Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 2352

Equivariant Neural Networks for Biomedical Image Analysis

KARL BENGTSSON BERNANDER

ACTA UNIVERSITATIS
UPSALIENSIS
2024

ISSN 1651-6214
ISBN 978-91-513-2004-5
urn:nbn:se:uu:diva-519611

UPPSALA
UNIVERSITET

Dissertation presented at Uppsala University to be publicly examined in
Ångströmlaboratoriet, 101121, Sonja Lyttkens, Lägerhyddsvägen 1, Uppsala, Friday, 1 March
2024 at 13:15 for the degree of Doctor of Philosophy. The examination will be conducted
in English. Faculty examiner: Professor Michal Kozubek.

Abstract

Bengtsson Bernander, K. 2024. Equivariant Neural Networks for Biomedical Image Analysis. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 2352. 82 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-513-2004-5.

While artificial intelligence and deep learning have revolutionized many fields in the last decade, one of the key drivers has been access to data. This is especially true in biomedical image analysis where expert annotated data is hard to come by. The combination of Convolutional Neural Networks (CNNs) with data augmentation has proven successful in increasing the amount of training data at the cost of overfitting. In this thesis, equivariant neural networks have been used to extend the equivariant properties of CNNs to more transformations than translations. The networks have been trained and evaluated on biomedical image datasets, including bright-field microscopy images of cytological samples indicating oral cancer, and transmission electron microscopy images of virus samples. By designing the networks to be equivariant to e.g. rotations, it is shown that the need for data augmentation is reduced, that less overfitting occurs, and that convergence during training is faster. Furthermore, equivariant neural networks are more data efficient than CNNs, as demonstrated by scaling laws. These benefits are not present in all problem settings and which benefits will occur is somewhat unpredictable. We have identified that the results to some extent depend on architectures, hyperparameters and datasets. Further research may broaden the performed studies to explain how the results occur with new theory.

Karl Bengtsson Bernander, Department of Information Technology, Computerized Image Analysis and Human-Computer Interaction, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.

ISSN 1651-6214
ISBN 978-91-513-2004-5
URN urn:nbn:se:uu:diva-519611 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-519611)

Dedicated to you who struggle. There is always hope.

List of papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I **K.B. Bernander**, J. Lindblad, R. Strand, and I. Nyström. "Replacing data augmentation with rotation-equivariant CNNs in image-based classification of oral cancer". In: *Iberoamerican Congress on Pattern Recognition (CIARP 2021).* Springer, 2021

II **K.B. Bernander**, J. Lindblad, R. Strand, and I. Nyström. "Rotation-Equivariant Semantic Instance Segmentation". In: *Medical Image Understanding and Analysis (MIUA 2022).* Springer, 2022

III **K.B. Bernander**, I. Sintorn, R. Strand, and I. Nyström. "Classification of Viruses in Transmission Electron Microscopy Images using Equivariant Neural Networks". Submitted for journal publication

IV K. Bylander, I. Nyström and **K.B. Bernander**. Equivariant Neural Networks for Biomedical Images Improves Data Efficiency. Submitted for publication

Reprints were made with permission from the publishers.

Summary of Contributions

The Roman numerals correspond to the numbers in the list of papers.

I I designed the experiments and implemented the methods. I wrote the paper jointly with Joakim Lindblad, Robin Strand and Ingela Nyström.

II Robin Strand, Joakim Lindblad, Ingela Nyström and I jointly developed the main idea. I implemented the method, designed the experiments and wrote the paper. The co-authors reviewed the paper.

III I conducted the experiments, analyzed the results and wrote the manuscript. All authors conceived the experiments and reviewed the manuscript.

IV I. Nyström and I proposed the project and provided supervision of the work of K. Bylander. I designed most of the experiments together with K. Bylander. K. Bylander conducted all of the experiments. K. Bylander provided initial analysis of the results, which I extended. I wrote the manuscript. All authors reviewed and contributed to the manuscript.

Related Work

In addition to the papers included in this thesis, the author has also contributed to the following works:

Papers

R1 **K.B. Bernander***, K. Gustavsson*, B. Selig, I. Sintorn, C. Luengo Hendriks. "Improving the stochastic watershed". In: *Pattern Recognition Letters*. Elsevier, 2013.

R2 O. Sunneborn Gudnadottir, D. Gedon, C. Desmarais, **K.B. Bernander**, R. Sainudiin, R. Gonzalez Suarez. "Distributed training and scalability for the particle clustering method UCluster". In: *25th International Conference on Computing in High-Energy and Nuclear Physics (CHEP 2021)*. EPJ Web of Conferences, 2021.

Extended Abstract

R3 T. Asplund, **K.B. Bernander**, E. Breznik. "CNNs on Graphs: A New Pooling Approach and Similarities to Mathematical Morphology". In: *Swedish Symposium on Deep Learning (SSDL 2019)*. Swedish Society for Automated Image Analysis, 2019.

Thesis

R4 **K.B. Bernander**. "Improving training of deep learning for biomedical image analysis and computational physics". In: *degree of Licentiate of Philosophy in Computerised Image Processing*. Uppsala University, 2021.

* Authors contributed equally.

Code

C1 **K.B. Bernander (2021).** Code used in Paper I: Replacing data augmentation with rotation-equivariant CNNs in image-based classification of oral cancer. GNU General Public License Version 3. Github. https://github.com/kbbernander/rot-equivariant-cnn-oral-cancer

C2 **K.B. Bernander (2022).** Code used in Paper II: Rotation-Equivariant Semantic Instance Segmentation. GNU General Public License Version 3. Github. https://github.com/kbbernander/eq-ins-seg

C3 **K.B. Bernander (2023).** Code used in Paper III: Classification of Viruses in Transmission Electron Microscopy Images using Equivariant Neural Networks. GNU General Public License Version 3. Github. https://github.com/kbbernander/TEM-equivariance

Contents

Acknowledgements

Here, I would like to acknowledge those around me who in various ways played important parts during my PhD project.

First, I thank my mother, who was always there and provided support in so many ways, and her husband Herman. My brother, who I first saw in Oslo so many years ago. My late father, who I think about so much. I thank my paternal grandfather Arne and my uncle Pontus, none of whom I ever got to meet. They left infinite voids on both sides of the family, and their fates inspired me to do the things I did. I thank my grandmothers Gun and Viola who both passed away during the course of the project along with the husband of my paternal grandmother, Rune. I thank my maternal grandparent Arne who always makes time for me, and the late aunt of my father, Gunhild. I thank my aunt Inger and my cousins Viktor and Markus - we had some interesting Christmas celebrations together. Thanks to Lina, Moa and Maja Westbergh for being part of a family. Likewise, thanks to Helena and Per Högstorp. I also thank my family dogs and rabbits: Stella, Linus, Sally, Mindy and Mysan.

Thanks you to Evelina Rosenqvist, Sebastian Shashahani, Ilya Prokofiev, Katarina Blomstrand along with the Real Vetis Smålands nation pub quiz team. The student associations of Uppsala have changed how I think so many times. Many thanks to Tilda Jacobson Holmström and Josef Alhomsi of the Poetry Pub of Uppsala, to Hanna Déak and others from Verdandi, and to Lina Lantz, Erik Gjessing, Filippa Eklund, Alexander Albo, Jocke Björnfalk, Robin Lagelius and countless others from Demiratus. For long badminton sessions, thanks to Kalle Johansson, Linnea Lindahl and Filip Singh.

In parallel with my time in the project, I have worked with mental health in several ways. Even though there is a long way to go to combat stigma and increase knowledge and available help, progress has been made. I thank especially the organisations SPES, Mind and Suicidezero for their efforts in a tough environment. Thanks to Johanna Edström, Frida Runhagen, Frida Winnberg, Maria Ahlin, Marcus Eriksson, Björn Eklund, Kaisa Nordquist, Elisabeth Lindström, Kerstin Ahlgren, Charlotte Skjöldebrand, Göteborgs nation, Malin Tyberg and all the coauthors of the book "Att förlora en far : sönernas berättelser" for your efforts and support. Thank you to the all people I met in the support groups of SPES and Universitetskyrkan, although I cannot name all of you.

Thanks to Arvid Smeedsaas and Anders Bankefors for the gaming sessions. Also thanks to David Höjenberg, Sasch Bengtsson, Hussam Bashir, Henrik

Bådagård, Mikael Sunde, Maria Sinnikova, Emma Hagforsen, Alva Rosengren Blom, Annelie Bolldén, Eugenia Kalliontzi, Emily Thorselius, Nora Hansson, Thérèce Bonnier and Patric Bonnier. I thank Ann-Christine Troberg for long interesting walks in Stockholm. From the Engineering Physics programme, I thank Kenneth Gustavsson, Ola Söderström, Adam Saxén, Emil Jansson, Sofia Berglund, Mikaela Åhlén, Karl-Johan Åhs, Jennifer Leijon and Cheewin Pisanupoj. Discussing and sharing with previous colleagues proved invaluable to me. Thanks to, among others, Moysis Tsamsakizoglou, Monica Alaniz, Fredrik Persson, Daniel Skaborn, Christer Andersson, Francesco Vallegra, Per Brolin, Robert Lindegren, Nickolay Ivchenko, Stefan Strålsjö, Mattias Örth, Mattias Lindgren, Jan Persson, Martin Lembke, Magnus Bergström, Helen Ghattas, Viktória Szilva, Ananya Muddukrishna, Bettina Selig, Vladimir Curic and Cris Luengo Hendriks.

Throughout the project I worked alongside many who contributed to the thesis project. The most important ones have been my supervisors: Nataša Sladoje, Joakim Lindblad, Carolina Wählby, Ingela Nyström and Robin Strand. I would also like to thank Ida-Maria Sintorn, Eva Breznik, Teo Asplund, Olga Sunneborn Gudnadottir, Daniel Gedon, Colin Desmarais, Raazesh Sainudiin and Rebeca Gonzales Suarez for the publications we accomplished together. I thank the student advisors and program coordinators of the IT department for their important work and for teaching me so much: Liselott Dominicus, Olga Kaj, Anders Kronberg, Olle Gällmo, Anna Eckerdal and Matteo Magnani. I also thank the student advisors at the faculty and university, including Moses Högback, Emma Svensson and Therese Wallin. I thank Karl Bylander, along with all other students I teached and advised along the way, not least in the Data Science program. Thank you to Filip Malmberg, Diana Malakhova, Nadezhda Koriakina, Erik Hallström, Swarnadip Chatterjee, Andrea Behanova, Eduard Chelebian and Ankit Gupta for the intense teaching we did together. Thank you to Mats Lind, Anders Persson, Diane Golay, Anton Axelsson, Anders Hast, Anders Arweström Jansson, Magdalena Stadin, Li Hedenmalm, Sandor Albrecht, Pontus de Laval, Anna Wigren, Calle Andersson and Justin Pearson for interesting and supporting conversations throughout my time at the IT department. Thanks to Marco Chiodaroli, Michael Papenbrock, Ragnar Seton and Mauricio Perez for the efforts in the Saco-S union club at Ångström.

I thank the following people at CBA for all kinds of reasons: Raphaela Heil, Elisabeth Wetzer, Håkan Wieslander, Johan Öfverstedt, Nicolas Pielawski, Leslie Solorzano, Gabriele Partel, Axel Andersson, Can Deniz Bezek, Marc Fraile Fabrega, Anindya Gupta, Love Nordling, Ewert Bengtsson and Gunilla Borgefors. I also thank those I met while abroad, including Emma Robinson, Taco Cohen, Erik Bekkers, Gabriele Cesa, David Romero, Michael Yeung, Adrian Krenzer, Alexandra Walter and Anna Breger.

Finally, I thank the following people from WASP for sharing a very interesting journey on the new research program: Georg Bökman, Jimmy Aron-

sson, Pavlo Melnyk, Daniel Ahlsén, Inês Lourenço, Jonathan Styrud, Lars Svensson, Martin Danelljan, Lucas Brynte, Niklas Gunnarsson, Gizem Çaylak, Martin Andersson, Amanda Olmin, Oscar Carlsson, Axel Berg, Yaroslava Lochman, Jonas Nordlöf, Niklas Åkerblom, Rasmus Kjær Hœier, Magnus Oskarsson, Michael Felsberg as well as the other students, teachers and researchers who accompanied me along the way.

The project was funded by WASP (Wallenberg AI, Autonomous Systems and Software Program) which is funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Chalmers University of Technology partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Sammanfattning på svenska

Artificiell intelligens (AI), vilket oftast syftar på maskininlärning, har de senaste åren slagit igenom inom naturvetenskaplig forskning och i samhället i stort. Inte minst inom biomedicinsk bildanalys har metoderna varit framgångsrika, vilket banar vägen för snabbare och mer träffsäker diagnostik. Detta kan bidra till tidigare upptäckt av elakartad cancer och ökad chans till överlevnad bland patienter. En av förutsättningarna för hög träffsäkerhet är stora mängder uppmärkt träningsdata. Inom biomedicinsk bildanalys är detta ofta svårt att uppnå. Tillgänglig expertkunskap inom exempelvis cellbiologiska förändringar som tyder på elakartad cancer är sällsynt.

För att öka mängden träningsdata används därför ofta dataförstärkning, där kunskapen om att bilder med små förändringar, till exempel roterade eller spegelvända kopior, indikerar samma typ av objekt eller tillstånd som originalbilden. Denna metod riskerar dock att leda till överanpassning, alltså att nätverken lär sig att känna igen detaljer i träningsdata till för hög grad. Detta leder till att okända exempel oftare felklassificeras.

För att komma till rätta med dessa problem kan ekvivarianta neurala nätverk användas. Faltande neurala nätverk är ekvivarianta mot translationer, det vill säga lodräta och vågräta förflyttningar, av objekt i bilder. Efter ett faltningslager är resultatet detsamma som om förflyttningen skett på motsvarande sätt efter faltningsoperationen. Detta innebär att translationen och faltningsoperationen kommuterar. Ekvivarianta neurala nätverk utvidgar denna egenskap till en större mängd symmetrigrupper, exempelvis rotationer och speglingar. Detta minskar antalet parametrar och behovet av dataförstärkning.

I avhandlingen har ekvivarianta nätverk designats för populära nätverk för klassificering och segmentering som VGG16 och U-net. De har tränats och utvärderats på bland annat ljusmikroskopibilder av celler som indikerar cancer i munhålan, och elektronmikroskopibilder på olika typer av virus. Fördelarna med ekvivarianta nätverk är att dataförstärkning kan minskas, att mindre överanpassning uppträder och att nätverken konvergerar snabbare under träningsfasen. Utöver detta är de mer effektiva på att lära sig från träningsdata, vilket illustreras med brantare lutningar i skalningsdiagram. Dock är det oklart vilka fördelar som uppträder i vilket sammanhang, och vidare forskning bör fokusera på att studera fler arkitekturer, dataset och hyperparametrar. Utöver detta behövs mer teoretisk forskning för att förklara resultaten.

I ett vidare sammanhang kommer AI sannolikt att utvecklas snabbt i takt med ökad tillgång till beräkningskapacitet och data. Vårt framtida samhälle bör fokusera på att lära oss systemens förmågor, sprida kunskapen, föra etiska diskussioner och instifta globala lagar.

Artikel I visar hur en VGG16-klassificerare kan modifieras till att bli ekvivariant mot p4-gruppen av translationer och multiplar av rotationer på 90 grader. Metoden tränas och utvärderas på ett cytologiskt dataset bestående av bilder utifrån ljusmikroskopi. Cellerna kommer från patienter som antingen är friska eller har cancer i munhålan. Den ekvivarianta klassificeraren minskar överanpassning och behovet av dataförstärkning i jämförelse med ett faltande nätverk.

Artikel II visar hur ett nätverk för semantisk instanssegmentering, U-Net med en urskiljande förlustfunktion, kan göras ekvivariant mot p4-gruppen. De ekvivarianta egenskaperna bevisas teoretiskt och illustreras. Metoden tränas och utvärderas på ett syntetiskt dataset bestående av bilder på pinnar och ett riktigt dataset med bilder på celler med olika former och ursprung. Resultaten tyder på att träffsäkerheten är liknande som för ett faltande nätverk, men att det ekvivarianta nätverket konvergerar snabbare under träningsfasen.

Artikel III visar hur en VGG16-klassificerare kan modifieras till att bli ekvivariant mot p4-gruppen bestående av translationer och multiplar av rotationer på 90 grader, på motsvarande sätt som i Artikel I. Metoden tränas och utvärderas på ett dataset bestående av bilder på olika typer av virus tagna med transmissionselektronmikroskopi. Resultaten visar att dataförstärkning med motsvarande transformationer kan minskas och att konvergenshastigheten är högre under träningsfasen. Samtidigt visar ett skalningsdiagram att det ekvivarianta nätverket lär sig mer effektivt från träningsdata i jämförelse med ett faltande nätverk.

Artikel IV visar hur både VGG16-klassificerare och ett mindre nätverk kan modifieras till att bli ekvivarianta mot flera olika symmetrigrupper. Nätverket tränas och utvärderas på samma dataset som i Artikel III. Detta dataset består av bilder på olika typer av virus tagna med transmissionselektronmikroskopi. Satsnormalisering används också under träningsfasen. Resultaten visar att med betydligt längre träningstider minskar överanpassning för de ekvivarianta nätverken i jämförelse med ett faltande nätverk. Dessutom visar ett skalningsdiagram att en ekvivariant version av VGG16 med D4-gruppen lär sig mer effektivt från träningsdata i jämförelse med ett faltande nätverk.

1. Introduction

At the time of writing this thesis in late 2023, the field of artificial intelligence (AI) is making headlines regularly and has found its way into public discourse. Be it generative models such as text-to-image models or chatbots, automated analysis of medical images, or visions of a future society (dystopian or utopian), some things seems clear. These technologies will continue to progress, their impact on society will be significant, and it is difficult tell how. How we use the technologies will partly be up to the reader. How would you like society to function in terms of e.g. surveillance, task automation or ownership of data? The more you know about how these technologies work, and the more aware you are of your own values, the more informed your opinions will be.

I first came into contact with the field of computer assisted image analysis in 2012 when taking an introductory course. Back then, the focus was on what we today call classical methods: signal processing, statistics and conventional programming [39]. Solving a problem typically involved manual design of an algorithm divided into steps. We were 45 students in this course instance. The number of published articles at the largest related scientific conference, CVPR (Conference on Computer Vision and Pattern Recognition) was 1933. The best top-5 test error on the ImageNet challenge was 15.3 % [24]. This challenge involved letting the program analyze one image at a time, outputting the five most likely objects contained in the image, which were then compared to known labels to calculate the error rate. The winner was AlexNet [55], which used a very different approach than the current standards.

At the time, a number of technologies were mature enough to start benefitting from each other, including massive datasets of labelled images and computational hardware capable of parallel processing. However, the techniques behind AlexNet had been developed in steps since the 1950s [90]. A superset of these technologies is often called Artificial Intelligence (AI), which is a broad term that encompasses the hypothetical development of sentient or superintelligent machines, but which does not reflect their current capabilities. Machine learning is more precise, aiming to let programs find algorithmic solutions on their own. Artificial Neural Networks (ANN), which are loosely inspired by the neurons of human brains, form the backbone of the current best performing systems. They are often referred to simply as neural networks. Here, signal-processing neurons are connected to each other in layers. If there are multiple hidden layers between the inputs and outputs of the system, the technology is called deep learning [40]. This hierarchy of technologies is illustrated in Figure 1.1.

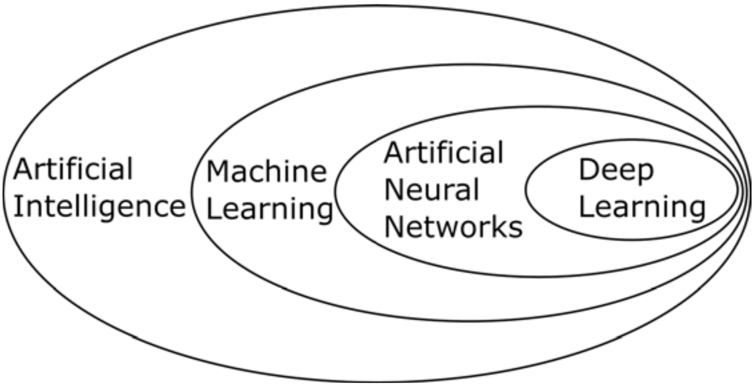


Figure 1.1. The hierarchy of technologies related to deep learning.

Developing deep learning solutions means setting up a specific ANN model, defining the inputs (such as images), defining the target outputs (such as a cancer diagnosis), and starting the training process. During training, which can last for days or weeks, the weights of the connections between the layers are adjusted in an optimization process which gradually brings the outputs of the network close to the target outputs (also referred to as labels). Finally, the trained model can be applied to new data. The same methodology has proven successful in numerous other fields where digital outputs need to be predicted from digital inputs.

At the time of writing this thesis, I am teaching in the same course that I enrolled in myself almost twelve years ago. The number of students is 125, an increase of 178 %. The number of CVPR submissions has increased by 358 % to 9155 in the year 2022. The top-5 test error on the ImageNet challenge has been reduced from 15.3 % to 0.98 %. These numbers are certainly greatly affected by hype and marketing. But the ImageNet numbers do not lie - the results are reproducible and similar results have been seen in many other fields.

1.1 Research background

In spite of the success of deep learning in recent years, the methodology has a number of drawbacks. As already mentioned, large amounts of annotated data is required. Expert knowledge is needed for labelling e.g. malignant or healthy cells in microscopy data. Since these domain experts can be hard to find, and their skills are also in demand for clinical work, biomedical datasets are often in short supply [31].

Prior knowledge about the problem can be exploited to increase the amount of training data. Since the rotation of the sample under the microscope is irrelevant for diagnosis, images can be rotated while keeping the labels intact. However, larger networks with more parameters are needed to learn each rotation. This can lead to overfitting, where the methods are adapted to learning the training examples, but struggle to generalize to unseen data when deployed.

Another issue is the amount of computations needed when training. Training a state of the art model can take days or weeks, increasing in time with the amount of data and the size of the models. It is often not feasible to fine-tune the models for optimal performance, as the amount of computations and time requirements are too large [78].

1.2 Thesis aims

The aims of this thesis are to mitigate the issues posed by limited training data (Aim 1), overfitting (Aim 2) and the costs associated with the training process (Aim 3) in modern biomedical image analysis by using equivariant neural networks in combination with empirical deep learning and predictable scaling.

1.3 Thesis outline

The rest of this thesis is organized as follows.

Chapter 2 aims to provide an overview of AI. This can be seen as positioning the contributions of the thesis in a wider context. This chapter can safely be skipped by a reader knowledgeable in deep learning.

Chapter 3 provides the necessary knowledge about common problem settings in biomedical image analysis. It also explains how Papers I, II and III contribute to solving three related problems by deep learning.

Chapter 4 introduces geometric deep learning. It explains how equivariant neural networks incorporate geometric information by design to mitigate issues commonly seen in convolutional neural networks. Then, for Papers I, II and III, it explains how to design equivariant neural networks for the problems introduced in the previous chapter.

Chapter 5 introduces empirical deep learning and predictable scaling for mitigating training costs. Furthermore, the experimental results from all the papers are presented. Additionally for Papers III and IV, it shows how equivariant neural networks can be used to reduce the costs of training for biomedical image analysis using scaling laws.

Chapter 6 summarizes the findings of the papers, revisits the aims of the thesis, proposes the way forward for further research, and finishes with the author’s outlook on the technology.

2. Background

This chapter provides an overview of deep learning and other types of machine learning, starting with the basics of learning paradigms and models. Then it introduces the current state of applications. After that, possible societal impacts are examined and the important drivers of leading AI technology are analyzed. The chapter ends with a survey of ethical theory.

2.1 Supervised learning

In supervised learning, labels are associated to each input. An example is K nearest neighbors [109] or neural networks, described more in detail below.

2.1.1 Neural networks

The fundamentals of modern deep learning is a multilayer neural network with inputs on one side and outputs on the other [40]. The inputs can take varying forms, such as age and height of a person, or images which exhibit spatial relationships between pixels. When a new sample is fed through the network each neuron performs calculations on its inputs. Typically a nonlinear sum is calculated by using activation functions. The results are then passed along to their outputs in the next layer. These intermediate layers are called hidden layers. The last layer is the output layer, which, like the inputs, can take many forms. Examples include images (for reconstruction using autoencoders) or concepts such as a disease diagnosis.

Each neuron has weights associated to its inputs. During the training process, which fine-tunes the output of the network to match the labels, the weights are updated by backpropagation. First, model outputs are calculated in a forward pass. These are then compared with the labels in a loss function, which is then derived with respects to the weights layer by layer using the chain rule. The weights are then updated by mathematical optimization, most commonly gradient descent. This is known as a backward pass. The forward and backward pass for all training data is called an epoch, and this cycle is repeated until the loss has decreased sufficiently.

The training process can be complicated, and different combinations of hyperparameters often need to be tried to reach high accuracy. Important hyperparameters and settings include the learning rate of the optimizer, regularization and batch normalization. Transfer learning is a technique that has

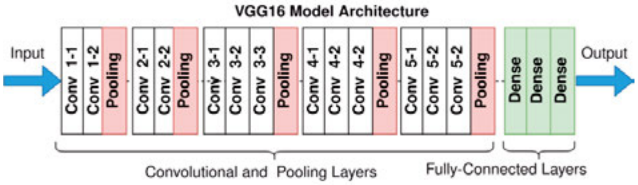


Figure 2.1. The VGG16 convolutional neural network architecture. The white blocks are combined convolutional and ReLu layers, the red blocks are max pooling layers, and the green blocks are combined fully connected and ReLu layers. Image sourced from [42].

proven useful to increase accuracy and reduce the training time. This involves pretraining a model on a large standardized dataset and then retraining, or finetuning, the final layers on the new data.

2.1.2 Convolutional neural networks

Common network architectures in image analysis are based on Convolutional Neural Networks (CNNs), which can exploit the spatial relationships between pixels efficiently. Here, each neuron has a restricted 2D field of view of the outputs of the previous layer. The field of view can have a window size of e.g. 3 by 3 pixels, with a parameter associated to each input, constituting a kernel k. Simultaneously, all the neurons in a channel share the kernel, reducing the number of parameters significantly. During a forward pass, this corresponds to a convolution C of the inputs f with the kernel. This can be expressed mathematically in the following way:

$$z(x,y) = k * f(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} k(i,j) f(x-i, y-j) \tag{2.1}$$

where z is the output, x and y are image coordinates, and i and j are kernel coordinates in the window of size $[-a,a]$ by $[-b,b]$.

An example is shown in Figure 2.1, which visualizes the VGG16 architecture [96]. Each layer has multiple channels. Activation functions are used to amplify or attenuate different outputs. Max pooling layers reduce the dimensionality of the inputs in steps. After the final pooling layers, only one dimensional points remain, which are fully connected in the later layers. The final output layer assigns the most likely classes from the points using a softmax activation function.

After training, it has been observed that CNNs tend to learn a hierarchy of features. Earlier maps tend to learn simple shapes like lines or edges. Later layers tend to learn more complex structures, like parts of facial structures. This mechanism is similar to how visual perception functions in the visual cortex of the human brain. This could also explain how the models learn to distinguish e.g. cats from dogs. Dogs can look very different from each other but share fundamental characteristics which the models learn to identify and separate from the characteristics of cats.

2.1.3 Transformers

In the last years, transformers [105] have become increasingly competitive in relation to CNNs. They first made a strong impact in natural language processing. Previous state of the art models were mainly based on CNNs and Recurrent Neural Networks (RNN) [93]. RNNs process sequential data efficiently through the use of bidirectionality and long short-term memory units.

Transformers are instead based on attention mechanisms. When applied to computer vision tasks, the image is divided into fixed-size patches which are converted to a vector along with their positions. Attention mechanisms, which loosely mimic cognitive attention in human brains, then amplify the important relations among the patches and attenuate less important relations. Finally, classification heads proceeds similarly as in the end layers of CNN classifiers.

2.2 Reinforcement learning

Another important machine learning paradigm is reinforcement learning [100]. Here, a reward function is used to steer the actions of an agent. If the goal is to teach a bot to play a computer game, the reward function is designed to increase when the agent takes actions that wins the game, and decrease when the agent takes actions that loses the game. Algorithms are designed to balance between exploration, where new actions are tested by the agent, and exploitation, which reinforces current knowledge. Besides games [95], reinforcement learning is often used in e.g. robotics [44] and energy management [74].

2.3 Unsupervised learning

In contrast to supervised learning and reinforcement learning, unsupervised learning uses data with no explicit labels or reward function. Instead, some kind of underlying structure to the data is presumed, and algorithms are designed to find this structure automatically. Examples of unsupervised methodologies include clustering methods such as Kmeans [40]. Applications of unsupervised learning include image segmentation [8] and experimental particle physics [98].

It has been suggested that unsupervised learning is a promising avenue for further gains in machine learning [48]. Humans rarely rely on supervisory signals for many cognitive tasks such as perception. This is illustrated by

Table 2.1. *Overview of different machine learning paradigms and models.*

Type of learning	**Supervisory signal**	**Example models**
Supervised	Inputs and Labels	Discriminative
Reinforcement	Reward Function	Agent-based
Unsupervised	Inputs	Generative, SNN
Semisupervised	Inputs and small # Labels	Discriminative

the Hebbian principle [45], where neurons that fire together, wire together. Learning occurs irrespectively of errors.

Self-supervised learning is a similar paradigm that uses the statistics of the data to predict hidden parts of the input. For example, having only seen similar sentences without any labels, given an uncomplete sentence the networks can predict the final words.

2.4 Semisupervised learning

Semisupervised learning [19] falls in between supervised and unsupervised learning. Here, unlabelled data is combined with small portions of labelled data, which has proven effective in many instances.

A summary of machine learning paradigms, as well as example models, can be seen in Table 2.1.

2.5 Other machine learning methods

Besides deep learning and neural networks, there are many other machine learning models. One is Support Vector Machines (SVM) and decision trees [69]. SVM uses hyperplanes to separate different clusters of data into classes. Decision trees divide data into two or more branches depending on data features. At the bottom of the tree, different classes are found at the leafs. Bayesian networks use graphical models to infer probable causes of signals [46]. The edges between nodes represent conditional probabilities. This methodology is useful for causal modelling [83], while deep learning can only uncover correlations between data. Still, determination of causality can only come from external experiments, such as do-calculus [101].

2.6 Spiking neural networks

Artificial neural networks are inspired by biological neural networks in the human brain, but there are important differences between the two. In biological brains, the timing and rate of signals in both the input and output neurons are very important for learning. Spiking neural networks (SNNs) are designed to emulate this behavior [77]. Also, ANNs rely on numerical representations of data for information transmission, while SNNs operate on spikes which are of a sparse and binary nature. ANNs are also very power inefficient, as a team of 5 humans consume around 100 W, while the OpenAI Five consumed around 10 MW, around 10 000 times more. More modern systems are even more power hungry. Modern SNNs aim to work around this by being implemented on analog neuromorphic computers. However, learning algorithms such as backpropagation are hard to implement in SNNs due to the non-differentiable nature of spikes and the lack of correspondence to weight transport by feedback. While SNNs show promise, they so far have not delivered the same impressive results as ANNs for standard tasks.

2.7 Generative models

In contrast to discriminative models, which only aim to determine meaningful information from data, generative models are probabilistic and can be used to produce new data of some desired form. This typically means that the user inputs text guiding the desired output.

2.7.1 Image generation

An example is Generative Adversarial Networks (GAN) [41]. Here, a discriminative network and a generative network compete against each other, with the generator creating images that are as similar to the training data as possible, while the discriminator tries to distinguish between the two sets.

Recently, generative models like Stable Diffusion, Midjourney and Dall-E have been released to the public, being able to create realistic images of great variety. These are powered by diffusion models [87], which first learn the process of creating gaussian noise from the training set. After the training is completed, new images are generated by reversing the process, sampling new images starting from gaussian noise. Text is embedded in the same latent representation as the images, making it possible to guide the denoising process to the desired outputs. An example of a generated image is shown in Figure 2.2.

2.7.2 Text generation

Large language models, which today are mainly powered by transformers, aim to process and generate text. Chatbots like ChatGPT [78] are currently able to

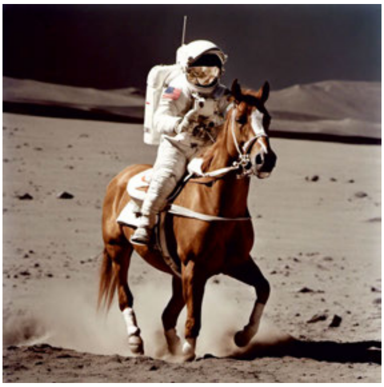


Figure 2.2. An example output from the Stable Diffusion text-to-image model. The prompt was: "a photograph of an astronaut riding a horse". Image sourced from [27].

roughly mimic humans conversations, which can be useful as e.g. sounding boards of new ideas.

2.7.3 Generative models compared to humans

Due to the impressive breakthroughs in e.g. image and text generation, discussions about uniquely human qualities like creativity and empathy have shifted in the last years. It is possible that the perceived qualities of the systems are driving this discussion to a large extent, rather than their actual capabilities [52]. Human creativity and efforts is what the results are ultimately owed to. Also, the systems are prone to hallucinate, i.e. fabricate facts, since what they ultimately are designed to do is function approximation in contrast to a deeper understanding of e.g. the logic of mathematics.

2.8 Artificial General Intelligence

Concerns have been raised about AI becoming sentient or superintelligent which could pose a risk to human civilization. An example of such a system in fiction is shown in Figure 2.3. Current systems lack these capabilities, but such outcomes remain possible. Quantitative assessments of this risk [16] have a similar structure to the Drake equation [29], which estimates the probability of the number of communicative alien civilizations in our galaxy by multiplying seven factors representing independent probabilities.

While some of these factors are known or well approximated, others are very difficult to estimate. The outcome is limited by the most uncertain factors, which makes it hard to draw any meaningful conclusions from the equation. The same criticism holds for current estimates of the emergence of superintelligence, as the uncertainties of the factors are substantial. Theoretical under-

Figure 2.3. HAL 9000 from the movie 2001. Image sourced from [23].

standing of AI or intelligent beings in general is lacking, and the definition of intelligence is controversial [57]. More research is needed.

2.9 Examples of applications

This section introduces some current and potential applications of AI.

2.9.1 Robotics and autonomous systems

One of the key topics of potential impacts to society from AI has been automation. Functions requiring motorics has proven difficult. Robots perform well at specialized, standardized tasks in controlled environments, but are not as proficient in e.g. setting a table [86]. This is probably related to Moravec's paradox from the 1980s: 'for computers it is easy to perform well in intelligence tests or games, but hard to perform well in tasks related to mobility or perception' [70]. Research in self-driving cars has shifted from fully autonomous vehicles in the near future to more controlled environments, where autonomous vehicles have been deployed for a long time [66].

2.9.2 Protein folding

One of the most successful recent applications has been AlphaFold [51], a model for predicting protein structure from chains of amino acids. This technology paves the way for use cases in biotechnology and medicine. An example of the model output can be seen in Figure 2.4.

2.9.3 Creative work

AI is already affecting creative endeavours such as writing and drawing. AI models can be used to quickly propose a number of drafts, which could be selected as is or used as foundations for further work [50]. Those who can adapt to this workflow are likely to benefit from the technology.

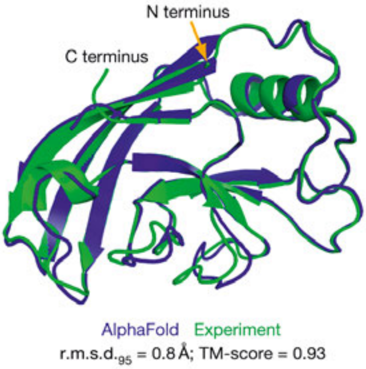


Figure 2.4. An output of the Alphafold model, which shows the CASP14 target T1049 (PDB 6Y4F, blue) compared with the true (experimental) structure (green). Image sourced from [33].

2.9.4 Law and Journalism

Generative models producing text and images are likely to impact fields like law [2]. However, it is unlikely to replace lawyers. Rather, lawyers are likely to change their workflows to incorporate AI as tools. A similar line of reasoning can be used for journalism.

2.9.5 Medicine

Medical doctors such as radiologists are increasingly using AI as assistance during diagnosis to free up time for other tasks [84]. Another area that could benefit from AI is mental health. Mental disorders are the leading cause of years lived with disability [79]. Meanwhile, mental health treatment is severely underfunded worldwide with an average of 2 % of total healthcare budgets. Seeking treatment can be unaffordable, inaccessible or lead to discrimination and ostracization. One example where AI could potentially help is specialized chatbots [15].

2.10 Societal risks

This section summarizes some of AI's most important risks to society.

2.10.1 Misinformation

One big danger of AI is the risk of spread of misinformation, as it becomes much easier to generate and spread false information [53]. Photorealistic images, like deepfakes [76], can be generated of people doing things that have never happened. This could be used for propaganda to e.g. sway opinion in political elections or court processes. There is also a risk of lower faith in voice recordings, photographs and videos as evidence of something actually having happened. Still, the capabilities of algorithms themselves to change our behaviours have been exaggerated and debunked in recent years [97]. They instead tend to reinforce whatever bias is already present.

2.10.2 Emissions

Another risk of AI is the increase of carbon emissions. As dataset and model sizes grow, so do the emissions when developing, training and deploying the systems [81]. More efficient ways of computation are needed if system capabilities are to progress while reducing emissions.

2.10.3 Bias and Hegemony

AI systems risk reinforcing biases and stereotypes, as their outputs are a reflection of their training data [71]. Examples include recommendation algorithms that showed more open job positions to men than women. This outcome was based on the fact that men tended to be more active in searching for new jobs [102]. Furthermore, AI systems risk creating cultural homogenization, as they more and more generate the content that we consume [11]. As an example, large languages like English are by far more common than smaller ones like Swedish. This risks strengthen whatever is already very visible and domineering.

2.10.4 Inequality

A big societal risk is an increase of inequality and the digital divide [17]. This power gap comes with great risks, fearmongering to manipulate others being one of them. Therefore, one of the most important tasks ahead is to provide access to education in critical thinking, AI and the means to take part in it.

2.10.5 Safety and Explainability

If AI systems are deployed in e.g. medical diagnosis, treatment, or self-driving cars, their safety is paramount. Current systems are still sensitive to outlier conditions, or adversarial attacks which fool systems by changing inputs in a way that is imperceptible to humans [59]. Also, systems need to be explainable to avoid harmful decisions and if they happen, to provide clarity in terms of what party bears responsibility [108].

2.10.6 Weapons

AI could be used for further weapons research, including autonomous weapon systems and harmful biological agents. Such weapons, like heat-seeking missiles, have existed for a long time. However, recent autonomous weapons have been facing similar issues as self-driving cars [6].

2.11 The AI ecosystem

The current development of AI is to a large extent being driven by two factors: data and computational power [99]. Algorithm development is certainly important as well, but computer code can easily be shared and replicated. By a similar line of reasoning, human ability is important, but the contributions of individuals tend to even out in large numbers.

2.11.1 Data

Data itself constitutes a societal risk. Mass surveillance has been implemented by states and companies to various extents [34]. Digital technologies, such as smartphones with cameras, have allowed for an exponential increase of the amount of data generated and saved per day [80]. The Internet has meanwhile provided an easy way of spreading and accessing this data. Many of the recent high performing models have scraped data without permission or knowledge from the creators [82]. Furthermore, annotations have been outsourced to workers in developing countries for very little pay under precarious conditions [61].

2.11.2 Computational Resources

As the amount of data grows, so does the demands on the computational hardware to process it. This manifests itself in the form of CPUs, memory and to an increasing extent, GPUs [68] from companies such as NVIDIA and AMD.

Cloud computing generally offers improved flexibility and scaling capabilities when compared to desktop computers at the cost of loss of control. This infrastructure is in the form of data centers with computational hardware. These resources are accessed through the internet by multiple users and managed by a central instance. Data centers are sometimes run by national actors, such as NAISS of Sweden which provides cloud computing to university-affiliated researchers in machine learning. A significant amount of cloud computing actors are companies. These include Databricks, Amazon Web Services, Microsoft Azure and Google Cloud [72]. An example of a datacenter in operation can be seen in Figure 2.5.

Figure 2.5. A datacenter at CERN. Image sourced from [49].

2.12 Ethics

There are many ethical theories, but broadly, they fall into two categories [4]. The first one is Teleology, which is concerned with ends and consequences. One example is Utilitarianism, which aims to maximize happiness for as many people as possible. One saying from a more practical perspective is ’the ends justify the means’. The other broad category is Deontology, which is more concerned with duty and doing what is right for its own sake. One example is virtue ethics, which puts more emphasis on the person’s character. Ethics does not in itself provide an answer to what system is ’the right one’ for any situation. However, studying it sheds light on what your own values are and what system you are reasoning by, because you are always implicitly following some system of values. If values are communicated and understood, it can also facilitate tolerance and conversation even when they are different. This is different from relativism, i.e. the belief in no moral principles [22]. Relativism can manifest itself as blindly following your own conscience or obeying current norms. Scientists need to be objective and reason from facts, but they do not have to be neutral in their opinions.

2.12.1 Legislation

Regulation concerning e.g. AI safety and weaponry should build on experience with previous technologies. This has shown repeatedly that legislation need to be clear. A worldwide ban on technology is much more efficient than encouraging responsible and restrained use [92]. This is one of the explanations of why chemical, biological or nuclear warfare is rare along with human cloning research. There is a consensus that such use cases are always wrong or too risky. Those who break these rules are ostracized and punished. However, sometimes there are big risks with what we are already used to, and this is where legislation has one of its biggest challenges ahead.

3. Biomedical Image Analysis

This chapter begins by providing an overview of computerised image analysis applied to biomedical data. Then, two of the most common problem settings are described: classification and segmentation. The remainder of the chapter is devoted to three different problems together with my contributions to solving them.

3.1 Preliminaries

Computerised image analysis is defined as extracting some type of information from digital images [39]. The focus of this thesis is image analysis, which is not the same as video analysis. The latter means evaluation of temporal data, i.e. the change of structure or appearance in images over time. The fundamentals of video analysis is the same as for static images.

Images can be both two dimensional (2D) and three dimensional (3D). In biomedical settings, 2D images are usually the result from microscopy or radiology. 3D images can be the result of optical imaging by e.g. CT (Computed Tomography). They can also come from magnetic imaging by MRI (Magnetic Resonance Imaging) or radioactive scans of the brain or body structures by e.g. PET (Positron Emission Tomography). Images can also be constructed by visualizing ultrasound data. In this thesis, the focus is on 2D images, but the analysis of 3D images is carried out in an analogous fashion.

Images can be in different modalities, which usually refer to the method for generating the images. As an example, in light microscopy, visible light is used to view samples and take images. In contrast, fluorescence microscopy uses higher intensity light which triggers a fluorescence reaction in the samples that is then viewed or imaged. Modalities can be combined for more advanced analysis using e.g. registration methods. This thesis focuses on single modalities of varying types.

Digital images can be synthetic or artificial, i.e. generated by computers. This thesis instead concerns images generated by a sampling of a continuous signal from an object in the real world via emitted or reflected photons. This process is exemplified in Figure 3.1. Each image pixel represents a sample of the signal both in the intensity and spatial domain. As images are usually manipulated by e.g. zooming and rotation, interpolation needs to be performed if the image dimensions are to be retained. This corresponds to a resampling operation, which needs to be performed with care in order to not introduce too many interpolation artifacts due to e.g. aliasing.

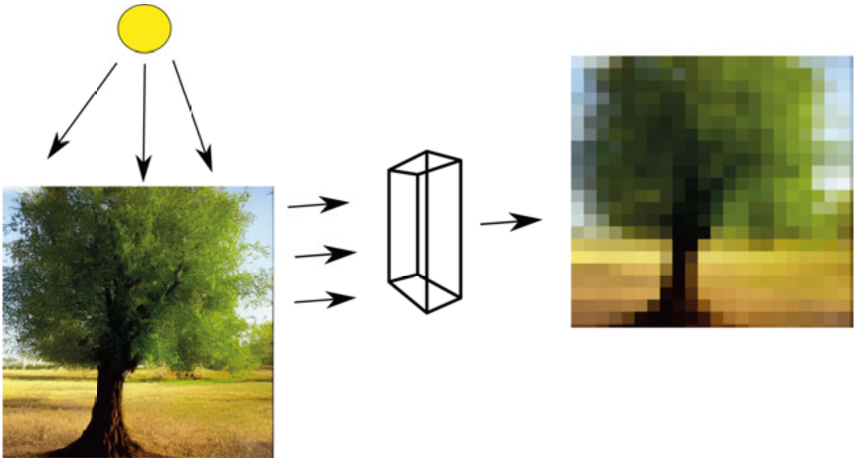


Figure 3.1. The formation of an image of a real-world object. In the top left, a source emits light. This is reflected by a real world object in the bottom left. This is captured by an imaging system in the middle. Finally, a digital image of the object is captured in the right.

3.2 Classification

In an image analysis context, classification typically involves determining what kind of object is present in an image. These objects can be tangible things, such as different species of bacteria. Classification can also indicate quality, such as the presence of an underlying disease in a tissue sample. In both cases, the images need to be analyzed in terms of their features, like the textures, shapes, colors and distributions of items.

3.2.1 Classification by classical methods

Consider as an example the problem of increasing amounts of resident space objects (RSO), such as space junk, in orbit around the earth. It is desirable to track these objects to avoid collisions and for potential cleaning missions. Tracking can be performed by satellite onboard cameras that take images in specific directions when objects are expected to pass [7]. An image might look like Figure 3.2. Here, stars can be seen as bright points. A passing object can be seen as a bright line, due to its movement during the exposure.

The features in the image could be classified using manually constructed detection algorithms. A line detection filter, i.e. the Laplacian of Gaussian, could highlight the presence of lines in the image. False detections such as stars could be handled using connectivity analysis, i.e. removing the smaller detected objects. The methodology of handcrafted solutions to specific problems is today referred to as classical methods in image analysis.

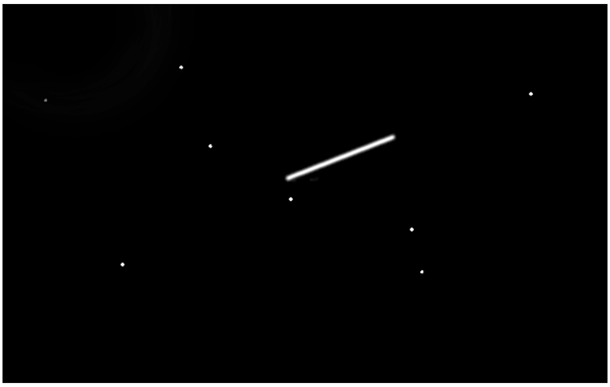


Figure 3.2. A simulated image of stars and a Resident Space Object passing by, visible as a line. The image contrast has been enhanced for clarity.

These approaches are suited for problems characterized by low variation. However, there are drawbacks. The manual design is time-consuming, and algorithms tend to generalize poorly when the conditions are varying, such as when objects are occluded. It is difficult to manually design algorithms that can handle all probable cases.

3.2.2 Classification by deep learning

To solve the same problem using neural networks, a CNN could be used [40]. First, images would be loaded into the network. They would preferably number in the thousands to capture sufficient variation of the data. During training, filters that indicate the presence of relevant features in the images would be automatically learnt. Finally, the network could be used to classify unseen images. Still, the network would have to be retrained if the imaging conditions changed. Therefore, the key is large amounts of data, which in turn means larger networks are needed to learn the variations of the data, which in turn means larger amounts of computational power is needed for processing.

3.2.3 Performance metrics

The performance is assessed using a number of different metrics. In a binary classifier, four outcomes are possible: True Positive (TP), where an object is present and has been classified as present; False Positive (FP), when an object is not present but has been classified as present; True Negative (TN), where an object is not present and has not been classified as present; and False Negative (FN), where an object is present and has not been classified as present. These can be presented in a 2D confusion matrix.

More detailed metrics can also be used. *Accuracy* is defined as the number of correctly classified samples divided by the total number of samples:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.1}$$

Two other common metrics are sensitivity (the true positive rate) and specificity (the true negative rate). They are useful for disease prediction but have to be combined with disease prevalence in clinical practice [1]. They are defined as follows:

$$\text{Sensitivity} = \frac{TP}{TP+FN}, \tag{3.2}$$

$$\text{Specificity} = \frac{TN}{TN+FP}. \tag{3.3}$$

3.2.4 Datasets

The data is typically divided into at least two partitions: a training set and a test set [30]. The training set is usually the biggest partition and is used in the training phase. The test set is used after training to see how the trained model performs on unseen data. Validation sets are often used as well to check the generalization performance of the model during training. It is crucial to design the partitions with care to prevent data leakage [54]. Data leakage occurs when the model learns irrelevant features for classification. For the RSO detection example, if the test set contain the same RSO as the training set, even if the images themselves are different, the classifier might pick up subtle features that only exist in these particular objects. This would give a misleadingly high accuracy on the test set.

3.2.5 Data augmentation

In biomedical image analysis, datasets are usually difficult to find as expertise is needed for high quality sample preparation and labelling [32]. To increase the amount of training data, data augmentation is usually employed [94]. Augmentation can be performed in many ways, such as by adding small amounts of noise, cropping, or by rotations while keeping the labels intact. Some examples are shown in Figure 3.3. Data augmentation by rotations is usually performed in microscopy, as classification should be invariant to rotations of the sample under a microscope. However, each rotation of the object has to be learnt separately, meaning the number of parameters increases. This can increase the risk of overfitting.

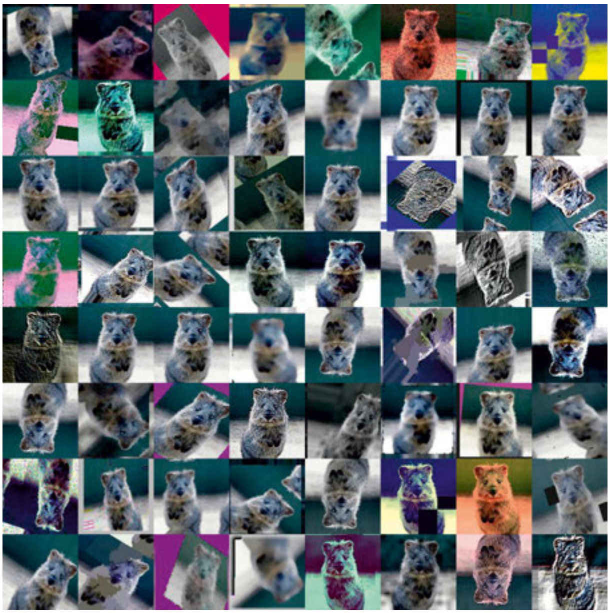


Figure 3.3. Data augmentation examples. The original image is transformed by e.g. noise additions or rotations while keeping its label intact. Image sourced from [67].

3.2.6 Overfitting and underfitting

One common problem when constructing classifiers is overfitting [26]. Here, the training data is learnt and classified to a high degree of accuracy, but the performance on the test set is significantly lower. What usually has happened in these cases is that the network has learnt the variations of the training data to a too high extent, and cannot generalize to unseen data which typically looks slightly different. The risk of overfitting can be explored using the cross-validation technique, where the training and test sets are recombined into e.g. five different folds and assessed separately.

When the network is unable to learn the features, low training accuracy occurs and the network is said to be underfitting. The network is biased by making erroneous assumptions about the relationships between inputs and outputs. An optimally trained network has found a balance between bias and variance [28]. Overfitting and underfitting can be quantified by comparing the empirical risk with the testing accuracy. The empirical risk for a classifier $R_{emp}(h)$ [104] is defined as:

$$R_{emp}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i) \tag{3.4}$$

where h is the hypothesis (i.e., our network), L is the loss function, and x_i and y_i are n independent input and output samples, respectively. Since a low empirical risk is directly proportional to a high classification accuracy on the

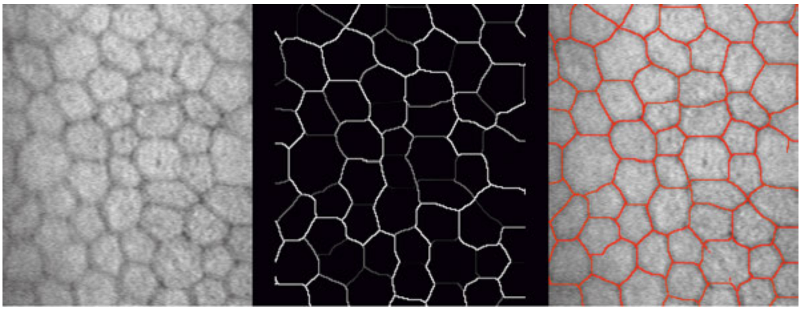


Figure 3.4. Segmentation by the stochastic watershed algorithm. The image shows, from left to right: endothelial cells, segmentation image after stochastic watershed algorithm, segmentation after thresholding the result superimposed on the original image.

training set, the training accuracy can be used as a proxy for the empirical risk. From this, the overfitting ratio is constructed in Paper I:

$$Overfitting\ Ratio = \frac{Training\ Accuracy}{Testing\ Accuracy} \tag{3.5}$$

Higher values indicate a higher degree of overfitting.

3.3 Segmentation

Segmentation involves dividing images into different regions [39]. This can provide a more holistic understanding of the image, such as separating the background from the foreground. Segmentation can propose candidate regions for further analysis, such as identifying not only regions where objects are located in the image, but also what classes the regions represent. Overall, there are three types of segmentation: semantic, instance, and semantic instance.

3.3.1 Semantic segmentation

Semantic segmentation aims to assign a label to each pixel in the image, where similarly labelled pixels share some common characteristic, or class. This can be performed by e.g. border delineation, which determines the boundaries between regions of the image. An example is the watershed algorithm [10], which finds the local intensity ridges between different regions. The ridges are found by analysis of the topographic maps of intensities between seeds. An example of applying the watershed algorithm on an image of endothelial cells is shown in Figure 3.4.

A commonly used deep learning architecture for semantic segmentation is U-Net [89]. It consists of two parts: the encoder and the decoder. The

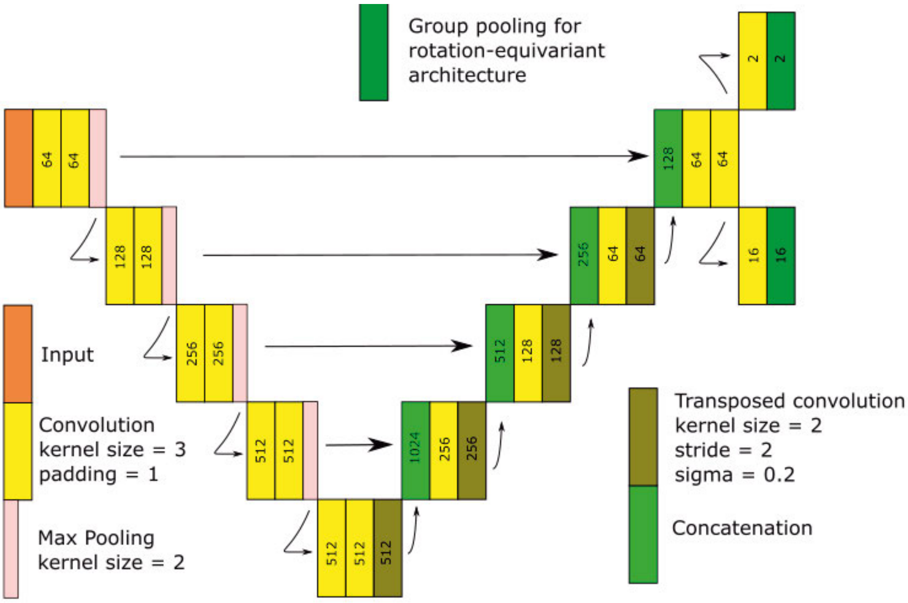

Group pooling for rotation-equivariant architecture
Input
Convolution kernel size = 3 padding = 1
Max Pooling kernel size = 2
Transposed convolution kernel size = 2 stride = 2 sigma = 0.2
Concatenation
64
64
128
128
256
256
512
512
512
512
512
1024
256
256
512
128
128
256
64
64
128
64
64
16
16
2
2

Figure 3.5. The U-Net architecture modified with an extra head for semantic instance segmentation.

encoder takes the image as input and downscales it in steps to a latent space where similar features are grouped closely. The decoder then upscales the image in steps, reconstructing the original dimensions from the latent space and by skip-connections from the decoder. Finally, each pixel is assigned a label, completing the semantic segmentation. Training of U-Net is performed similarly as for classification networks, with the difference that the labels are segmentation maps instead of scalars.

3.3.2 Instance segmentation

The objective of instance segmentation [43] is to label the different object instances regardless of what class they belong to. Consider an image of multiple cats and dogs. In semantic segmentation, the cats would be labelled different than the dogs, but the individual cats would be indistinguishable from one another. The same would hold for the dogs. In instance segmentation, all individuals would be labelled separately from each other. This is hard to achieve for networks designed for semantic segmentation when instances are overlapping in image space.

3.3.3 Semantic instance segmentation

The goal of semantic instance segmentation is to output both the classes and instances in the image. One way of accomplishing this is by adding a separate head to the U-Net architecture, right before the semantic segmentation layer. This additional head contains a 16-channel representation. At inference time, this representation is clustered using Kmeans to extract the different instances. The results are mapped to the original image pixels, yielding both the classes and instances. An example of the U-Net architecture modified for semantic instance segmentation is shown in Figure 3.5.

During training, a discriminative loss function [12] is added to the cross-entropy loss used for semantic segmentation. The purpose of the discriminative loss is to enforce a mapping of the training data to a latent space in the instance head. The mapping enforces features originating from different instances to be separated from each other, while features originating from the same instances are enforced to cluster tightly. This loss term has the following form:

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[||\mu_c - x_i|| - \delta_v]_+^2 \tag{3.6}$$

$$L_{dist} = \frac{1}{C(C-1)}\sum_{c_A=1}^{C}\sum_{c_B=1}^{C}[2\delta_d - ||\mu_{c_A} - \mu_{c_B}||]_+^2(c_A \neq c_B) \tag{3.7}$$

$$L_{reg} = \frac{1}{C}\sum_{c=1}^{C}||\mu_c|| \tag{3.8}$$

$$L = \alpha \cdot L_{var} + \beta \cdot L_{dist} + \gamma \cdot L_{reg} \tag{3.9}$$

Equation 3.6, L_{var}, corresponds to a term that enforces features originating from the same instance to minimize the distance to their center. Equation 3.7, L_{dist}, pushes different clusters apart from each other. Equation 3.8, L_{reg}, prohibits cluster terms from growing too large. The number of clusters is C, N_c is the number of elements in cluster c, x_i is an embedding, and μ_c is a cluster center. $||\cdot||$ is the L1 or L2 norm, and $[x]_+$ = $\max(0,x)$. δ_v and δ_d constitute margins for the variance and distance terms, respectively. The constants α, β and γ control the contribution of each term in equation 3.9.

3.3.4 Performance metrics

Segmentation models are evaluated differently than classification models as each output pixel has to be compared with its label. One common way of doing this is with the Dice score [9]:

$$Dice = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.10}$$

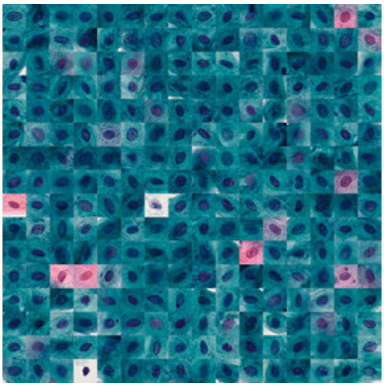


Figure 3.6. Example cells from the Oral Cancer dataset.

This is easily calculated for classes, but the equation cannot be directly applied to instances as they are permutation invariant. This means that the instance labels are without meaning, except in the sense that they differ from other labels. This can handled in the following way. First, the output instances are compared with all the labels, and the best overlap is selected. Then, the selected instance and label are used to update the Dice score and removed. This process is repeated until no instances and labels are left.

3.4 Oral cancer

Oral cancer can develop in the lips, mouth or upper throat. In 2017, 389 760 people had the disease and 193 696 people died from it [85]. Survival is heavily dependent on the stage at which the cancer is detected. There are a number of causes, including tobacco and alcohol use [37], as well as infection by the human papillomavirus [38]. The latter also causes cervical cancer and can be prevented with vaccines. Treatment is typically surgery, as well as radiotherapy and chemotherapy [63].

Oral cancer is usually diagnosed by biopsy followed by tissue analysis [91]. As early detection is important for survival, minimally invasive screening methods are being tested. They follow a similar procedure as screening for cervical cancer. For oral cancer screening, a brush is used to scrape cells from the inside of patients' mouths. The samples are then put in liquid vials, smeared on glass slides and stained. Cytotechnologists then examine the sample, typically looking at around 100 000 cells in different magnifications. This can take around 10-15 minutes. There is high demand to automate this process as it can be difficult and costly to allocate this expertise when needed. Furthermore, malignancy associated changes (MACs) might be detected with automated screening. This refers to changes in chromatin structure and morphology of the nucleus in cells, which are imperceptible to humans [36].

3.4.1 Dataset

The oral cancer dataset originates from a collaboration between Uppsala University, Karolinska Universitetssjukhuset and Södersjukhuset [60]. After the samples were stained, they were imaged and color information was removed. Before cytological analysis, each image was centered around a nucleus and cropped to 80 by 80 pixels. The images came from twelve patients, six of which were healthy, and six of which had a cancer diagnosis. These were partitioned into training data consisting of 8508 images and test data containing 9942 images. The sets were kept apart at a patient level to avoid data leakage. Images were given weak labels, meaning individual cells were given the diagnosis of the patient even though not all cells would be affected. Example images are shown in Figure 3.6.

3.4.2 Paper I

Paper I describes research conducted into automated analysis of oral cancer. A sufficiently deep network was chosen to allow for a hierarchy of features to emerge. The VGG16 classifier network, illustrated in Figure 2.1, was selected as it had been established in the biomedical image analysis community. The model was implemented in PyTorch. Sufficiently large amounts of training training data was needed for convergence during training. The training data was augmented by rotations of 0, 90, 180 and 270 degrees, multiplying the number of training samples by four.

3.5 Segmentation of cell nuclei

One way to develop segmentation algorithms is to use challenge datasets. One of them is BBBC038 from the website Kaggle, where segmentation methods are ranked according to their performance on the test set [14]. The BBBC038 dataset consists of 670 training images and 65 test images. The training images are accompanied by pixel masks which vary in numbers, shapes and sizes. An example is shown in Figure 3.7.

The images are highly varied and have been collected from multiple universities, companies and hospitals. They originate from different organisms such as humans, mice and flies. The nuclei are in different states, undergoing e.g. cell division, genotoxic stress or differentiation. They are present in cultured mono-layers, tissues and embryos. The images are of varying quality, magnifications, illumination, and are imaged in different modalities like fluorescent and histology stains.

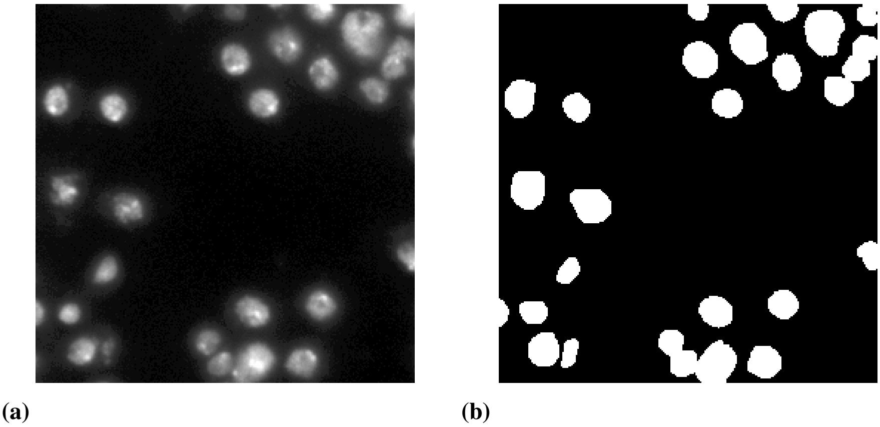


Figure 3.7. An example from the BBBC038 dataset. (a) The raw image of cells. (b) The instance masks.

3.5.1 Paper II

The aim of Paper II was to develop a higher performing and more data efficient algorithm for semantic instance segmentation in biomedical image analysis. The U-Net architecture was chosen as a baseline for its capabilities in semantic segmentation. It was modified with an extra head for instance segmentation according to the procedure in Subsection 3.3.3 for use in combination with a discriminative loss function. The architecture is shown in Figure 3.5.

The BBBC038 images had to be selected or cropped to fit the U-Net implementation, which only worked on images of certain fixed sizes. The cutouts were chosen in sizes of 256 by 256 pixels. Additionally, each cutout could only contain a fixed number of instances, as this was a parameter used by the Kmeans clustering method. The images were therefore searched for patches containing this number of nuclei. This meant that some images had to be discarded while others provided multiple patches. 500 images were used for the training set and 16 were used for the test set. The images were converted to greyscale.

3.6 TEM images of viruses

Viruses are usually around 20-300 nanometers in size. This is too small to image by conventional microscopy methods. Therefore, Transmission Electron Microscopy (TEM) is usually applied [56]. In TEM, an electron beam is transmitted through a sample suspended on a grid. During transmission, the beam interacts with the sample. This signal is magnified and focused onto a sensor, e.g. a fluorescent screen, producing the final image.

TEM is commonly used to characterize subcellular structures and new pathogens when combined with other methods. As an example, starting in the year

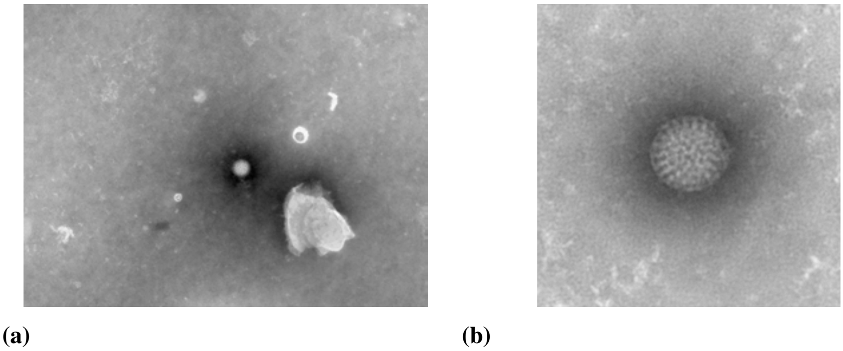


Figure 3.8. TEM image of a rotavirus. (a) The virus specimen is found in the center of the image. (b) Cutout image centered around the rotavirus particle.

2020, the COVID-19 pandemic impacted the world, resulting in millions of deaths. TEM played an important role in identifying the cause as the SARS-CoV-2 virus, a type of corona virus, with its characteristic corona of glycoproteins [58]. Furthermore, TEM confirmed biochemical data that localized the entry pathways into host cells. Methods for diagnosis in a clinical setting include molecular tests such as real-time reverse transcriptase-polymerase chain reaction (rRT-PCR) or antigen tests on, e.g., nasal specimens [35].

Epidemics caused by new or unknown pathogens are expected in the future. Therefore, methods for more efficient and accurate characterizations of novel samples are valuable. Similarly as for diagnosis of oral cancer and many other diseases, it can be difficult to find the experts and allocate the data needed for manual analysis. Machine learning methods could automate this classification. However, this requires large amounts of training data which can be hard to find. More data efficient methods are needed.

3.6.1 Dataset

A new dataset was prepared with these goals in mind. It consists of 14 different virus species imaged by transmission electron microscopy (TEM) [64, 65]. The images are cropped around the particles in sizes of 256 by 256 pixels. Each image is labelled with its corresponding virus particle contained in the image. The training data has 93 images per class for a total of 1302 images. An additional augmented training set has the training samples rotated by 0, 90, 180 and 270 degrees, in addition to flips around one axis or no flips. The validation data has 2249 samples, while the test data has 1900 samples. For the validation and test sets the classes have varying amounts of samples. To prevent data leakage, all partitions have been kept separate at the image level. An example image can be seen in Figure 3.8.

3.6.2 Papers III and IV

In Papers III and IV, the aim was to investigate how data efficiency could be improved using equivariant neural networks. Paper IV extended the experiments from Paper III with more optimized networks and varied experimental conditions [13]. These papers focused more on collecting empirical results than innovation in comparison to Papers I and II.

Similarly as for Paper I, the VGG16 architecture was chosen for its familiarity within the biomedical image analysis community. To gather more empirical results, a custom architecture was selected as well. This second network was similar in design to the ResNet network, but with significantly fewer parameters to learn. The custom network is illustrated in Figure 3.9.

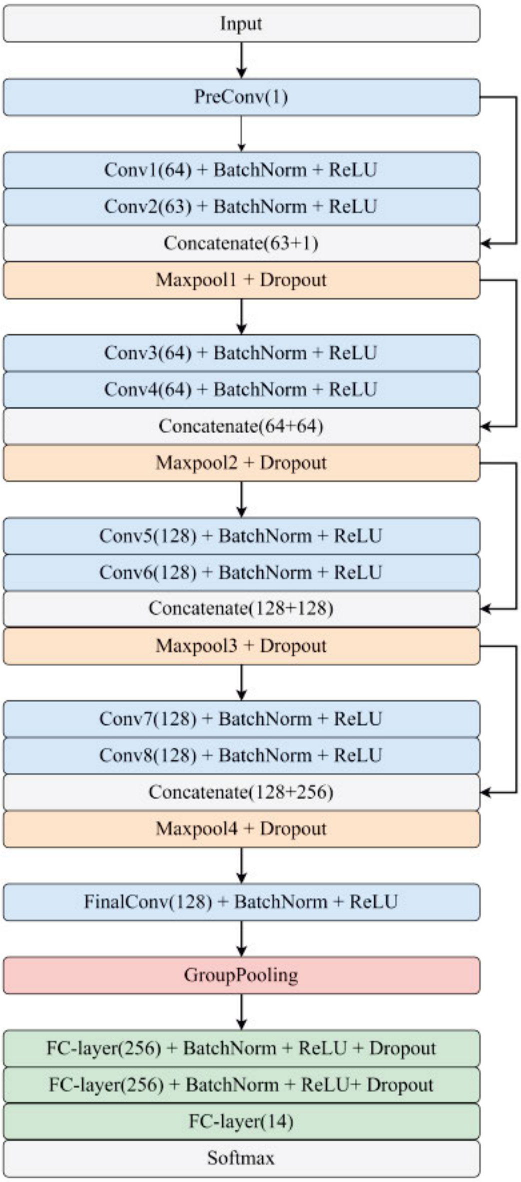

Input
PreConv(1)
Conv1(64) + BatchNorm + ReLU
Conv2(63) + BatchNorm + ReLU
Concatenate(63+1)
Maxpool1 + Dropout
Conv3(64) + BatchNorm + ReLU
Conv4(64) + BatchNorm + ReLU
Concatenate(64+64)
Maxpool2 + Dropout
Conv5(128) + BatchNorm + ReLU
Conv6(128) + BatchNorm + ReLU
Concatenate(128+128)
Maxpool3 + Dropout
Conv7(128) + BatchNorm + ReLU
Conv8(128) + BatchNorm + ReLU
Concatenate(128+256)
Maxpool4 + Dropout
FinalConv(128) + BatchNorm + ReLU
GroupPooling
FC-layer(256) + BatchNorm + ReLU + Dropout
FC-layer(256) + BatchNorm + ReLU+ Dropout
FC-layer(14)
Softmax

Figure 3.9. The custom architecture used in the experiments. The group pooling layer is only used in the equivariant version of the architecture.

4. Equivariant Neural Networks

This chapter introduces equivariance and how it manifests in convolutional neural networks. Building on this knowledge, equivariant neural networks are then presented. The second half of the chapter is devoted to the practicalities of how to design and test equivariant neural networks. This includes rotation-equivariant versions of the classifiers and segmentation models from Chapter 3.

4.1 Equivariance in CNNs

CNNs are equivariant to translations of objects in the image f [40]. This means that if the object is shifted left-right or up-down, the output will be shifted equivalently after a convolution layer. This can be expressed in the following way:

$$T(C(f)) = C(T(f)) \tag{4.1}$$

Here, the translation operator T commutes with the convolution operator C. Translating the output of the convolution is identical to performing the convolution first followed by a translation.

The reason for this property is weight sharing. That is, filter kernels are identical across the same layer and channel. CNNs can additionally become invariant to small translations of the inputs by adding pooling layers, which remove the spatial information from the detected features. This property does not hold for other transformations, such as rotations or reflections.

4.2 Group Equivariant Neural Networks

There are several approaches to extend the equivariant properties of CNNs. One of the most prominent is Group Equivariant Convolutional Networks [21]. Here, convolutions are generalized to cover isometric, i.e. distance-preserving, transformations as well as translations of the kernel k across the input. The G-convolution is defined as follows for the input layer and the transformation g:

$$z(x,y) = k * f(x,y)[g] = \sum_{i=-a}^{a} \sum_{j=-b}^{b} k(i,j) f[g^{-1}(x-i, y-j)] \tag{4.2}$$

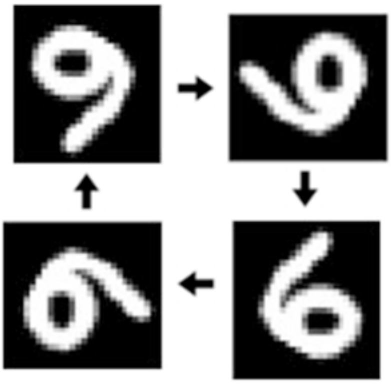


Figure 4.1. The rotations of the p4 group, going clockwise: 0 degrees (top left), 90 degrees, 180 degrees and 270 degrees. The original handwritten image depicts the digit nine, but its rotation by 180 degrees resembles the digit six.

where z is the output, x and y are image coordinates, and i and j are kernel coordinates in the window of size $[-a,a]$ by $[-b,b]$. For transformations g in the symmetry group G the output is a stack of feature maps. For subsequent layers, the G-convolutions are defined similarly on the stacks. Networks can be constructed to be equivariant to the transformations of the chosen symmetry group to arbitrary depth. Group pooling layers can be used to make the network invariant to the group transformations.

A commonly used symmetry group is p4, consisting of all compositions of translations and rotations by 90 degrees about any center in a square grid. The rotations of this group are illustrated in Figure 4.1, which also shows that the rotation of an object sometimes carries meaning. Another group is p4m, which in addition to the transformations in the p4 group also contain mirror reflections. These groups are subgroups of the Euclidean group in two dimensions $E(2)$, consisting of all isometric transformations [3]. Other common groups are the 2D orthogonal group consisting of all reflections and rotations $O(2)$, the 2D special orthogonal group consisting of all rotations $SO(2)$, and the translation group $(\mathbb{R}^2,+)$.

4.3 Other approaches

There are multiple other ways to achieve equivariance to more transformations than translations. One method applies the transformations to the data or features directly instead of the kernels [25]. Another approach is CFNet which uses the 2D discrete Fourier transform combined with conic convolutions, achieving equivariance to the p4 group [20]. Another method is to learn steerable atomic basis filters for continuous resolution in orientation [107]. Recently, the attention mechanisms of transformers have been combined with G-convolutions [88].

One common way of implementing equivariant neural networks is to use the General E(2) - Equivariant Steerable CNNs framework [106]. As the name hints, it implements the E(2) isometric transformations in 2D. It does this by solving kernel constraints throughout the layers in the network. A kernel constraint looks like the following:

$$k(gx) = \rho_{out}(g)k(x)\rho_{in}(g^{-1}) \quad \forall g \in G, \quad x \in \mathbb{R}^2 \tag{4.3}$$

where ρ_{out} and ρ_{in} are the output and input representations respectively. Each input and output has to have its representation defined. As an example for a G-convolution in the first layer, the input representation will be the trivial representation, while the output will be the regular representation, specifying the stack of feature maps determined by the chosen symmetry group. The E(N)-equivariant steerable CNNs framework implements the same principle on $\mathbb{R}^3$ [18].

4.4 Designing equivariant neural networks

When constructing an equivariant neural network, the workflow should be structured. A good starting point is to select a deep learning framework that is user friendly and easily integrated with other tools. In my projects, I used PyTorch, a library on top of the Python programming language. Secondly, a framework for equivariant neural networks is highly recommended. I used e2cnn, which is an extension of Pytorch.

The equivariant property cannot be assumed even if you have a solid grasp of what group you want to design the network to be equivariant to. A test driven development is therefore recommended. For an invariant classifier, a recommended approach is to take a few images from the test set and perform the component transformations of the symmetry group on them. That is, if the network should be invariant to e.g. rotations of 0, 90, 180 and 270 degrees, these rotations should be performed on some of the test images. Then, the images should be loaded into a forward pass of the model. The classifier should yield identical results for any image regardless of its input orientation. There can still be small errors due to e.g. numerical errors.

Similar tests should be used for other types of models. For a rotation-equivariant segmentation network such as U-Net, the resulting segmentation should be identical if the input image is rotated by the component transformations of the symmetry group. The results can be compared for all pixels in the image. Similar to classification models, a small amount of pixels, preferably less than one percent, can differ due to e.g. numerical errors.

The layers of the network have to be designed in a way that does not break the equivariance properties. One common issue is that pooling layers are not aligned with the input feature dimensions. This will cause the stack of transformations in the group to pool over different areas in the input feature map, producing different results depending on the orientation. The dimensions and the strides of the pooling layers together with the dimensions of the input features have to be designed to account for this.

4.5 Papers I, III and IV - equivariant classifiers

In Papers I, III and IV, the VGG16 classifier was redesigned to be equivariant to more transformations than translations using the e2cnn library. The details of designing the network in Paper I is hereby described.

The first layer lifts a trivial representation (the input images) to a regular representation, using the p4 symmetry group of translations and multiples of 90-degree rotations. Subsequent layers use the same symmetry group. A group pooling operation follows the final G-convolution to make the classifier invariant to the transformations of the group. The final layers are fully connected to perform the classification of the detected features. Details of the training settings and architecture are shown in Table 4.1.

An extended custom architecture is described in Paper IV. The classifier is designed layer by layer in a similar process as for Paper I. Furthermore, as this architecture contains skip connections, the direct sum function from e2cnn is used to concatenate tensors of the same type. A PreConvolution layer is used to prepare the layer for subsequent concatenation. The kernel is of size 1, i.e. a scalar, to convert the input representation to a regular representation. The architecture is shown in Figure 3.9.

The test of equivariance to the transformations of the symmetry group showed an error of typically less than one percent in all cases. This was in contrast to baseline CNNs which typically showed errors around ten percent.

4.6 Paper II - equivariant neural networks for semantic instance segmentation

In Paper II, a rotation-equivariant version of the U-Net architecture for semantic instance segmentation is described. The design follows the same principles as the equivariant classifiers in section 4.5. Layer by layer, the convolution operations have been replaced by G-convolutions using the p4 symmetry group of translations and multiples of 90 degrees. Direct sums are used to perform the concatenation operations for the skip connections.

As the U-Net is modified with an extra head in combination with a discriminative loss for instance segmentation, the network needs to be equivariant to

Table 4.1. *Settings for the VGG16 classifier architecture and training procedures.*

Parameter	Setting
Loss function	Cross entropy
Weight initialization	He
Optimizer	Adam
Learning rate	0.00001
Batch normalization	Batches of size 128
No. of epochs	200
Validation frequency	1/5 epochs
Activation functions	ReLu
Dropout	0.5 between linear layers
No. of channels	16-16-32-32-64-64-64-128-128-128-128-128-128
Convolution layers (size, stride, padding)	Layers 1-12: (3,1,1) Layer 13: (4,0,0)
Maxpooling layers (size, stride, padding)	Layers 1-4: (2,2,0) Layer 5: (2,1,0)
Linear layer parameters (input, output sizes)	(128,4096) - (4096,4096) - (4096,2)

$$\begin{tikzcd}
\text{Input} \arrow[r, "\textit{U-net}"] \arrow[d, "\textit{Rotation, 90 degrees}"'] & \text{Instance representation} \arrow[r, "\text{Clustering}"] \arrow[d, "\textit{Rotation, 90 degrees}"] & \text{Output} \arrow[d, "\textit{Rotation, 90 degrees}"] \\
\text{Input}_{90} \arrow[r, "\textit{U-net}"] & \text{Instance representation}_{90} \arrow[r, "\text{Clustering}"] & \text{Output}_{90}
\end{tikzcd}$$

Figure 4.2. Commutative diagram for the U-Net semantic instance segmentation network.

90-degree rotations not only for the segmentation output but for the instance output as well. This is proved in the following way. In the instance head, each pixel is associated with 16 scalar values. These values span a space which looks the same regardless of the rotation of the underlying pixels. Therefore, the clustering step is invariant to rotations, as well as any transformation acting only on the pixel coordinates. This also holds for any clustering algorithm not relying on pixel coordinates. The commutative diagram for the network is shown in Figure 4.2.

Testing the equivariance of the semantic output of the U-Net revealed that only around 167 pixels on average differed from what was expected, i.e. 0.25 % of the total number of pixels.

5. Empirical Results

This chapter introduces the emergent field of empirical deep learning, including topics like scaling laws, data efficiency and convergence time. The chapter begins with some lessons learned about setting up and working with an appropriate computational environment. This is followed by the experimental setups and results comparing equivariant neural networks with baseline CNNs.

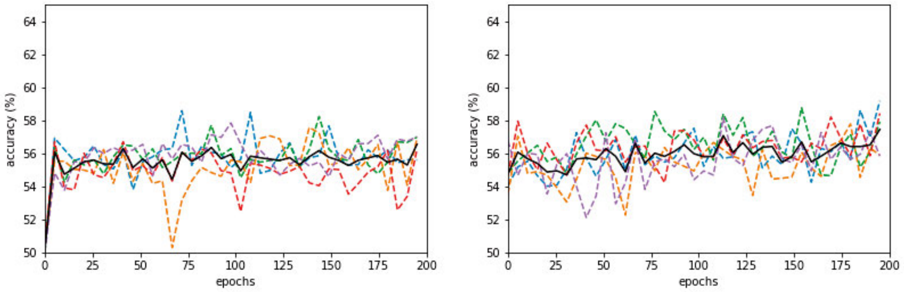
5.1 Deep learning computational and development workflow

During the course of the thesis project, I changed both the computational hardware and the software development environment several times. I began by working on my laptop, equipped with a Quadro P2000 GPU with 4GB memory and 8 GB system RAM. This was sufficient for simple models using moderate amounts of data. Moving e.g. the backpropagation computations to the GPU from the CPU resulted in a significant speedup, reducing the time taken from days to hours. However, the GPU memory proved to be the primary bottleneck for heavier computations when more data was added.

In my second year, I moved to a stationary desktop computer with a much more powerful GPU which could handle the needed computations. I accessed this computer remotely using SSH and SCP to transfer files. The coding was mainly performed in the Spyder development environment. In my third year, I tried out different cloud computing environments, including Ericsson Openstack, Databricks and Google Colab, which was selected for Paper II. However, this system proved to be memory-limited for larger equivariant networks. Cloud computing owned by private interests also comes with privacy concerns when using e.g. confidential patient data.

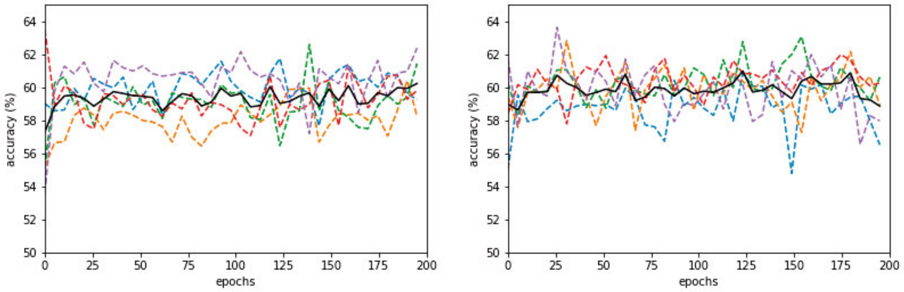
In my final phase of my projects, I moved to the NAISS (National Academic Infrastructure for Supercomputing in Sweden), specifically the Alvis cluster for machine learning research. This provided access to hardware (an A40 GPU and 64 GB system RAM) and storage that was not limiting the models or data I could work on. The system also provided transparent schedulers and computational budgets. Around the same time I switched to Visual Studio Code, a development environment with a debugger and integrations for remote development and file access.

For installing software, python package management through pip combined with virtual environments allowed for isolation of each project. I also used



(a) **(b)**

Figure 5.1. Accuracy on the test set using the VGG16 baseline classifier in Paper I. The colored lines indicate individual runs, and the black line indicates the mean of the runs. (a) Using unaugmented training data. (b) Using augmented training data.



(a) **(b)**

Figure 5.2. Accuracy on the test set using the VGG16 classifier in Paper I. The network was modified to be equivariant to the p4 group. The colored lines indicate individual runs, and the black line indicates the mean of the runs. (a) Using unaugmented training data. (b) Using augmented training data.

Horovodrunner on the Databricks cloud computing platform for distributed deep learning over multiple GPUs on particle collision data from CERN [98]. While this is not related to the papers in this thesis, the same approach can be used on biomedical image datasets, which can be expected to grow in size in time. Code with new implementations was uploaded to github open repositories to aid reproducibility efforts.

5.2 Data augmentation vs equivariance

As described in Section 3.2.5, data augmentation can be used to increase the amount of training data in biomedical image analysis. This provides the networks with additional examples, letting it generalize better. However, this strategy comes at cost of learning the component transformations. One of the

Table 5.1. *Overfitting measurements in Paper I. CNN is the baseline network and GCNN is the rotation-equivariant network.*

Network	Overfitting ratio
CNN with data augmentation	**1.82**
GCNN without data augmentation	**1.69**

main aims of the thesis was to investigate if equivariant networks trained on unaugmented data could replace baseline networks trained using augmented data. In control experiment one, baseline networks were expected to drop in accuracy when training on unaugmented data instead of augmented data. In control experiment two, equivariant networks were expected to not benefit much from training on augmented data if the augmentations were to match the transformations of the symmetry group.

5.2.1 Paper I

In Paper I, the VGG16 model was used to classify oral cancer from cytological images. The p4 symmetry group was used along with augmentations consisting of rotations of 0, 90, 180 and 270 degrees. The results of the augmentation-equivariance experiments are illustrated in Figure 5.1 and Figure 5.2 for the baseline and equivariant networks respectively. It can be seen that the baseline with data augmentation yielded accuracies of around 55-56 % on the test set, while the equivariant network without data augmentation yielded accuracies of around 59-60 %. These results were verified to be statistically significant. For control 1, curiously, the baseline did not benefit from data augmentation. For control 2, as expected, the equivariant network did not benefit from data augmentation.

Both the baseline and equivariant networks yielded 100 % accuracy on the training set. Combining the two accuracies, the overfitting ratio could be calculated. This is seen in Table 5.1. As expected, the equivariant network without data augmentation resulted in less overfitting (1.69) than the baseline (1.82). Similarly, the equivariant network with data augmentation yielded both higher sensitivity (0.62) and specificity (0.60) than the baseline (0.57 and 0.56 respectively).

5.2.2 Paper II

In Paper II, augmentation-equivariance experiments were carried out using a U-Net model modified for instance segmentation. The experiments were first carried out on a dataset of synthetic sticks. The mean instance Dice score on the test set was 0.88 for the last 20 epochs.

Then, experiments were conducted on the modified BBBC038 dataset consisting of cells of varying types. For the equivariant network, the p4 symmetry group was used. Augmentations by rotations of 0, 90, 180 and 270 degrees were also performed. The result are presented in Table 5.2. It can be seen that the equivariant network performed slightly worse than the baseline with data augmentation. The Dice scores were 0.63 and 0.66, respectively. The results of Control 1 was as expected, increasing in Dice score from 0.62 to 0.66 when data augmentation was added. Control 2 yielded slightly lower accuracy when using data augmentation, dropping from 0.63 to 0.62.

Table 5.2. *Results for the instance segmentation experiments in Paper II on the modified Broad BBBC038 dataset. Initial Dice score is the mean of the first 10 epochs on the test set. Final Dice score is an average for the last epoch on the test set. Data augmentation refers to rotations of 0, 90, 180 and 270 degrees. The number of epochs was 100 except for experiment 2, which used 300 (rows 3–6). CNN refers to the baseline network and GCNN refers to the rotation-equivariant network. TD stands for the number of training samples, and AT is short for data augmentation type.*

# TD	Network	AT	Initial Dice	Final Dice
400	**CNN**	**None**	**0.54**	**0.65**
400	**GCNN**	**None**	**0.59**	**0.65**
500	**CNN**	**None**	**0.44**	**0.62**
500	**CNN**	**Rotations**	**0.52**	**0.66**
500	**GCNN**	**None**	**0.52**	**0.63**
500	**GCNN**	**Rotations**	**0.52**	**0.62**
100	**CNN**	**None**	**0.46**	**0.61**
200	**CNN**	**None**	**0.49**	**0.63**
300	**CNN**	**None**	**0.50**	**0.62**
400	**CNN**	**None**	**0.54**	**0.65**
100	**GCNN**	**None**	**0.53**	**0.61**
200	**GCNN**	**None**	**0.53**	**0.62**
300	**GCNN**	**None**	**0.55**	**0.64**
400	**GCNN**	**None**	**0.57**	**0.62**

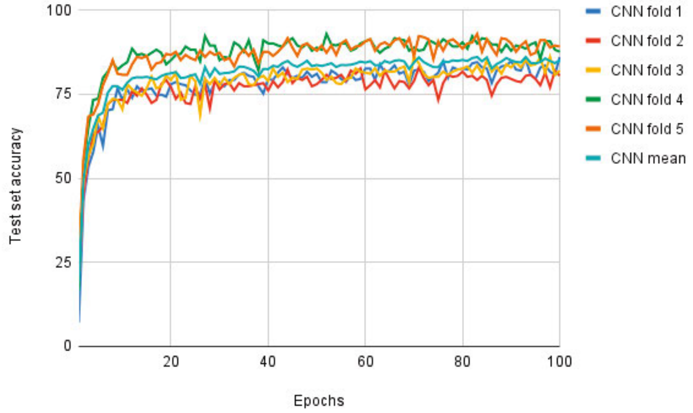


Figure 5.3. The results of Cross-validation for the VGG16 architecture.

5.2.3 Papers III and IV

In Paper III, the VGG16 classifier was used to classify TEM images of viruses. Again, the p4 group was used, but the augmentations included reflections about one axis in addition to rotations by multiples of 90 degrees. The results can be seen in Table 5.3. It can be seen that the rotation-equivariant network using no data augmentation outperformed the CNN using data augmentation, as expected. The best accuracies on the test sets were 78.80 % and 75.27 % respectively. Also for control 1, the CNN improved from 57.64 % to 75.27 % when data augmentation was added. For control 2, when data augmentation was added to the equivariant network, the accuracy increased from 78.80 % to 82.56 %. Note that batch normalization was not used in any of these experiments to reduce memory requirements.

To control for overfitting or selection bias, cross-validation was performed. The training and test datasets were merged. Five folds were constructed by random sampling of the combined dataset. The images were augmented following the same procedure as in the main experiments. Results can be seen in Figure 5.3. It can be seen that there is a considerable difference in test set accuracy depending on which folds are used as training data, indicating the data could benefit from further rebalancing.

In Paper IV, more experimental conditions were carried out, varying both the architectures and the symmetry groups. The D4 and C8 symmetry groups were tested both for the VGG16 and custom architectures. The D4 symmetry group consists of rotations of multiples of 90 degrees and reflections about one axis. The C8 symmetry group consists of rotations of multiples of 45 degrees. The results for the custom architecture, which was designed to reduce the number of weights significantly, can be seen in Table 5.4. It can be seen that, for 300 epochs on unaugmented training data using the custom architecture,

Table 5.3. *Results from the main experiments in Paper III. The numbers are averages and standard deviations over five runs.* Best accuracy *is the percentage correctly classified on the test set and* Time to stability *is in seconds.*

Augmentation scheme	**CNN**	**GCNN**
No augmentation		
Best accuracy	**57.64** **± 2.65**	**78.80** **± 2.23**
Epochs to stability	**31**	**24**
Time to stability	**1577** **± 336.23**	**2145** **± 717.85**
Data augmentation		
Best accuracy	**75.27** **± 0.751**	**82.56** **± 2.44**
Epochs to stability	**15**	**12**
Time to stability	**4317** **± 641.31**	**5469** **± 2670.41**

the D4 group with 72.90 % outperformed the baseline with 69.90 %, which outperformed the C8 group with 64.10 %. When training on augmented data, similar accuracies (88.4 0%, 87.80 % and 88.70 %) were reached regardless of equivariant designs of the network. Still, they all increased in accuracy when augmented data was added, which could be partly explained by the fact that the augmentations were different from the component transformations of the symmetry group C8. However, they matched the transformations of the D4 group, which makes the results curious. The results were similar when using the VGG16 network, as can be seen in Table 5.5.

5.3 Longer training

In Paper IV, both the custom and the VGG16 architectures were additionally trained for 2400 epochs on the unaugmented training sets. This corresponded to eight times 300 epochs, which matched the number of transformations in the augmented data. These experiments were performed as the equivariant networks trained on unaugmented data received one eighth of the training op-

Table 5.4. *Results from the main experiments in Paper IV for the custom architecture.* Best accuracy *is the percentage correctly classified on the test set and* Time to stability *is in hours.*

Custom network	**Baseline**	**D4**	**C8**
No augmentation **300 epochs**			
Best accuracy	**69.90**	**72.90**	**64.10**
Epochs to stability	**274**	**275**	**234**
Time to stability	**0.85**	**7.32**	**3.90**
2400 epochs			
Best accuracy	**83.30**	**89.80**	**89.00**
Epochs to stability	**2260**	**1886**	**2126**
Time to stability	**7.06**	**48.72**	**55.61**
Augmented training **300 epochs**			
Best accuracy	**87.80**	**88.40**	**88.70**
Epochs to stability	**244**	**212**	**2 56**
Time to stability	**4.46**	**33.54**	**40.54**

portunities when compared to CNNs trained on augmented data. The results can be seen in Table 5.4 and Table 5.5. It can be seen that the equivariant networks significantly outperform the baseline network in both cases. Also, the accuracies match the previous state of the art accuracies on the dataset [64]. This shows that the benefits of equivariant networks sometimes manifest after very long training times.

5.4 Convergence speed

Equivariant neural networks have proven to converge faster than baseline CNNs, since they do not have to learn the symmetries of the data explicitly [106]. Similar effects have been seen using equivariant MDP homomorhic in reinforcement learning [103], and in facial classification based on invariant Zernike moments [62].

Table 5.5. *Results from the main experiments in Paper IV for the VGG16 architecture.* Best accuracy *is the percentage correctly classified on the test set and* Time to stability *is in hours.*

VGG16 network	**Baseline**	**D4**	**C8**
No augmentation			
300 epochs			
Best accuracy	**81.90**	**85.50**	**78.20**
Epochs to stability	**266**	**262**	**184**
Time to stability	**1.33**	**12.80**	**9.04**
2400 epochs			
Best accuracy	**87.00**	**92.40**	**92.50**
Epochs to stability	**1478**	**1474**	**1667**
Time to stability	**7.57**	**73.67**	**84.01**
Augmented training			
300 epochs			
Best accuracy	**91.10**	**91.00**	**92.00**
Epochs to stability	**145**	**253.4**	**275**
Time to stability	**4.28**	**75.09**	**81.28**

5.4.1 Paper II

In Paper II, the convergence speed was measured by calculating the Dice score on the test set for the initial ten epochs during training. It can be seen in Table 5.2 that the initial Dice score is higher for the equivariant networks compared to the baseline networks for varying amounts of data and types of augmentations. As an example, the equivariant network on the BBBC038 dataset without data augmentations has an initial Dice score of 0.52 vs the baseline's 0.44.

5.4.2 Papers III and IV

To measure convergence time in the classification tasks in Papers III and IV, the time until stability metric was defined as the time until 95 % of the top accuracy was achieved for the first time during training. The results for Paper III can be seen in Table 5.3. It can be seen that the equivariant network using no data augmentation converged in 2145 seconds. This was roughly half the time of the baseline network trained on data augmented data, which took 4317 seconds to reach the stable accuracy.

The results for Paper IV can be seen in Table 5.4 and in Table 5.5. It can be seen that across a range of conditions both the baseline and equivariant networks converge in a similar number of epochs. However, as the equivariant networks demand more computations than CNNs when the number of channels are the same, the equivariant networks take longer time to converge. This is in contrast to the results from Paper III. One difference between Papers III and IV is that batch normalization was not used in Paper III to reduce the memory requirements. When batch normalization was turned off in Paper IV, the baseline networks failed to converge at all while the equivariant networks achieved significantly lower accuracies [13]. This shows that designing networks for extended equivariance and enabling batch normalization improve data efficiency during training through different means.

5.5 Scaling Laws

The success of deep learning is largely unexplained. While there is no theory that contradicts the empirical results, there are also no theoretical guarantees. This contrasts with e.g. the simplex method in linear programming [73], where theoretical proofs of finding the global optimum exist if the problem is modelled correctly. Computational learning theory [69] has explained some machine learning results, but more theory is needed.

One way forward to building a theory of deep learning is to collect empirical results of model behaviours. For instance, it has been observed that test errors versus model sizes tend to follow a double descent curve [75]. Classical statistics would expect that after a minimum has been found, the error would increase with the number of parameters. However, in deep learning applications, the error increases, hits a peak, and then starts to decrease again, reaching even lower errors than in the previous valley. This methodology is not only useful for theoretical developments. State of the art large language models use similar procedures for determining how to optimize the models, as it is not feasible to do extensive hyperparameter tuning and model testing due to computational and time costs [78]. Instead, the model behavior is predicted using similar curves, tuning is made in the smaller model regions, and then the model is scaled to its production size.

By similar experiments, the behaviour of models have been examined when instead varying the amount of data [47]. It has been seen that the error y versus the amount of data x follows a power law:

$$y = ax^k \tag{5.1}$$

where a and k are constants. Taking the logarithm and rearranging yields:

$$log(y) = log(a) + k * log(x) \tag{5.2}$$

This can be modelled with linear regression. As the slope k also is the exponent in the power law, this is a practical way of determining how fast the error is reduced when adding more training data. It has been observed that different CNN models do not affect the slope, instead shifting the curve up or down. Interestingly, this behaviour has not been seen with equivariant networks in modelling of molecular dynamics [5]. Instead, the equivariant networks exhibit a steeper slope, hinting at improved data efficiency.

5.5.1 Papers III and IV

In Papers III and IV, the scaling behaviours for varying amounts of unaugmented training samples were investigated. The TEM virus dataset was used, and baseline and equivariant versions of the VGG16 classifiers were deployed, as described in Section 3.6. The results can be seen in Figure 5.4 for Paper III and Figure 5.5 for Paper IV. In the first case, the slope of the equivariant model using the p4 symmetry group was -0.43, and for the baseline it was -0.26. In the second case, the slope of the equivariant model using the D4 symmetry group was -1.05, and for the baseline it was -0.75. In both cases, the slopes were steeper for the equivariant models. This means that the equivariant networks learn faster when adding more training data, i.e. they are more data efficient. Also in Paper IV, the same experiments were carried out using a custom model. Here, the slopes of both the baseline and equivariant models were -0.66. It seems that the difference in data efficiency is architecture dependent.

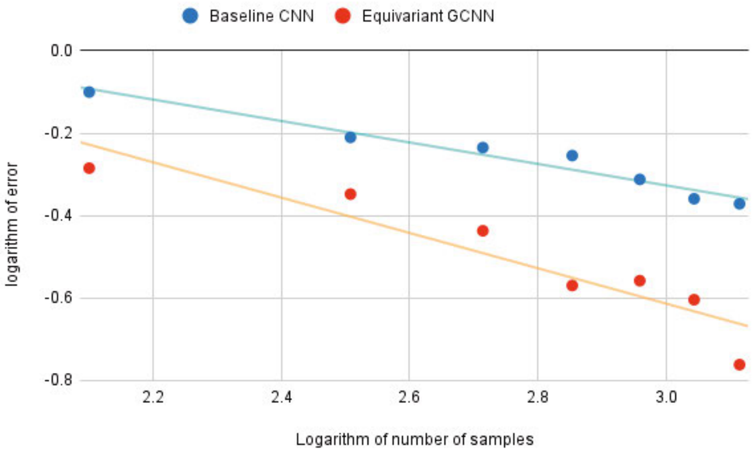


Figure 5.4. Log-log chart of test set error vs number of unaugmented training samples for Paper III. Here the VGG16 was used to classify TEM viruses. The equivariant network using the p4 symmetry group (lower line) has a steeper slope than the baseline, indicating higher data efficiency.

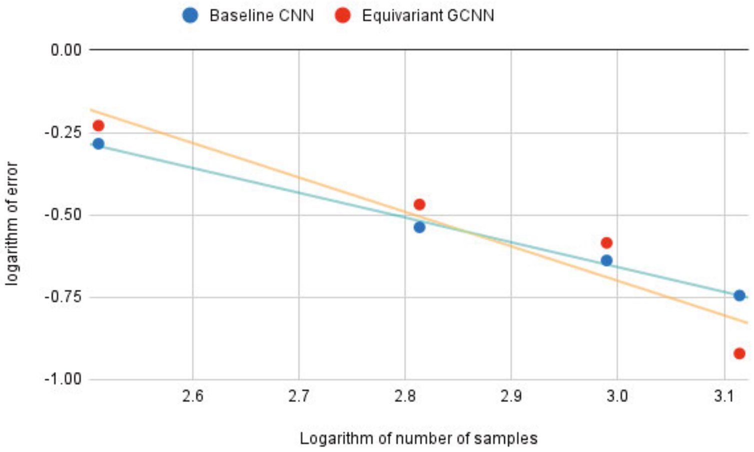


Figure 5.5. Log-log chart of test set error vs number of unaugmented training samples for Paper IV. Here the VGG16 was used to classify TEM viruses. The equivariant network using the D4 symmetry group (upper line in the extreme left of the chart) has a steeper slope than the baseline, indicating higher data efficiency.

6. Closing Remarks

This chapter begins by summarizing the papers of the thesis. This is followed by the conclusions from the research, as well as suggested further research. The chapter and thesis ends by the author's outlook on the field of AI.

6.1 Summary of papers

In Paper I, the VGG16 network was used to classify malignancy in images of cells from patients' oral cavities. It was shown that an equivariant neural network using the p4 symmetry group resulted in higher classification accuracy on the test set than a baseline CNN. It was also shown that data augmentation by 90 degree rotations could be omitted.

In Paper II, the U-Net architecture was used to semantically segment images from the BBBC038 dataset which pictures cells in varying conditions. When the network was modified with an extra head and a discriminative loss function, instance segmentation could be performed as well. It was proven that changing the network to be equivariant to transformations of the p4 symmetry group resulted in rotation-equivariant semantic instance segmentation. Test set Dice scores were similar regardless of using the rotation-equivariant or the baseline network with or without data augmentation. Still, the rotation-equivariant networks provided faster convergence during training.

In Paper III, the VGG16 network was used to classify virus samples in TEM images. It was shown that an equivariant neural network using the p4 symmetry group instead of the baseline CNN resulted in higher classification accuracy. It was also shown that data augmentation by multiples of 90 degree rotations could be omitted with little cost to classification accuracy. Furthermore, the rotation-equivariant network without data augmentation converged in around half the time of the baseline with data augmentation. Finally, it was shown in a log-log plot that the rotation-equivariant network was more data efficient than the baseline when training on varying amounts of unaugmented data.

In Paper IV, the VGG16 network and a smaller custom network were used to classify the same virus samples as in Paper III. In contrast to Paper III, the batch normalization optimization was used as well. Furthermore, variants of the networks were designed to be equivariant to the C8 and D4 symmetry groups, respectively. It was shown that results in terms of accuracy and convergence speed were similar when training for 300 epochs no matter the choice of architecture, symmetry group or augmentation strategy. However, when training for 2400 epochs on unaugmented data, the equivariant networks outperformed the baseline networks significantly. Also it was shown in a log-log plot that the equivariant VGG16 network using the D4 symmetry group was more data efficient than the baseline when training on varying amounts of unaugmented data. When the custom architecture was used, both the baseline and equivariant networks exhibited similar slopes in a log-log plot.

6.2 Conclusions

The aims of this thesis as stated in Section 1.2 are to reduce the issues associated with the annotated training data in biomedical image analysis using equivariant neural networks in combination with empirical deep learning and predictable scaling. These issues are lack of data (Aim 1), overfitting (Aim 2) and computational costs (Aim 3) such as convergence time during training.

The issues were addressed in the papers using equivariant neural networks. Both classification and semantic instance segmentation models were investigated along with several symmetry groups. Different problem settings, such as oral cancer classification and semantic instance segmentation of cells, were explored. Furthermore, predictable scaling methods were employed to aid in the training process.

A number of conclusions can be drawn from the results. Data augmentation, a technique for increasing the amount of training data, can be reduced if the component transformations of the symmetry group match the augmentations (Aim 1). Overfitting, which was measured by a novel overfitting metric, is lessened in many cases (Aim 2). One exception to this can be seen with the semantic instance segmentation networks in Paper II. In some cases, such as in Paper IV, higher accuracy on the test set and lower overfitting are observed for equivariant networks only after training for much longer times than what is considered as standard. In most cases, the convergence to a stable accuracy during training is faster for the equivariant networks (Aim 3).

There seems to be benefits of using equivariant neural networks in biomedical image analysis owing to the extended symmetries usually found in the associated datasets, such as rotational symmetry. However, for new problem settings it is unclear what these benefits will be in practice, such as lower overfitting or faster convergence. For medical diagnostics, equivariant networks could be used either for improved accuracy in production settings. Equivariant networks could also be used for development and fine-tuning in the training process to reduce time consumption.

6.3 Further research

Future research into reducing overfitting and improving data efficiency using equivariant neural networks could focus on ablation studies, similar to how performance depended on batch normalization in Paper IV. Further varying datasets, architectures, symmetry groups and hyperparameters could provide further insight into how general the observed effects are. The equivariant networks could also be designed with different symmetry groups in different layers to exploit underlying symmetries in varying scales.

In applied research, e.g. oral cancer classification, using equivariant networks instead of data augmentation techniques could be explored further. In theoretical research, more empirical results should be collected. Eventually and hopefully, theorems could be developed to explain the results, such as the different slopes of equivariant networks compared to CNNs in log-log plots of test set errors versus varying amounts of training data.

6.4 Future Outlook of AI

In the last years, AI methods, alone or in combination with other methods, have largely superseded classical methods in biomedical image analysis for state of the art accuracy. While classical methods can come with a number of advantages, such as stronger explainability and simpler implementations, one of the main strengths of AI methods is flexibility. Neural networks can be retrained to learn and reproduce patterns, as long as sufficient data and computational power are available. This is part of why they have exploded in use in many other fields, like natural language processing or molecular modelling. While neural networks as implemented today are very different from biological neural networks, they are more similar than classical methods. Neural networks are adaptable and closer to how we learn to make new predictions from the changing conditions around us.

We cannot know today how AI systems and our understanding of them will develop, but as computational hardware performance and the amount of available data grow and spread, so will probably the the capacities of AI. Those with the means to control and exploit the data and computational infrastructure have significant power. This is one of the biggest dangers ahead. Therefore efforts should focus on discussing and regulating access to resources and systems, and what we do with them. Understanding AI system capabilities and limitations, spreading this knowledge, ethical discussions and global legislation are critical for our future society.

References

[1] Anthony K Akobeng. Understanding diagnostic tests 1: sensitivity, specificity and predictive values. *Acta paediatrica*, 96(3):338–341, 2007.

[2] John Armour, Richard Parnham, and Mari Sako. Augmented Lawyering. *SSRN Electronic Journal*, 01 2020.

[3] MA Armstrong. The Euclidean Group. In *Groups and Symmetry*, pages 136–144. Springer, 1988.

[4] Warren Ashby. Teleology and deontology in ethics. *The Journal of Philosophy*, 47(26):765–773, 1950.

[5] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, May 2022.

[6] Samuel Bendett. Russian Unmanned Vehicle Developments: Syria and Beyond. *Improvisation and Adaptablity in the Russian Military*, pages 38–47, 2020.

[7] Karl Bengtsson Bernander. A Method for Detecting Resident Space Objects and Orbit Determination Based on Star Trackers and Image Analysis, 2014.

[8] Karl Bengtsson Bernander, Kenneth Gustavsson, Bettina Selig, Ida-Maria Sintorn, and Cris L. Luengo Hendriks. Improving the stochastic watershed. *Pattern Recognition Letters*, 34(9):993–1000, 2013.

[9] Jeroen Bertels, Tom Eelbode, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B. Blaschko. Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory and Practice. In Dinggang Shen et al., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 92–100, Cham, 2019. Springer International Publishing.

[10] Serge Beucher. The watershed transformation applied to image segmentation. *Scanning microscopy*, 1992(6):28, 1992.

[11] Georgina Born, Jeremy Morris, Fernando Diaz, and Ashton Anderson. Artificial intelligence, music recommendation, and the curation of culture. Schwartz Reisman Institute for Technology and Society, 2021.

[12] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic Instance Segmentation with a Discriminative Loss Function. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2017)*, 2017.

[13] Karl Bylander. *More efficient training using equivariant neural networks*. Number 23004 in UPTEC X. 2023.

[14] Juan C. Caicedo, Allen Goodman, Kyle W. Karhohs, Beth A. Cimini, Jeanelle Ackerman, Marzieh Haghighi, Cherkeng Heng, Tim Becker, Minh Doan,

Claire McQuin, Mohammad Hossein Rohban, Shantanu Singh, and Anne E. Carpenter. Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature Methods*, 16:1247–1253, 2019.

[15] Per Carlbring, Heather Hadjistavropoulos, Annet Kleiboer, and Gerhard Andersson. A new era in Internet interventions: The advent of Chat-GPT and AI-assisted therapist guidance. *Internet Interventions*, 32, 2023.

[16] Joseph Carlsmith. Is Power-Seeking AI an Existential Risk? *arXiv preprint arXiv:2206.13353*, 2022.

[17] Ismail Celik. Exploring the Determinants of Artificial Intelligence (AI) Literacy: Digital Divide, Computational Thinking, Cognitive Absorption. *Telematics and Informatics*, 83:102026, 2023.

[18] Gabriele Cesa, Leon Lang, and Maurice Weiler. A Program to Build E(N)-Equivariant Steerable CNNs. In *International Conference on Learning Representations*, 2022.

[19] O. Chapelle, B. Scholkopf, and A. Zien, Eds. Semi-Supervised Learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[20] Benjamin Chidester, Tianming Zhou, Minh N Do, and Jian Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537, 07 2019.

[21] Taco Cohen and Max Welling. Group Equivariant Convolutional Networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR.

[22] Lauren Collier-Spruel, Ashley Hawkins, Eranda Jayawickreme, William Fleeson, and R. Michael Furr. Relativism or tolerance? Defining, assessing, connecting, and distinguishing two moral personality features with prominent roles in modern societies. *Journal of personality*, 87(6):1170–1188, 2019.

[23] Cryteria. Creative Commons Attribution 3.0 Unported. `https://commons.wikimedia.org/wiki/File:HAL9000.svg`. Accessed: 2023-10-30.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[25] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting Cyclic Symmetry in Convolutional Neural Networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1889–1898, New York, New York, USA, 20–22 Jun 2016. PMLR.

[26] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.

[27] Stable Diffusion. Public domain. `https://upload.wikimedia.org/wikipedia/commons/d/d3/Astronaut_Riding_a_Horse_%28SDXL%29.jpg`. Accessed: 2023-10-30.

[28] Pedro Domingos. A unified bias-variance decomposition. In *Proceedings of*

17th international conference on machine learning, pages 231–238. Morgan Kaufmann Stanford, 2000.

[29] F. D. Drake. Project Ozma. *Physics Today*, 14(4):40–46, 04 1961.

[30] Tom Eelbode, Pieter Sinonquel, Frederik Maes, and Raf Bisschops. Pitfalls in training and validation of deep learning systems. *Best Practice and Research Clinical Gastroenterology*, 52-53:101712, 2021. Artificial intelligence in GI-endoscopy.

[31] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25, 01 2019.

[32] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25, 01 2019.

[33] John Jumper et al. Creative Commons Attribution 4.0 International. `https://commons.wikimedia.org/wiki/File:AlphaFold_2.png`. Accessed: 2023-10-30.

[34] Steven Feldstein. *The global expansion of AI surveillance*, volume 17. Carnegie Endowment for International Peace Washington, DC, 2019.

[35] Olena Filchakova, Dina Dossym, Aisha Ilyas, Tamila Kuanysheva, Altynay Abdizhamil, and Rostislav Bukasov. Review of COVID-19 testing and diagnostic methods. *Talanta*, 244:123409, 2022.

[36] Gustav Forslid, Håkan Wieslander, Ewert Bengtsson, Carolina Wählby, Jan-Michael Hirsch, Christina Runow Stark, and Sajith Kecheril Sadanandan. Deep Convolutional Neural Networks for Detecting Cellular Changes Due to Malignancy. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 82–89, 2017.

[37] Sara Gandini, Edoardo Botteri, Simona Iodice, Mathieu Boniol, Albert B Lowenfels, Patrick Maisonneuve, and Peter Boyle. Tobacco smoking and cancer: a meta-analysis. *International journal of cancer*, 122(1):155–164, 2008.

[38] Maura L Gillison, Anil K Chaturvedi, William F Anderson, and Carole Fakhry. Epidemiology of human papillomavirus–positive head and neck squamous cell carcinoma. *Journal of clinical oncology*, 33(29):3235, 2015.

[39] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., USA, 2006.

[40] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[41] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[42] Gorlapraveen. Creative Commons Attribution-ShareAlike 4.0 International. `https://commons.wikimedia.org/wiki/File:VGG16.png`. Accessed: 2023-10-30.

[43] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance

segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.

[44] Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A Survey on Deep Reinforcement Learning Algorithms for Robotic Manipulation. *Sensors*, 23(7), 2023.

[45] D.O. Hebb. *The organization of behavior; a neuropsychological theory*. Wiley, 1949.

[46] David Heckerman. *A Tutorial on Learning with Bayesian Networks*, pages 33–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[47] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv*, 2017.

[48] Geoffrey E. Hinton and Terrence J. Sejnowski. Unsupervised learning: foundations of neural computation. In *Unsupervised learning: foundations of neural computation*, 1999.

[49] Hugovanmeijeren. Creative Commons Attribution 4.0 International. `https://en.wikipedia.org/wiki/File:Cern_datacenter.jpg`. Accessed: 2023-10-30.

[50] Nanna Inie, Jeanette Falk, and Steve Tanimoto. Designing Participatory AI: Creative Professionals' Worries and Expectations about Generative AI. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2023.

[51] John M. Jumper et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583 – 589, 2021.

[52] Joohee Kim and Il Im. Anthropomorphic response: Understanding interactions between humans and artificial intelligence agents. *Computers in Human Behavior*, 139:107512, 2023.

[53] Sarah Kreps, R Miles McCain, and Miles Brundage. All the news that's fit to fabricate: AI-generated text as a tool of media misinformation. *Journal of experimental political science*, 9(1):104–117, 2022.

[54] Nikolaus Kriegeskorte, William Simmons, Patrick Bellgowan, and Chris Baker. Circular analysis in systems neuroscience: The dangers of double dipping. *Nature neuroscience*, 12:535–40, 05 2009.

[55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[56] Gustaf Kylberg. *Automatic Virus Identification using TEM: Image Segmentation and Texture Analysis*. Number 1122 in Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. 2014.

[57] Shane Legg, Marcus Hutter, et al. A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and applications*, 157:17, 2007.

[58] Kendra Leigh and Yorgo Modis. Imaging and visualizing SARS-CoV-2 in a new era for structural biology. *Interface Focus*, 11, 12 2021.

[59] Teng Long, Qi Gao, Lili Xu, and Zhangbing Zhou. A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions.

Computers & Security, page 102847, 2022.

[60] Jiahao Lu, Nataša Sladoje, Christina Runow Stark, Eva Darai Ramqvist, Jan-Michaél Hirsch, and Joakim Lindblad. A deep learning based pipeline for efficient oral cancer screening on whole slide images. In *International Conference on Image Analysis and Recognition*, pages 249–261. Springer, 2020.

[61] Clément Le Ludec, Maxime Cornet, and Antonio A Casilli. The problem with annotation. Human labour and outsourcing between France and Madagascar. *Big Data & Society*, 10(2):20539517231188723, 2023.

[62] Vijayalakshmi G.V. Mahesh, Alex Noel Joseph Raj, and Zhun Fan. Invariant moments based convolutional neural networks for image analysis. *International Journal of Computational Intelligence Systems*, 10:936–950, 2017.

[63] Robert E Marx, Diane Stern, et al. *Oral and maxillofacial pathology: a rationale for diagnosis and treatment*. Quintessence Publishing Company Hanover Park, 2012.

[64] Damian J. Matuszewski and Ida-Maria Sintorn. TEM virus images: Benchmark dataset and deep learning classification. *Computer Methods and Programs in Biomedicine*, 209:106318, 2021.

[65] Ida-Maria Matuszewski, Damian; Sintorn. TEM virus dataset. *Mendeley Data* `https://data.mendeley.com/datasets/x4dwwfwtw3`, 2021.

[66] Michael Milford, Sam Anthony, and Walter Scheirer. Self-Driving Vehicles: Key Technical Challenges and Progress Off the Road. *IEEE Potentials*, 39(1):37–45, 2020.

[67] MIT. MIT License. `https://github.com/aleju/imgaug/blob/master/LICENSE`. Accessed: 2023-11-07.

[68] Sparsh Mittal and Shraiysh Vaishay. A survey of techniques for optimizing deep learning on GPUs. *Journal of Systems Architecture*, 99:101635, 2019.

[69] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[70] Hans Moravec. *Mind Children: The Future of Robot and Human Intelligence*. Harvard University Press, USA, 1988.

[71] Dena F. Mujtaba and Nihar R. Mahapatra. Ethical Considerations in AI-Based Recruitment. In *2019 IEEE International Symposium on Technology and Society (ISTAS)*, pages 1–7, 2019.

[72] Neelesh Mungoli. Scalable, Distributed AI Frameworks: Leveraging Cloud Computing for Enhanced Deep Learning Performance and Efficiency. *arXiv*, 2023.

[73] K.G. Murty. *Linear Programming*. Wiley, 1984.

[74] Taha Abdelhalim Nakabi and Pekka Toivanen. Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable Energy, Grids and Networks*, 25:100413, 2021.

[75] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: where bigger models and more data hurt*. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, dec 2021.

[76] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M. Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022.

[77] João D. Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S. Cardoso. Spiking Neural Networks: A Survey. *IEEE Access*, 10:60738–60764, 2022.

[78] OpenAI. GPT-4 Technical Report, 2023.

[79] World Health Organization. *World mental health report: transforming mental health for all.* Geneva: World Health Organization, 2022.

[80] Harald Øverby and Jan Arild Audestad. *Big Data Economics*, pages 305–322. Springer International Publishing, Cham, 2021.

[81] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon Emissions and Large Neural Network Training, 2021.

[82] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11), 2021.

[83] Judea. Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Penguin Books Limited, 2018.

[84] Filippo Pesapane, Marina Codari, and Francesco Sardanelli. Artificial intelligence in medical imaging: threat or opportunity? Radiologists again at the forefront of innovation in medicine. *European Radiology Experimental*, 2, 10 2018.

[85] Zhen-Hu Ren, Chuan-Yu Hu, Hai-Rong He, Yuan-Jie Li, and Jun Lyu. Global and regional burdens of oral cancer from 1990 to 2017: Results from the global burden of disease study. *Cancer Communications*, 40(2-3):81–92, 2020.

[86] Robert Riener, Luca Rabezzana, and Yves Zimmermann. Do robots outperform humans in human-centered domains? *Frontiers in Robotics and AI*, 10, 2023.

[87] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[88] David W. Romero, Erik J. Bekkers, Jakub M. Tomczak, and Mark Hoogendoorn. Attentive Group Equivariant Convolutional Networks. In *ICML*, pages 8188–8199, 2020.

[89] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015)*, pages 234–241. Springer International Publishing, 2015.

[90] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[91] Robert T Sataloff. *Sataloff's Comprehensive Textbook of Otolaryngology: Head & Neck Surgery: Pediatric Otolaryngology*, volume 6. JP Medical Ltd, 2015.

[92] Paul Scharre. *Army of None: Autonomous Weapons and the Future of War*. Tantor Audio, Old Saybrook, CT, 2018.

[93] Alex Sherstinsky. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[94] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[95] David Silver et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature*, 529(7587):484–489, jan 2016.

[96] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*. OpenReview, 2015.

[97] David Sumpter. *Outnumbered: From Facebook and Google to Fake News and Filter-bubbles - the Algorithms that Control Our Lives*. Bloomsbury sigma series. Bloomsbury Sigma, 2018.

[98] Olga Sunneborn Gudnadottir, Daniel Gedon, Colin Desmarais, Karl Bengtsson Bernander, Raazesh Sainudiin, and Rebeca Gonzalez Suarez. Distributed training and scalability for the particle clustering method UCluster. *EPJ Web Conf.*, 251:02054, 2021.

[99] Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1), 2019.

[100] Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2010.

[101] Robert Tucci. Introduction to Judea Pearl's Do-Calculus. *arXiv*, 04 2013.

[102] Scientific United Nations Educational, Cultural Organization, OECD, and Inter-American Development Bank. *The Effects of AI on the Working Lives of Women*. 2022.

[103] Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4199–4210. Curran Associates, Inc., 2020.

[104] V. Vapnik. Principles of Risk Minimization for Learning Theory. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 831–838. Morgan-Kaufmann, 1992.

[105] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[106] Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[107] Maurice Weiler, Fred Hamprecht, and Martin Storath. Learning Steerable Filters for Rotation Equivariant CNNs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 849–858, June 2018.

[108] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable AI: A brief survey on history, research areas, approaches and challenges. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference*, pages 563–574. Springer, 2019.

[109] Zhongheng Zhang. Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, 4:218–218, 06 2016.

Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 2352

Editor: The Dean of the Faculty of Science and Technology

Distribution: publications.uu.se
urn:nbn:se:uu:diva-519611

ACTA UNIVERSITATIS
UPSALIENSIS
2024