Two of the errors were produced by the post processing algorithm not connecting correct components, and one was deemed to have been caused by incorrect segmentation around an imaging artifact. The errors made by the radiologists were mostly annotations outside the canal path. Out of the 1200 total annotations, these errors were present in 29 of them (2.4%).

5. Machine Learning for Neonatal Mortality and Morbidity Prediction (Publication V)

Infants under the age of 28 days are called *neonates* and those with a birth weight under 1500 g are called very low birth weight (VLBW) infants. The VLBW infants have an increased risk of neonatal mortality and morbidity [120], and are treated in neonatal intensive care units (NICUs). In the Western Europe and USA, the VLBW neonates have a mortality rate around 11% [121], in comparison to the overall neonatal mortality rate around 1.8% [122]. Neonatal morbidities are associated with an increased risk of mortality, as well as possibly permanent ailments [123, 124, 125], and thus the early detection of the morbidities is of paramount interest for timely treatment. Several risk scores have been developed to aid in the estimation of the health of neonates [126, 127, 128], such as the SNAP-II and SNAPPE-II scores [129], however, these scores have been found to perform poorly on patient level prediction [130, 131].

In Publication V, machine learning approaches were studied for the prediction of VLBW neonatal in-hospital mortality, bronchopulmonary dysplasia (BPD), necrotizing enterocolitis (NEC), and retinopathy of prematurity (ROP). BPD is a chronic condition of the lungs that affects those born preterm, as the lungs are underdeveloped and prone to injury [132]. It can lead to long-term ailments such as persistent pulmonary dysfunction [123]. NEC is a disease of the gastrointestinal tract that can lead to bowel necrosis and death [133], but also has long-term issues such as gastrointestinal and neurodevelopmental problems [124]. ROP is retinopathy that affects preterm babies due to their underdeveloped retinas, which can lead to blindness [125].

5.1 Prior Work

Machine learning approaches have been proposed for neonatal mortality and morbidity prediction. Neonatal mortality prediction has been explored using a variety of machine learning models with static admission data [130, 134] and using decision trees with real-time data [135]. BPD

has been predicted using neural networks and SVMs with a combination of static and real-time data [136, 137]. In Saria et al. [138], nonlinear Bayesian models and logistic regression were proposed for neonatal "high morbidity" prediction, which was defined as a single label for many morbidities including the BPD, ROP, and NEC, with static and real-time data.

In Publication V, a continuation work of Rinta-Koski et al. [139] and Rinta-Koski et al. [140] was presented. In Rinta-Koski et al. [139], a Gaussian process classifier was used to predict the BPD, NEC, and ROP status of VLBW neonates. The data included static and time-series variables, and comparison was made between using the first 24 and 72 hours of time-series data. For the BPD classification, the classifier achieved 0.85 AUROC with 24h of time-series data and 0.87 AUROC with 72h of the data. For the NEC classification, the AUROC was 0.72 for 24h and 0.74 for 72h datasets, and for the ROP classification, these values were 0.80 and 0.84, respectively. It was observed that the sensitivity of the classifier was close to zero for the NEC and ROP tasks.

In Rinta-Koski et al. [140], a similar analysis was conducted for the in-hospital mortality of VLBW neonates. Four Gaussian process classifiers with different kernels and an SVM classifier were used with 12, 18, 24, 36, 48, and 72 hours of time-series data, along with static data. It was observed that all the models performed roughly equally, the Gaussian process classifier with linear plus constant kernel being the best achieving 0.949 AUROC with 48 and 36 hours of data. The classifier performance was also compared to the medical risk scores and it was found that the machine learning approaches outperformed them, as the best score "SNAPPE-II" achieved 0.875 AUROC.

5.2 Helsinki University Hospital Data

The dataset used in Publication V was collected in clinical work in the NICU of Children's Hospital of Helsinki University Hospital between 1999 and 2013. It was partly the same as the dataset used in Rinta-Koski et al. [139] and Rinta-Koski et al. [140], however, it included more patients. A detailed description of the collection and data format can be found in Rinta-Koski [141]. The static and time-series variables were chosen as in Rinta-Koski et al. [140]: SNAP-II, SNAPPE-II, birth weight, and gestational age as the static variables, and as the time-series variables the oxygen saturation, heart rate, and systolic, diastolic, and mean blood pressure. The SNAP-II and SNAPPE-II scores were chosen as input variables, because an ablation result in Rinta-Koski et al. [140] suggests that they improve the performance slightly.

Some patients were excluded from the dataset. Firstly, patients who had died or were discharged less than 72 hours after admission were

excluded. This was to ensure that the mortality label was not leaked by decayed vital signs. Secondly, if any of the time-series had less than 50 measurements the patient was rejected. This was to limit the noise in the feature extraction step that is described later. The resulting dataset had 977 patients out of which 63 had died, 275 had BPD, 31 had NEC, and 77 had ROP.

Similar to Rinta-Koski et al. [139] and Rinta-Koski et al. [140], the first 72 hours of time-series data was taken into account to examine the early prediction of the different end-points. As the mean days of hospitalization was over 20 days for mortality and all of the morbidities, the 72h period could be considered as an early prediction period. In addition, the classifier performance was also examined conditional to the length of the time-series, by using the first 12, 18, 24, 36, 48, or 72 hours them. Features were extracted from the time-series in a similar manner as in Rinta-Koski et al. [140]. These features were the mean and standard deviation of each time-series, and they were concatenated with the static data to create a vector input for the models.

As the dataset was relatively small and very imbalanced, stratified 8-fold nested cross validation with 8 repeats was used. The training of the models was implemented in MATLAB [142] using the built-in functions, except for the Gaussian processes that were implemented using the GPstuff package [143]. The performance measures were computed using scikit-learn package [144] in Python.

Results

A large set of machine learning classifiers was selected to comprehensively study the machine learning approach for the different tasks. The models were the logistic regression (LR), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-nearest neighbor (KNN), support vector machine (SVM), three different Gaussian processes, and random forest (RF) classifier. The three GP classifiers differed in the choice of the kernel function. All of them included the sum of a constant kernel, linear kernel, and either a squared exponential (GP-SE), Matérn with $\nu = 3/2$ (GP-M32), or Matérn with $\nu = 5/2$ kernel (GP-M52). Description of these kernels can be found in Rasmussen and Williams [44].

In Rinta-Koski et al. [139] and Rinta-Koski et al. [140], it was observed that the sensitivity of the classifiers were either moderate or close to zero, especially for the ROP classification. In Publication V, multiple data preprocessing methods were used to examine if the predictive performance on the positive cases could be improved. Namely, 1. standardization of the irregularly sampled time-series by repeating the last observation to fill missing observations, 2. excluding the first 6 hours of time-series data to examine if the neonatal adaptation period degrades the quality of the

extracted features, and 3. subsampling the majority, i.e. the negative, class examples of the training set to achieve uniform class distribution. All possible combinations of the steps 1-3 were used in addition to using none of them, and thus there were in total 8 sets of data. In addition, the F1-score was used as an alternative to the AUROC for classifier selection, in order to determine if it could be used to improve the performance on the positive cases.

When the best classifier was selected based on the highest AUROC on the 8 sets of data, the results were generally similar in the mortality classification to Rinta-Koski et al. [140] and in the BPD, NEC, and ROP classification to Rinta-Koski et al. [139]. However, the RF classifier was observed to outperform other models in mortality and NEC classification, and obtaining the same AUROC as GP classifiers in ROP classification, but with a higher sensitivity. Similar to the previous works, the sensitivities of the GP and SVM classifiers were low or zero on the NEC and ROP classification tasks. The highest AUROC was achieved for most models with the full set of data without the majority subsampling strategy.

In the second experiment, the F1-score was used to select the classifiers. There was a favourable tradeoff between the AUROC and sensitivity as a result, since no classifier had zero sensitivity in any task, but the corresponding AUROC values decreased only slightly. For those classifiers that had very low sensitivity on the NEC and ROP tasks, the F1-score classifier selection preferred the classifiers trained on the majority subsampled data. Indeed, the best GP classifiers and the SVM were all trained on the subsampled data in all the tasks, whereas in the AUROC selection they were all trained with all the data.

Out of all the models, the RF had a consistently high performance in every task. In the experiments where the RF classifier was selected based on AUROC, it had the highest AUROC in mortality and NEC classification, highest AUROC with the GP classifiers in ROP classification, and the second highest AUROC in BPD classification. In terms of F1-score, the RF classifier had the best performance in every experiment, even when selected based on the AUROC value. The RF classifier is compared to the GP classifiers presented in Rinta-Koski et al. [139], Rinta-Koski et al. [140], and Publication V in Table 5.1 that also illustrates the improved sensitivity of the GP classifiers when using majority class subsampled data.

When examining the performance of the classifiers on time-series with different lengths, results for the mortality, BPD, and ROP classification showed only minor improvements when using a longer period. However, on the NEC classification, all the classifiers had some benefits of using longer periods, the RF the most in terms of the F1-score and the QDA in terms of the AUROC. The results suggest that the mortality, BPD, and ROP classification can be performed with as short as 12 hours of patient

Table 5.1. Classification results for GP and RF classifiers. Results in AUROC (AUC), F1-score (F1), sensitivity (Sens.), precision (Prec.), and Specificity (Spec.). Criterion denotes the classifier selection criterion. "Data" is in format time-series length (hours) - empirical or uniform class distribution (E/U).

Task	Method	Criterion	Data	AUC	F1	Sens.	Prec.	Spec.
In-hospital mortality	GP-SE [140]	-	48h-E	**0.95**	-	0.46	**0.66**	0.98
	GP-M32 [140]	-	48h-E	**0.95**	-	0.45	0.65	0.98
	GP-M52 [140]	-	48h-E	**0.95**	-	0.44	**0.66**	0.98
	GP-SE Pub. V	AUC	18h-E	0.92	0.23	0.16	0.49	**0.99**
	GP-M32 Pub. V	AUC	72h-E	0.92	0.33	0.24	0.62	**0.99**
	GP-M52 Pub. V	AUC	36h-E	0.92	0.32	0.24	0.59	**0.99**
	RF Pub. V	AUC	36h-E	0.92	0.48	0.67	0.38	0.92
	GP-SE Pub. V	F1	72h-U	0.91	0.39	0.85	0.25	0.82
	GP-M32 Pub. V	F1	72h-U	0.91	0.39	**0.86**	0.25	0.82
	GP-M52 Pub. V	F1	72h-U	0.91	0.39	**0.86**	0.25	0.82
	RF Pub. V	F1	72h-E	0.92	**0.49**	0.71	0.38	0.92
BPD	GP-SE [139]	-	72h-E	0.87	-	0.52	0.67	**0.93**
	GP-SE Pub. V	AUC	72h-E	**0.89**	0.69	0.68	0.69	0.88
	GP-M32 Pub. V	AUC	72h-E	**0.89**	0.69	0.68	**0.70**	0.88
	GP-M52 Pub. V	AUC	72h-E	**0.89**	0.69	0.68	**0.70**	0.88
	RF Pub. V	AUC	72h-E	0.88	**0.70**	0.77	0.65	0.83
	GP-SE Pub. V	F1	72h-U	0.88	0.69	**0.86**	0.58	0.75
	GP-M32 Pub. V	F1	72h-U	0.88	0.69	0.85	0.58	0.75
	GP-M52 Pub. V	F1	72h-U	0.88	0.69	0.85	0.58	0.76
	RF Pub. V	F1	72h-E	0.88	**0.70**	0.77	0.65	0.83
NEC	GP-SE [139]	-	72h-E	0.74	-	0.17	0.11	0.98
	GP-SE Pub. V	AUC	72h-E	0.78	0.0	0.0	0.0	**1.0**
	GP-M32 Pub. V	AUC	72h-E	0.79	0.0	0.0	0.0	**1.0**
	GP-M52 Pub. V	AUC	72h-E	0.79	0.0	0.0	0.0	**1.0**
	RF Pub. V	AUC	48h-E	**0.81**	0.19	0.25	0.16	0.96
	GP-SE Pub. V	F1	72h-U	0.78	0.13	0.70	0.07	0.69
	GP-M32 Pub. V	F1	72h-U	0.78	0.16	**0.71**	0.07	0.68
	GP-M52 Pub. V	F1	72h-U	0.76	0.12	0.67	0.07	0.70
	RF Pub. V	F1	72h-E	0.79	**0.22**	0.27	**0.20**	0.96
ROP	GP-SE [139]	-	72h-E	0.84	-	0.05	0.50	0.99
	GP-SE Pub. V	AUC	72h-E	**0.85**	0.0	0.0	0.0	**1.0**
	GP-M32 Pub. V	AUC	72h-E	**0.85**	0.0	0.0	0.0	**1.0**
	GP-M52 Pub. V	AUC	72h-E	**0.85**	0.0	0.0	0.0	**1.0**
	RF Pub. V	AUC	48h-E	**0.85**	**0.37**	0.64	**0.26**	0.84
	GP-SE Pub. V	F1	72h-U	0.84	0.33	**0.84**	0.21	0.72
	GP-M32 Pub. V	F1	72h-U	0.84	0.33	**0.84**	0.21	0.72
	GP-M52 Pub. V	F1	72h-U	0.84	0.33	**0.84**	0.21	0.72
	RF Pub. V	F1	72h-E	**0.85**	**0.37**	0.64	**0.26**	0.84

monitoring data. There was generally no difference when excluding the first 6 hours of data.

Feature Importance Analysis

The random forest classifier had generally the best performance in all the tasks. To examine what features the classifier used, the feature importances were computed with random feature permutation and out-of-bag error. It permutes randomly one feature across the patients and computes the error on the out-of-bag, i.e. a held out subset, and the error is considered to be the measure of importance to the classification. The feature importances are presented in Figure 5.1.

For the mortality classification, the birth weight had the largest importance, and for the BPD and NEC classification, it had the second largest importance. This observation is in line with the general understanding that low birth weight increases the risk of mortality [145]. In addition, for the mortality classification, the standard deviation of blood oxygen, mean of systolic blood pressure, and mean of mean blood pressure had generally high importances. For the BPD classification, the gestational age had the highest importance and the SNAPPE-II had the third highest importance, however, with a similar value as the second most important variable birth weight. In NEC classification, the mean of blood oxygen had the highest importance and the mean of systolic blood pressure had only slightly lower importance than the birth weight. Interestingly, for the ROP classification, the birth weight had negative importance, which indicates that the feature is not used in a meaningful manner. In addition, all the important features were based on the time-series features, the highest being the standard deviation of diastolic blood pressure, which was followed by the standard deviation of systolic blood pressure, mean of blood oxygen, and mean of mean blood pressure.

(a) Mortality **(b)** BPD

(c) NEC **(d)** ROP

Figure 5.1. Feature importances. μ and σ denote the mean and the standard deviation components of the feature extraction. Adapted from Publication V.

6. Summary of Publications

This chapter summarizes Publications I–V and the contributions of this thesis.

6.1 Publication I

This publication presented a deep learning approach for classification of diabetic retinopathy and macular edema using a relatively small Finnish dataset of color fundus images. The work investigated two clinically used classification systems, as well as three simplified systems derived from them. It also systematically analyzed the relationship between the resolution of the images and the deep learning classifier performance. Higher resolution generally improved the performance in comparison to the lower resolutions, however, using the highest resolution 2095×2095 had negligible or decreasing effect to the performance. In addition, ensembling was found to improve the performance of the models trained with the 512×512 resolution. The novelty and the contributions of this publication are: demonstration that state-of-the-art CNN classifiers can be trained for diabetic retinopathy and macular edema classification using approximately 40 000 images, systematic investigation of the relationship between image resolution and classifier performance that provided evidence that 512×512 is sufficient for the classification tasks, and first ever results for a deep learning classification of a clinically used diabetic macular edema classification system using retinal fundus images.

6.2 Publication II

This publication investigated approximate Bayesian deep learning approaches for uncertainty-aware diabetic retinopathy classification. Nine BNN approximations were examined with a binary and a clinical 5-class diabetic retinopathy classification system. Furthermore, an uncurated

clinical hospital dataset was used in addition to three benchmark datasets. The publication validated results from previous studies on the benchmark datasets with the binary classification system. It was discovered that the methods developed for the benchmark datasets did not generalize to the clinical dataset when the clinical classification system was used. A classifier risk-based approach was used to develop an uncertainty measure, as an alternative to the entropy of the posterior predictive distribution, that improved the utility of the referral process on the clinical classification system for both the clinical and one of the benchmark datasets. The novelty and the contributions of this publication are: first ever results for approximate BNNs for the clinically used 5-class proposed international diabetic retinopathy classification system and clinical hospital data, derivation of a connection between uncertainty-based referral process and reject option classification, and development of the QWK-Risk uncertainty measure that could be used to improve the performance of the referral process for clinical data.

6.3 Publication III

This publication considered deep learning for mandibular canal segmentation from volumetric CBCT images. It was shown that the deep learning approach could outperform previously proposed methods, even though the training data was annotated with sparse control points and spline interpolation that resulted in noisy voxel labels. It was observed that the errors of the deep learning model correlated with the subjective visibility of the canal, which manifested as higher errors for those canals deemed to be not clearly visible. The novelty and the contributions of this publication are: proposal of a deep learning approach for segmentation of the mandibular canal from volumetric CBCT images, demonstration that a high voxel-level segmentation performance can be achieved with approximate training annotations, and setting a new state-of-the-art for the segmentation task.

6.4 Publication IV

This publication validated the model presented in Publication III in a more clinically oriented manner. The interobserver variability of four radiologists was compared to the variability between the radiologists and the deep learning model. It was found that the deep learning model had lower variability to the individual radiologists than the interobserver variability between the radiologists. In addition, the highest level of disagreement was larger between the radiologists than between them and the model. When comparing the radiologists and the deep learning model to a consen-

sus reference segmentation, the model showed the highest performance. The novelty and the contributions of this publication are: a first ever study comparing the interobserver variability of radiologists to a deep learning model in the mandibular canal segmentation task, demonstration that a deep learning model can segment the mandibular canal with a lower variability to the radiologists' estimates than the interobserver variability between them, and demonstration that a deep learning model can reach a more accurate segmentation performance than radiologists when comparing to a consensus segmentation.

6.5 Publication V

This publication systematically investigated machine learning for early detection of VLBW neonatal mortality, bronchopulmonary dysplasia, necrotizing enterocolitis, and retinopathy of prematurity. The data included patient demographic variables, medical indices, and physiological time-series measurements. Different data preprocessing methods were examined and it was found that the majority class subsampling could be used to improve the sensitivity of the models proposed in previous studies. The random forest classifier was found to be generally the best in every task. Increasing the length of the time-series was found to improve performance, however, only marginally, and in the end, even 12 hours of monitor data was sufficient for the mortality, BPD, and ROP classification tasks with most of the classifiers. Examination of the random forest variable importances revealed that for most of the tasks it utilized birth weight as the most important variable that is also used as a traditional predictive measure of neonatal well-being. The novelty and the contributions of this publication are: systematic analysis of different machine learning classifiers, data preprocessing methods, and time-series lengths for neonatal mortality and morbidity prediction, validation of the feature extraction method proposed in previous studies, demonstration that the prediction tasks are possible with even 12 hours of monitor data, and analysis of the discriminative features for the tasks using the random forest classifier.

7. Discussion and Concluding Remarks

Machine learning has the potential to automate many tasks that are currently performed manually by experts. In healthcare, machine learning could reduce the burden on medical professionals by automating screening or assisting in diagnosis, for example. The code associated with machine learning can be, and often is, executed in cloud-based services, and thus it would be possible to offer some form of diagnostic services with only an internet connection, given that the input data can collected without special equipment. This in turn would allow for a more wide coverage of healthcare services, for instance in developing countries.

However, in order for these systems to become widely used and accepted, extensive clinical validation is required to evaluate the clinical utility and risk. A weak point of some machine learning models, especially the deep neural networks, is the lack of human interpretable explainability of the approaches. For deep neural networks, this issue can be even worse due to the observed poor calibration of them. Publication I, Publication III, and Publication V have served to propose and analyze machine learning methods in healthcare applications. Publication II has analyzed the benefits of uncertainty information on clinical data, and also proposed a more clinically oriented approach for measuring the uncertainty. In Publication IV, a clinically oriented validation of deep learning segmentation was presented that also analyzed the human interobserver variability in the task.

These Publications leave space for future research to the subjects. In Publication I, the Finnish dataset was very homogeneous, as regards to the population and imaging devices. This aspect was addressed in Publication II comprehensively, by using a clinical dataset and three benchmark datasets, but the results did not turn out to be as good as in Publication I. This might be because the VGG-like network in Publication II is not as large in the terms of the parameters and depth as the Inception-v3 used in Publication I. A possible future research avenue would be to investigate the approximate BNNs with a more modern network architecture, such as the ResNet-50 used in Band et al. [99]. This could provide more accurate classification performance with the benefits of leveraging the uncertainty

information for referring the uncertain images. A limitation of the Publication II is that the approximate Bayesian methods cannot be compared against the true HMC solution due to computational complexity. Thus, it is possible that the entropy would work well as an uncertainty measure given the HMC posterior predictive distribution. However, this can only be addressed in the future when more computational power becomes cheaply available. On the other hand, developing better approximations to the BNN posterior distribution is of special interest and a possible avenue for future work. Planned future research within this domain also includes application of BNN approximations to medical segmentation tasks. Scalable approximations to the deep convolutional Gaussian processes [146] could also be used in these tasks.

Due to the large amount of experiments, data augmentation methods were not examined in Publication V. For example the SMOTE [147] or the generative adversarial network-based actGAN [148] could be used to create synthetic examples of the minority class, which would mean that data would not need to be excluded, as was done with the majority subsampling method. In future research, it is planned to utilize machine learning models that do not require the feature extraction from the time-series, i.e. time-series classifiers. Also, neonatal sepsis and other ICU prediction tasks are planned to be studied with machine learning methods.

An interesting future research avenue would be the use of the so-called wavelet scattering networks [149] for diabetic retinopathy and other medical classification tasks. They are deep convolutional networks that have predetermined wavelet convolutional filters, and have been found to perform well on small datasets. However, it has been observed that they require some learned layers on demanding tasks [150, 151]. As the wavelet scattering network is not learned, every component remains interpretable, which could increase the interpretability of a machine learning algorithm using the features generated by it. Indeed, interpretable and controllable medical segmentation is planned to be researched in future.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1026–1034.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[4] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 852–863.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[6] F. Rosenblatt, "The perceptron - a perceiving and recognizing automaton," Cornell Aeronautical Laboratory, Ithaca, New York, Tech. Rep. 85-460-1, January 1957.

[7] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, Jan 1982.

[8] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel, "The Helmholtz machine," *Neural Computation*, vol. 7, no. 5, pp. 889–904, 09 1995.

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[11] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster, "Development and validation of

a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 12 2016.

[12] C. Spampinato, S. Palazzo, D. Giordano, M. Aldinucci, and R. Leonardi, "Deep learning for automated skeletal bone age assessment in X-ray images," *Medical Image Analysis*, vol. 36, pp. 41–51, 2017.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015, pp. 234–241.

[14] X. Yi and P. Babyn, "Sharpness-aware low-dose CT denoising using conditional generative adversarial network," *Journal of Digital Imaging*, vol. 31, no. 5, pp. 655–669, Oct 2018.

[15] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King, "Key challenges for delivering clinical impact with artificial intelligence," *BMC Medicine*, vol. 17, no. 1, p. 195, Oct 2019.

[16] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 1321–1330.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2006.

[19] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[20] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[21] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Springer Series in Statistics. Springer, 2009.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014.

[23] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 13–18 Jul 2020, pp. 1691–1703.

[24] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning image restoration without clean data," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 2965–2974.

[25] Y. LeCun and I. Misra, "Self-supervised learning: The dark matter of intelligence," 2021, last accessed on 2022-01-12. [Online]. Available: https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan 2016.

[27] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, Feb 2020.

[28] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez, "Ordinal regression methods: Survey and experimental study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2016.

[29] E. Alpaydin, *Introduction to Machine Learning*, ser. Adaptive Computation and Machine Learning. The MIT Press, 2014, vol. Third edition.

[30] R. Szeliski, *Recognition*. Cham: Springer International Publishing, 2022, pp. 273–331.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[32] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*. CRC Press LLC, 2013.

[33] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013, no. 3.

[34] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed. MIT Press, 2018.

[35] G. Hripcsak and A. S. Rothschild, "Agreement, the F-measure, and reliability in information retrieval," *Journal of the American Medical Informatics Association*, vol. 12, no. 3, pp. 296–298, 05 2005.

[36] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[37] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[38] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[39] ——, "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit," *Psychological Bulletin*, vol. 70, no. 4, pp. 213–220, 1968.

[40] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, p. 29, Aug 2015.

[41] G. Kim, J. Lee, H. Lee, J. Seo, Y.-M. Koo, Y.-G. Shin, and B. Kim, "Automatic extraction of inferior alveolar nerve canal using feature-enhancing panoramic volume rendering," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 2, pp. 253–264, 2011.

[42] F. Abdolali, R. A. Zoroofi, M. Abdolali, F. Yokota, Y. Otake, and Y. Sato, "Automatic segmentation of mandibular canal in cone beam CT images using conditional statistical shape model and fast marching," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 4, pp. 581–593, 2017.

[43] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. 70, pp. 2079–2107, 2010.

[44] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[45] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[46] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015, arXiv:1412.6980v9 [cs.LG].

[48] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014.

[49] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[50] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[52] R. M. Neal, *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

[53] A. G. Wilson and P. Izmailov, "Bayesian deep learning and a probabilistic perspective of generalization," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 4697–4708.

[54] J. Knoblauch, J. Jewson, and T. Damoulas, "Generalized variational inference: Three arguments for deriving new posteriors," *arXiv preprint arXiv:1904.02063*, 2019.

[55] P. Izmailov, S. Vikram, M. D. Hoffman, and A. G. Wilson, "What are Bayesian neural network posteriors really like?" in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 18–24 Jul 2021, pp. 4629–4640.

[56] R. M. Neal *et al.*, "MCMC using Hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 2, no. 11, p. 2, 2011.

[57] N. P. Jouppi, D. H. Yoon, G. Kurian, S. Li, N. Patil, J. Laudon, C. Young, and D. Patterson, "A domain-specific supercomputer for training deep neural networks," *Communications of the ACM*, vol. 63, no. 7, pp. 67–78, 2020.

[58] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient Hamiltonian Monte Carlo," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1683–1691.

[59] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[60] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[61] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059.

[62] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[63] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, vol. 24. Curran Associates, Inc., 2011.

[64] A. Filos, S. Farquhar, A. N. Gomez, T. G. Rudner, Z. Kenton, L. Smith, M. Alizadeh, A. de Kroon, and Y. Gal, "A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks," *arXiv preprint arXiv:1912.10481*, 2019.

[65] S. Farquhar, M. A. Osborne, and Y. Gal, "Radial Bayesian neural networks: Beyond discrete support in large-scale Bayesian deep learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 108. PMLR, 26–28 Aug 2020, pp. 1352–1362.

[66] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1613–1622.

[67] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, pp. 429–449, 2002, 5.

[68] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04.

[69] Ministry of Social Affairs and Health, Finland, "Medical Research Act No. 488/1999," www.finlex.fi.

[70] ——, "Secondary Use of Health and Social Data Act No. 552/2019," www.finlex.fi.

[71] P. Saeedi, I. Petersohn, P. Salpea, B. Malanda, S. Karuranga, N. Unwin, S. Colagiuri, L. Guariguata, A. A. Motala, K. Ogurtsova, J. E. Shaw, D. Bright, and R. Williams, "Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas," *Diabetes Research and Clinical Practice*, vol. 157, p. 107843, 2019.

[72] H. Khalil, "Diabetes microvascular complications—a clinical update," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 11, pp. S133–S139, 2017, sI: Online Supplement - 1.

[73] R. Taylor and D. Batey, Eds., *Handbook of Retinal Screening in Diabetes: Diagnosis and Management*, 2nd ed. Wiley-Blackwell, 2012.

[74] N. Cheung, P. Mitchell, and T. Y. Wong, "Diabetic retinopathy," *The Lancet*, vol. 376, no. 9735, pp. 124–136, 2010.

[75] M. D. Davis, M. R. Fisher, R. E. Gangnon, F. Barton, L. M. Aiello, E. Y. Chew, r. Ferris, F L, and G. L. Knatterud, "Risk factors for high-risk proliferative diabetic retinopathy and severe visual loss: Early Treatment Diabetic Retinopathy Study report #18." *Investigative Ophthalmology & Visual Science*, vol. 39, no. 2, pp. 233–252, 02 1998.

[76] C. Wilkinson, F. L. Ferris, R. E. Klein, P. P. Lee, C. D. Agardh, M. Davis, D. Dills, A. Kampik, R. Pararajasegaram, and J. T. Verdaguer, "Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales," *Ophthalmology*, vol. 110, no. 9, pp. 1677–1682, 2003.

[77] Diabetic Retinopathy: Care Guidelines Abstract, 2014, "Diabetic Retinopathy. Current Care Guidelines. Working group set up by the Finnish Medical Society Duodecim, Suomen Silmälääkäriyhdistys and Finnish Diabetes Associations physician council. Helsinki: The Finnish Medical Society Duodecim, 2014 (referred March 4, 2022). Available online at: www.kaypahoito.fi," 2014.

[78] Kaggle, "Diabetic retinopathy detection challenge," 2015. [Online]. Available: https://www.kaggle.com/c/diabetic-retinopathy-detection

[79] Aravind Eye Hospital & PG Institute of Ophthalmology (managed and run by the Govel Trust) in support of the Asia Pacific Tele-Ophthalmology Society (APTOS) Symposium, "APTOS 2019 blindness detection," https://www.kaggle.com/c/aptos2019-blindness-detection/overview/aptos-2019, 2019, accessed: 2020-01-01.

[80] D. S. W. Ting, C. Y.-L. Cheung, G. Lim, G. S. W. Tan, N. D. Quang, A. Gan, H. Hamzah, R. Garcia-Franco, I. Y. San Yeo, S. Y. Lee, E. Y. M. Wong, C. Sabanayagam, M. Baskaran, F. Ibrahim, N. C. Tan, E. A. Finkelstein, E. L. Lamoureux, I. Y. Wong, N. M. Bressler, S. Sivaprasad, R. Varma, J. B. Jonas, M. G. He, C.-Y. Cheng, G. C. M. Cheung, T. Aung, W. Hsu, M. L. Lee, and T. Y. Wong, "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes," *JAMA*, vol. 318, no. 22, pp. 2211–2223, 12 2017.

[81] M. D. Abràmoff, M. Niemeijer, M. S. Suttorp-Schulten, M. A. Viergever, S. R. Russell, and B. van Ginneken, "Evaluation of a system for automatic detection of diabetic retinopathy from color fundus photographs in a large population of patients with diabetes," *Diabetes Care*, vol. 31, no. 2, pp. 193–198, 02 2008.

[82] M. D. Abràmoff, J. M. Reinhardt, S. R. Russell, J. C. Folk, V. B. Mahajan, M. Niemeijer, and G. Quellec, "Automated early detection of diabetic retinopathy," *Ophthalmology*, vol. 117, no. 6, pp. 1147–1154, Jun 2010.

[83] G. G. Gardner, D. Keating, T. H. Williamson, and A. T. Elliott, "Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool." *British Journal of Ophthalmology*, vol. 80, no. 11, pp. 940–944, 1996.

[84] R. Priya and P. Aruna, "SVM and neural network based diagnosis of diabetic retinopathy," *International Journal of Computer Applications*, vol. 41, no. 1, 2012.

[85] V. Lakshminarayanan, H. Kheradfallah, A. Sarkar, and J. Jothi Balaji, "Automated detection and diagnosis of diabetic retinopathy: A comprehensive survey," *Journal of Imaging*, vol. 7, no. 9, 2021.

[86] N. Tsiknakis, D. Theodoropoulos, G. Manikis, E. Ktistakis, O. Boutsora, A. Berto, F. Scarpa, A. Scarpa, D. I. Fotiadis, and K. Marias, "Deep learning for diabetic retinopathy detection and classification based on fundus images: A review," *Computers in Biology and Medicine*, vol. 135, p. 104599, 2021.

[87] J. Krause, V. Gulshan, E. Rahimy, P. Karth, K. Widner, G. S. Corrado, L. Peng, and D. R. Webster, "Grader variability and the importance of reference standards for evaluating machine learning models for diabetic retinopathy," *Ophthalmology*, vol. 125, no. 8, pp. 1264–1272, Aug 2018.

[88] A. Benzamin and C. Chakraborty, "Detection of hard exudates in retinal fundus images using deep learning," in *2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 2018, pp. 465–469.

[89] Y. Xu, Z. Zhou, X. Li, N. Zhang, M. Zhang, and P. Wei, "Ffu-net: Feature fusion u-net for lesion segmentation of diabetic retinopathy," *BioMed Research International*, vol. 2021, p. 6644071, Jan 2021.

[90] J. I. Orlando, E. Prokofyeva, M. del Fresno, and M. B. Blaschko, "An ensemble deep learning based approach for red lesion detection in fundus images," *Computer Methods and Programs in Biomedicine*, vol. 153, pp. 115–127, 2018.

[91] F. Arcadu, F. Benmansour, A. Maunz, J. Willis, Z. Haskova, and M. Prunotto, "Deep learning algorithm predicts diabetic retinopathy progression in individual patients," *npj Digital Medicine*, vol. 2, no. 1, p. 92, Sep 2019.

[92] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

[93] E. Decencière, X. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay *et al.*, "Feedback on a publicly distributed image database: the Messidor database," *Image Analysis & Stereology*, vol. 33, no. 3, pp. 231–234, 2014.

[94] M. D. Abràmoff, J. C. Folk, D. P. Han, J. D. Walker, D. F. Williams, S. R. Russell, P. Massin, B. Cochener, P. Gain, L. Tang *et al.*, "Automated analysis of retinal images for detection of referable diabetic retinopathy," *JAMA Ophthalmology*, vol. 131, no. 3, pp. 351–357, 2013.

[95] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[96] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. AAAI Press, 2017, p. 4278–4284.

[97] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Scientific Reports*, vol. 7, no. 1, p. 17816, Dec 2017.

[98] J. Cuadros and G. Bresnick, "EyePACS: An adaptable telemedicine system for diabetic retinopathy screening," *Journal of Diabetes Science and Technology*, vol. 3, no. 3, pp. 509–516, 2009.

[99] N. Band, T. G. Rudner, Q. Feng, A. Filos, Z. Nado, M. W. Dusenberry, G. Jerfel, D. Tran, and Y. Gal, "Benchmarking Bayesian deep learning on diabetic retinopathy detection tasks," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[100] M. S. Ayhan and P. Berens, "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," in *International Conference on Medical Imaging with Deep Learning*, 2018.

[101] M. S. Ayhan, L. Kühlewein, G. Aliyeva, W. Inhoffen, F. Ziemssen, and P. Berens, "Expert-validated estimation of diagnostic uncertainty for deep neural networks in diabetic retinopathy detection," *Medical Image Analysis*, vol. 64, p. 101724, 2020.

[102] T. Araújo, G. Aresta, L. Mendonça, S. Penas, C. Maia, Ângela Carneiro, A. M. Mendonça, and A. Campilho, "DR|GRADUATE: Uncertainty-aware deep learning-based diabetic retinopathy grading in eye fundus images," *Medical Image Analysis*, vol. 63, p. 101715, 2020.

[103] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.

[104] V. Franc and D. Prusa, "On discriminative learning of prediction uncertainty," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 1963–1971.

[105] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.

[106] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Computation*, vol. 16, no. 5, pp. 1063–1076, 05 2004.

[107] W. C. Scarfe, A. G. Farman, P. Sukovic *et al.*, "Clinical applications of cone-beam computed tomography in dental practice," *Journal of the Canadian Dental Association*, vol. 72, no. 1, p. 75, 2006.

[108] C. H. Kau, M. Božič, J. English, R. Lee, H. Bussa, and R. K. Ellis, "Cone-beam computed tomography of the maxillofacial region—an update," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 5, no. 4, pp. 366–380, 2009.

[109] T. Kondo, S. Ong, and K. W. Foong, "Computer-based extraction of the inferior alveolar nerve canal in 3-D space," *Computer Methods and Programs in Biomedicine*, vol. 76, no. 3, pp. 181 – 191, 2004.

[110] S. Rueda, J. A. Gil, R. Pichery, and M. Alcañiz, "Automatic segmentation of jaw tissues in CT using active appearance models and semi-automatic landmarking," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 167–174.

[111] D.-J. Kroon, "Segmentation of the mandibular canal in cone-beam CT data," Ph.D. dissertation, University of Twente, Netherlands, Dec. 2011.

[112] D. Kainmueller, H. Lamecker, H. Seim, M. Zinser, and S. Zachow, "Automatic extraction of mandibular nerve and bone from cone-beam ct data," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 76–83.

[113] B. Moris, L. Claesen, Yi Sun, and C. Politis, "Automated tracking of the mandibular canal in CBCT images using matching and multiple hypotheses methods," in *2012 Fourth International Conference on Communications and Electronics (ICCE)*, Aug 2012, pp. 327–332.

[114] F. Abdolali and R. A. Zoroofi, "Mandibular canal segmentation using 3d active appearance models and shape context registration," in *2014 21th Iranian Conference on Biomedical Engineering (ICBME)*, Nov 2014, pp. 7–11.

[115] S. Vinayahalingam, T. Xi, S. Bergé, T. Maal, and G. de Jong, "Automated detection of third molars and mandibular nerve by deep learning," *Scientific Reports*, vol. 9, no. 1, p. 9007, Jun 2019.

[116] J.-Y. Cha, H.-I. Yoon, I.-S. Yeo, K.-H. Huh, and J.-S. Han, "Panoptic segmentation on panoramic radiographs: Deep learning-based segmentation of various structures including maxillary sinus and mandibular canal," *Journal of Clinical Medicine*, vol. 10, no. 12, p. 2577, 2021.

[117] G. H. Kwak, E.-J. Kwak, J. M. Song, H. R. Park, Y.-H. Jung, B.-H. Cho, P. Hui, and J. J. Hwang, "Automatic mandibular canal detection using a deep convolutional neural network," *Scientific Reports*, vol. 10, no. 1, p. 5711, Mar 2020.

[118] S. Kurt Bayrakdar, K. Orhan, I. S. Bayrakdar, E. Bilgir, M. Ezhov, M. Gusarev, and E. Shumilov, "A deep learning approach for dental implant planning in cone-beam computed tomography images," *BMC Medical Imaging*, vol. 21, no. 1, p. 86, May 2021.

[119] F. Milletari, N. Navab, and S. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, Oct 2016, pp. 565–571.

[120] R. J. Baer, E. E. Rogers, J. C. Partridge, J. G. Anderson, M. Morris, M. Kuppermann, L. S. Franck, L. Rand, and L. L. Jelliffe-Pawlowski, "Population-based risks of mortality and preterm morbidity by gestational age and birth weight," *Journal of Perinatology*, vol. 36, no. 11, pp. 1008–1013, 2016.

[121] J. D. Horbar, E. M. Edwards, L. T. Greenberg, K. A. Morrow, R. F. Soll, M. E. Buus-Frank, and J. S. Buzas, "Variation in performance of neonatal intensive care units in the United States," *JAMA Pediatrics*, vol. 171, no. 3, pp. e164 396–e164 396, 03 2017.

[122] United Nations Inter-agency Group for Child Mortality Estimation (UNIGME), "Levels & trends in child mortality: Report 2018, estimates developed by the United Nations inter-agency group for child mortality estimation," *United Nations Children's Fund, New York*, 2018.

[123] L. M. Davidson and S. K. Berkelhamer, "Bronchopulmonary dysplasia: Chronic lung disease of infancy and long-term pulmonary outcomes," *Journal of Clinical Medicine*, vol. 6, no. 1, 2017.

[124] A. M. Thompson and M. J. Bizzarro, "Necrotizing enterocolitis in newborns," *Drugs*, vol. 68, no. 9, pp. 1227–1238, Jun 2008.

[125] A. Hellström, L. E. Smith, and O. Dammann, "Retinopathy of prematurity," *The Lancet*, vol. 382, no. 9902, pp. 1445–1457, Oct 2013.

[126] V. Apgar, "A proposal for a new method of evaluation of the newborn infant." *Anesthesia & Analgesia*, vol. 32, no. 4, 1953.

[127] D. K. Richardson, J. E. Gray, M. C. McCormick, K. Workman, and D. A. Goldmann, "Score for neonatal acute physiology: A physiologic severity index for neonatal intensive care," *Pediatrics*, vol. 91, no. 3, pp. 617–623, 1993.

[128] D. K. Richardson, C. S. Phibbs, J. E. Gray, M. C. McCormick, K. Workman-Daniels, and D. A. Goldmann, "Birth weight and illness severity: Independent predictors of neonatal mortality," *Pediatrics*, vol. 91, no. 5, pp. 969–975, 1993.

[129] D. K. Richardson, J. D. Corcoran, G. J. Escobar, and S. K. Lee, "SNAP-II and SNAPPE-II: Simplified newborn illness severity and mortality risk scores," *The Journal of Pediatrics*, vol. 138, no. 1, pp. 92 – 100, 2001.

[130] B. Zernikow, K. Holtmannspoetter, E. Michel, W. Pielemeier, F. Hornschuh, A. Westermann, and K. H. Hennecke, "Artificial neural network for risk assessment in preterm neonates," *Archives of Disease in Childhood - Fetal and Neonatal Edition*, vol. 79, no. 2, pp. F129–F134, 1998.

[131] O. Dammann, B. Shah, M. Naples, F. Bednarek, J. Zupancic, E. N. Allred, A. Leviton, and E. S. Investigators, "Interinstitutional variation in prediction of death by SNAP-II and SNAPPE-II among extremely preterm infants," *Pediatrics*, vol. 124, no. 5, pp. e1001–e1006, Nov 2009.

[132] E. Baraldi and M. Filippone, "Chronic lung disease after premature birth," *New England Journal of Medicine*, vol. 357, no. 19, pp. 1946–1955, 2007.

[133] M. A. Isani, P. T. Delaplain, A. Grishin, and H. R. Ford, "Evolving understanding of neonatal necrotizing enterocolitis," *Current Opinion in Pediatrics*, vol. 30, no. 3, 2018.

[134] M. Podda, D. Bacciu, A. Micheli, R. Bellù, G. Placidi, and L. Gagliardi, "A machine learning approach to estimating preterm infants survival: development of the preterm infants survival assessment (PISA) predictor," *Scientific Reports*, vol. 8, no. 1, p. 13743, 2018.

[135] J. Gilchrist, C. M. Ennett, M. Frize, and E. Bariciak, "Neonatal mortality prediction using real-time medical measurements," in *2011 IEEE International Symposium on Medical Measurements and Applications*, 2011, pp. 65–70.

[136] W. Wajs, P. Stoch, and P. Kruczek, "Radial basis networks and logistic regression method for prediction of broncho pulmonary dysplasia," in *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, 2007, pp. 551–555.

[137] M. Ochab and W. Wajs, "Bronchopulmonary dysplasia prediction using support vector machine and LIBSVM," in *2014 Federated Conference on Computer Science and Information Systems*, 2014, pp. 201–208.

[138] S. Saria, A. K. Rajani, J. Gould, D. Koller, and A. A. Penn, "Integration of early physiological responses predicts later illness severity in preterm infants," *Science Translational Medicine*, vol. 2, no. 48, pp. 48ra65–48ra65, 2010.

[139] O. Rinta-Koski, S. Särkkä, J. Hollmén, M. Leskinen, K. Rantakari, and S. Andersson, "Prediction of major complications affecting very low birth weight infants," in *2017 IEEE Life Sciences Conference (LSC)*, Dec 2017, pp. 186–189.

[140] O.-P. Rinta-Koski, S. Särkkä, J. Hollmén, M. Leskinen, and S. Andersson, "Gaussian process classification for prediction of in-hospital mortality among preterm infants," *Neurocomputing*, vol. 298, pp. 134 – 141, 2018.

[141] O.-P. Rinta-Koski, "Machine learning in neonatal intensive care," Doctoral thesis, School of Science, 2018.

[142] MATLAB, *version 9.6.0 (R2019a)*. Natick, Massachusetts: The MathWorks Inc., 2019.

[143] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari, "GPstuff: Bayesian modeling with Gaussian processes," *Journal of Machine Learning Research*, vol. 14, no. Apr, pp. 1175–1179, 2013.

[144] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[145] M. C. McCormick, "The contribution of low birth weight to infant mortality and childhood morbidity," *The New England Journal of Medicine*, vol. 312, no. 2, pp. 82–90, Jan 10 1985.

[146] K. Blomqvist, S. Kaski, and M. Heinonen, "Deep convolutional gaussian processes," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 582–597.

[147] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[148] A. Koivu, M. Sairanen, A. Airola, and T. Pahikkala, "Synthetic minority oversampling of vital statistics data with generative adversarial networks," *Journal of the American Medical Informatics Association*, vol. 27, no. 11, pp. 1667–1674, 09 2020.

[149] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[150] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. Blaschko, and E. Belilovsky, "Scattering networks for hybrid representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2208–2221, 2019.

[151] J. Zarka, F. Guth, and S. Mallat, "Separation and concentration in deep networks," in *ICLR 2021-9th International Conference on Learning Representations*, 2021.

Aalto-DT 161/2022

ISBN 978-952-64-1008-1 (printed)
ISBN 978-952-64-1009-8 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

**DOCTORAL
THESES**