On the Confluence of Machine Learning and Model-Based Energy Minimization Methods for Computer Vision

DISSERTATION
zur Erlangung des Grades eines Doktors
der Naturwissenschaften

vorgelegt von

M.Sc. Hannah Dröge

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2023

Betreuer und erster Gutachter
Prof. Dr. Michael Möller
Universität Siegen

Zweiter Gutachter
Prof. Dr. Florian Bernard
Rheinische Friedrich-Wilhelms-Universität Bonn

Tag der mündlichen Prüfung
17.04.2024

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer, nicht angegebener Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt. Es wurden keine Dienste eines Promotionsvermittlungsinstituts oder einer ähnlichen Organisation in Anspruch genommen.

Ort, Datum

Unterschrift

Abstract

Deep learning has achieved great success in the field of computer vision across a wide range of applications. However, learning-based methods still have several limitations, particularly in terms of interpretability and guarantees. In contrast, traditional model-based computer vision techniques, built on explicit models that are derived from our understanding of the specific problem domain, offer a different and interpretable approach on addressing these challenges.

In this work, we analyze and further develop hybrid approaches that combine model-based and learning-based methods in computer vision, introducing four different approaches. We analyze the capabilities of both model-based and learning-based methods, discuss the value of deep learning for underdetermined problems, present an extended approach to incorporate learning directly into the optimization process, and address problems where the challenge lies in the intrinsic formulation of the problem itself. Thereby we deal with different application areas in the field of computer vision. We start with studying segmentation problems on a single image, given only user input in the form of drawn scribbles in the color images, and analyze the performance of learning-based methods to incorporate the scribble information, compared to a cleverly designed model-based approach. Further, we address reconstruction problems, focusing on underdetermined computed tomography reconstructions of lung scans. We integrate a learning-based regularizer into the reconstruction process and explore the space of possible data-consistent reconstructions corresponding to various degrees of pathological malignancy. Also, to integrate neural networks into model-based approaches, we build on recent studies, which aim to learn iterative descent directions for minimizing model-based cost functions. By applying Moreau-Yosida regularization, we introduce a method that avoids the need for differentiability. This is a significant improvement over previous approaches, that are limited to continuously differentiable cost functions. For solving matching and assignment problems, we introduce an approach that approximates large permutation matrices and reduces computation and memory costs by non-linear low-rank matrix factorization. We experimentally demonstrate its performance across various model- and learning-based methods.

Zusammenfassung

Deep Learning hat im Bereich der “Computer Vision” für eine Vielzahl von Anwendungen große Erfolge erzielt. Allerdings weisen lernbasierte Methoden noch einige Einschränkungen auf, insbesondere in Bezug auf Interpretierbarkeit und Garantien. Im Gegensatz dazu bieten traditionelle, modellbasierte Techniken der “Computer Vision”, die auf expliziten Modellen basieren und aus unserem Verständnis des spezifischen Problembereichs abgeleitet sind, einen anderen und interpretierbaren Ansatz, um diese Herausforderungen anzugehen.

In dieser Arbeit analysieren und entwickeln wir hybride Ansätze weiter, die modellbasierte und lernbasierte Computer-Vision-Methoden kombinieren und stellen hierzu vier verschiedene Ansätze vor. Wir analysieren die Fähigkeiten sowohl modellbasierter als auch lernbasierter Methoden, diskutieren den Nutzen von Deep Learning bei unterbestimmten Problemen, präsentieren einen erweiterten Ansatz zur direkten Integration des Lernens in den Optimierungsprozess und befassen uns mit Problemen, in denen die Herausforderung in der intrinsischen Formulierung des Problems selber liegt. Dabei beschäftigen wir uns mit verschiedenen Anwendungsbereichen im Bereich der “Computer Vision”. Wir beginnen mit der Untersuchung von Segmentierungsproblemen auf einzelnen Bildern, die ausschließlich Benutzereingaben in Form von auf den Farbbildern gezeichneten Markierungen erhalten, und vergleichen die Leistung von lernbasierten Methoden zur Einbeziehung der Markierungen mit einem durchdachten modellbasierten Ansatz. Außerdem befassen wir uns mit Rekonstruktionsproblemen, insbesondere mit unterbestimmten Computertomographie-Rekonstruktionen von Lungenscans. Wir integrieren einen lernbasierten Regularisierer in den Rekonstruktionsprozess und erkunden den Raum möglicher, datenkonsistenter Rekonstruktionen, die verschiedenen Graden von pathologischer Bösartigkeit entsprechen. Um neuronale Netze in modellbasierte Ansätze zu integrieren, stützen wir uns auf aktuelle Studien, die die iterativen Abstiegsrichtungen zum Minimieren modellbasierter Kostenfunktionen erlernen. Durch die Anwendung der Moreau-Yosida-Regularisierung führen wir eine Methode ein, die die Notwendigkeit der Differenzierbarkeit umgeht. Dies ist ein bedeutender Fortschritt gegenüber früheren Ansätzen, die auf stetig differenzierbare Kostenfunktionen beschränkt sind. Zur Lösung von Matching- und Zuordnungsproblemen stellen wir einen Ansatz vor, der große Permutationsmatrizen approximiert und die Rechen- und Speicherkosten durch nichtlineare Matrixfaktorisierung mit niedrigem Rang reduziert. Wir demonstrieren experimentell die Leistungsfähigkeit dieses Ansatzes in verschiedenen modell- und lernbasierten Methoden.

Acknowledgments

I would like to take this moment to express my gratitude and appreciation to all who have accompanied and supported me on my journey to the Ph.D.

A sincere thanks goes to my family who always believed in me and supported me in all life decisions. A special thanks goes to Markus, who was not only a great supporter but was also there for me in the most difficult times. Thank you for always believing in me, even when I didn't anymore myself. You made this journey easier.

I would like to thank Michael for his guidance in my academic journey and for providing many opportunities. I thank you for the chance to do my Ph.D. in your group and am grateful for the academic support both before and during my time there. Thank you for being understanding in many situations and doing your best to make all your Ph.D. students feel comfortable in your group. Also, many thanks to Sarah, for her skillful problem solving and constant helpfulness.

I would like to thank all my colleagues for our great conversations during coffee break. To all my co-authors, the inspiring discussions with you have been invaluable to me and I have learned a lot through our collaboration. Special thanks go to Zorah, whose composure turned challenging deadlines into manageable moments, and to Yuval, whose expertise provided invaluable support throughout our joint efforts. And finally, a heartfelt thank you to Hartmut. Our collaboration has developed into a valued friendship. Thank you for standing by me as a reliable colleague and true friend.

I am very thankful to each one of you and look forward to carrying the friendships made and lessons learned into the next chapter of my professional and personal development.

Table of Contents

Part I

Introduction and Background

CHAPTER 1

Introduction

1.1 Motivation

The field of computer vision deals with the interpretation and processing of visual data by machines and has become an important component of many applications that influence our daily lives. From facial recognition in securing devices, over augmented reality applications, to autonomous vehicles that promise safer and more efficient transportation, the practical impact is obvious and demonstrates the importance of computer vision. The huge increase in available visual data over the past decade has led to the rise of deep learning in computer vision and has redefined the boundaries of what machines can perceive, understand, and recreate from visual data. This covers especially challenges that require a semantic understanding, where before the deep learning era, computer vision faced significant difficulties in tasks like object recognition, semantic image segmentation, and scene understanding.

Many breakthroughs in deep learning made these possible, where the introduction of the convolutional neural network (CNN) was a fundamental step towards highly accurate image interpretation and classification, enabling the recognition and categorization of a variety of objects. For example, in autonomous driving semantic interpretability enables a machine to interpret traffic signs, recognize obstacles, and therefore avoid accidents by making quick decisions. It can be useful in public security by detecting suspicious activities or face recognition. Also the medical field benefits from deep learning, for early disease detection or the support of diagnostic procedures. Especially in medical imaging, transfer learning techniques have improved the development of diagnostic models, possibly even with limited labeled medical data. In addition, many other domains benefit from deep learning. In the artistic domain, Generative Adversarial Networks (GANs) and diffusion models have shown the ability to create lifelike artworks, while machine translation and interpretation of visual content into natural language have been improved by Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks and by transformers.

While deep learning has many advantages, it comes with its limitations and drawbacks. Deep learning models need in general a huge amount of data to train effectively. The lack of *sufficient data* can lead to poor generalization in real-world scenarios. Especially in the

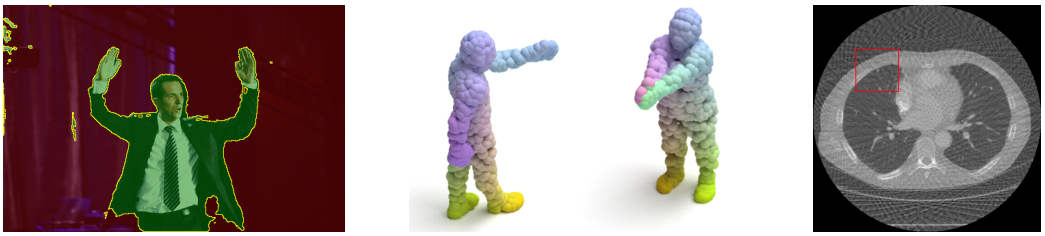


Figure 1.1: Computer vision applications that benefit from deep learning and its combination with model-based methods. (Left) Segmentation of an image using semantic information, highlighting distinct objects and regions, from [88]. (Middle) Shape matching using deep learning techniques, from [87]. (Right) Computed Tomography scan image with cancer classification in the highlighted area, from [86].

medical area there often does not exist enough labeled medical data, while the prevention of poor or wrong outcomes is of crucial importance. Further, large deep learning models, combined with memory-intensive data, result in *computationally demanding* training and application processes. Training these models is not only expensive due to the need for high-end hardware but also raises environmental concerns due to the high power consumption associated with long, computationally intensive training. The composition of training datasets plays a huge role in the ability of deep learning, as there is the risk of *maintaining biases* in the training data, which can lead to incorrect or discriminatory predictions [299]. For example, if traffic cameras with deep learning capabilities work with a model that is only trained on data of good weather conditions, it might fail if it is raining. Besides, deep learning models often exhibit a *lack of interpretability*, making it challenging to understand or trust their predictions, which is especially important in critical applications like healthcare, finance, or autonomous driving. Additionally, deep learning models are sensitive to *adversarial attacks*, where slight perturbations in the input can lead to drastical different outputs. In scenarios where self-driving cars use deep learning for object detection, slight disturbances in road signs by attackers could lead to a misinterpretation of the sign, where a "Stop" sign could be misclassified as a "Yield" sign, leading to a high risk of accidents.

So, despite its great success, deep learning has its limitations, which are also reflected in current events in the field of autonomous driving. According to [312], focusing on more traditional methods rather than artificial intelligence recently caused a German automobile manufacturer to be the first to get authorization for a level 3 autonomous driving system in California and Nevada – ahead of its competitors. This highlights the value of traditional computer vision techniques, that formulate explicit models, derived from our understanding of the specific problem domain, and provide a different, interpretable approach to solving these types of problems. While many good deep learning methods for most problems exist, the integration of the underlying structure and constraints of the specific problem can be used to improve the interpretability of deep learning systems.

I have studied different computer vision problems as e.g. shown in Figure 1.1, and the confluence of their model-based formulations and deep learning methods. We have shown that for segmentation scenarios with limited data, classical methods can provide fair segmentation, but the integration of learned semantic information, provides the best results. Further, for reconstructions problems, we were able to successfully combine a learning-based prior with classical model-based reconstruction to offer a solution for dealing with ambiguities in the reconstruction process and extended another hybrid method that incor-

porates deep learning into the optimization of classical methods to a wider range of explicit models. The evaluation of each of these methods has shown the benefits of combining deep learning with explicit models, such as improving results and offering semantic interpretations. In addition, we made progress in addressing memory-intensive assignment and matching problems, which affect both model-based and training-based methods.

I will introduce and discuss the aforementioned topics and methodologies on the confluence of learning- and model-based methods throughout this thesis.

1.2 Outline and Contribution

1.2.1 Thesis Outline

This dissertation has the following organizational structure. The remaining sections of this chapter give an overview of the key contributions to this thesis and are followed by an overview of publications on which I have worked, but that do not fit within the scope of this thesis. Chapter 2 focuses on the mathematical background, that covers the necessary foundation for optimization problems that are discussed through this dissertation. This includes a discussion of inverse problems, energy minimization, the nature of convex and smooth energy functions, an introduction to first-order optimization schemes, and a more detailed introduction to proximal methods and the Moreau-Yosida regularization, which becomes important throughout this essay. Chapter 3 addresses deep learning in imaging by introducing the concept of supervised learning, fundamental models for computer vision and discussing model robustness. In the context of the core topic of this thesis, the interplay between model-based and learning-based methods, Section 3.3 provides a categorization of different types of hybrid model-based and learning-based methods and discusses the underlying idea and related work of each one of them. Section 3.4 deals with the topic of neural networks that provide more than a single output, covering the exploration of the space of possible outcomes for reconstruction tasks. Chapter 4 addresses applications that become important throughout the thesis, namely computed tomography (CT), image segmentation, and assignment problems, including relevant work on model- and learning-based approaches, and introducing their general idea. The main contributions of this thesis are presented from Chapter 5 to Chapter 8, with each chapter containing a different approach on the confluence of learning-based and model-based approaches. Chapter 5 begins with a comparison of model-based and learning-based methods given different levels of information about a segmentation problem. It has been shown that the combination of model-based and learning-based methods works best, leading us to introduce pre-trained information into a model-based CT reconstruction problem in Chapter 6. Chapter 7 also deals with the combination of model- and learning-based methods, but this time, learning is used to guide the problem to the optimal solution while preserving the guarantees given by the models for non-smooth energies. In Chapter 8, we address memory-intensive assignment and matching problems and introduce a new representation of permutation matrices that can be useful for both learning- and model-based approaches. This dissertation concludes with a summary and a discussion on future work in Chapter 9.

1.2.2 Main Contribution

The publications listed below are the four main contributions to this dissertation, focusing on the interplay between model-based and learning-based methods in computer vi-

sion as the main topic. The first three works propose ways to integrate learning elements into model-based approaches for different computer vision applications. The last work addresses a method of memory reduction for both model-based and learning-based approaches.

All authors of the publications listed below have contributed significantly to their realization. Where necessary, I will highlight the contributions of individual authors, focusing exclusively on their role in the practical realization of the described methods.

- [88] Dröge, H., and Möller, M. (2021, September). Learning or Modelling? An Analysis of Single Image Segmentation Based on Scribble Information. In 2021 IEEE International Conference on Image Processing (ICIP) (pp. 2274-2278). IEEE.
- [86] Dröge, H., Bahat, Y., Heide, F., and Möller, M. (2022). Explorable Data Consistent CT Reconstruction. In 33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022. BMVA Press.
- [89] Dröge, H., Möllenhoff, T., and Möller, M. (2022, October). Non-Smooth Energy Dissipating Networks. In 2022 IEEE International Conference on Image Processing (ICIP) (pp. 3281-3285). IEEE.
- [87] Dröge, H., Lähner, Z., Bahat, Y., Martorell, O., Heide, F., and Möller, M. (2023). Kissing to Find a Match: Efficient Low-Rank Permutation Representation. arXiv preprint arXiv:2308.13252. (Accepted at NeurIPS 2023)

Learning and Modelling for Single Image Segmentation [88]

In Chapter 5, we address the challenge of single-image segmentation. As has hardly been done by any deep learning-related work in this direction, we compare the performance of neural networks with a cleverly designed model-based method. Each image in our study comes with prior information in the form of user-drawn scribbles. In particular, we analyze the effects of the information inherent in the scribbles on the segmentation and consider scenarios in which we only use the underlying color information from these scribbles on the one hand and further spatial and semantic information on the other. We discuss how to integrate this information in a weighted manner into the network training process and demonstrate that model-based methods work best when combining color data with semantic features, which are obtained from a pre-trained neural network. Overall, we show that for single image segmentation using user-drawn scribbles, a combination of model- and learning-based approaches benefits from the clever design of the model and access to semantic information from a neural network.

Guided Computed Tomography Reconstruction by a Learned Prior [86]

In Chapter 6, we regularize a model-based reconstruction method for sparse-view CT with a learned prior. Especially problematic for medical data is that in sparse-view CT, the limited data and the resulting underdetermined nature of the problem can lead to reconstruction ambiguities so that the correct interpretation of the reconstruction results cannot be guaranteed. On medical lung scans, we regularize the reconstruction process with a pre-trained classifier network and thus present several reconstructions of the same CT image but with different semantic meanings. In addition, we analyze to what extent the semantic meaning can be modified while maintaining data consistency with the CT scans. To

avoid artifacts in the reconstructions and ensure that their modifications are perceptible, we further discuss technical requirements for the training of the classification network and its input.

Non-Smooth Energy Dissipating Networks [89]

Similar to the approach for guiding CT reconstructions in Chapter 6, we combine model-based and learning-based techniques in Chapter 7. Unlike the previous method, where a learned prior is an explicit regularizer in the model-based minimization, we integrate the neural network into the optimization process itself. In our approach, a neural network predicts suitable descent directions for semi-convex and non-smooth cost functions, ensuring convergence to a minimizer of the cost function, while this is typically not possible in classical deep learning based approaches. In Chapter 7 we shortly introduce a prior work that works on differentiable cost functions and further propose an extension for semi-convex and non-smooth cost functions. To this end, we employ the Moreau-Yosida regularization and demonstrate convergence and applicability on denoising and deblurring tasks.

To the practical realization of this project, Michael Möller contributed significantly through a convergence analysis that ensures the reliability of the method, while my main contribution was adapting our theoretical framework into a practical model for empirical testing, ensuring its applicability.

Efficient Low-Rank Permutation Representation [87]

Challenging tasks are those where the problem lies in their representation itself. Chapter 8 addresses this challenge for permutation matrices, whose representation size scales quadratically with the size of the problem, implying significant computational and memory requirements in larger matching and assignment tasks. We introduce a representation of permutation matrices through low-rank matrix factorization, coupled with a nonlinearity, enabling a stochastic learning approach that enables large permutation matrices and reduces memory requirements. As this representation still allows an accurate prediction of permutation matrices, we demonstrate its applicability and accuracy for model-based linear and quadratic assignment problems, as well as for learning-based shape matching problems, and show a reduction in memory consumption for both scenarios.

Please note the specific contributions to the experiments in this project. Zorah Lähner primarily focused on the experiments involving point cloud alignment on spectral point representation, as well as on quadratic and dense linear assignment problems. My own contributions focused on the experiments related to point cloud alignment, sparse linear assignment problems, and shape matching. In addition, Onofre Martorell supported our work by providing code to load and integrate a memory-intensive dataset for shape matching.

1.2.3 Additional Publications

Throughout my doctoral studies, I participated in publishing additional papers that explore different research areas, distinct from the focus of this thesis. As a result, these publications are not included herein.

- Geiping, J., Bauermeister, H., Dröge, H., and Möller, M. (2020). Inverting gradients-how easy is it to break privacy in federated learning? Advances in Neural Information Processing Systems, 33, 16937-16947.

- Dröge, H., Yuan, B., Llerena, R., Yen, J. T., Möller, M., and Bertozzi, A. L. (2021). Mitral valve segmentation using robust nonnegative matrix factorization. Journal of imaging, 7(10), 213.
- Gandikota, K. V., Chandramouli, P., Dröge, H., and Möller, M. (2023, April). Evaluating Adversarial Robustness of Low dose CT Recovery. In Medical Imaging with Deep Learning.

Inverting gradients-how easy is it to break privacy in federated learning? [107]

In [107], Geiping *et al.* explored the numerical reconstruction of input images from a deep neural network based on their gradient to the network weights. This is relevant for federated learning, where a network is trained on a server, offering potential privacy since only parameter gradients are shared, not the actual data. However, our research raises concerns about the security implications of sharing these gradients. We investigated the extent to which a single image can be reconstructed from the network parameter updates across various network structures and scenarios. Furthermore, we demonstrated the privacy risks, even when the gradient is averaged over a batch of images. In this work, I primarily contributed to the experiments on single image reconstruction from a single gradient.

Mitral valve segmentation using robust nonnegative matrix factorization [90]

In our work in [90], which is based on the results of my Master's thesis, we introduced a method for segmenting non-rigid moving structures within medical videos with a focus on the mitral valve in two-dimensional echocardiographic videos. Unlike previous techniques, our objective was an automatic and unsupervised segmentation method for the mitral valve in two-dimensional echocardiographic videos that can operate with only optional prior information regarding the valve's size. The core idea is to use non-negative matrix factorization to distinguish between static and rigidly moving muscle structures and the non-rigidly moving mitral valve in the videos. To improve the accuracy of the segmentation, we applied additional methods to remove noisy segmentations and adopted a localization approach to refine the results.

Evaluating Adversarial Robustness of Low dose CT Recovery [105]

In this work [105], Gandikota *et al.* proposed an analysis of the robustness against adversarial attacks of multiple deep learning and classical methods for CT reconstruction. Thereby, the reconstruction methods are attacked with different types of approaches, showing the vulnerability of deep learning methods to untargeted attacks and the vulnerability of learning methods and classical methods to local attacks that seek to modify only local regions of the reconstruction. In this work, I contributed by providing code on identifying areas of clinical significance in CT scans and by training a network on CT reconstruction.

CHAPTER 2

Theoretical Foundations

In the domain of computer vision, many tasks aim to reconstruct or extract information from corrupted real world obervations. For instance, tasks such as denoising [257, 37] seek to recover a clear image from a noisy obervation, or deblurring [166, 320] need to retrieve a sharp image or video from a blurred measurement. Beyond these, image segmentation [47, 49] often involves classification tasks within the image itself. These challenges have a common goal of extracting the original or underlying information from corrupted or partial observations and can typically be formulated as inverse problems.

2.1 Inverse Problems

The general formulation of inverse problems involves a measurement or data vector f, and an unknown u, which has to be recovered. In the context of computer vision, f and u could be images, e.g. f corresponding to a blurred image for deblurring tasks. The relationship between f and u is described by a forward operator $A : \mathbb{A} \to \mathbb{B}$, where $\mathbb{A}$ and $\mathbb{B}$ are appropriate function spaces. The forward operator maps the unknown u to the space of measurements,

$$f = A(u) + e, \tag{2.1}$$

where $e \in \mathbb{B}$ represents the presence of noise or measurement errors. The objective in solving an inverse problem is to recover u given f by inverting the effect of the forward operator and unknown e.

In general, it can be difficult or impossible to solve such problems. To gain a better understanding of solvability, there exists the concept of ill-posed/well-posed problems:

Well-Posed Problems Inverse Problems are called well-posed if they meet the three Hadamard criteria, namely,

(a) that there needs to exist a solution to the problem,

(b) the solution needs to be unique,

(c) and the solution is continuously dependent on the input data,

which indicates that the solution is robust against small perturbations in the input data. If one of the criteria is violated, the problem is ill-posed [10]. Establishing well-posedness is essential to ensure that the inverse problem can be reliably solved and that the solution is meaningful and accurate.

In particular, the third criterion of continuous dependence is of great importance when discussing inverse problems, as in the presence of noisy data, the process of solving the inverse problem can significantly amplify this noise. To address this, one needs to find an approximation of the results that considers additional constraints.

2.2 Energy Minimization

A classical way to approach these problems are maximum a-priori (MAP) estimates, which motivate the reconstruction of u as the argument that minimizes a suitable cost function

$$\hat{u} \in \arg\min_u -\log p(f|u) - \log p(u), \tag{2.2}$$

where $p(f|u)$ refers to the conditional probability of observing the measurement f given the true image u and $p(u)$ is the (data-independent) prior probability of u. Such *energy minimization methods* therefore consist of a *data fidelity term*, $-\log(p(f|u))$ that depends on the distribution of the noise, and a regularizer $-\log(p(u))$. A Gaussian noise distribution affected measurement would give the following conditional probability

$$p(f|u) = \frac{1}{\sqrt{2\pi}\lambda} \exp(-\frac{1}{2\lambda^2}\|A(u) - f\|^2), \tag{2.3}$$

whereby λ is the standard deviation of the distribution, and a Gaussian distributed prior probability of u, with $p(u) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}\|u\|^2)$ leads to the very popular example of Tikhonov Regularization

$$\hat{u} \in \arg\min_u \|A(u) - y\|^2 + \alpha\|u\|^2, \tag{2.4}$$

where α is the ratio of λ and σ. Furthermore, there is a large body of literature for solving ill-posed inverse problems with a known data formation process (see (2.1)) via energy minimization methods (see (2.2)) with different regularization terms, including the popular total variation (TV) regularization [257]. The TV regularization term is particularly useful for segmentation and denoising tasks, especially in the context of images, as it effectively preserves edges. It assumes a Laplacian distribution of the gradient of u, $p(u) = \frac{1}{2\beta} \exp(-\frac{1}{\beta}\|Du\|_1)$, where D represents a finite difference matrix. Consequently after applying log, we obtain the following expression for the TV functional, except for a scaling:

$$\mathrm{TV}(u) := \|Du\|_1 \tag{2.5}$$

Further existing regularization terms are, for instance, extensions of TV regularization [33], wavelets [199], shearlets [94], or dictionary learning approaches [3].

There are many well-established optimization techniques for finding the minimizer of an energy minimization method, where the selection of an optimization method depends on the properties of the function under consideration. In the following sections, we will discuss conditions and useful function properties concerning the optimization of an energy function, followed by a short introduction to iterative first order descent methods.

2.3 Optimality Condition

In the following, let us denote the cost function by $E(u)$, s.t. we aim to find the minimizer $\hat{u} \in \arg\min E(u)$. The necessary optimality condition to find the minimizer states that the gradient of the energy at the optimum has to be zero, $\nabla E(\hat{u}) = 0$, applying for energies, differentiable in $\hat{u}$. But considering a function $E : \mathbb{R}^n \to \mathbb{R}$ with points where E is not differentiable, then rather than having a unique gradient, there might exist a set of vectors that satisfy the subgradient property.

Definition 1. *A vector $g \in \mathbb{R}^n$ is subgradient of E at point u if, for all $x \in dom(E)$, the following holds:*

$$E(x) \geq E(u) + g^T(x - u) \tag{2.6}$$

Here $\mathrm{dom}(E) := \{u \in \mathbb{R}^n | E(u) < \infty\}$ ist the domain of E. The inequality in (2.6) ensures that the linear function defined by the subgradient g at u always lies below (or on) the graph of E. ∂E is the set of all subgradients, and leads to the nessecary optimality condition of

$$0 \in \partial E(\hat{u}). \tag{2.7}$$

A desired property for optimization problems is the convexity of the given cost function E, simplifying the search process for the optimum of the cost function. Assuming that a minimizer exists, a local minimum of a convex functions is also a global one, while for strictly convex functions the minimum is unique [29].

Definition 2. *A set C is convex if the line segment between any two points in C lies entirely within the set,*

$$\lambda u + (1 - \lambda)v \in C, \tag{2.8}$$

for all $u, v \in C$ and all $\lambda \in [0, 1]$.

Definition 3. *A function $E : C \to \mathbb{R}$ is convex if for all $\lambda \in [0, 1]$:*

$$E(\lambda u + (1 - \lambda)v) \leq \lambda E(u) + (1 - \lambda)E(v), \tag{2.9}$$

for all u, v in its domain, whereby the domain C is a convex set. A function is called strictly convex if the strict inequality holds for all $\lambda \in]0, 1[$ and $v \neq u$.

For an energy function to converge towards an optimum, it is essential that the minimizer $\hat{u}$ exists at all. For a continuous and convex energy function, coercivity implies the existence of a minimizer.

Definition 4. *A function $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is coercive, if*

$$\lim_{\|u\| \to \infty} E(u) \to \infty. \tag{2.10}$$

So, the coercivity of a function ensures that it does not decrease indefinitely as its input grows and guarantees the existence of a minimizer for continuous, convex and coercive functions E.

2.4 First-Order Optimization

In the following we will first define two function properties (*L-smoothness* and *lower semi-continuity*), that play an important role for the opimization of convex energy functions, followed by the introduction of iterative descent methods with the objective to find the minimizer of an energy formulation. It follows a section about proximal methods and the Moreau-Yosida regularization, which become an important component in Chapter 7.

2.4.1 Function Properties: Smoothness and Semi-Continuity

Lipschitz continuity is an important function property, which intuitively prevents a function from having steep parts. Essentially, it is a measure of the behavior of a function in an interval between two points, where the distance of its function values in the two function points is bounded.

Definition 5. *A function $E : C \subset \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous with a Lipschitz constant L if the following holds true,*

$$\|E(u) - E(v)\|_2 \leq L\|u - v\|_2, \tag{2.11}$$

for all $u, v \in C$.

Moreover, if the gradient of a differentiable function ∇E is Lipschitz continuous, the function is called *L-smooth*. The property of L-smoothness states that gradients of an energy function E do not change too fast and is useful for optimization methods, e.g. it is an important factor for the convergence of the gradient descent.

Definition 6. *A function $E : C \subset \mathbb{R}^n \to \mathbb{R}^m$ is L-smooth with a Lipschitz constant L if the following holds true,*

$$\|\nabla E(u) - \nabla E(v)\|_2 \leq L\|u - v\|_2, \tag{2.12}$$

for all $u, v \in C$.

The property of *lower semi-continuity* in a function is a weaker property than continuity. Continuous functions are always lower semi-continuous, but for discontinuous functions, lower semi-continuity ensures that the point at which the jump happens takes the smallest possible value. Formally, this can be formulated as follows, using the limit inferior [206].

Definition 7. *A function $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is lower semi-continuous, if for all v*

$$\liminf_{u \to v} E(u) \geq E(v), \tag{2.13}$$

2.4.2 Iterative Descent Methods

Iterative descent methods are often used to solve energy minimization methods, as given in (2.2). The objective is to find a minimizer of an energy function E, with update steps of the form

$$u^{k+1} = u^k - \tau d^k, \tag{2.14}$$

where $\tau \in \mathbb{R}$ is the step size, and $d^k \in \mathbb{R}^n$ the update step. Widely used to solve computer vision-related optimization problems are the optimization method gradient descent and its variants. Gradient descent offers an iterative approach where the update steps are computed using the gradient of the objective function at the current point u^k, resulting in the update step $d^k = \nabla E(u^k)$. It converges to a minimizer under the condition that E has a minimizer, is convex and L-smooth for a step size $\tau \in]0, \frac{2}{L}[$, depending on the Lipschitz constant of ∇E.

Backtracking Line-Search

If the Lipschitz constant L is unknown, methods like backtracking line-search [29] provide a possibility to approximatly calculate the step size τ each iteration. The basic idea is to decrease τ when a certain condition is not fulfilled in each iteration. For gradient descent τ will be decreased by a predefined factor $0 < \beta < 1$ as long as the inequality

$$E(u^k - \tau \nabla E(u^k)) \leq E(u^k) - \alpha\tau \|\nabla E(u^k)\|^2 \tag{2.15}$$

for $0 < \alpha < 1$ does not hold. For a more general formulation refer to [29].

Related Variants

Besides gradient descent, there are many related variants, such as Nesterov's accelerated gradient descent [305], originally published in [220], the gradient projection method [282, 258], which is suitable for set-constrained problems for convex, closed sets $C \in \mathbb{R}^n$, projecting after each iteration onto the given set $u^{k+1} = \Pi_C(u^k - \tau \nabla E(u^k))$, or the conditional gradient descent [102], also for minimizing the objective function over a set C [32]. Further iteratives techniques are stochastic methods, e.g. stochastic gradient descent, and are often used in learning based approaches (see Chapter 3). While gradient descent converges for convex, L-smooth functions, the property of smoothness is not always given. The proximal gradient descent [40], which will be discussed in the next section, is convergent for functions that can be divided into two subproblems: a convex, smooth problem and a possibly non-differentiable problem. An accelerated variant is given in [21].

2.4.3 Proximal Methods

Let us consider problems that are not differentiable, but can be split into two functions of the form $G = F + E$, for convex and L-smooth functions F, and convex, possibly non-differentiable functions E. The proximal gradient descent [40] is appropriate for such problem scenarios. This form of problems is particularly interesting for energy minimization problems with an L-smooth data term and a non-smooth and convex regularization term, as e.g. TV. The update step of proximal gradient descent, calculates the gradient descent step, surrounded by the proximal operator prox_E,

$$u^{k+1} = \text{prox}_{\tau E}(u^k - \tau \nabla F(u^k)). \tag{2.16}$$

The proximal operator can be been as a tradeoff between minimizing the energy E, and predicting v to be close to u, and is defined by

$$\text{prox}_{\mu E}(u) = \arg\min_v E(v) + \frac{1}{2\mu}\|u - v\|^2, \tag{2.17}$$

for a fixed parameter μ, and besides the (accelerated) proximal gradient method [21], is at the heart of several techniques such as the primal-dual algorithm [48] or the alternating direction method of multipliers (ADMM) [103].

In the special case of $F = 0$, the proximal gradient descent algorithm is called the proximal point algorithm:

$$u^{k+1} = \text{prox}_{\tau E}(u^k), \tag{2.18}$$

2.4.4 Moreau-Yosida Regularization

Closely related to the proximal operator (2.17) is the Moreau-Yosida regularization (or Moreau envelope), also defined in [162, 250].

Definition 8. *Let $E : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a lower semi-continuous, proper function, then $E_\mu(u)$ is the Moreau-Yosida regularization of E with respect to a regularization parameter $\mu > 0$, with*

$$E_\mu(u) = \inf_v \; E(v) + \frac{1}{2\mu}\|u - v\|^2. \tag{2.19}$$

In a convex setting, E_μ can be seen as a convex and smooth approximation of a possibly non-differentiable function E [250]. The parameter $\mu > 0$ influences the degree of its smoothness. While a small μ yields a smooth approximation of the original function E, for decreasing μ, the approximation E_μ becomes a closer representation of E. This smooth behavior plays a role in specifying the Moreau-Yosida regularization's L-smoothness, where the Lipschitz constant of E_μ depends on the choice of μ, i.e. $L = \mu^{-1}$.

Example on the Huber Function

The Moreau-Yosida regularization of $E(u) = |u|$, is the popular Huber function [137],

$$E_\mu(u) = \inf_v \left\{ |v| + \frac{1}{2\mu}(u - v)^2 \right\} = \begin{cases} \frac{1}{2\mu}u^2, & |u| \le \mu, \\ |u| - \frac{\mu}{2}, & |u| > \mu \end{cases} \tag{2.20}$$

and is a prominent example for the Moreau-Yosida regularization. Figure 2.1 visually demonstrates this example and shows the behavior of $E(u)$ and the Huber function in (2.20), showing the transition between a linear to a quadratic function and maintaining differentiability at $u = 0$. This example highlights the key property of the Moreau-Yosida regularization E_μ, having the same optimum as the corresponding energy function E given that E is proper, convex, and lower semi-continuous.

Relation to the Proximal Point Algorithm

The replacement of non-smooth energies by their Moreau-Yosida regularization will be of central importance over the course of this work to make the prediction of iterative update steps on non-smooth energies possible. As the proximal point algorithm can be interpreted as a gradient descent method on the Moreau-Yosida regularization, their relation is a key component of the methology in Chapter 7. To demonstrate this relationship in the following section, we first need the concept of the conjugate function.

Definition 9. *Given a proper function $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, then the conjugate function $E^* : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\}$ of E is*

$$E^*(u) = \sup_{v \in dom(E)} \left(u^T v - E(v) \right). \tag{2.21}$$

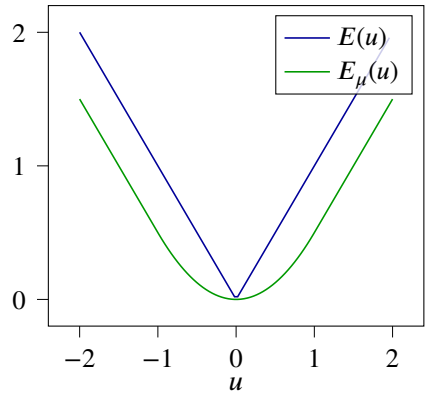


Figure 2.1: Example for Moreau-Yosida regularization, showning $E = |\cdot|$ and its Moreau-Yosida regularization E_μ for $\mu = 1$, also known as the Huber function (2.20).

Following this definition, the Fenchel inequality as stated in [29], can be directly derived,

$$E^*(u) + E(v) \geq u^T v, \tag{2.22}$$

which holds for all u and v. An important relation linking the gradient of a function to the gradient of its conjugate function is given in [248] (Theorem 23.5), which states that the gradient of the conjugate function E^*, at a point u, is the argument v that maximizes $u^T v - E(v)$.

To demonstrate the relationship between the Moreau-Yosida regularisation and the Proximal Point algorithm, let us consider the definition of the Moreau-Yosida regularisation in Definition 8.

$$E_\mu(u) := \min_{v \in V} E(v) + \frac{1}{2\mu}\|v - u\|^2 \tag{2.23}$$

$$= \frac{\|u\|^2}{2\mu} + \min_{v \in V} \left\{ E(v) + \frac{\|v\|^2}{2\mu} - \frac{\langle v, u\rangle}{\mu} \right\} \tag{2.24}$$

$$= \frac{\|u\|^2}{2\mu} - \sup_{v \in V} \left\{ \frac{\langle v, u\rangle}{\mu} - \left(E(v) + \frac{\|v\|^2}{2\mu} \right) \right\} \tag{2.25}$$

$$= \frac{\|u\|^2}{2\mu} - \frac{1}{\mu} \sup_{v \in V} \left\{ \langle v, u\rangle - \left(\mu E(v) + \frac{\|v\|^2}{2} \right) \right\} \tag{2.26}$$

$$= \frac{\|u\|^2}{2\mu} - \frac{1}{\mu} \left(\mu E + \frac{\|\cdot\|^2}{2} \right)^* (u). \tag{2.27}$$

The term inside the supremum from the rearranged expression can be identified as a shifted and scaled version of the conjugate function. Let now $h := \mu E + \|\cdot\|_2^2$. As h is strongly convex, we know that h^* is differentiable and the gradient is

$$\operatorname{argmax}_{v \in V} \left\{ \langle v, u\rangle - \left(\mu E(v) + \frac{\|v\|^2}{2} \right) \right\} = \operatorname{prox}_{\mu E}(u). \tag{2.28}$$

Therefore the gradient of E_μ can be calculated as

$$\nabla E_\mu = \frac{u}{\mu} - \frac{\operatorname{prox}_{\mu E}(u)}{\mu} \tag{2.29}$$

$$= \frac{u - \operatorname{prox}_{\mu E}(u)}{\mu}, \tag{2.30}$$

where the proximal operator is derived from its definition. Rearranging the equation to have the proximal operator on the left hand side,

$$\text{prox}_{\mu E}(u) = u - \mu \nabla E_\mu(u), \tag{2.31}$$

shows that the proximal point algorithm (2.18) can be written in terms of gradient descent on E_μ with step size μ, i.e.

$$u^{k+1} = \text{prox}_{\mu E}(u^k) = u^k - \mu \nabla E_\mu(u^k). \tag{2.32}$$

So the proximal point algorithm (2.18) can be interpreted as a conventional gradient descent method on the Moreau-Yosida regularization of the original costs (see, e.g., [234]).

We will use this insight in Chapter 7 to extend a hybrid model- and learning-based methodology from a setting that requires differentiability to a non-differentiable setting.

CHAPTER 3

Deep Learning in Imaging

While solving inverse problems by classical model-based techniques requires a fundamental understanding of the forward process, in recent decades, many (inverse) problems have been solved by data-driven neural networks, as image denoising [331, 318, 333, 292], classification [156, 169], super-resolution [81, 151, 273], magnetic resonance imaging [325] or image segmentation [193]. Feedforward networks can be seen as nonlinear functions $\mathcal{G}_\theta$, that are composed of multiple layers and aim to map given (corrupted) input data f to an appropriate output u,

$$\mathcal{G}_\theta(f) = u, \tag{3.1}$$

imitating the inversion of a forward model of an imaging problem, where θ is the parametrization of the neural network. Please note the difference between machine learning and classical energy minimization methods, that the latter ones optimize the variable u, which in the end also reflects the solution, while the goal of deep learning techniques is to learn a parametrization θ of an unknown function $\mathcal{G}_\theta$.

The training of neural networks can be categorized into supervised and unsupervised learning strategies. In supervised learning strategies, the network is trained on a labeled dataset where each network input is paired with the corresponding known output/label. Unsupervised deep learning techniques train neural networks without the use of labeled data, allowing networks to find patterns or clusters in input data on their own. We, however, will not discuss unsupervised learning since it is outside the scope of the learning methods in this thesis. Please see [58] for additional information on this subject.

3.1 Supervised Learning and Core Models

In the context of supervised learning, neural networks are designed to learn from a dataset or set of observations $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^m$, where $\mathcal{D}_t$ is the training dataset, consisting of m data elements. The elements y_i are annotations that correspond to specific input data x_i. The primary objective of the training is to optimize the neural network's weights θ, so that it correctly maps the data points x_i to their corresponding outputs y_i,

$$\mathcal{G}_\theta(x_i) = y_i, \tag{3.2}$$

by minimizing the expected error of the neural network with respect to the parameters θ,

$$\min_{\theta} E(\theta), \quad \text{with} \quad E(\theta) = \mathbb{E}_{(x_i,y_i)\sim D_t} \mathcal{L}(\mathcal{G}_\theta(x_i), y_i), \tag{3.3}$$

where the data samples are drawn from the training dataset. The loss function $\mathcal{L}$ depends on the task and measures the error between the network's prediction $\mathcal{G}_\theta(x_i)$ and the true annotation y_i. Common choices of loss functions for regression tasks such as image deblurring [224] and image denoising [331, 142] include the squared ℓ_2 norm, measuring the average squared difference between the network outputs and the true annotations, and the ℓ_1 norm, known for its robustness to outliers. If a balance between the ℓ_1 and ℓ_2 norm is desired, the Huber loss [137] offers a good alternative.

Most widely used for classification tasks, including image segmentation, is the cross-entropy loss, which has been used in [14] for the *SegNet* segmentation network. The cross-entropy loss measures the dissimilarity between the true distribution p and the predicted distribution q over all classes k of the classification or segmentation task and is given by $\mathcal{L}(p, q) = -\sum_k p(k) \log q(k)$. Ronneberger *et al.* [255] and Cicek *et al.* [66] leverage a weighted variance of the cross entropy loss for *U-Net* and *3D U-Net*, while a balanced cross-entropy loss, is discussed in [319]. Other works use the dice loss [212, 265], which is, among further loss functions, discussed in [293]. For more detail on loss functions for deep neural networks, refer to [22]. There also exist several optimization algorithms for minimizing loss functions, including the widely used stochastic gradient descent and the adaptive moment estimation (Adam) optimizer [152], which we extensively used in our work. Please see [285] for a discussion on further optimization methods.

Widely used for image-related applications, is the convolutional neural network (CNN) architecture, which consists of convolutional layers to learn spatial features via learnable filters. One of the pioneering CNN, introduced by LeCun *et al.* [163], was designed for handwritten digit recognition. As research progressed, there was an interest to further improve the architecture's structure for better prediction accuracy and reduction of computational costs. In the following, we will highlight those CNN architectures that will become important in the course of this work and refer for a more complete overview to [117, 173]. An important work, proposed by He *et al.* [129], addresses the observation that deep neural networks often show a decrease in accuracy with increasing depth. He *et al.* discussed the Residual Network *ResNet* as a solution and introduced the idea of skip connections, that allow direct connections between layers, skipping some intermediate layers. Another widely known CNN architecture is the *U-Net* [255], which was originally designed for biomedical image segmentation and consists of a downsampling and an upsampling path, the latter containing connections with the corresponding feature map from the downsampling path. With a focus on more computational efficiency and less memory requirement, Paszke *et al.* [236] proposed the *E-Net*, designed for more lightweight applications, such as mobile applications. For a more extensive receptive field of the model, without the need for increasing the number of parameters, the *DeepLabv3* network [55] consists of atrous/dilated convolutions, where the kernel has gaps between its elements. In recent developments within the field of computer vision, new architectures are based on transformer networks that were originally developed for natural language processing [301] and have adapted their attention mechanisms for images [85]. In addition, the combination of vision transformers with CNNs has been the focus of various research studies [76, 93, 322]. While one strand of research has focused on improving the network architecture, other research focused on the robustness of neural networks against adversarial attacks, which will be discussed in the following section.

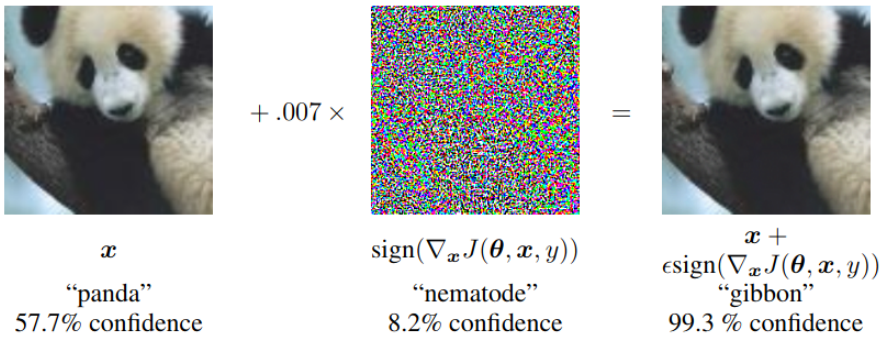


Figure 3.1: This figure is taken from Goodfellow *et al.* [112]. It shows an example of an adversarial attack using fast gradient sign method (FGSM), which leads to the misclassification of the adversarially perturbed image. While the image on the left is (correctly) identified as a *panda*, the corresponding adversarial example on the right, displaying a perturbed version of the panda from the left image, is classified as a *gibbon* with high confidence.

3.2 Model Robustness

Nowadays, although neural networks are being widely used with great success, they can be vulnerable to adversarial attacks, where small changes in the input image are shown to completely alter the prediction by the neural network. This becomes an important aspect as learning-based approaches are used for safety-critical applications, such as human tracking [277], robotics [118], autonomous driving tasks [27], and in connection to that, sign recognition [67].

In Chapter 6, we apply adversarial training on a classification network to mitigate the effects of minor, imperceptible changes in the input on the neural network's classification results. This step is crucial for the success of our proposed method.

3.2.1 Adversarial Attacks

The goal of adversarial attacks is to find adversarial examples x^{adv}, within a proper distance from the original input sample, that lead to incorrect predictions by a neural network. The most popular distance metric to regulate the distance of the adversarial examples from the original sample is the ℓ_∞ norm [246].

It was first discovered by Szegedy *et al.* [290] that adding a perturbation to an image can lead to its misclassification by a classification network, thereby highlighting the vulnerability of these networks to adversarial examples. They demonstrated that the box-constrained L-BFGS method can identify such adversarial examples, where the perturbation may not even be visible for a human. Subsequently, Goodfellow *et al.* [112] conducted experiments on adversarial training and proposed the fast gradient sign method (FGSM) for improving robustness against adversarial attacks by creating adversarial examples $x^{\text{adv}} = x + \eta$, within a ℓ_∞ distance, by

$$\eta = \epsilon \, \text{sign}(\nabla_x \mathcal{L}(\mathcal{G}_\theta(x), y)) \tag{3.4}$$

for small values of ϵ. Here, x represents the network input, y the corresponding annotation, and θ the network weights. A popular example of an image misclassified by FGSM is

illustrated in Figure 3.1. To not only aim for misclassification by a neural network but also to increase the probability that a neural network predicts a specific target y^{target}, there exist targeted attacks [159] such as targeted FGSM,

$$x^{\text{adv}} = x - \epsilon \operatorname{sign}(\nabla_x \mathcal{L}(\mathcal{G}_\theta(x), y^{\text{target}})), \tag{3.5}$$

predicting a gradient step towards the prediction of target label y^{target}.

While one-step methods, like FGSM, generate adversarial examples in a single step, multi-step methods aim to predict adversarial examples iteratively. A popular iterative attack, derived from FGSM, is the basic iterative method (BIM) introduced by Kurakin *et al.* [160] ,

$$x_{k+1}^{\text{adv}} = \text{clip}_{x,\epsilon}(x_k^{\text{adv}} + \alpha \text{sign}(\nabla_x \mathcal{L}(\theta, x_k^{\text{adv}}, y))), \tag{3.6}$$

where a clipping operation in each iteration ensures that the result remains elementwise in the ℓ_∞ ball neighborhood, e.g. $(\text{clip}_{x,\epsilon}(x^{\text{adv}}))_i \in [x + \epsilon, x - \epsilon]$. Similar to (3.6), but with different initialization strategy, is the projected gradient descent (PGD) attack of Madry *et al.* [198]. Further attack models are variations of FGSM, like the Momentum-based Iterative Fast Gradient Sign Method (MI-FGSM) [83], distributionally adversarial attack [341], Carlini and Wagner (C&W) attacks [46], the Jacobian based saliency map attack (JSMA) [233], and the DeepFool [215]. All of these attacks imply that the attacker is familiar with the targeted neural network and its weights, which is the exact description of *White-Box attacks* – a categorization based on the attacker's knowledge. Further categories are *Gray-Box attacks*, where the attacker only partially knows the target model, and *Black-Box attacks*, where the attacker is unaware of the network. The ability to transfer adversarial examples between models has been investigated in [232, 190], making them adaptable for both Gray-Box and Black-Box scenarios. Since adversarial examples were introduced by Szegedy *et al.* [290], intensive research was devoted to methods that perform adversarial attacks, e.g. by learning [20]. For a summary of various attack strategies, please refer to [246, 19].

3.2.2 Adversarial Defence

A common strategy to make neural networks robust against adversarial attacks is to train them with adversarial examples, which we also do in Chapter 6. This can be formulated as a min-max problem as proposed by Madry *et al.* [198],

$$\min_\theta E(\theta), \quad \text{where} \quad E(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\eta \in S} \mathcal{L}(\mathcal{G}_\theta(x + \eta), y) \right] \tag{3.7}$$

where the inner problem focuses on maximizing the loss $\mathcal{L}$ by adding perturbations η from a set of allowed perturbations S to the data sample x, drawn from a distribution $\mathcal{D}$. The inner problem can be addressed using adversarial methods, as discussed in Section 3.2.1, such as the FGSM. The outer problem refers to the training of a neural network by minimizing $E(\theta)$ with respect to the network weights θ. A robust training strategy using FGSM, proposed by Goodfellow *et al.* [112], forms a weighted composition of the training loss with sample x, and the corresponding adversarial sample x_{adv}. More robustness has been shown for adversarial training using PGD [198]. Several other defence mechanisms against adversarial attacks build on approaches like denoising, which aim to reduce the impact of noise on the network features [323, 175], provable defense strategies, which seek to maintain a given accuracy level [242], and the sparsification of network models [121]. A summary of various defence approaches is given in [246].

3.2.3 Model Robustness in Computer Vision

In addition to the original idea of attacking classification networks [112], adversarial attacks and robustness were explored in various computer vision-related topics. This includes the field of object tracking [140, 174, 316], where Xie *et al.* [316] combined an adversarial attack on object tracking, with one on semantic segmentation, leading to incorrect image segmentations. Further research in segmentation also demonstrated the possibility of removing a target class from the result while keeping the rest of the segmentation correct [132]. Studies in super-resolution induced perturbations into low-resolution images to degrade the quality of high-resolution images [64]. Moreover, there are adversarial attacks on graph matching, e.g. Zhang *et al.* [338] induced perturbations into the graph structure to push matching nodes towards dense regions, making them indistinguishable from their neighbors and reducing the quality of matching. A lot of work has been done in the direction of face recognition, which is important for surveillance or access control, where Dong *et al.* [84] discussed black-box adversarial attacks on decision-based face recognition, and Zhong *et al.* [342] investigated and improved the transferability of adversarial examples for face recognition. An interesting approach is proposed by Sharif *et al.* [270], discussing an adversarial attack on face recognition systems that allows a person to impersonate another person by wearing printed eyeglasses. Adversarial attacks have also been studied for medical-related imaging, mostly for classification and segmentation tasks [82, 101, 235].

3.3 Deep Learning for Model-Based Methods

While deep learning models have shown strong performance in computer vision tasks, they are limited by a lack of theoretical understanding and an absence of guarantee and control over network's predictions. To simultaneously take advantage of the mathematical understanding of the behavior of classical model-based methods and data-based learned networks, efforts in the direction of integrating data-based learned networks into iterative optimization algorithms have been made in the past few years. An effective way for many applications to combine model-based and learning-based methods is by incorporating a learning-based denoiser into the model-based approach, as has been done for *Plug-and-play Priors* [302], and for *Regularization by Denoising* [253]. Besides the inclusion of learning-based denoisers, in the following sections we will cover further approaches that exploit learned regularizers based on training-data in model-based methods, and *deep image prior (DIP)* [296], where the structural design of a neural network serves as an implicit prior. The structural design of a neural network also plays a key role in *unrolling* techniques, where model-based iteration steps are encoded by neural networks. Since convergence is only partially achieved by the aforementioned methods, further discussion will be extended to hybrid methods that come with convergence guarantees.

3.3.1 Plug-and-Play Prior

The Plug-and-play prior framework was first introduced by Venkatakrishnan *et al.* [302] in 2013, aiming to integrate prior denoising methods, including learned networks, into optimization problems. The framework is based on the concept of decoupling the forward

model from the prior $\mathcal{V}$ in an energy minimization formulation,

$$\hat{u} = \arg\min_{u} E(u) + \alpha\mathcal{V}(u), \tag{3.8}$$

which is then reformulated as

$$(\hat{u}, \hat{v}) = \arg\min_{u,v} E(u) + \alpha\mathcal{V}(v), \text{ s.t. } u = v. \tag{3.9}$$

This formulation can be optimized using ADMM [28, 103]. In this method, the forward model and the prior are optimized separately, with the key modification of replacing the proximal operator in ADMM with a denoising algorithm. Such, the Plug-and-play framework combines proximal algorithms with denoising priors.

For this purpose several denoising algorithms were proposed, especially denoising by data-driven neural networks [207, 247], such as Zhang *et al.* [330], introducing a deep denoiser prior specifically for Plug-and-play image restoration. The concept of Plug-and-play has been used in many algorithms. This includes the primal-dual algorithm [48, 229], used in [131] for demosaicking, and FISTA [21, 144]. This approach has also been used for several applications, such as super-resolution [35, 332], Poisson denoising [254], color denoising and deblurring [330].

Other studies have focused on the convergence analysis of Plug-and-play methods. Under assumption of a nonexpansive denoiser, Sun *et al.* [287] analyzed the convergence of a variant of Plug-and-play based on the proximal gradient method. Followed by this, Ryu *et al.* [259] analyzed the convergence properties of two other Plug-and-play variants (Plug-and-play forward-backward splitting and Plug-and-play-ADMM), demonstrating convergence under the condition that the learned denoiser satisfies a Lipschitz condition. In their paper they provided guidance on enforcing this condition during the training of the denoiser. More recently, Al-Shabili *et al.* [6] discussed the convergence of Plug-and-play on the Bregman proximal gradient method, assuming a strongly convex data-fidelity term and a Lipschitz continuous network.

3.3.2 Regularization by Denoising

Another approach for the explicit integration of (possibly pre-trained) denoiser priors into iterative algorithms has been introduced by Romano et al. [253]. They proposed a method called Regularization by Denoising (RED) to generate regularizers $\mathcal{V}(u)$ from denoisers $D(u)$,

$$\mathcal{V}(u) = \frac{1}{2}u^T(u - D(u)), \tag{3.10}$$

penalizing the residual difference and cross-correlation between u and the residual. This work has inspired several subsequent studies using neural network architectures for noise reduction (or similar) in the same framework. For example, Metzler *et al.* [210] used the framework of RED on phase retrieval problems, while instead of denoising, Liu *et al.* [181] proposed regularization by artifact-removal and showed their results on 3D magnetic resonance imaging (MRI). Further research [313, 286] discussed the application of RED on large images, while including a study on the convergence. A condition on the convergence of RED has been given in [245], showing that denoisers as DnCNN [331] do not guarantee convergence, as they need a symmetric Jacobian. While many denoisers do not meet the necessary criteria [245], Cohen *et al.* [69] proposed a class of denoisers that fulfill this condition.

3.3.3 Learned Regularizer

Besides the inclusion of learning-based denoisers, the integration of learning into model-based methods can also be realized by using neural networks as regularization functions. This approach is discussed by Lunz *et al.* [195] on a network that is trained to distinguish between training images and unregularized reconstructions. The objective of this approach can be expressed as:

$$\arg\min_{u} E(u) + \lambda \mathcal{G}_\theta(u), \tag{3.11}$$

for a pre-trained neural network $\mathcal{G}_\theta$ with fixed weights θ. Li *et al.* [168] proposed the Network Tikhonov (NETT) method, which defines the regularizer through a learned neural network. They also conducted a convergence analysis, demonstrating both convergence and convergence rate under mild assumptions. Following from this, Obmann *et al.* introduced a deep synthesis version of NETT and augmented NETT to overcome restrictions by using an augmented form of the regularizer [227, 226]. Alberti *et al.* [7] investigated unsupervised and supervised learning frameworks to learn the optimal Tikhonov regularizer. Other type of learned regularizers are discussed by Kobler *et al.* [154] who developed a learnable generic multi-scale regularizer, while Mukherjee *et al.* [217] studied data-driven convex regularizers. Duff *et al.* [91] has explored the integration of generative regularizers, incorporating a generative adversarial network (GAN) into the regularization framework.

Efforts to use a data-driven, learned regularizer have also been proposed in [86] for computed tomography (CT) reconstruction. For additional details, please refer to Chapter 6, where we discuss the regularization of an optimization problem with a learned prior.

3.3.4 Deep Image Prior

Moreover, certain methods use the structural design of neural networks to act as a form of regularization. An important publication on this topic is the work of Ulyanov *et al.* [296], introducing DIP networks, where the network structure, even without training, favors the generation of images with realistic high-level features. To reconstruct an image $\hat{u}$ from its corrupted version f one uses an untrained network $\mathcal{G}_\theta$ which takes random noise z as input and typically optimizes a cost function E (without an explicitly defined regularization term) over the network:

$$\hat{\theta} = \min_{\theta} E(\mathcal{G}_\theta(z); f) \tag{3.12}$$

Here the goal is to recover the image by $\hat{u} = \mathcal{G}_{\hat{\theta}}(z)$. Efforts to improve DIP have led to the integration of additional regularizations. Mataev *et al.* [205] included an explicit prior based on the RED regularizer (see (3.10)), while Liu *et al.* [182] explicitly added TV regularization to DIP for image restoration. Similarly, Van *et al.* [298] included a learned regularizer to DIP for compressed sensing. To improve the potential of DIP for image denoising, Asim *et al.* [11] propose to apply DIP on noisy image patches, exploiting self-similarities in images. Besides the work on additional priors to DIP, Chen *et al.* [60] worked on network architectures based on U-Net that capture stronger deep image priors.

DIP has been applied in various applications, as for image decomposition tasks such as dehazing and binary segmentation [104], for adversarial defense methods [147, 75], for hyperspectral unmixing [244] and for improving image quality of undersampled photoacoustic microscopy (PAM) images [304].

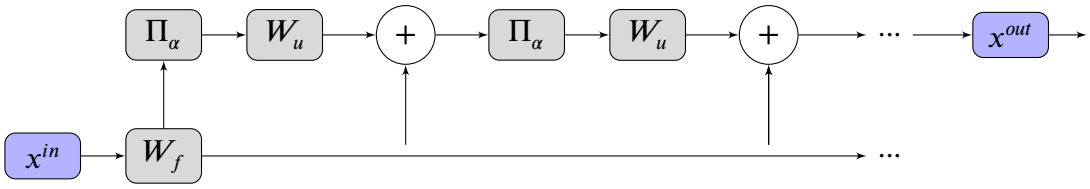


Figure 3.2: This illustration shows the conceptual mapping of each ISTA iteration onto a distinct neural network layer, where the learnable parameters W_f and W_u are integrated within the layers.

3.3.5 Algorithm Unrolling

Algorithm unrolling or unfolding refers to the connection of learning approaches and iterative optimization algorithms by encoding each step of an iterative algorithm as a layer within a neural network. Originally Gregor and LeCun [113] developed the unrolling technique for the Iterative Shrinkage Algorithms (ISTA) [21], called Learned ISTA (LISTA), to improve the efficiency of sparse coding. Their basic idea lies in optimizing

$$\min_u \|Au - f\|_F^2 + \lambda \|u\|_1 \tag{3.13}$$

by unrolling the update steps of the corresponding descent algorithm, calculating descent steps on the data term and applying soft thresholding on these by

$$\begin{aligned} u^{k+1} &= \Pi_\alpha(u^k - \tau\nabla(\|Au - f\|_F^2)) && (3.14)\\ &= \Pi_\alpha(u^k - \tau A^T(Au^k - f)) && (3.15)\\ &= \Pi_\alpha((I - \tau A^T A)u^k - (\tau A^T)f) && (3.16)\\ &= \Pi_\alpha(W_u u^k - W_f f) && (3.17) \end{aligned}$$

where $(\Pi_\alpha(v))_i = \text{sign}(v_i)\max(|v_i| - \alpha_i, 0)$ denotes soft thresholding with the threshold α_i depending on λ, and W_u, W_f are substitutions of $(I - \tau A^T A)$ and (τA^T) respectivly. The general idea of LISTA is illustrated in Figure 3.2, showing the structure of the corresponding unrolling. Here the parameters α, W_u and W_f are learned by a given dataset. Unlike shown in Figure 3.2, it is also possible that each iteration (layer) can be parameterized differently.

Following [113], several studies have explored the unrolling concept, as Hershey *et al.* [133] propose the idea of unrolling iterations as layers within a deep neural network, while untying model parameters across these layers, where each iteration can have its own set of learnable parameters. Classical optimization algorithms have been adopted as network frameworks in a variety of image processing and reconstruction applications. These include blind image deblurring [172], super-resolution of images [311], and image restauration [266, 61]. Moreover, these algorithms have been used for reconstruction tasks that are especially interesting in the medical field such as CT [2] and MRI image reconstruction [284, 324, 135]. Over the years, several classical optimization algorithms have been reformulated as network frameworks, as Hammernik *et al.* [124] proposed unrolled gradient descent scheme for MRI reconstruction, Chen *et al.* [61] proposed a trainable diffusion model by unrolling gradient descent steps for image restauration, and further analysis on the learned step size for unrolled gradient descent has been done by Takabe and Wadayama [291]. In addition to gradient descent, unrolling techniques have also been applied to other optimization algorithms, such as PDHG (Primal-Dual Hybrid

Gradient) [2]. Further work [284, 324] discussed architectures, which are derived from the iterative procedures in ADMM, Lohit *et al.* [191] unrolled the projected gradient descent (PGD) algorithm, and Mardani *et al.* [201] discussed unrolled proximal gradient descent, Hosseini *et al.* [135] with skip connections.

Few works also deal with the convergence analysis of unrolling schemes. In 2018, Chen *et al.* [57] performed a convergence analysis for the unrolling strategy LISTA by inducing constraints on its structure and thereby proving linear convergence, while Liu *et al.* [185] proposed a converging unrolling framework. To increase the number of unrolling iterations that can be covered by unrolling methods which are often limited by computational resources, Gilton [109] proposed to use deep equilibrium architectures [18] for unrolling with provable convergence guarantees.

3.3.6 Further Hybrid Methods with Convergence Guarantees

In addition to approaches on learning-based priors and unrolling schemes, there are methods in which the iterative update step of a minimization approach is predicted by a data-driven neural network, that still ensure convergence to a minimizer of an energy function. In this context, Heaton *et al.* [130] demonstrated convergence in a learning-to-optimize scheme. This is achieved by examining their data-driven update steps and, when required, substituting them with conventional update steps to guarantee convergence. Moreover, studies by Liu *et al.* [184, 187, 186] proposed the usage of network-based update steps and controlled and corrected convergence behavior through a feedback mechanism. Specifically, in [187] they focused on the application of compressive sensing in MRI, while in [186] they targeted image enhancements. In contrast to these methods, the work of Möller *et al.* [214] does not rely on fallback mechanisms, but instead each update is exclusively predicted by a neural network. Möller *et al.* proposed training a neural network to predict the update directions of a energy function and guarantee convergence, under certain conditions on the energy function. This approach is combined with Plug-and-play networks in the subsequent work of Sommerhoff *et al.* work [280]. Yet, both approaches fundamentally rely on the ability to differentiate the energy and obtain reasonable step sizes (e.g. via Lipschitz continuous gradients with reasonably small Lipschitz constants).

In our work on energy dissipating networks for non-smooth energies [89], we tackle the aforementioned drawbacks by harnessing the properties of the Moreau-Yosida regularization, e.g. for applications using regularizers and robust losses, involving non-smooth functions. We address this topic in Chapter 7.

3.4 Diverse and Explorable Reconstruction

In learning-based approaches to image restoration or reconstruction, a neural network typically maps each input to one single result. But instead of a single result, there could be multiple feasible reconstructions for image restoration problems. For instance for super-resolution tasks, multiple high-resolution images could feasibly correspond to the same low-resolution input, as shown in Figure 3.3. Recently, there has been some research focused on expressing and exploring the diverse space of valid solutions to computer vision tasks, including super-resolution, imagine decompression, inpainting, and deblurring.

Super-resolution refer to the problem of reconstructing a high-resolution image $\hat{u}$ from

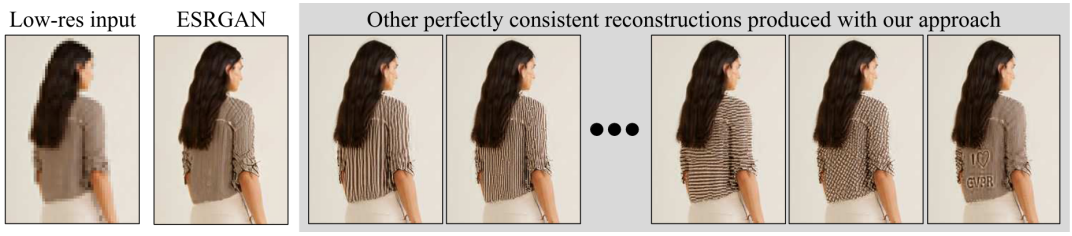


Figure 3.3: This figure is taken from Bahat *et al.* [16] and shows on the left a low-resultion image, and multiple corresponding high-resultion images, resulting from the exporable super-resolution approach in [16], where all high-resultion images are consistent with the low-resultion image.

a given low-resolution image f, s.t.

$$f = H\hat{u} \tag{3.18}$$

with H being the matrix corresponding to a blur and a subsequent downsampling operator. The possibility of multiple high-resolution images $\{\hat{u}_1, \hat{u}_2, \dots\}$ corresponding to the same low-resolution image f, has led to research focused on sampling from the space of potential high-resolution images. In this context, Lugmayr *et al.* [194] proposed an approach for sampling high-resolution images from a learned conditional distribution $p(u|f, \theta)$, given a low-resolution image f. Menon *et al.* [209] discussed a slightly different method to obtaining samples of high-resolution images. This method includes sampling from the latent space of a generative model and identifying images that match the corresponding low-resolution image when downsampled. Other methods go beyond randomly sampling the solution space and develop tools to enable users to explore it. Bahat *et al.* [16] proposed a method that allows a user to explore the solution space in GAN-based image super-resolution. They proposed to control the manipulation of the output of a super-resolution model with a control signal, while still guaranteeing data consistency by introducing the consistency enforcing module (CEM). This means that after applying a blurring and downsampling operation to the higher-resolution image, the CEM ensures that the downsampled image matches the corresponding low-resolution image. Additionally, Bühler *et al.* [39] proposed to make an exploration in the context of super-resolution semantically controllable using a GAN.

In the field of image decompression, Bahat *et al.* [17] proposed using a pre-trained digit classifier to automatically explore the potential decodings corresponding to a compressed picture of a numerical digit. Furthermore, Guo *et al.* [120] presented the "one-to-many" network for image decompression for JPEG-compressed pictures, capable of providing several reconstruction outcomes for a single compressed input image.

Besides super-resolution and image decompression, methods for diverse reconstruction have been proposed for several other tasks. Dey *et al.* [78] have been working on inpainting problems, dealing with images of faces, where regions of the faces, e.g. mouth or the eye, are obscured. They aimed to produce multiple 3D reconstructions of the faces by exploring the latent space representations of obscured areas and pushing towards diverse results for those regions yet remaining consistent with the visible area. Cai *et al.* [43] proposed work on image deblurring, addressing multiple levels of image degradation. To do so, they use a GAN, controlled by a condition vector, allowing modifications to the restored image. In other work on diverse denoising, Prakash *et al.* [239] proposed to use a variational autoencoder for sampling denoising solution from a predicted a distribution

in latent space.

In Chapter 6, which is based on [86], we discuss an explorable reconstruction approach, focusing on the reconstruction of CT scans. We explore a variety of underdetermined reconstructions that maintain high data consistency, all referring to different levels of medically concerning reconstructions.

CHAPTER 4

Imaging Basics and Applications

In the following sections, we will introduce all applications that will be important to this thesis. We will address the forward problem of computed tomography (CT), discuss filtered backprojection (FBP) and classical iterative reconstruction methods, as well as learning-based methods. A section about segmentation will cover model- and learning-based image segmentation approaches and dive deeper into a relevant iterative reconstruction method on spatially varying color distributions. Section 4.3 addresses assignment problems, including approaches to permutation learning, the linear assignment problem (LAP), and the quadratic assignment problem (QAP). The last section introduces 3D shape matching and functional maps.

4.1 Computed Tomography

CT scanning is an imaging technique in which multiple X-ray beams pass through an object's body from various angles and offsets, capturing the internal composition by measuring the attenuated radiation on the opposite side. The reconstruction of CT scans, based on the measured attenuations, can be written as a linear inverse problem, where the Radon transform serves as the forward operator.

4.1.1 Radon Transform

The Radon transform of a two-dimensional function, e.g. an image $u : \mathbb{R}^2 \to \mathbb{R}$, is computed by integrating along the lines that intersect u. These lines, corresponding to the X-rays in CT, are defined by their angles ρ and offsets s, such that a line l can be parameterized by the tupel (ρ, s):

$$l(\rho, s) = \{z \in \mathbb{R}^2 | \langle z, \omega(\rho) \rangle = s\}, \tag{4.1}$$

where $\omega(\rho) = (\cos(\rho), \sin(\rho))$ is a unit vector, whose direction depends on the angle ρ. Consequently, any point z on l satisfies the condition where the inner product with the unit vector $\omega(\rho)$ is equal to the line offset s.

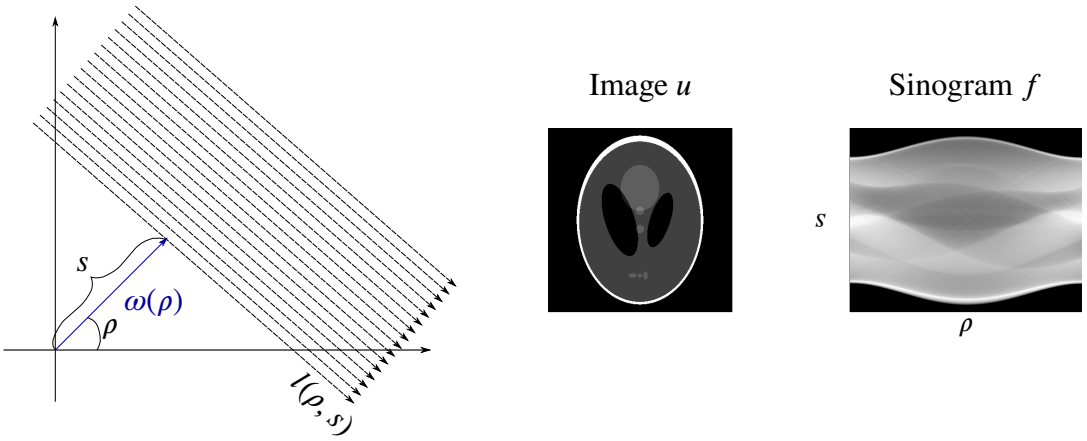


Figure 4.1: Illustration of the idea of the Radon transform (left) and an example of an image (Shepp-Logan Phantom [272]) u (middle) and its corresponding sinogram f (right).

Moreover, l can be defined by considering the unit vector that is perpendicular to $\omega(\rho)$, which is expressed as $\hat{\omega}(\rho) = (-\sin(\rho), \cos(\rho))$:

$$l(\rho, s) = \{s\omega(\rho) + t\hat{\omega}(\rho) \,:\, t \in \mathbb{R}\} \tag{4.2}$$

The underlying intuition is that by varying the parameter t, it is possible to move along the line and thus to reach different points on it. For a visualization, please refer to the illustration in Figure 4.1 on the left, which shows multiple lines $l(\rho, s)$ at a given angle ρ and multiple offsets s. By integrating along these lines, the Radon transform accumulates the values of the function u along $l(\rho, s)$:

$$\mathcal{R}(u)(\rho, s) = \int_{-\infty}^{\infty} u(s\cos(\rho) - t\sin(\rho), s\sin(\rho) + t\cos(\rho))\, dt \tag{4.3}$$

In a discrete setting, these lines are sampled at multiple points. To capture various angles and offsets, the Radon transform is expressed as a matrix-vector multiplication of the matrix version of the Radon operator R and the vectorized image u,

$$Ru + e = f, \tag{4.4}$$

where e represents additive noise, and the resulting measurement f, known as sinogram, records the attenuated radiation of the X-ray beams. An example of an image and its corresponding sinogram is shown in Figure 4.1 in the middle and on the right.

4.1.2 Filtered Backprojection

For an infinite number of projection angles ρ, the image u can be recovered from a noise-free sinogram f (under some mild assumptions) using the inverse Radon transform [241]. In practical settings, however, where the target is projected from a discrete set of p angles and the measurements are noisy, u cannot be perfectly reconstructed. One commonly used approach for approximating u is the filtered backprojection (FBP) [99] method, which is a discretized approximation of the inverse Radon transform and yields a single output $\hat{u}$ solving the inverse problem (4.4).

Let $g(\rho, s)$ be a function representing projection data, where s is the distance from the origin to the line of projection, and ρ the angle of the projection. The backprojection

operation B, when applied to g, reconstructs the image at point z and can be defined as:

$$B(g)(z) = \frac{1}{\pi} \int_0^{\pi} g(\rho, \langle z, \omega(\rho) \rangle)\, d\rho \tag{4.5}$$

Here the integral accumulates the contributions from all projection lines that intersect at point z, aiming to reconstruct the image from its projections. The FBP operation $\mathcal{F}$ applied to g is the backprojection of the filtered projection data and can be expressed as

$$\mathcal{F}(g)(z) = \frac{1}{\pi} \int_0^{\pi} F(\rho, \langle z, \omega(\rho) \rangle)\, d\rho, \tag{4.6}$$

where $F(\rho, \cdot)$ represents the filtered projection at angle ρ. This data is obtained by first applying the Fourier transform $\mathcal{T}$ to the projection,

$$(\mathcal{T} g)(\rho, t) = \int_{-\infty}^{\infty} g(\rho, s) e^{-its}\, ds, \tag{4.7}$$

and applying a filter function in the frequency domain. The ramp filter, for example, is a widely used filter in CT imaging and is expressed in the frequency domain as $h(t) = |t|$. The application of the inverse Fourier transform to the product of the ramp filter and the Fourier-transformed projection yields the filtered projection:

$$F(\rho, \langle z, \omega(\rho) \rangle) = \frac{1}{2\pi} \int_{-\infty}^{\infty} |t| (\mathcal{T} g)(\rho, t) e^{it\langle z, \omega(\rho) \rangle} dt \tag{4.8}$$

4.1.3 Iterative Reconstruction

Given the health risk that is associated with long exposure to X-rays, there is a high interest in reducing the exposure for CT scans [34]. Physically, this can be realized by sending a reduced number of X-ray beams through the measured object/body e.g. by descreasing the number of projection angles, although reconstructions using methods like FBP and fewer projection angles ρ typically contain artifacts. To address this problem, many regularized iterative methods have been developed for CT reconstruction to suppress artifacts and noisy results. Especially, regularization through TV [136] and its variations [294, 189, 275, 276, 63, 24, 52] are popular for reducing artifacts. Commonly the data term states that the difference between the Radon transform of u and the given data f should be minimal in a least-squares sense while searching for a result that minimizes the TV penalty:

$$\tilde{u} \in \arg\min_u \mathrm{TV}(u) \text{ s.t. } \|Ru - f\|_2^2 \text{ small} \tag{4.9}$$

We also worked on an energy formulation based on this concept in Chapter 6. As TV itself is not able to distinguish between noise and small structures, and as such possibly removes structures in a noisy recording, there has been work on integrating dictionary learning [321, 59] to the iterative reconstruction process, or employing nonlocal regularization [65].

4.1.4 Learning Based Reconstruction

Since the turn of the millennium, researchers have already been applying neural networks for (sparse-view) CT reconstruction [197, 314, 317], and several approaches have been developed to incorporate learning methods into the reconstruction. On the one hand, there

are *data-to-image* approaches, consisting of end-to-end neural networks, directly predicting a reconstruction based on a sinogram. In this context He *et al.* [126] proposed a CT reconstruction network which is pre-trained on the ImageNet dataset [77] and finetuned on medical data. Other types of methods are *image-to-image* approaches [337, 157, 143, 53], that apply neural networks as postprocessors to noisy CT reconstructions e.g. from FBP,

$$f = \mathcal{G}_\theta(\mathcal{F}(u)), \tag{4.10}$$

using different type of networks, like U-Net [143], DenseNet [337] or encoder-decoder architectures [53]. Instead of reconstructing CT images by deep learning, Han *et al.* [125] proposed to learn the residual in the reconstruction, introducing a fast and successful way of removing artifacts in CT. Xia *et al.* [315] discussed a framework for both approaches, *data-to-image* and *image-to-image*, that tackles multiple geometries and radiation dose levels in CT. In another line of work, neural networks are designed to support the iterative reconstruction schemes via a learned prior as shown in [150] or by using a neural network to support an iterative reconstruction algorithm using (relaxed) projected gradient descent, by replacing the projection step with a neural network [122]. Adler *et al.* proposed CT reconstruction by an unrolled primal-dual approach [2], and He *et al.* [127] focused on developing a learned prior for CT reconstruction using Plug-and-play ADMM.

4.2 Image Segmentation

Image segmentation refers to the problem of dividing an image into meaningful non-overlapping regions and is a crucial component in many image processing applications. It plays an important role in the area of computer vision for various tasks like autonomous driving [167, 295, 274, 100], satellite image segmentation [149, 13], agriculture segmentation [196, 211], gesture recognition [62, 51], and in medical applications [334, 271, 243, 90, 255]. The segmentation process can be categorized into semantic segmentation, where each pixel is assigned to a specific object class from a predefined set of categories, or instance segmentation, also differentiating between separate objects of the same class [213].

4.2.1 Model-Based Approaches

Well known classical techniques have formulated image segmentation in terms of an energy minimization problem, e.g., in the form of graph cuts [114, 30], where an image is expressed as a weighted graph where nodes represent pixels, and edges represent neighborhood relationships between pixels, and segmentation is performed by cutting the graph. Further, the edge-based segmentation with Snakes [146] iteratively deforms a given contour to fit along an object boundary. Particularly influential in that respect is the model of Mumford and Shah [218], which forms the basis of the successful variational two-region segmentation method of Chan and Vese [49] and has been extended to multiple regions in various works, see e.g. [47, 15]. A common formulation of such approaches involves to determine a one-hot representation $\hat{u} \in \mathbb{R}^{n_y \times n_x \times I}$ of an I-region segmentation for an image $f \in \mathbb{R}^{n_y \times n_x \times n_c}$ via

$$\hat{u} \in \operatorname*{arg\,min}_{u_{i,j,l} \in \{0,1\}, \sum_l u_{i,j,l} = 1} \langle u, c(f) \rangle + \alpha \mathcal{V}(u), \tag{4.11}$$

where $\mathcal{V}$ denotes a suitable regularization that penalizes irregularities, e.g. the (weighted) TV [257], and $c(f) \in \mathbb{R}^{n_y \times n_x \times I}$ are the unary costs with the entry $(c(f))_{i,j,l}$ having some sort of inverse relation to the estimated likelihood of pixel (i, j) belonging to class l.

The unaries (or data terms) of variational methods have been modeled in various forms including optimizing for suitable thresholds [49], and estimating it via spectral methods [221, 71]. Due to the difficulty of generating meaningful unaries $c(f)$ without additional information, several works have considered interactive segmentation methods, in which the user provides clues about the object to be segmented, e.g., in form of bounding boxes [256], or scribbles [80, 222]. For video segmentation, there are works that use motion cues [145].

Notably, the work by Nieuwenhuis and Cremers [222] (later extended to textural [223] information and depth segmentation [80]) demonstrated that a faithful segmentation is possible based on a few user scribbles by constructing *spatially varying* color histograms for each label, and estimating the likelihood of a pixel having a certain label by computing probabilities of the pixels' RGB values being a part of the corresponding histogram. In the next section, we will discuss this approach in more detail, as it plays a key role in our experiments in Chapter 5.

4.2.2 Spatially Varying Color Distributions

Nieuwenhuis and Cremers [222] introduced an interactive, multi-label segmentation approach using user-induced scribble data in images. Instead of performing segmentation solely based on the color in each image, they take a spatial color distribution into account.

Given a color image $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$, that is supposed to be segmented into I regions $\Omega = \{\Omega_1, \dots, \Omega_I\}$, and a labeling $v : \Omega \to \{1, \dots, I\}$ assigning each pixel to a class. Then, for every class $i \in \{1, \dots, I\}$, a user can mark/scribble some pixels, as shown in the left image in Figure 4.2. These scribbeled pixels are represented by the tupel (x_{ij}, f_{ij}), where x_{ij} indicates the locations of the m_i scribbeled pixels, for $j \in \{1, \dots, m_i\}$. Meanwhile f_{ij} stands for the corresponding color values of those pixels.

With the given information, Nieuwenhuis and Cremers [222] formulate an energy data term $F(x)$ based on the scribble positions x and the underlying color information f of the image, aiming to compute the propability of a pixel having a certain color, assuming that the pixel is part of a specific class i:

$$F_i(x) = -\log\left(\frac{1}{m}\sum_{j=1}^{m_i} G_{\rho(x)}(x - x_{ij})G_\sigma(f(x) - f_{ij})\right) \tag{4.12}$$

Here, the central concept of segmentation lies in the Gaussian kernels $G_{\rho(x)}$ and G_σ, with their respective standard deviations $\rho(x)$ and σ. The values of these standard deviations influence the impact of spatial and color information on the process of assigning each pixel x to its correct categorization.

The second part of the equation, $G_\sigma(f(x) - f_{ij})$, represents the Gaussian kernel applied to the distance between a pixel's color and the color of the scribbled pixel in class i. Generally the closer the colors are to each other, the higher the probability that the respective pixel will belong to class i. This probability is affected by the Gaussian standard deviation σ. The larger σ, the less important the color distance becomes for the data term. This probabiliy is also weighted by $G_{\rho(x)}(x - x_{ij})$. Here, the key idea is that the closer a pixel location x is to a scribble location x_{ij}, the greater the weight given to its color, such that pixels that are close to a scribble have color information that is more important. Now, with a larger standard deviation in $G_{\rho(x)}$, the pixel distance has less impact on the probability of a pixel being part of a specific region. $\rho(x)$ is computed for each pixel location x individually and depends on its spatial distance to the closest scribbled pixel: Its value is small if

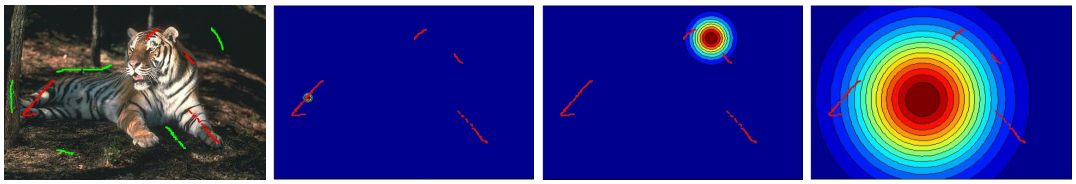


Figure 4.2: This figure (©2012 IEEE) from Nieuwenhuis and Cremers [222] illustrates the impact of the distance between a pixel and its closest scribble on the standard deviation $\rho(x)$ of the Gaussian kernel $G_{\rho(x)}$. Viewed from left to right, the first image is a color image with scribbles, while the subsequent three images show a Gaussian kernel within the image, each corresponding to another pixel location.

the distance to the closest scribble pixel is small, and high if the closest scribble pixel is farther away. This concept is also illustrated in Figure 4.2. The left image shows the RGB image with scribbles, whereby red scribbles denote the foreground, while green scribbles indicate the background. In the subsequent images on the right, the standard deviation of $G_{\rho(x)}$ is color-coded to show that it decreases as a pixel gets closer to a foreground scribble. So the underlying idea is that if a pixel is close to a specific scribble location, only nearby scribbles are given high weight, and the pixel location becomes important. If a pixel is located between multiple scribbles, but not close to a specific one, the distance between the pixel and the scribble becomes less important to the probability.

Nieuwenhuis and Cremers [222] solve for a segmentation by optimizing the data term across all pixels in each region, $\Omega_i = \{x | v(x) = i\}$ for every class i, and regularizing the perimeter of each segmented region $\text{per}(\Omega_i)$ with an emphasis on aligning the edges of the segmentation with those in the image. The energy function for this segmentation process is expressed as:

$$E(\Omega_1, \ldots, \Omega_I) = \frac{1}{2} \sum_{i=1}^{I} \text{per}(\Omega_i) + \lambda \sum_{i=1}^{I} \int_{\Omega_i} F_i(x)\, dx \tag{4.13}$$

Here, $\text{per}(\Omega_i)$ is implemented using a weighted version of TV on the binarized segmentation mask. This weighting ensures the alignment of segmented object edges with those in the image, and is determined by the gradient of the color image.

Nieuwenhuis and Cremers further discuss the optimization and the convergence of their approach in their publication [222].

4.2.3 Learning Based Approaches

With the rise of deep learning methods, researchers have developed image segmentation networks with great success [14, 54, 193, 56]. Currently, end-to-end image segmentation systems largely represent the state-of-the-art performance in this field. In 2015, Long *et al.* [193] proposed to use a fully convolutional neural network (FCN) architecture for semantic, supervised image segmentation. This approach started to gain popularity and was further used and extended [306, 183, 171]. For example, Li *et al.* [171] proposed to integrate information on the pixels' position to object instances in the images by combining FCNs with instance-sensitive score maps [73]. Liu *et al.* [188] introduced the idea of including global context into convolutional layers by averaging the features of network layers. Further work as *DeepLab* by Chen *et al.* [54] and works in [179, 340] improve the ability of localization in deep neural networks, by integrating conditional random fields [161],

or using dilated convolution [55, 56]. To improve memory efficiency and reduce computational time, encoder-decoder networks have been developed, such as SegNet [14], or the popular U-Net [255], originally intended for biomedical image segmentation. Additionally, for real-time semantic segmentation, lightweight versions of encoder-decoder segmentation networks have been proposed in [310, 50, 343]. A popular segmentation network for instance segmentation is *Mask-RCNN* [128], predicting segmentation masks by extending the object detection model in [110] with an additional segmentation branch. Following from this, Zhang *et al.* [335] discussed an adaption for improved segmentation. More recently transformer networks [301] have been used for segmentation [281, 297, 139, 138]. We refer to [170] for an overview of segmentation transformer networks. An overview of deep learning segmentation methods can be found in [213, 108], and [307] for medical applications.

Rather than estimating object properties on separate images, these networks are usually trained on thousands of examples and are therefore able to learn common shapes of objects from their training data. Yet, such networks require large training datasets, which are expensive to generate and annotate, and the resulting networks are limited to exactly those classes they have been trained on. In response to the challenge of extensive annotation, weakly supervised methods were developed. In weakly supervised methods not every pixel is annotated, but weaker sources of information such as image labels [134], scribbles [178] or bounding boxes [74], are used. Still, the aforementioned approaches require large training datasets and do not generalize to previously unseen categories.

Reducing the amount of supervision even further, several researchers have investigated learning-based clustering methods. Such techniques can also be applied to image segmentation, e.g. in [4], without even knowing the number of classes a-priori. Related to this, the intention of zero-shot segmentation is to segment non-annotated objects that have not been seen by a neural network before [38, 119].

While these approaches are not applied to scribbles, we discuss in Chapter 5 the segmentation of a single image that we aim to segment using only drawn scribbles, and compare the performance of learning-based methods with that of the classical segmentation method in [222].

4.3 Assignment Problems

Assignment problems appear in vision-related tasks in the form of feature matching between images [336, 141, 327], or shape matching [203, 45], and address the objective of finding a permutation between two ordered sets given certain assignment costs [41].

A permutation p, corresponding to the bijection from a set $\{1, \dots, n\}$ onto itself, can be represented efficiently by merely enumerating the n elements

$$(p(1),\ p(2), \dots, p(n)) \in \mathbb{N}^n. \tag{4.14}$$

However, this representation is unsuitable for most computer vision problems that involve estimating p through optimization since this representation

(i) is inherently discrete, yielding combinatorial problems for which no natural relaxation exists, and

(ii) induces a solution space with a meaningless distance metric, as element i in the set generally is not 'closer' to element $i + 1$ than it is to any other element j.

As a result, almost all methods for predicting permutations, including learning-based methods, favor a *permutation matrix* representation instead, i.e., formulating a permutation as an element in the set

$$\mathcal{P}_n = \{P \in \{0,1\}^{n\times n} \mid \sum_i P_{ij} = 1, \ \sum_j P_{ij} = 1 \ \forall i,j\}, \tag{4.15}$$

with $p(i) = j$ in representation (4.14) corresponding to $P_{ij} = 1$ in the matrix representation form (4.15).

Still, the representation of permutation matrices, shown in (4.15) is discrete and not differentiable, which makes it unsuitable for optimization. For optimization problems, a common approach is to approximate solutions by relaxing permutation matrices to continuous domains, such as the Birkhoff polytope [25]. The Birkhoff polytope represents a set of doubly stochastic matrices, where a matrix $P \in \mathbb{R}^{n\times n}$ possesses the similar properties as a permutation matrix, as defined in (4.15), but with relaxed constraints, allowing matrix entries to have values $P_{ij} \in [0,1]$ [200]:

Definition 10. *A $n \times n$ matrix $S \in \mathbb{R}^{n\times n}$ is called doubly stochastic if*

$$S_{ij} \in [0,1], \quad \sum_{i=1}^{n} S_{ij} = 1, \quad \sum_{j=1}^{n} S_{ij} = 1, \quad \forall i,j \le n. \tag{4.16}$$

In the following sections, we will introduce the LAP and the QAP and discuss further learning-based approaches for learning permutation matrices in general. Section 4.3.3 deals with shape matching as a fundamental application of matching problems with a particular focus on the widely utilized framework of functional maps [230].

4.3.1 Linear- and Quadratic Assignment Problems

The most common versions of assignment problems are LAPs and QAPs, which are based on element-wise and pair-wise costs, respectively.

The LAP is the task of assigning a set of agents to an equally sized set of tasks under minimal cost [41]. These costs are represented in a cost matrix C, where specific assignments are indicated by a permutation matrix P. The objective is to minimize the cost for each pairing across a set of permutations $\mathcal{P}_n$,

$$\min_{P\in\mathcal{P}_n} \operatorname{tr}(C^\top P), \tag{4.17}$$

for $C \in \mathbb{R}^{n\times n}$, which can be solved in cubic time using the Hungarian algorithm [158]. Nevertheless, while LAP is restrictive due to its inability to capture higher-order relationships, QAP offers an approach to modeling complex interactions. The QAP was first introduced by Koopmans and Beckmann [155] and addresses the challenge of optimally assigning elements from one set to another by

$$\min_{P\in\mathcal{P}_n} \operatorname{tr}(APBP^\top), \tag{4.18}$$

for $A, B \in \mathbb{R}^{n\times n}$, where A is the cost between elements of the first set to match, and B is the distance function between elements in the second set. The QAP has been proved to be NP-hard so no polynomial time solution can be expected for general cases. As a result, many relaxations of the problem exist, for example by relaxing the permutation

constraint [111, 251], or by lifting the problem to higher dimensions [148, 339]. A survey on various relaxation approaches can be found in [192]. While the relaxations do ease some aspects of the problems, they normally do not decrease the dimensionality of the problem which remains demanding for large n.

4.3.2 Permutation Learning

Permutation learning aims to develop a model capable of predicting the optimal or best permutation matrix corresponding to a specific input data or label. While doubly stochastic matrices exist in a continuous domain and are therefore well suited for optimization, enforcing the sum-to-one constraints still poses a challenge, which is commonly addressed by approximating permutation matrices using Sinkhorn layers [1]. In 2011, Adams *et al.* [1] proposed the *Sinkhorn propagation* method, to iteratively learn doubly-stochastic matrices that allow the prediction of permutation matrices. This approach is based on the iterative *Sinkhorn normalization* [278], which aims the transformation of a non-negative square matrix $\tilde{P}$ to a doubly stochastic matrix by iteratively normalizing its rows and columns,

$$\tilde{P}^{k+1} = N_c\left(N_r\left(\tilde{P}^k\right)\right), \tag{4.19}$$

where N_c refers to the operation of column-wise, and N_r to the row-wise normalization. Conditions on the convergence of iterative Sinkhorn normalization were shown in [279]. So, to learn a doubly stochastic matrix, Adams *et al.* [1] proposed to backpropagate through a series of unrolled Sinkhorn normalization steps. Following this algorithm, Cruz *et al.* [262] proposed *Sinkhorn networks*, where they realize the Sinkhorn normalization inside the last layer of a CNN. Their objective is to predict a permutation matrix by a CNN, that gets as an input a permuted data sequence $\tilde{x}$, and is trained to match a ground truth permutation matrix $\hat{P}$, that was applied to generate the permuted sequence $\tilde{x}$:

$$\min_{\theta} \mathcal{L}(\hat{P}, \mathcal{G}_\theta(\tilde{x})) \tag{4.20}$$

Following, Mena *et al.* [208] proposed *Gumble-Sinkhorn networks*, where the predicted matrix is seen as a distribution of doubly stochastic matrices, which are sampled by adding Gumbel noise to the Sinkhorn layer. They demonstrated their approach on tasks like sorting and jigsaw puzzles. Motivated by the relaxation of permutation matrices to doubly stochastic matrices, Grover *et al.* [116] made efforts to relax a sorting operation to a unimodal row-stochastic matrix, such that the matrix rows must maintain a total sum of one while having a distinct arg max. More recent studies suggest circumventing the constraint of row and column sums being one by learning permutations in Lehmer code, whose matrix form is row-stochastic [79]. The Lehmer code represents a permutation by a vector $(c_p(1), c_p(2), \ldots, c_p(n))$, where each entry contains the number of elements with a smaller index, but a higher rank in the permutation.

4.3.3 3D Shape Matching and Functional Maps

3D shape matching refers to the task of finding correspondences between given 3D shapes, represented as e.g. meshes or point clouds. Those correspondences set different shapes in relation to each other by assigning their respective points. For example, they might connect specific points of a human figure to the corresponding points of another figure. There

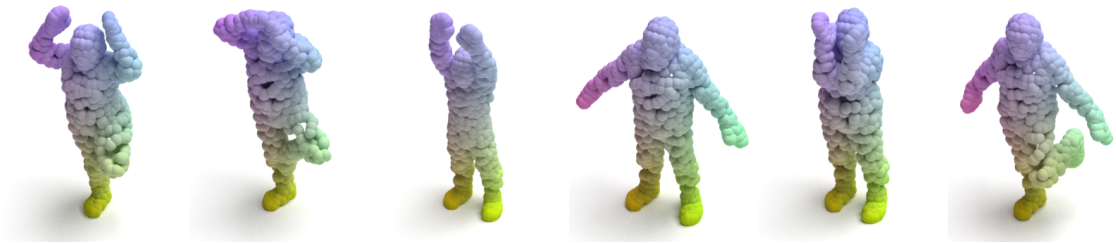


Figure 4.3: Six human shapes of different poses, generated from point-cloud data of the FAUST dataset [26].

are two primary types of shape matching, where the first is the matching of shapes with *rigid deformations*, where shapes are related to each other through rotation and translation. The second, more complex type is shape matching of *non-rigid deformations*, which also includes shapes that have different forms or poses from each other, as illustrated in Figure 4.3 [260].

For shape matching of non-rigidly deformed shapes, a suitable distance measure between points on the shapes is the geodesic distance. The geodesic distance of two points on a 3D shape describe the length of shortest path between those points along the shape's surface [72]. While the exact calculation of the geodesic distance is computationally expensive, faster approximations approaches exist [289]. Several methods of calculating the geodesic distance are discussed in [72].

3D Shape Correspondences

The 3D shape correspondence problem can be posed as an assignment problem between the sets of vertices of a pair of shapes, for example through point descriptors, capturing the shape properties, matched by a LAP, or a QAP aiming to preserve distances between all point pairs.

However, 3D shapes are often discretized with thousands of vertices which makes optimization for a permutation computationally challenging. Hence, the permutation constraint is often relaxed [251] and, even though the tightness of relaxation might be known [23, 92], the optimization variables still scale quadratically with the problem size. In [106] and [303] the QAP is deconstructed into smaller problems and then each of them is optimized with a series of LAPs, while [268] solve for permutations as a series of cycles that gradually improve the solution. Because permutation constraints for large resolution become infeasible, and, hence, the restriction to cases with the same number of vertices, recent methods often do not impose these constraints at all. Lines of work rely on a given template to constrain the solution [115, 288], impose physical models that regularizes the deformation between inputs [96, 95, 231], or learn a solution space through training approaches [45, 180, 203, 97].

In particular *functional maps* [230] play an important role in order to circumvent the high dimensions of permutations.

The Functional Maps Framework

The functional maps framework, as proposed by Ovsjanikov *et al.* [230] in 2012, simplifies the shape matching problem by avoiding the direct computation of point-to-point correspondences. Instead, it focuses on the relationship between functions defined over

the shapes, which are correlated by a matrix C. Especially for shape matching problems with a large number of vertices, the functional map framework is of high interest, as it opens the possibility to transform the problem into one that can be handled by a much smaller matrix C (if a low-dimensional basis set is chosen).

Let us discuss the basic idea behind functional maps. Given the shapes X and Y, which can be represented, for example, by a set of vertices, and assuming a bijective mapping T between these 3D shapes capturing their relation,

$$T : X \to Y. \tag{4.21}$$

Within the context of functional maps, this mapping induces a new linear mapping T_F,

$$T_F : \mathcal{F}(X) \to \mathcal{F}(Y) \tag{4.22}$$

in the function spaces of X and Y, where $\mathcal{F}(X)$ and $\mathcal{F}(Y)$ represent the spaces of real-valued functions on X and Y, respectively.

The function spaces $\mathcal{F}(X)$ and $\mathcal{F}(Y)$ are each defined by a set of basis functions, $\{\phi_i^X\}$ and $\{\phi_j^Y\}$ respectively, s.t. within this framework, the transformation T_F can represented by a matrix C. Under the assumption that the basis vectors are orthonormal, the entries of the functional map matrix C are given by the inner products:

$$C_{i,j} = \langle T_F(\phi_i^X), \phi_j^Y \rangle. \tag{4.23}$$

C can be calculated using additional information, including known correspondences between points or shapes, or specific conditions on the transformation. For instance, in [203] the mapping has to preserve learned point descriptors. A common approach of calculating the mapping $C \in \mathbb{R}^{m \times m}$ is by minimizing

$$\arg\min_C \|C(\phi^X)^{-1} K_X - (\phi^Y)^{-1} K_Y\|_2 \tag{4.24}$$

for given bases and n pointwise descriptor functions K_X and K_Y, which are assumed to be preserved by the unknown mapping. The matrices $(\phi^X)^{-1} K_X$ and $(\phi^Y)^{-1} K_Y$ in $\mathbb{R}^{m \times n}$ represent the coefficients of the descriptor functions in the chosen bases and the minimization in (4.24) aims to find the matrix C that aligns the shapes in the corresponding function spaces. As base functions, originally the authors propose to use the eigenfunctions of the Laplace-Beltrami operator [238], which has been also used in the majority of follow-up works, as e.g. in [252, 225, 180].

In the context of this thesis, we aim to calculate the pointwise mappings in the form of permutation matrices. Assuming X and Y have the same number of vertices, n, and the mapping T is bijective, we aim to find the underlying $n \times n$ permutation matrix P. For orthogonal $\{\phi_i^X\}$ and $\{\phi_j^Y\}$, each containing m basis functions, the relation between $C \in \mathbb{R}^{m \times m}$ and $P \in \{0,1\}^{n \times n}$ can be formulated by

$$C = (\phi^Y)^{-1} P \phi^X \iff P^T = \phi^Y C (\phi^X)^{-1}, \tag{4.25}$$

where C can be interpreted as an alignment of the base functions, and the original permutation can be recovered by calculating nearest neighbor of ϕ^X and $\phi^Y C$.

As for $m \ll n$, in practice, it is not always possible to recover the optimal correspondence, leading to research containing learning techniques. First, only the optimal features were learned [180, 123], followed by work on learning basis functions. Instead of using

the Laplace-Beltrami eigenfunctions as the choice of basis, Marin *et al.* [203] introduced to learn the basis from data, demonstrating improved robustness and accuracy for point cloud-based shapes. They proposed to learn the basis by optimizing

$$\min_{\theta} \sum_{(X,Y)\in\mathcal{D}_t} \|P(\phi^X, \phi^Y)X - \hat{P}X\|_2^2 \tag{4.26}$$

from some dataset $\mathcal{D}_t$ where $\phi^X = \mathcal{G}_\theta(X)$, and $\phi^Y = \mathcal{G}_\theta(Y)$ are predicted by a neural network, $\hat{P}$ is the ground truth permutation, and $P(\phi^X, \phi^Y)$ is calculated from the learned basis functions. In an ℓ_2 norm loss, a second network is trained to learn the optimal transformation by generating the linear transformation matrix C from predicted point descriptors and matching it with a given ground truth matrix. Subsequently to [203], Cao and Bernard proposed a learning approach for both modalities (point cloud and mesh) [45].

Part II

Methodology

CHAPTER 5

Learning and Modelling for Single Image Segmentation

This chapter is based on the publication in [88] and focuses on the analysis and the comparison of classical model-based techniques for image segmentation and modern learning-based approaches. Specifically, we focus on the role of user-induced scribbles and semantic information in single-image segmentation. For an overview of previous segmentation approaches, Section 4.2 provides an overview of classical model-based methods, especially in terms of energy minimization problems, and successful learning-based segmentation methods.

5.1 Introduction

Image segmentation refers to the problem of dividing an image into meaningful non-overlapping regions. Unlike the classical model-based methods, learning-based models are trained on extensive datasets to learn image segmentation. Yet, the limitation to only predict those semantic labels (and objects) present in the training database limits the applicability of such models, particularly because the sole definition of image segmentation as a division into "meaningful" non-overlapping regions makes image segmentation an ill-posed task by definition: What is a meaningful region? The answer, of course, is highly subjective and depends on the specific intended application, as illustrated in Figure 5.1.

This is the reason why we believe that image segmentation on only one image based on scribbles, i.e., the prediction of regions in a single image that have previously been marked by a few strokes, so-called *scribbles*, by a user (see Figure 5.2), remains highly relevant, e.g. for image editing software.

Unfortunately, hardly any works in the area of deep learning focus on single image segmentation from scribbles. This poses the fundamental question if model- or learning-based approaches represent the state-of-the-art in this field, along with the quest for network architectures and regularization schemes that are well-suited for single image segmentation.

We study these questions in two different scenarios: 1) The case where the current scribbled image represents the sole source of information, and 2) the case in which prior information from a segmentation benchmark may be utilized in the form of transfer learning or accessing features of segmentation networks.

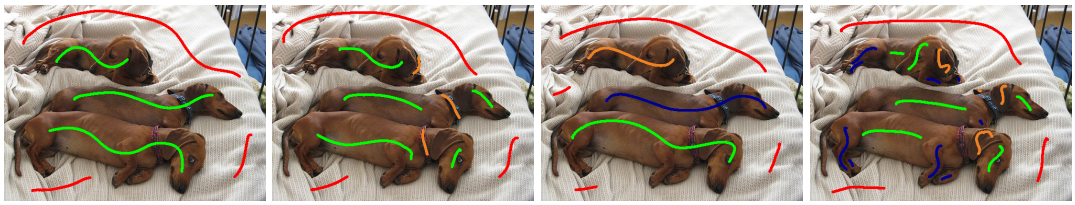


Figure 5.1: Four different ways to place scribbles into an image for segmentation, depending on the user's specific intention.

scribble | SVCD [222] | ...with semantic information

CNN | ...with spatial information | ...with semantic information

Figure 5.2: Single image segmentation based on scribbles that rely on image color and on optional spatial or semantic information: The upper row shows the segmentation by SVCD, the second row shows the segmentation by a CNN.

For the first scenario we demonstrate that the additional inclusion of spatial information in a neural network improves the segmentation compared to color-only images. We also present how color and spatial information can be optimally weighted against each other for segmentation using double backpropagation. Yet, the model-based method remains superior to neural networks. In the second scenario, we propose a hybrid technique that combine the (model-based) spatially varying color histogram from [222] with learning-based soft semantic features from [4] and yields results that outperform the segmentation by neural networks and the stand-alone model-based approach. Here we focus on segmenting one image with scribbles without using any semantic information about the objects in the scene.

Spatially varying color distributions [222] for $c(f)$ in (4.11), and using an edge-weighted (or even nonlocal) TV regularization marked the state of the art in 2014. Since then, deep learning has proven to dominate any segmentation application for which at least a moderate number of training examples is available. Thus, we believe it is high time to ask if modern network architectures are able to outperform model-based approaches *even on a single image*.

We first consider an image segmentation problem for which all methods rely solely on

the given scribbled image (such that, on a pixel level, we still have have several hundred training examples). Subsequently, we study how to incorporate prior information in the form of a different dataset with ground information. As transfer learning appears to fail (detailed in Section 5.3.1), we propose to instead use *soft semantic features* of Aksoy *et al.* [4], that have been successfully used in soft semantic segmentation without scribble information.

5.2 Model- and Learning-based Segmentation Methods

In both settings (with and without prior information), we compare the following approaches: *Spatially varying color distributions (SVCD)* [222] are used to model smoothly changing histograms in space to approach the scribble-based segmentation problem via solving a convex relaxation of (4.11) followed by a thresholding. It does not involve any learning. To integrate additional semantic information, we concatenate the RGB values with the soft semantic features prior to applying the method.

To mimic the behavior of color histogram-based approaches, we train *pixel-wise networks (PWNs)* $\mathcal{G}_\theta(x)$ with learnable parameters θ that get the vector $x \in \mathbb{R}^{n_c}$ of RGB values at a single pixel as an input and are suggesting a class label solely based on color. To additionally incorporate the idea of spatially varying color distributions in a learning-based setup, we also train PWNs with a 5-dimensional input vector consisting of the RGB values as well as the xy-coordinates of the image (normalized to a range of $[0, 1]$). Similarly, semantic features are just concatenated with other inputs. In terms of the network architecture, an extensive empirical search resulted in surprisingly small and shallow structure, consisting of two layers with 16 neurons per layer and Leaky-ReLU activations.

As PWNs might have too little spatial context to make faithful predictions, we additionally consider *CNNs* with larger receptive fields: Using larger convolution kernels and increasing the depth of the networks allows us to provide the network with more and more non-local information. Again, we evaluate networks that use the plain RGB input image as an input and concatenated it in the channel dimension with its xy-coordinates and/or the semantic features. In an ablation study detailed in Section 5.3.2 we again found a rather shallow network of *depth 2*, *width 16*, and a *kernel size of 3* to be most successful.

In our experiments we found that the inclusion of spatial information often dominates the results of the CNNs, such that objects close to the scribble are incorrectly segmented. For this reason, we introduce a regularization term similar to the idea of *double backpropagation* from [164] by computing

$$\min_\theta \mathcal{L} + 10^{-7}\beta||\nabla_{x_s}\mathcal{L}||_1,\ \mathcal{L} = C(\mathcal{G}_\theta(x_c, x_s), sc). \tag{5.1}$$

Here the network loss $\mathcal{L}$ is computed by the cross-entropy (C) between our scribbles and the output of the network $\mathcal{G}_\theta$, using the color information x_c and the spatial information x_s. We regularize the ℓ_1 norm of the gradient of our loss function, a form of TV regularization, in the spatial direction to attenuate the influence of x_s on the our final result. Figure 5.3 visualizes the effect of increasing the regularization on the segmentation, whereby without regularization, structures close to the object are segmented as those, and with a too strong regularization, the color information predominates. We refer to this regularized approach as *CNN+reg.*.

As one of the most famous architectures for semantic image segmentation, we finally consider the *U-Net* architecture from [255]. It is a convolutional architecture with a receptive field that spans large portions of the image while still being able to preserve fine

$\beta = 0$ $\beta = 0.05$ $\beta = 0.1$

Figure 5.3: Segmentation with color and weighted spatial information via double back-propagation as network input.

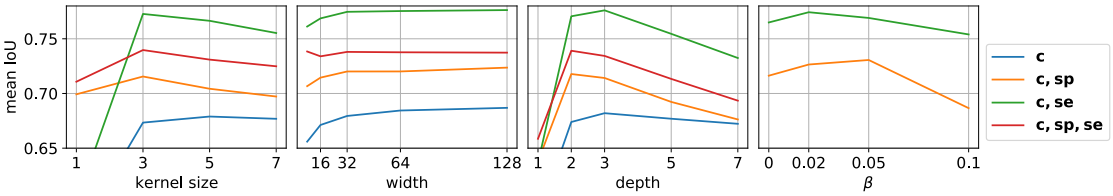


Figure 5.4: Study of the impact of the convolutional neural network from left to right: kernel size, width, and depth on the segmentation accuracy for different types of network input: **c**olor image, **sp**atial pixel position and **se**mantic information. The most right image shows the regularization of the spatial impact on the segmentation as defined in (5.1) (orange plot), as well as the corresponding penalty w.r.t. variations of the color channels (green plot).

details. While this architecture would clearly be superior in a fully supervised setting with sufficient training data, our investigations aim at an understanding how the strong overparameterization of such an approach in comparison to the small number of labeled (=scribbled) pixels in a single image affects its accuracy. We took the U-Net architecture from [255] as a basis and conducted a study on the number of downscaling steps of the network architecture. We tested U-Net architectures without any downscaling steps up to four steps and observed a decrease in segmentation accuracy for U-Net networks with more downscaling steps.

5.3 Numerical Evaluation

To evaluate the above approaches, we use scribbled images from [178] of the Pascal VOC2012 dataset [98] for which ground truth segmentation are available. We tune the hyperparameters of each method on a fixed set of 200 images of this dataset. Table 5.1 shows the best results we were able to attain for each class of methods, with the left part of the table depicting the single image segmentation results without additional information (using color (**c**) and spatial (**sp**) information only), and the right part additionally allowing the use of semantic (**se**) features from [4].

As we can see, the model-based SVCD approach outperformed all learning-based approaches in terms of pixel accuracy as well as mean intersection over union (IoU). While other fields, e.g., in image reconstruction with pioneering work on DIPs in [296], indicated that the architecture of common convolutional networks has a regularizing effect that is well suited for natural images and thus allows a self-supervised training on a single image, we cannot confirm that similar effects appear in image segmentation.

	c	**c** & **sp**	**c** & **se**	**c** & **sp** & **se**
SVCD	–	0.775	–	0.845
PWN	0.607	0.721	0.689	0.731
CNN	0.673	0.715	0.772	0.739
CNN + reg.	–	0.731	0.774	–
U-Net	0.654	0.642	0.658	0.648

Table 5.1: Summary of the best mean intersection over union each of the pure single image segmentation methods could attain for different types of network input: **c**olor image, **sp**atial pixel position and **se**mantic information. As we can see, the additional inclusion of spatial information improves segmentation, which in turn is outperformed by the inclusion of semantic information in the neural network. SVCD still outperforms the learning based methods.

Interestingly, the inclusion of the spatial coordinates as inputs to the neural network helped to improve all learning-based approaches except the U-Net. Moreover, including our proposed regularization to avoid an overfitting to the spatial information only, gave the best result among the learning-based approaches. Based on the rather small CNNs that our ablation study in Section 5.3.2 found to be optimal, and the surprisingly bad performance of a U-Net architecture which is not even influenced by the inclusion of semantic information, we conclude that overfitting remains a significant problem in single image segmentation with neural networks.

Except for U-Net, the combination of semantic and color information increases the performance of all methods significantly. In particular, the combination of the model-based creation of spatially varying color histograms with semantic soft features achieves excellent results. Interestingly, the additional inclusion of spatial coordinates on top of semantic features does not appear to be beneficial for CNNs anymore. The slight improvement of CNN+reg. over CNN was obtained similar to (5.1), but using the gradient with respect to the color input instead of the spatial coordinates. As the proposed approach of using semantic soft features is only possible if a second annotated dataset is available, *transfer learning* is a natural baseline for such approaches.

5.3.1 Transfer Learning

A common method in semantic segmentation is the fine-tuning of a neural network pre-trained on a given fully supervised dataset. Thus, we finetune the architectures E-Net [236] (pre-trained on the CityScape [70] dataset) and DeepLabv3 [55] (pre-trained on the CityScape dataset [70]) , on single scribbled images. Despite varying the amount of parameters to freeze and train, the best mean IoUs were found to be 0.52 for E-Net and 0.59 for Deeplabv3. Surprisingly, these values are not even close to the results seen in Table 5.1 even without semantic features. We conclude that – at least for the significantly different datasets of CityScape and VOC2012 – it is not straightforward to utilize transfer learning for single image segmentation with scribbles.

5.3.2 Ablation study for Convolutional Neural Networks

To study the impact of the architecture, we train simple CNNs with alternating convolution and ReLU layers of varying width and depth along with a cross-entropy loss on the scrib-

bled pixels only. By expanding the kernel size from 1×1 convolutions (which is equivalent to our PWNs), the segmentation networks start to include information from neighboring pixels in the predicted segmentation. Figure 5.4 shows how the mean IoU depends on the neural networks widths, depths, kernel size, and parameter β of our proposed regularization for different inputs using color (**c**), semantic (**se**), and spatial (**sp**) information. Here the non-variable values in the graphs are fixed to *width = 16*, *depth = 2*, *kernel size = 3*, and $\beta = 0$. Given the above parameters and all the input variations shown in Figure 5.4 we could measure a variance of the mean IoU of ±0.002 in our experiments. As we can see, rather shallow networks of only 2 layers are more successful than deep ones, while the width has little effect as long as the network consists of at least 16 channels. Finally, the kernel size was found to be optimal for 3×3 convolutions, and moderate values of the regularization parameter β do allow to increase the mean IoU by over 0.01.

5.4 Conclusion

In this chapter, we have shown that image segmentation based on user-drawn scribbles is a challenging problem where model-based approaches still perform better than machine learning. Instead of transfer learning approaches, including soft semantic features as additional input channels to an energy minimization approach using spatially-varying histograms showed the most promising performance. While our modifications, which include incorporating spatial coordinates as inputs and simultaneously applying regularization, have improved the mean IoU of learning-based approaches, it remains an interesting challenge for future research to develop architectures, regularizations, and training schemes that can outperform model-based approaches even on single image segmentation without prior information.

CHAPTER 6

Guided Computed Tomography Reconstruction by a Learned Prior

The previous chapter demonstrated that cleverly designed model-based approaches not only can outperform learning-based methods but also benefit from semantic information obtained from previous training processes. This chapter is based on the publication in [86] and handles the induction of a pre-trained classifier in a model-based reconstruction method. Unlike in the previous chapter, we do not compare model- and learning-based methods for reconstruction, but shift the optimization of an underdetermined problem using learned semantic information. Specifically, we work on CT reconstruction problems, which are introduced in Section 4.1.

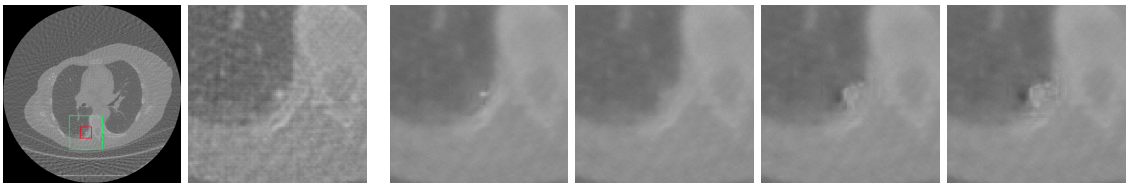
6.1 Introduction

CT plays an important role in medical imaging with many applications, such as diagnosing various health conditions and devising appropriate treatment plans [326, 177]. For recording CT data, the target (e.g. a patient) is projected with X-ray radiation from various directions comprising half a circle around it, while a detector measures the attenuated radiation at the other side of the target. Measurements corresponding to all projections are then organized as an array termed sinogram, from which the CT image can be reconstructed using different reconstruction methods.

However, the exposure of a patient to the ionizing X-ray radiation is known to present significant health risks such as cancer. This fuels a substantial research effort for reducing radiation exposure, for example by using *sparse-view CT*, where the target is radiated with fewer projection angles, typically distributed uniformly around it [150]. Unfortunately however, reconstructing the CT image from the recorded sparse-view data becomes an underdetermined problem, which often manifests itself as significant ambiguities in the tomographic reconstruction process.

The reconstruction of a tomographic image u from a measured sinogram f captured using q projection angles can be formulated as a linear inverse problem of the form

$$f = Ru + e, \tag{6.1}$$



(a) Reconstruction by FBP (b) Data Consistent Reconstructions (our approach)

Figure 6.1: Reconstruction methods such as filtered backprojection (FBP) [99] typically yield only one data-consistent output (a). However, there are many possible reconstructions (b) that are consistent with the data term, but differ in their pathological categorization, i.e. with respect to their malignancy. This ambiguity, which increases when using sparse-view CT, is demonstrated here for $q = 50$ projection angles.

where $\boldsymbol{R} \in \mathbb{R}^{q \cdot d \times N}$ corresponds to the discrete *Radon transform*, which computes $q \cdot d$ line integrals through image u (with a total number of N pixels) along all projection directions[1]. Here, d is the number of pixels in the one-dimensional X-ray detector and $e \in \mathbb{R}^{q \cdot d}$ is some additive noise. As the number of projection angles q decreases, the problem of recovering image u in (6.1) becomes increasingly underdetermined.

Over the years many methods attempted to tackle this challenge, typically producing a reconstruction $\hat{u}$ which strives to be as close as possible to the ground truth image u. However, due to the underdetermined nature of the problem, there are many different valid image reconstructions $\hat{u}$ whose Radon transform $\boldsymbol{R}\hat{u}$ matches the measured sinogram f. This is demonstrated in Figure 6.1 for the case of a lung nodule captured using $q = 50$ projection angles. While all four reconstructed images on the right are consistent with the sinogram f (satisfying $\frac{1}{qd}\|f - \boldsymbol{R}\hat{u}\|^2 < 3 \cdot 10^{-5}$), their appearances, and more importantly their medical interpretations vary dramatically, with an increasing level of malignancy from left to right.

In this chapter we point out the ambiguity that is inherent to medical data reconstruction and argue that enabling exploration of the space of consistent reconstructions, rather than producing a single arbitrary image, is essential in medical applications. We propose the first method to allow this, which enables exploring the range of possible image reconstructions $\hat{u}$ that are consistent with the measurement f, while potentially corresponding to different pathological findings. Our method operates by optimizing for different solutions, whose Radon transform matches with the measured sinogram while corresponding to semantically different interpretations, obtained from a pre-trained CT image classifier. In particular, we use gradient descend to minimize the data term induced by (6.1), as well as a term that encourages the resulting image $\hat{u}$ to be classified into different malignancy levels by a classifier that was trained to distinguish between malignant and benign tissues. We introduce technical novelties such as the use of an adversarially trained classifier and the sole use of energy minimization for solution exploration, which is easier and typically more stable for training than, e.g., GAN frameworks. We demonstrate our method on the case of reconstructing human lung CT images with pulmonary nodules, such that they correspond to various degrees of pathological malignancy while maintaining consistency with the measurements f. Nonetheless, extending our approach to other use-cases, as well as to other medical imaging modalities, would be fairly straight forward.

[1] With a slight abuse of notations, we use u and f when referring either to the two-dimensional or to the column-stacked versions of the target image and the recorded sinogram, respectively.

6.2 Explorable Computed Tomography Reconstruction

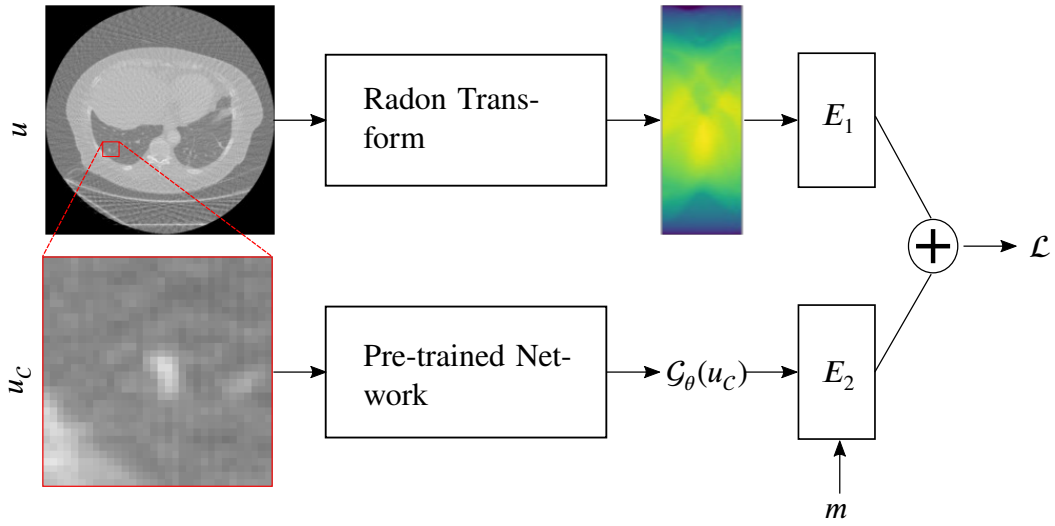


Figure 6.2: Overview of the proposed approach. Loss function $\mathcal{L}$ is composed of two energy terms E_1 and E_2, which take as input the Radon transform of the desired reconstruction x and the output of a pre-trained classification network, respectively, as well as the desired malignancy level m. The input of the classification network is a crop u_C from image u.

Exploring Data Consistent Reconstructions The goal of our work is to provide medical experts that are interpreting images for diagnostic purposes with a better understanding of the actual information the recorded data contains about the object of interest. While our idea extends to any property that can be captured by a classification (or scalar regression) network, we exemplify our method by exploring the space of possible CT reconstructions of nodules associated with different degrees of malignancy as predicted by a given classification method. An overview of our method is depicted in Figure 6.2. As a malignancy classifier, we use a classification network $\mathcal{G}_\theta : \mathbb{R}^{h\times w} \to [0,1]$ pre-trained for classifying nodules in chest CT. The network predicts the malignancy of a nodule from the region of interest u_C, which is manually chosen and cropped from a CT image u around the nodule, as shown by the red box in Figure 6.1 (a).

Because we want to predict physically plausible, i.e., data consistent, solutions only, we constrain our reconstructions to the solution space $S := \{u \mid \frac{1}{qd}\|Ru - f\|^2 \leq \delta^2\}$ of images u whose sinogram Ru differs from the measured data f by a noise-level dependent constant δ. To allow comparing reconstructions with different number of projection angles q, we normalize the squared ℓ_2 norm by $1/(qd)$ (approximating the ℓ_2 norm in function space more realistically). Within S we explore possible solutions using a target malignancy level m for our classification network $\mathcal{G}_\theta$ by finding

$$\min_{u\in[0,1]^N} H_\epsilon(\mathcal{G}_\theta(u_C) - m) \text{ s.t. } u \in S, \tag{6.2}$$

where H_ϵ is the Huber loss [137] with $\epsilon = 0.01$, which we found to work best empirically, being a trade-off between the ℓ_1 and ℓ_2 norms:

$$H_\epsilon(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \epsilon, \\ \epsilon \cdot \left(|a| - \frac{1}{2}\epsilon \right), & \text{otherwise.} \end{cases} \tag{6.3}$$

While (6.2) could be optimized (at least locally) using a projected gradient descent approach, the projection is rather computationally intense such that we propose to instead consider the regularized problem

$$\min_{u \in [0,1]^N} \frac{1}{qd} \| Ru - f \|^2 + \lambda H_\epsilon(\mathcal{G}_\theta(u_C) - m) \tag{6.4}$$

with λ indicating a weighting of the malignancy prediction of interest.

Transformations Our goal is to produce realistically looking (rather than unnatural) CT reconstructions corresponding to malignancy levels m. We therefore utilize transformed versions $T_j(u_C)$, where $\{T_j\}_{j=1}^J$ is a set of natural image transformations like different rotations and scalings, which do not affect the semantic interpretation of the image. This reduces the chance of yielding an unrealistic u_C that manages to "fool" the classifier $\mathcal{G}_\theta$, as we visualize in Figure 6.3, highlighting that using transformations can oppress noisy results. This leads to the modified objective

$$\hat{u}(m) = \arg\min_{u \in [0,1]^N} \underbrace{\frac{1}{qd} \| Ru - f \|^2}_{=E_1(u)} + \underbrace{\lambda_1 \frac{1}{J} \sum_j H_\epsilon(\mathcal{G}_\theta(T_j(u_C)) - m) + \lambda_2 \mathrm{TV}(u_C)}_{=E_2(u)}, \tag{6.5}$$

where we use notation $\hat{u}(m)$ to stress the dependency of the reconstructed $\hat{u}$ on our exploration parameter m. To further encourage smoothness, we add TV regularization [257] to our energy function.

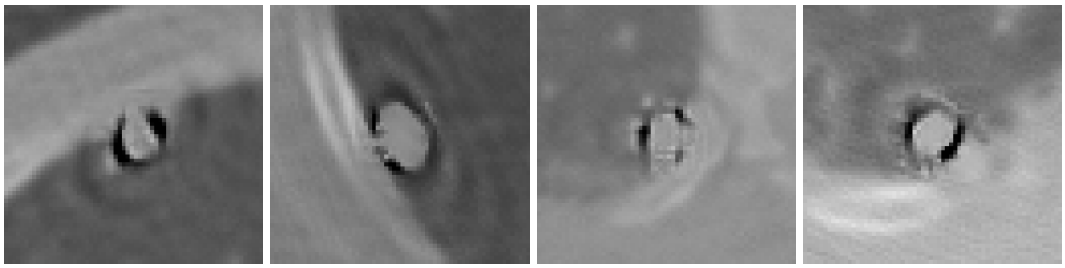
Soft Cropping We empirically found that hard cropping u to obtain u_C often results in visible artifacts in $\hat{u}$ around the cropping boundary.

To encourage a smooth transition of the crop to the remaining part of $\hat{u}$, we attenuate the gradient of E_2 in (6.5) with a Gaussian mask G, so that modifications to the peripheral pixels of u_C are attenuated, as we visualize in Figure 6.3, clearly showing a hard border around the crop, that could be prevented by using soft cropping. Our gradient descent update can be written as

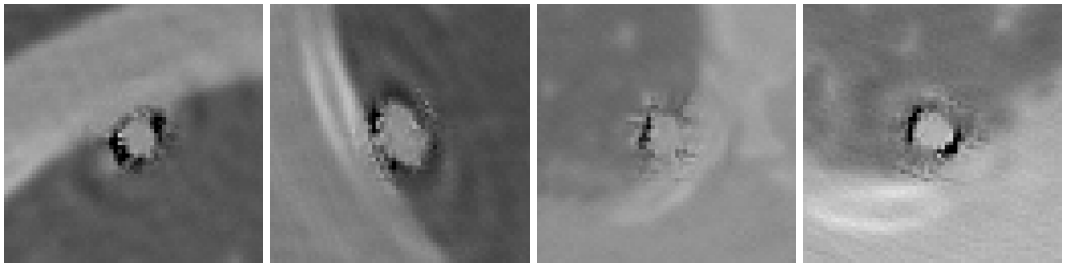
$$u^{i+1} = u^i - \tau(\nabla E_1(u^i) + G \odot \nabla E_2(u^i)), \tag{6.6}$$

where $\odot$ denotes point-wise multiplication.

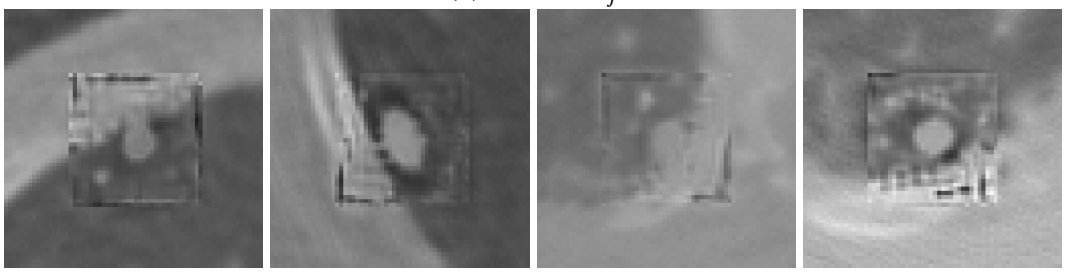
Training Suitable Classification Networks Modern image classification models can be susceptible to adversarial examples – small perturbations in the input image that cause misclassification. To further encourage the reconstruction to be meaningful and realistic, and to prevent slight imperceptible changes in $\hat{u}$ from affecting the classification by $\mathcal{G}_\theta$, we utilize adversarial training for classifier $\mathcal{G}_\theta$ using the Fast Gradient Sign Method [112], thus making the classifier more robust.



(a) Full method



(b) Without T_j



(c) Without soft cropping

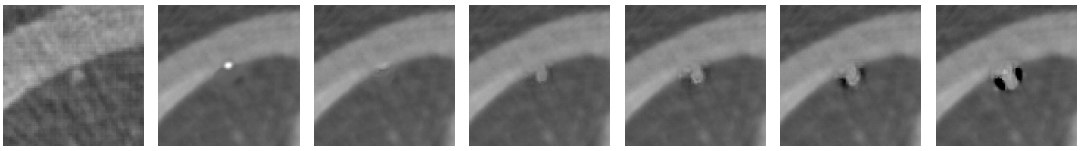
Figure 6.3: Results for different versions of our method. All reconstructions are obtained from 100 projection angles. The images in the first row show reconstructions of our full method. The images in the middle row are obtained without using the transformations T_j. The images on the right are created without soft cropping (but with an application of the transformations).

Parallels to Adversarial Techniques Note that although our method involves optimizing over the reconstructed image u to achieve the desired prediction by a pre-trained classifier (similarly to adversarial attacks), it does not entail any alteration of the recorded sinogram f, and instead operates on the reconstructed image u in a data consistent manner. This makes our method different from adversarial attacks, despite the mathematical similarity manifested in (6.2). Note that set S of possible inputs to the classifier is unbounded, which is a significant difference on the technical side as well.

6.3 Numerical Evaluation

6.3.1 Preparation

Dataset For our experiments, we use the Lung Image Database Consortium Image Collection (LIDC-IDRI) [9, 68], including over 1000 cases that were annotated by four radiologists independently. There are 5 levels of grading, depending on how certain radiologists are that the nodule is malignant (or benign). To create our training annotation, we aver-



(a) FBP (b) $m = 0.0$ (c) $m = 0.1$ (d) $m = 0.2$ (e) $m = 0.3$ (f) $m = 0.5$ (g) $m = 1.0$

Figure 6.4: Nodule reconstructions using FBP (a), and various degrees of malignancy with our proposed approach in (b) – (g).

aged the annotated levels for each CT image, and discarded the data whose classification is the closest to *indeterminate* (level 3), as it can be considered neither malignant nor benign

From the remaining data, we extract those 2d slices that contain the annotated nodules and get a training dataset of 244 scans with malignant nodules and 729 scans with benign nodules. All CT images are normalized to a range of $[0, 1]$. For optimizing ((6.5)), we use a validation dataset containing a total of 100 scans with 50 malignant and 50 benign cases, which was not used for training the classification network.

Classification Network To classify nodules of sparse-view CT reconstructions, we use BasicResNet, since it has shown superior results for classifying the malignancy of nodules in [5], by adapting their hyper-parameter settings (with minor changes) and training for 350 epochs using the Adam optimizer and a learning rate of 0.0005. All training input images were normalized by subtracting their mean and dividing by their standard deviation.

6.3.2 Solution Space Exploration

In all our experiments described below, we optimize (6.5) along with the Gaussian damping of (6.6) with a variance of 11, using gradient descent with a learning rate of 1.0 and $\lambda_1 = 1.0$ and $\lambda_2 = 0.01$ for 50000 iterations, with the stopping criterion triggered when the energy no longer decays.

To start our method with an u that already results in low energy, we beforehand calculated the FBP (u^{FBP}) of our input and minimized E_1 for 600 iterations using gradient descent with a learning rate of 0.0005 and a momentum of 0.9. The choice of parameters were obtained empirically. In each optimization step, we normalize the input u_C of $\mathcal{G}_\theta$ with a fixed mean and variance, taken from the FBP reconstruction.

Realistic Solution Space We explore the space of underdetermined CT reconstructions, with $q = 50$ projection angles, and consider reconstructions of different malignancies, which we control by setting the variable m in (6.5). Figure 6.4 shows an example of reconstructions of a nodule for multiple levels of malignancy, starting from a benign ($m = 0.0$) to a malignant nodule ($m = 1.0$) and several values in between. The prediction of the nodule reconstructed with FBP (a) is classified as benign with $\mathcal{G}_\theta(u_C^{FBP}) = 0.2$. It can be seen that in most extreme cases and especially the cases with strong deviation from the predicted malignancy towards increasing malignancy, artifacts appear in the nodule, which here appears as black areas around the nodule. From $m = 0.5$ onward they become visually unrealistic. Numerically we found that rather small changes of m differing by the prediction on the FBP reconstruction by around ± 0.1 yields realistic images while still causing significant changes in the appearance of the nodule, see Figure 6.4, (c) – (e).

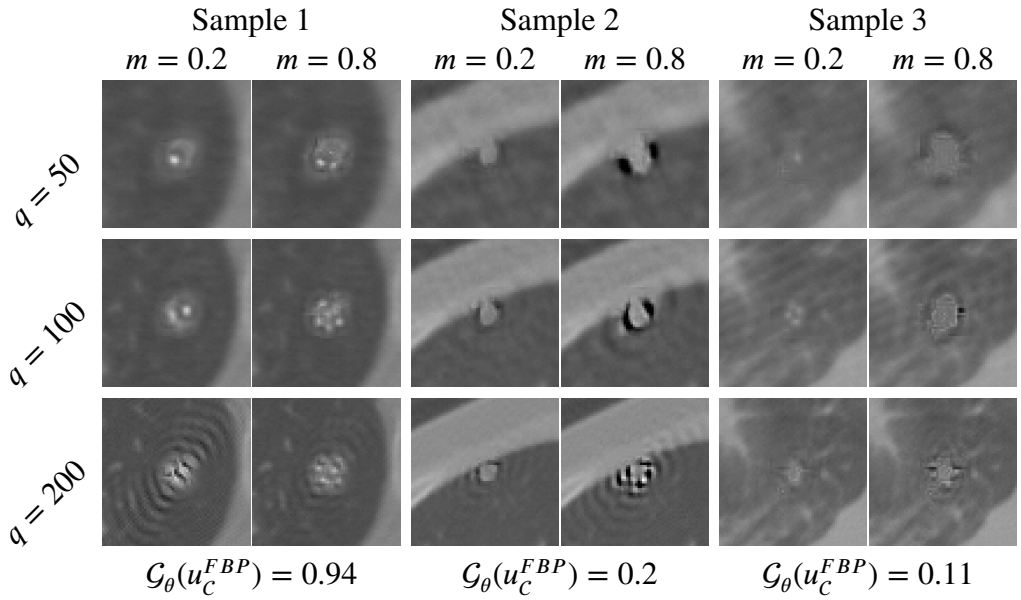


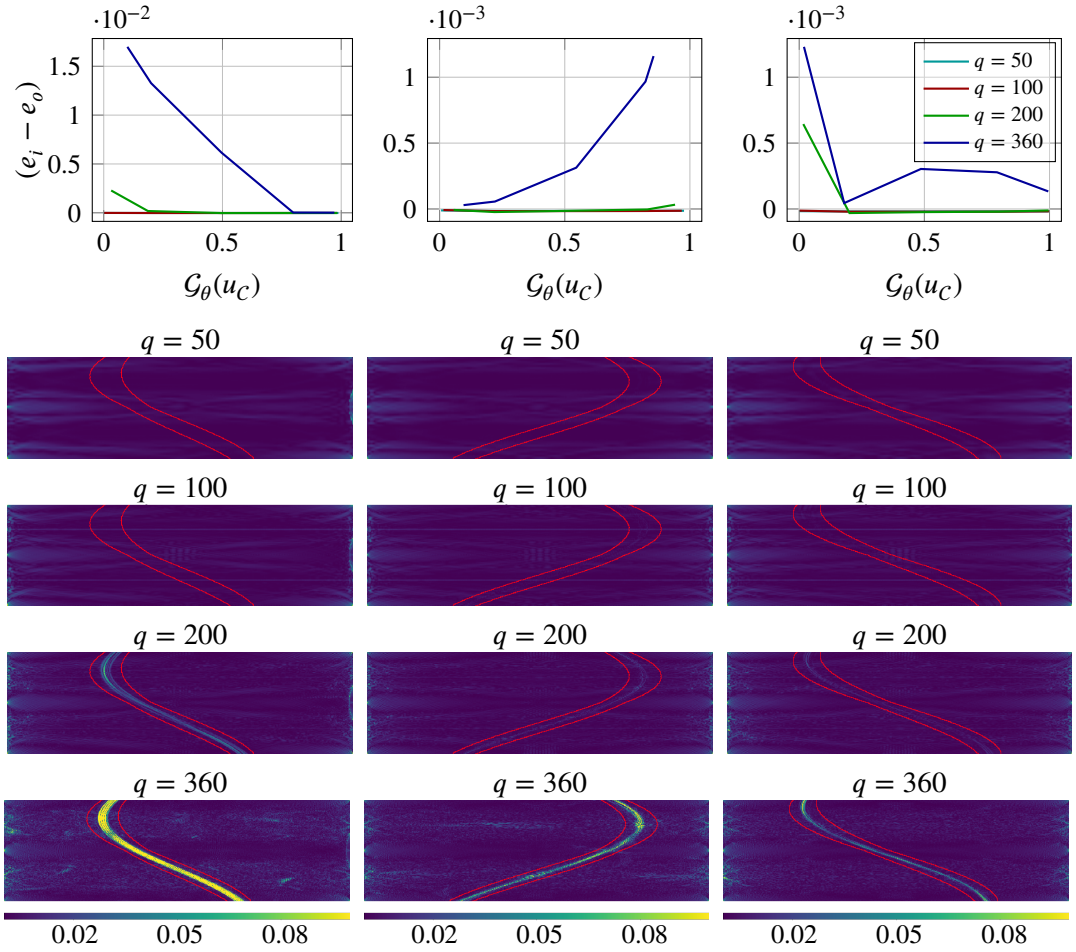
Figure 6.5: Examples of data consistent reconstructed nodules for a varying number of projection angles q for $m = 0.2$ and $m = 0.8$, such that the reconstructed nodules are categorized by the classification network $\mathcal{G}_\theta$ into different classes with respect to their malignancy.

In the following, we refer to the prediction of the reconstruction by FBP as the original malignancy.

Investigation on the Residuals An important question is how much the nodules can change in their appearance and malignancy while still maintaining data consistency. Because this has to depend on the number of projections recorded in the sinogram, we consider reconstructions with 50, 100, 200, and 360 angles and optimize (6.5) towards malignant and benign reconstructions. Exemplary reconstructions with rather large variations of the malignancy level m are illustrated in Figure 6.5 for varying numbers of projection angles q. One can see that fewer projections tend to allow larger variations in the reconstructions, e.g. allowing the nodule to almost disappear for $q = 50$ projections in *Sample 3*. For a large number of $q = 200$ projections strong deviations from the original malignancy $\mathcal{G}_\theta(u_C)$ can lead to severe artifacts that do not correspond to a medically realistic reconstruction anymore.

The severe visual artifacts raise the question to what extent such reconstructions even remain data consistent. Therefore, we analyze the behaviour of the residual $r = |Ru - f|$, that shows the pointwise distance of the sinogram of the reconstruction Ru and the measurement f. It is visualized in Figure 6.6 for each aforementioned sample for the malignancy that is opposite to its original classification. Here, the red marking indicates the area in the residual that has an influence on the nodule in the reconstruction. We can see that a modification in the nodule (within the red marking) is easier to recognize the more projection angles were used for the reconstruction. For fewer projection angles, such as $q = 50$, it is possible to modify the nodule to another malignancy without any sign of the exploration in the residuum.

To quantify this effect we compute the mean squared error of all points insight the red track of each nodule as well as outside of it, and denote them by e_i and e_o, respectively.



(a) Sample 1 (b) Sample 2 (c) Sample 3

Figure 6.6: Residual and error measurement for three examples of data consistent reconstructed nodules for a varying number of projection angles q. Top: difference of the interior and exterior error when optimized towards different malignancies. Bottom: Pointwise residual (in sinogram space) for different q, when optimizing originally benign samples to become malignant and vice versa. The values of the residual are clipped at 0.1.

The plots on the top of Figure 6.6, show the difference between the interior and the exterior error $(e_i - e_o)$ as a function of the malignancy we were able to enforce on the reconstruction. As we can see, small values of q allow to not only access the entire range of malignancies without compromising data fidelity, but also do not lead to any recognizable difference between the errors of rays that pass through the nodule and those that do not. As q increases, the differences of errors for *Sample 1* and *Sample 2* increases for malignancies in opposite to their original classification. In contrast, the plot of *Sample 3* shows that there also exist cases, where the difference of errors is not monotone in its classification result.

To go beyond the exemplification on three particular samples, Table 6.1 shows measurements on the growth of the error and the on the malignancy prediction, when optimizing towards the extrema of malignancy $\mathcal{G}_\theta(u_C) = 1$ or $\mathcal{G}_\theta(u_C) = 0$ for the validation dataset of 100 reconstructions. Here we differentiate between two sets of reconstructions: those whose corresponding reconstruction by FBP are classified as benign $S^{\mathcal{B}} = \{u | \mathcal{G}_\theta(u_C^{FBP}) < 0.5\}$ and those whose are classified as malignant $S^{\mathcal{M}} = \{u | \mathcal{G}_\theta(u_C^{FBP}) \geq 0.5\}$. Figure 6.5 shows results for exploring reconstruction in the direction of the classification opposite to the classification obtained on the classical FBP reconstruction. We compare their mean

	set	q	$\frac{1}{qd}\|Ru - f\|^2 \cdot 10^5$	$\mathcal{G}_\theta(u)$	$(e_i - e_o) \cdot 10^5$
optimizing for small $\mathcal{G}_\theta$	$S^{\mathcal{B}}$	50	2.56/1.36	0.008/0.009	−1.86/1.08
		100	3.45/2.01	0.015/0.025	0.23/12.82
		200	6.82/3.93	0.048/0.047	22.01/29.92
		360	9.59/5.73	0.067/0.060	52.47/58.06
	$S^{\mathcal{M}}$	50	2.09/1.24	0.003/0.002	−1.50/1.00
		100	3.36/3.48	0.099/0.296	−1.63/2.08
		200	30.16/38.51	0.423/0.375	309.01/461.44
		360	62.15/66.95	0.526/0.395	658.71/801.45
optimizing for large $\mathcal{G}_\theta$	$S^{\mathcal{B}}$	50	5.58/2.58	0.960/0.040	−3.73/2.12
		100	3.30/1.76	0.957/0.109	−1.26/1.74
		200	5.82/2.77	0.922/0.168	9.40/18.38
		360	13.45/9.68	0.802/0.286	86.85/98.89
	$S^{\mathcal{M}}$	50	4.55/2.65	0.973/0.031	−2.97/2.22
		100	3.35/3.54	0.986/0.017	−1.83/1.64
		200	2.07/1.71	0.989/0.013	−0.04/3.80
		360	6.55/5.71	0.979/0.026	5.88/15.79
	set	q	$\frac{1}{qd}\|Ru - f\|^2$	$\mathcal{G}_\theta(u)$	$(e_i - e_o)$
FBP	$S^{\mathcal{B}} \cup S^{\mathcal{M}}$	50	5.23/9.86	0.54	−1.00/5.43
		100	2.73/8.65	0.55	−1.72/5.34
		200	2.52/8.50	0.55	−1.81/5.32
		360	2.52/8.49	0.55	−1.82/5.31

Table 6.1: Mean/standard deviation of the data consistency loss, network prediction, and distance between the interior and exterior error $(e_i - e_o)$ of the residual r, for originally benign nodules $u \in S^{\mathcal{B}}$ and malignant nodules $u \in S^{\mathcal{M}}$ optimized towards the most extreme malignancies. For comparison, the last rows of this table show the results of the reconstructions obtained by FBP.

data consistency loss, their mean prediction, and their mean distance of the interior and the exterior error $(e_i - e_o)$ for each number of projection angles q. As we can see, the data consistency loss as well as the distance of the interior and the exterior error increase with increasing q. This applies especially when optimizing the originally malignant classified nodules towards a benign classification. Note also, that large values of q make it impossible to reach extreme values of $\mathcal{G}_\theta(u)$ in our setting. Finally, the widely used FBP reconstructions lead to a reconstruction error that is at least *four orders of magnitude higher than all explorable reconstructions*. Thus, trusting a FBP reconstruction would mean that an extreme range of possible alternate solutions would have to be considered as well.

6.4 Conclusion

In this chapter, we have shown the semantic guidance of an iterative CT reconstruction by a data-driven classification network. By conditioning the reconstruction with a pre-trained classification network, we explored the solution space of ambiguous sparse-view CT reconstruction, focusing on the classification of lung nodules. In our experiments,

we have shown to which extent the perceived malignancy of lung nodules can be altered and analyzed the range of alterations for different levels of ambiguity in CT images. We observed that the lower the number of projections, the easier it is to semantically modify reconstructions without having artifacts.

While many methods aim to predict the most realistic reconstruction (typically derived from a large set of training data), we argue that an exploration towards the pathologically most and least concerning reconstruction is significantly more informative to a medical expert interpreting the images: A healthy-looking result is a stronger indication of a healthy patient when obtained by optimizing for the most pathologically concerning image, compared to when optimizing for the most realistic one. This holds particularly in a medical context where great caution needs to be taken of any possible bias arising from the set of training images.

CHAPTER 7

Non-Smooth Energy Dissipating Networks

In the previous chapter, we have shown the guidance of an iterative reconstruction by a data-driven network. Similarly, in this chapter, we will discuss the data-driven guidance of a model-based method, but unlike before, we will not use a neural network as an explicit regularizer. Instead, the neural network will predict the update steps of an energy minimization problem, focusing particularly on non-smooth energies.

After a brief introduction to the problem formulation, we will discuss a previously published method for predicting updated steps, which forms the foundation for our approach, before presenting our own. This chapter is based on a paper published in [89].

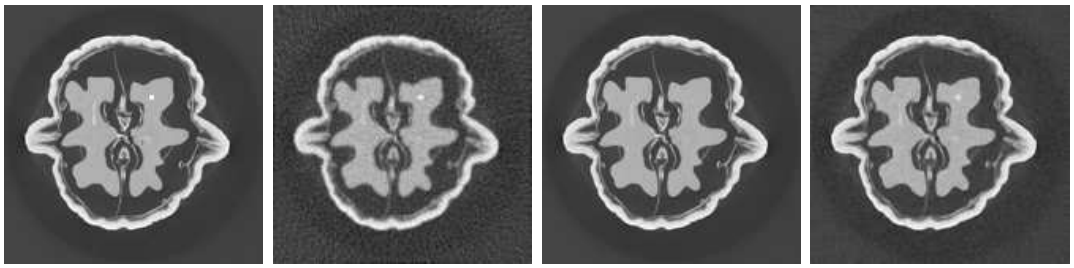
7.1 Introduction

Many image processing problems, e.g. in medical imaging or the reconstruction of impaired corrupted images such as down-sampled, noisy, or blurred images, can be written as linear inverse problems, where a desired quantity $\hat{u}$ ought to be recovered from measured data f that relates to the true solution via (2.1), and the problem can be approached by an energy minimization as shown in (2.2). Given for example a differentiable energy like $E(u) = \|Au - f\|^2$, a simple reconstruction can be approached by using gradient descent, optimizing the update step

$$u^{k+1} = u^k - \tau^k d^k, \text{ where } d^k := \nabla E(u^k). \tag{7.1}$$

Considering the reconstructions from [214] in Figure 7.1, one can see that the reconstruction by gradient descent shows all important features of the reconstructed object, but still is accompanied by noise.

Over the past decade, approaches like (2.2) have largely been outperformed and therefore replaced by deep learning based techniques that directly predict a suitable estimate $\hat{u} = \mathcal{G}_\theta(f)$ for a (deep convolutional) neural network $\mathcal{G}_\theta$ with learnable parameters θ. Despite their performance, it is, however, difficult for such network to *guarantee* a certain *constraint* on its output. For example, the reconstruction in Figure 7.1c, produced by a trained neural network, exhibits missing details in its upper right region, which could potentially be important. This can be a severe limitation particularly for safety-critical



(a) ground truth (b) gradient descent, PSNR 27.5 (c) learned, PSNR 40.2 (d) energy dissipating, PSNR 33.0

Figure 7.1: Illustration from Möller *et al.* [214], showing reconstructions of a walnut using gradient descent (b), using a pre-trained neural network (c), and by using their proposed energy dissipating network (d). While the neural network provides a clean reconstruction with a high peak signal-to-noise ratio (PSNR), it does not guarantee to capture all details of the walnut, as the highlight in the walnuts over right part. The energy dissipating network combines the advantages of the model-based approach and the data-driven neural network.

applications, where one at least needs to ensure that – if the distribution of the noise can be characterized as $p(u|f) \propto \exp(-\text{dist}(Au, f))$ – the prediction $\hat{u}$ respects the data up to the expected noise level $\delta = \text{dist}(A\hat{u}, f)$, i.e.

$$\text{dist}(A\hat{u}, f) \leq \delta. \tag{7.2}$$

Previous works [214, 280] have addressed similar problems by training networks safeguarded with a suitable cost function, ensuring bounds like those in (7.2). In the following section, let us discuss the work of Möller *et al.* [214] in more detail.

7.2 Energy Dissipating Networks

Möller *et al.* [214] introduced a method where a neural network iteratively predicts the descent direction for a given cost function E. This approach integrates successful deep learning techniques into model-based minimization, while also guaranteeing convergence to a minimizer of the continuously differentiable cost function and maintaining data consistency. They predict the descent steps d^k for minimizing a cost function as shown in (7.1) using a neural network $\mathcal{G}_\theta$:

$$d^k = \mathcal{G}_\theta(u^k, \nabla E(u^k), f) \tag{7.3}$$

The network prediction depends on the current iterate u^k, the gradient of the cost function $\nabla E(u^k)$, and the input data f.

The authors propose training a network $\mathcal{G}_\theta$, to iteratively predict update directions that lie in a suitable convex set $C(\zeta_1, \zeta_2, \nabla E(u^k))$ of descent directions of a continuously differentiable cost function E at the current estimate u^k to ensure convergence to a minimizer of the cost function E. This set is defined for the fixed parameters ζ_1 and ζ_2 as:

$$C(\zeta_1, \zeta_2, g) = \{d \,|\, \langle d, g\rangle \geq \zeta_1 \|g\|^2, \|d\| \leq \zeta_2 \|g\|\} \tag{7.4}$$

The intuition behind the projection onto the above set is to control the angular deviation between the network's predicted direction and the gradient direction g of the energy E by

changing ζ_1 and ζ_2. The projection onto the defined set is performed in the last layer of the neural network as

$$z \mapsto \hat{\eta} g + \Pi_B(z - \eta g), \tag{7.5}$$

where $\eta = \frac{\langle z,g \rangle}{\|g\|^2}$ is the magnitude of z in the direction of the gradient g and $\hat{\eta}$ is the clamped version of η that lies between ζ_1 and ζ_2. Moreover, Π_B represents the projection onto

$$B = \left\{ d \mid \|d\| \leq \sqrt{\zeta_2^2 - \hat{\eta}^2 \|g\|^2} \right\}. \tag{7.6}$$

To confirm that the projection meets the first condition specified in (7.4), we examine whether it fulfills the following condition:

$$\langle (\hat{\eta} g + \Pi_B(z - \eta g)), g \rangle \geq \zeta_1 \|g\|^2 \tag{7.7}$$

For this purpose, we will split the dot product $\langle \hat{\eta} g + \Pi_B(z - \eta g), g \rangle$ into two separate components. The first inequality,

$$\langle \hat{\eta} g, g \rangle = \hat{\eta} \|g\|^2 \geq \zeta_1 \|g\|^2 \tag{7.8}$$

holds, as $\hat{\eta}$ is clamped to be larger than or equal to ζ_1. Additionally,

$$\langle \Pi_B(z - \eta g), g \rangle = 0, \tag{7.9}$$

as $z - \eta g$ is the projection of z orthogonal to g. The second condition in (7.4),

$$\|\hat{\eta} g + \Pi_B(z - \eta g)\| \leq \zeta_2 \|g\| \tag{7.10}$$

follows from

$$\begin{aligned} \|\hat{\eta} g + \Pi_B(z - \eta g)\|^2 &= \hat{\eta}^2 \|g\|^2 + \|\Pi_B(z - \eta g)\|^2 + 2\langle \hat{\eta} g, \Pi_B(z - \eta g) \rangle \\ &\leq \zeta_2^2 \|g\|^2, \end{aligned} \tag{7.11}$$

where $\langle \hat{\eta} g, \Pi_B(z - \eta g) \rangle = 0$, as $\Pi_B(z - \eta g)$ is orthogonal to g. Because $\hat{\eta}$ is clamped to be less than or equal to ζ_2, and the norm of the projection $\Pi_B(z - \eta g)$ is bounded, the inequality holds, and finally shows $\hat{\eta} g + \Pi_B(z - \eta g) \in C(\zeta_1, \zeta_2, g)$. In [214], the authors show linear convergence under assumption that the given energy E is L-Lipschitz differentiable, and further mild assumptions (see *Assumption 1* in [214]), for the predicted update step $d^k \in C(\zeta_1, \zeta_2, g)$ (see *Proposition 2* in [214]).

In order to train the network $\mathcal{G}_\theta$, training data u^k is sampled from the potential inputs that $\mathcal{G}_\theta$ might face during inference, while predicting the descent steps. Initially the training data is sampled by calculating a random, variable number of descent steps using gradient descent with line search. As the network training progresses, repetitively after a certain number of epochs, the prediction of the descent steps used for sampling the training data is replaced by the network trained so far. For training, the loss function minimizes the error between the networks predicted direction and the direction pointing from u^k to the ground truth $\hat{u}$:

$$\min_\theta \mathcal{L}(\mathcal{G}_\theta(u^k, \nabla E(u^k), f), \hat{u} - u^k) \tag{7.12}$$

During inference Möller *et al.* conduct a line-search algorithm to find τ^k for an update of the form

$$u^{k+1} = u^k - \tau^k \mathcal{G}_\theta(u^k, \nabla E(u^k), f) \tag{7.13}$$

for input data f and network parameters θ, to ensure a monotonic decrease of energy and a convergence to minimizers of the energy E.

Unfortunately, the approach of Möller *et al.* [214] is limited to the case where $E(u)$ is *continuously differentiable* in u, e.g. the investigated classical case of Gaussian noise where $E(u) = \|Au - f\|^2$. Yet, some distributions of high practical relevance such as the Laplace (or even more heavy-tailed) distributions cannot be tackled with their approach. Moreover, for cost functions that do not possess a Lipschitz-continuous gradient with reasonably small Lipschitz constant, the stated convergence can become very slow.

In the following, we extend the method from [214] to apply to α-semi-convex and non-differentiable costs by descending on their *Moreau-Yosida regularization*, a smooth approximation with identical minimizers. We discuss appropriate step size rules of the resulting descent scheme to ensure convergence, and showcase the importance of using non-smooth loss functions in two exemplary applications.

7.3 Non-Smooth Energy Dissipating Networks

For the remainder of this chapter, let $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be a proper, lower semi-continuous cost function that has a minimizer (e.g. by being coercive). The minimization of such costs is very well studied in the literature, particularly in the case where E is convex, with customized versions of the proximal point method [249, 234]

$$u^{k+1} = \text{prox}_{\tau E}(u^k), \tag{7.14}$$

with

$$\text{prox}_E(u) = \arg\min_v E(v) + \frac{1}{2}\|u - v\|^2. \tag{7.15}$$

As in the convex setting the proximal point algorithm (7.14) can be interpreted as a conventional gradient descent method on a smoothed version of the original costs (see Section 2.4.4), the explicit gradient descent on the Moreau-Yosida regularization of a convex non-smooth function E is interesting for the framework of energy dissipating networks. To go beyond the fully convex case, let us assume that E is α-semi-convex, i.e., that there exists a constant α such that $E(u) + \frac{\alpha}{2}\|u\|^2$ is convex.

Proposition 1. *Let E be proper, lower semi-continuous, and α-semi-convex. For $\frac{1}{\mu} > \alpha$ the gradient of the Moreau-Yosida regularization E_μ,*

$$\nabla E_\mu(u) = \frac{1}{\mu}(u - prox_{\mu E}(u)). \tag{7.16}$$

is L-Lipschitz continuous with a constant of at most $\frac{1}{\mu}(1 + \frac{1}{(1+\mu\alpha)^2})$.

Proof. Let us denote $\tilde{v} = \text{prox}_{\mu E}(v)$, $\tilde{z} = \text{prox}_{\mu E}(z)$, and note that $\tilde{E}(u) = E(u) + \frac{\alpha}{2}\|u\|^2$ is convex. Given the definition of the proximal operator in (2.17), the optimality condition yields:

$$0 = p_v - \alpha\tilde{v} + \frac{1}{\mu}(\tilde{v} - v), \text{ for } p_v \in \partial\tilde{E}(\tilde{v}) \tag{7.17}$$

Subtracting two equations derived from the optimality conditions, we get

$$0 = p_v - \alpha\tilde{v} + \frac{1}{\mu}(\tilde{v} - v) - p_z + \alpha\tilde{z} - \frac{1}{\mu}(\tilde{z} - z) \tag{7.18}$$

$$= (p_v - p_z) - \alpha(\tilde{v} - \tilde{z}) + \frac{1}{\mu}(\tilde{v} - \tilde{z}) - \frac{1}{\mu}(v - z) \tag{7.19}$$

$$\Rightarrow (\frac{1}{\mu} - \alpha)(\tilde{v} - \tilde{z}) = \frac{1}{\mu}(v - z) - (p_v - p_z), \tag{7.20}$$

$$p_v \in \partial\tilde{E}(\tilde{v}),\ p_z \in \partial\tilde{E}(\tilde{z}). \tag{7.21}$$

Multiplying by μ, taking the inner product with $\tilde{v} - \tilde{z}$, and using that $\langle p_v - p_z, \tilde{v} - \tilde{z}\rangle \geq 0$ yields

$$(1 - \mu\alpha)\|\tilde{v} - \tilde{z}\|^2 \leq \langle v - z, \tilde{v} - \tilde{z}\rangle \tag{7.22}$$

and taking young's inequality for inner product of $|\langle u, v\rangle| \leq \frac{\lambda^2}{2}\|u\|^2 + \frac{1}{2\lambda^2}\|v\|^2$, we get

$$\langle v - z, \tilde{v} - \tilde{z}\rangle \leq \frac{1}{2}(1 - \mu\alpha)\|\tilde{v} - \tilde{z}\|^2 + \frac{1}{2(1 - \mu\alpha)}\|v - z\|^2 \tag{7.23}$$

such that

$$\|\tilde{v} - \tilde{z}\|^2 \leq \frac{1}{(1 - \mu\alpha)^2}\|v - z\|^2. \tag{7.24}$$

As this shows that $\mathrm{prox}_{\mu E}$ is $\frac{1}{(1-\mu\alpha)^2}$-Lipschitz continuous, the assertion follows by simple addition of Lipschitz constants. □

Therefore, we propose the following approach: Let E be a given semi-convex but possibly non-smooth cost function with which we'd like to control the behavior of a data driven (deep learning) approach.

We design an arbitrary (e.g. deep convolutional) neural network $\mathcal{G}_\theta$ of our choice, that gets as an input the current estimate u^k, the input data f and the gradient of the Moreau-Yosida regularization at the current estimate $\nabla E_\mu(u^k)$ and predicts a descent direction $\mathcal{G}_\theta(u^k, \nabla E_\mu(u^k), f)$, s.t.

$$u^{k+1} = u^k - \tau\mathcal{G}_\theta(u^k, \nabla E_\mu(u^k), f) \tag{7.25}$$

converge to a minimizer of a non-smooth energy E.

To satisfy the descent constraints for a non-smooth energy E we use a surjective mapping onto $C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$ (see (7.4)) as the last layer of $\mathcal{G}_\theta$, which is given by

$$z \mapsto \Pi_{[\zeta_1,\zeta_2]}(\eta)g + \Pi_B(z - \eta g), \tag{7.26}$$

with Π_B being a projection onto $B = \{d \,|\, \|d\| \leq \sqrt{\zeta_2^2 - \eta^2}\|g\|\}$, with $\eta = \langle z, g\rangle / \|g\|^2$ and in this setting $g = \nabla E_\mu$, in comparison the mapping $g = \nabla E_\mu$ proposed in [214]. This mapping satisfies the network prediction to lie in $C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$, similary as detailed in Section 7.2.

In order to train the network on data that it could face during descent, prior to each training step, the data is transformed into a potential sample generated from the space of possible inputs as shown in Algorithm 1. For this purpose, starting from the input data u^0 (e.g. $u_i^0 = \frac{1}{n}\sum_j^n f_j$ in Section 7.4.1 and $u^0 = f$ in Section 7.4.2), an arbitrary number of

Algorithm 1: Learned descent steps by an energy dissipating network satisfying the descent constraints for an energy E_μ.

Data: starting point u^0, network $\mathcal{G}_\theta$, gradient of the moreau envelope ∇E_μ, input data f, stepsize τ, maximal number of iterations N
Result: $u^{\tilde{k}}$
$\tilde{k} \in \{0, \dots, N\}$
for $k \in \{0, \dots, \tilde{k}\}$ **do**
  $u^{k+1} \leftarrow u^k - \tau \mathcal{G}_\theta(u^k, \nabla E_\mu(u^k), f)$

descent steps (7.25) are performed to generate a sample that is potentially visited during descent with the current model. This potential sample $u^{\tilde{k}}$ for $\tilde{k} \in \{0, \dots, N\}$ is used as input for training the neural network. The network is trained by minimizing the sum of losses

$$\|\mathcal{G}_\theta(u^{\tilde{k}}, \nabla E_\mu(u^{\tilde{k}}), f; \theta) - (\hat{u} - u^{\tilde{k}})\|_2^2 \tag{7.27}$$

over all training examples for θ, where $\hat{u}$ represent the desired (ground truth) predictions. Please note the increased computational cost by computing new training samples, in comparison to other training based networks.

Proposition 2. *For a Moreau-Yosida regularization with L-Lipschitz continuous gradient, the descent steps in (7.25) converge with constant step size $\tau^k < \frac{\zeta_1}{\zeta_2 L}$ for a model $\mathcal{G}_\theta(u^k, \nabla E_\mu(u^k), f)$ that satisfies*

$$\mathcal{G}_\theta(u^k, \nabla E_\mu(u^k), f) \in C(\zeta_1, \zeta_2, \nabla E_\mu(u^k)) \tag{7.28}$$

Proof. According to Taylor's theorem it holds that

$$\begin{aligned} E_\mu(u^{k+1}) =& E_\mu(u^k) + \langle \nabla E_\mu(u^k), u^{k+1} - u^k \rangle \\ &+ \langle \nabla E_\mu(\xi) - \nabla E_\mu(u^k), u^{k+1} - u^k \rangle, \end{aligned} \tag{7.29}$$

for some ξ on the line segment between u^k and u^{k+1}. Using that $u^{k+1} = u^k - \tau^k d^k$, we get

$$E_\mu(u^{k+1}) - E_\mu(u^k) = -\tau^k \langle \nabla E_\mu(u^k), d^k \rangle + \langle \nabla E_\mu(\xi) - \nabla E_\mu(u^k), u^{k+1} - u^k \rangle. \tag{7.30}$$

Using the Cauchy-Schwarz inequality, we establish the subsequent inequality:

$$\langle \nabla E_\mu(\xi) - \nabla E_\mu(u^k), u^{k+1} - u^k \rangle \leq \|\nabla E_\mu(\xi) - \nabla E_\mu(u^k)\| \|u^{k+1} - u^k\| \tag{7.31}$$

Furthermore, it is given that $d^k \in C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$, which guarantees $\langle \nabla E_\mu(u^k), d^k \rangle \geq \zeta_1 \|\nabla E_\mu(u^k)\|$. Considering the L-smoothness of E_μ, it holds that $\|\nabla E_\mu(\xi) - \nabla E_\mu(u^k)\| \leq L\|\xi - u^k\|$, leading us to the following inequality:

$$\begin{aligned} &- \tau^k \langle \nabla E_\mu(u^k), d^k \rangle + \langle \nabla E_\mu(\xi) - \nabla E_\mu(u^k), u^{k+1} - u^k \rangle \\ &\leq -\tau^k \langle \nabla E_\mu(u^k), d^k \rangle + \|\nabla E_\mu(\xi) - \nabla E_\mu(u^k)\| \|u^{k+1} - u^k\| \\ &\leq -\tau^k \zeta_1 \|\nabla E_\mu(u^k)\|^2 + \tau^k L \|\xi - u^k\| \|d^k\| \end{aligned} \tag{7.32}$$

As ξ lies on the line segment between u^k and u^{u+1}, it holds that $\|\xi - u^k\| \leq \|\tau^k d^k\|$. In the following steps we also use that $d^k \in C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$ and such it holds that

$\|d\| \leq \zeta_2 \|\nabla E_\mu(u^k)\|$:

$$\begin{aligned} &\leq -\tau^k \zeta_1 \|\nabla E_\mu(u^k)\|^2 + \tau^k L \|\xi - u^k\| \|d^k\| \\ &\leq -\tau^k \zeta_1 \|\nabla E_\mu(u^k)\|^2 + (\tau^k)^2 L \|d^k\|^2 \\ &\leq -\tau^k \zeta_1 \|\nabla E_\mu(u^k)\|^2 + (\tau^k)^2 L \zeta_2 \|\nabla E_\mu(u^k)\|^2, \\ &= \tau^k \|\nabla E_\mu(u^k)\|^2 \cdot (-\zeta_1 + \tau^k L \zeta_2) \end{aligned} \tag{7.33}$$

From this, we derive that $E_\mu(u^{k+1}) - E_\mu(u^k) \leq \tau^k \|\nabla E_\mu(u^k)\|^2 \cdot (-\zeta_1 + \tau^k L \zeta_2)$. Given that $\|\nabla E_\mu(u^k)\|^2 > 0$ and $\tau^k > 0$, we are descending in the energy for $\tau^k < \frac{\zeta_1}{L\zeta_2}$. Second, coercivity of the energy ensures the existence of a convergent subsequence. Third, since $d^k \in C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$, it holds that $\|\nabla E_\mu(u^k)\| \leq \frac{1}{\zeta_1}\|d^k\| = \frac{1}{\zeta_1 \tau^k}\|u^{k+1} - u^k\|$. The convergence then follows from standard results in descent-based methods such as *Theorem 2.9 (Convergence to a critical point)* in [12]. □

7.4 Numerical Evaluation

For proof of concept we implemented salt and pepper denoising to demonstrate energy dissipating networks on non-smooth energies and binary deblurring to show energy dissipating networks on non-smooth and semi-convex energies.

7.4.1 Salt and Pepper Denoising

For denoising images with salt and pepper noise using dissipation neural networks, an appropriate convex data fidelity term is the ℓ_1 norm of the distance to the noisy image f:

$$E(u) = \|u - f\|_1. \tag{7.34}$$

We construct the Moreau-Yosida regularization of (7.34) as

$$E_\mu(u) = \sum_i e_\mu(u_i) \tag{7.35}$$

with

$$e_\mu(u_i) = \begin{cases} |u_i - f_i| - \frac{\mu}{2} & \text{if } |u_i - f_i| > \mu \\ \frac{1}{2\mu}(u_i - f_i)^2, & \text{otherwise,} \end{cases} \tag{7.36}$$

and train an energy dissipating network on noisy data with a surjective mapping to the gradient ∇E_μ in its last layer. Based on Proposition 2, the dissipating network minimizes the data fidelity term (7.34), but takes a path that tries to get as close as possible to the noise-free image in a data-driven way. Thus, there is a point during the minimization where the image is denoised best, and afterwards, due to convergence to the minimizer of (7.34), approaches the noisy image v.

To stop minimization when denoised best, a popular a posteriori stopping rule is the discrepancy principle [216]: Similar to (7.2) we stop the algorithm at the minimum distance of the expected noise level $\delta = \|\hat{u} - f\|_p$ for $p = 1$ to the distance of the calculated image to the noisy image

$$\arg\min_k |\|u^k - f\|_p - \delta|. \tag{7.37}$$

noise	$\|u-v\|_1$	$\|u-v\|_2^2$	median filter	TV
1%	**39.87/0.97**	39.07/0.97	30.25/0.87	34.24/0.97
5%	**34.99/0.95**	32.17/0.89	29.34/0.85	30.00/0.92
10%	**28.62/0.82**	16.93/0.45	26.61/0.80	27.83/0.86
25%	15.13/0.39	15.13/0.39	14.87/0.24	**24.75/0.72**

Table 7.1: Measured mean PSNR and structural similarity index measure (SSIM) value on the validation dataset of BSDS500 for energy dissipation network algorithm with an ℓ_1 and an ℓ_2 data fidelity term for $\zeta_1 = 0.05$ and $\zeta_2 = 30$, for running a median filter with kernel size 3, and for running TV denoising.

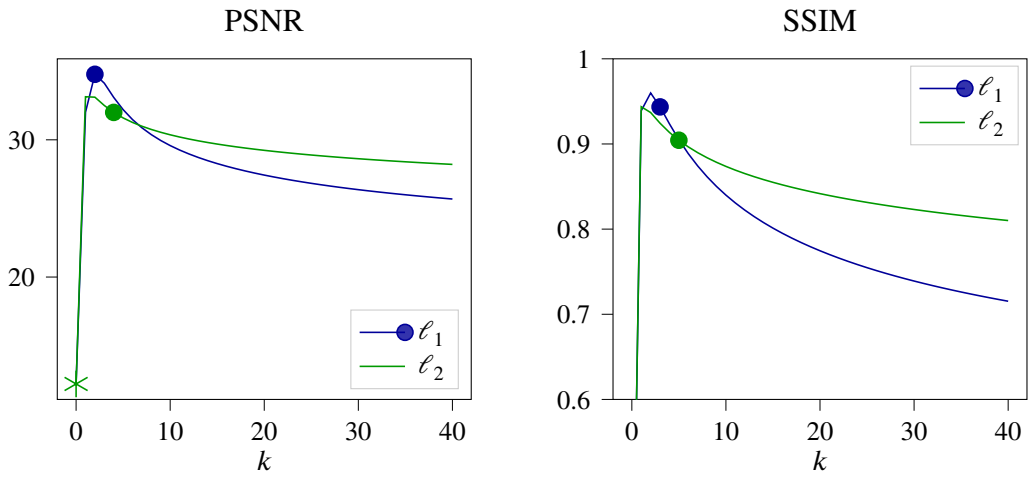
(a) Noisy image (b) Median filter (c) TV (d) ℓ_1 fid. term (e) ℓ_2 fid. term

Figure 7.2: Exemplary denoising results of noisy images (a) by running median filter with kernel size 3 (b), by running TV denoising (c), by running descent on an energy dissipating network satisfying descent constraints for an ℓ_1 fidelity term (d) and an ℓ_2 fidelity term (e).

In the following experiments, we train an energy dissipating network on salt and pepper denoising with mapping on the gradient of the Moreau-Yosida regularization (7.35) and compare the approach with using dissipating networks on the ℓ_2 norm $E_{\ell_2}(u) = \|u-f\|^2$. For the latter, the predicted descent direction is mapped to $\nabla E_{\ell_2}(u) = 2(u^k - f)$, s.t. the update step becomes $d^k = \mathcal{G}_\theta(u^k, \nabla E_{\ell_2}(u^k), f)$. We also compare our results to denoising using a median filter and TV denoising by $\min_u \|Du\|_1$ s.t. $\|u-f\|_1 \leq \delta$, with D being a finite difference matrix.

As train and validation data, we use the given images from BSDS500 [8] and apply salt and pepper noise with a probability of 5% each that a pixel takes the value 0 or 1. For training, we extract patches of dimension 52×52 and use them to train the network with the architecture of [331] for 30000 iterations on the loss function in (7.27) using Adam optimizer.

Since the step size given in Proposition 2 turns out to be too small at the beginning to efficiently minimize the energy, we choose our step size as $\tau_i = \max\left(\frac{1}{i+1}, \tau_{\min}\right)$. In validation, we compare PSNR and SSIM averaged over 100 validation images for different fractions of noise, i.e., images that originated outside the potential sampling space of the training. A quantitative evaluation is given in Table 7.1, where the values are measured at the stopping point triggered by the criterion in (7.37). It shows good performance for the ℓ_1 norm as surrogate energy, better than for the ℓ_2 norm (up to 8.8% in PSNR for 5% noise), the median filter (with kernel size 3), or TV denoising, except for the highly degraded data with 25% noise, which appears to be too far outside of the range of our



(a) PSNR over the iterations

(b) SSIM over the iterations

Figure 7.3: Exemplary comparison of the PSNR and SSIM for running descent on an energy dissipating network satisfying descent constraints for an ℓ_1 fidelity term and an ℓ_2 fidelity term with circles shaped markings of the iteration where the stopping criterion is triggered for $p = 1$ and a star shaped marking where the stopping criterion is triggered for $p = 2$.

training examples, as in our method the network was trained on data with 5% probability of noise. Exemplary denoising results for 5% noisy data are shown in Figure 7.2, with the corresponding PSNR and SSIM curve (for the upper images) over the iterations in Figure 7.3, showing the peak of PSNR and SSIM at a certain point during minimization.

7.4.2 Binary Deblurring

To demonstrate the concept of non-smooth and semi-convex energy dissipating networks, we consider the deblurring of binary images u with pixels that are supposed to be either zero or one $u_i \in \{0, 1\}$, e.g. having the reconstruction of bar-codes or QR-codes in mind. To ensure binary outputs, we consider the function

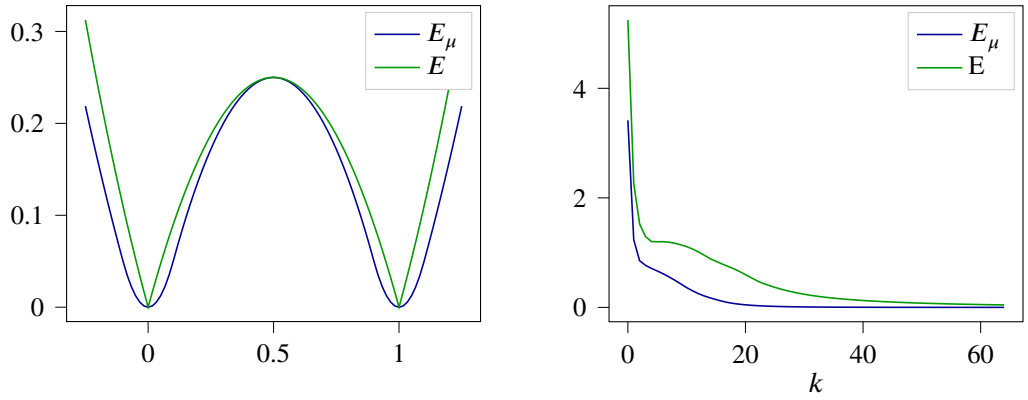
$$E(u) = \|(u - 0.5)^2 - 0.25\|_1, \tag{7.38}$$

and its Moreau-Yosida regularization, which is illustrated in Figure 7.4 (a) and train a dissipating network that satisfies the descent constraints for (7.38).

As dataset we use generated bar-codes $b_i \in \{0, 1\}^{180}$ of type *Code 128*, by encoding randomly chosen sequences of 5 numbers and letters, and blur them using a Gaussian filter with radius 1.5. Our training set consists of 40960 arrays and our test set of 1024 arrays. In our experiments, we use the network architecture of [331] and decrease the network depth to 12 convolutional layers and the width to 32. We train the network using losses of the form (7.27) for 30000 iterations.

As shown in Figure 7.4 (b), the network-based updates lead to a monotonically decreasing cost function, converging to zero, i.e., to binary predictions in about 60 iterations.

Figure 7.5 shows an input bar code with a blur radius of 1.0 and the result of a dissipating network in comparison to an unconstrained network, with the same network architecture and trained on the same blurred data with a radius of 1.5, just without energy dissipation. Here, for 2D visualization, the arrays were repeated along the height and



(a) Illustration of E and E_μ (b) Energy over the iterations

Figure 7.4: Illustration of Energy E (7.38) and its Moreau-Yosida regularization E_μ for $\mu = 0.1$ (a) and their energy curve over the iterations when running descent on an energy dissipating network (b).

(a) Blurred bar-code (b) Dissipating network (c) Neural network $\mathcal{G}_\theta$

Figure 7.5: Deblurring results for the blurred input bar-code using a Gaussian filter with radius 1.0 (a), by running descent on an energy dissipating network (b), satisfying (7.38), and by applying a trained neural network (c).

cropped in width. As it turns out, the unconstrained network, which has no guarantee of predicting a deblurred binary image on unknown data, fails to predict a binary image, unlike the constrained energy dissipating network.

7.5 Conclusion

This chapter dealt with the inclusion of data-driven networks to model-based energy minimization approaches, in the form of the prediction of descent steps by (non-smooth) energy dissipating neural networks.

We first discussed the mapping of a neural network's prediction onto a suitable convex set, as proposed by Möller *et al.* [214], which ensures that iterative updates converge to a minimizer of the corresponding energy function. Subsequently, we demonstrated how to extend their framework of energy dissipating networks to non-smooth energies by exploiting the equivalence of the proximal point algorithm and gradient descent on a cost functions' Moreau-Yosida regularization. In numerical experiments, we applied this approach to the ℓ_1-norm for salt and pepper denoising, and to a non-smooth (semi-convex) function for binary deblurring, demonstrating improved performance compared to an unconstrained network.

CHAPTER 8

Efficient Low-Rank Permutation Representation

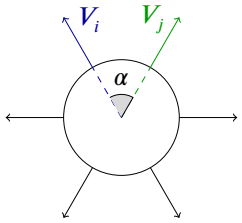
While the previous chapters concern the combination and comparison of model- and learning-based methods, we now deal with problem formulations, that can appear in model- and learning-based approaches likewise. This chapter is based on [87], where we study the memory-intensive permutation matrix for assignment problems in data-driven and classical applications and discuss another representation of the permutation matrix by two matrices and a nonlinearity, with their dimensions motivated by the so-called Kissing number.

8.1 Introduction

Permutation matrices, which encode the reordering of elements, arise naturally in any problem that can be phrased as a bijection between two equally sized sets. As such, they are fundamental to many important computer vision applications, including matching semantically identical key points in images [329, 308, 309, 328], matching 3D shapes or point clouds [148, 303, 204], estimating scene flow on point clouds [240] and solving jigsaw puzzles [208], as well as to various sorting tasks [23, 116]. As briefly discussed in the introduction to assignment problems in Section 4.3, permutations can be alternatively representated by an enumeration of the permuted elements. However, it is quickly concluded that this representation is unsuitable for optimization tasks due to its discrete nature. Therefore, most methods for predicting permutations, especially learning-based approaches, prefer a *permutation matrix* representation as given in (4.15).

Yet, the advantages of the matrix form representation (4.15) come at the cost of a prohibitive increase in memory, as it requires storing n^2 binary numbers $P_{ij} \in \{0, 1\}$, or – after commonly used relaxations – even n^2-many floating point numbers instead of the n integers in (4.14). This renders matching problems with $n > \sim 10^4$ largely infeasible as their corresponding permutation matrix P constitutes over one hundred million entries. To handle large matrices whose size prohibits explicit processing and storage, existing approaches typically either turn to *sparse representations*, i.e., storing only a small portion of matrix values in the form of $(i, j, P_{i,j})$ triplets, where $P_{i,j} \neq 0$, or employ *low rank representations*, i.e., forming a large matrix P as a product of matrices

$$P = VW^T, \tag{8.1}$$



$$2\max\left(\begin{pmatrix} 1.0 & 0.0 \\ \mathbf{0.5} & \mathbf{0.87} \\ \mathbf{-0.5} & \mathbf{0.87} \\ -1.0 & -0.0 \\ -0.5 & -0.87 \\ 0.5 & -0.87 \end{pmatrix} \cdot \begin{pmatrix} -1.0 & -0.0 \\ 1.0 & 0.0 \\ \mathbf{0.5} & \mathbf{0.87} \\ -0.5 & -0.87 \\ 0.5 & -0.87 \\ \mathbf{-0.5} & \mathbf{0.87} \end{pmatrix}^T - \tfrac{1}{2}, 0\right) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{0} \\ 0 & 0 & \mathbf{0} & 0 & 0 & \mathbf{1} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 8.1: Geometric intuition behind our approach on a 2D unit sphere. For well-distributed vectors $V \in \mathbb{R}^{\text{Kiss}(2)\times 2}$, where the number of vectors is determined by the Kissing number ($\text{Kiss}(2) = 6$),the cosine angle between different vectors $V_{i,:}$ and $V_{j,:}$, $i \neq j$, is $\langle V_{i,:}, V_{j,:}\rangle = \cos(\alpha) \leq 0.5$, while $\langle V_{i,:}, V_{i,:}\rangle = 1$ for the same vector. Thus, for any permutation P, the matrix-matrix product of V and $(PV)^T$ merely has to be thresholded suitably to represent the permutation P, i.e. $P = 2\max(V(PV)^T - 0.5, 0)$.

with $V, W \in \mathbb{R}^{n\times m}$ and $m << n$.

Unfortunately, neither of these approaches is applicable to permutation matrices: sparse representation cannot be used as the sparsity pattern is not only unknown a-priori but actually the sought-after solution to the problem. On top of that, since permutation matrices are by definition full rank, a low-rank representation (8.1) can yield only a crude approximation at best.

In this chapter, we alleviate the limitation on problem size by harnessing the well-studied problem of (bounds for) the so-called *Kissing number*, which, in practice, translates to introducing a simple adaptation to the matrix factorization approach (8.1). In particular, we exploit the fact that for row-normalized matrices V and W, the entries of VW^T correspond to the cosines of the angles between the matrix rows. We then apply a pointwise non-linearity on the product of the matrices in (8.1), which allows representing any permutation while using $m << n$. We use the Kissing number theory to provide an estimate for how small an m we can use. While previous work on the approximation of sparse and non-negative matrices by nonlinear matrix decomposition [264], and further (accelerated) methods on this subject [269] have been proposed in the last few years, including an analysis of the geometric relationship between the sparse and low-rank matrices [263], we exploit this problem of low-rank approximation for permutation matrices and show its relationship to the Kissing number. We elaborate on our theoretical considerations in Section 8.3 and provide an illustration of the geometric intuition for our approach in Figure 8.1. We then demonstrate the applicability of the proposed approach through several numerical experiments tackling various problems that involve estimating permutations, including a study on point alignment, LAPs, QAPs, and a real-world shape matching application. We find that the proposed approach trades off only little accuracy to offer significant memory saving, thus enabling handling bijection mapping problems that are larger than was previously possible, when full permutation matrices had to be stored.

8.2 Kissing Number Theory

The origin of the Kissing number is said to have arisen in the late 1600s from a dispute between Isaac Newton and David Gregory, who were discussing how many billiard balls could touch a given other billiard ball at once, whereby the balls are called to "kiss" if they touch. Newton said 12, Gregory claimed 13, whereas in 1953 Schütte and Waerden [267]

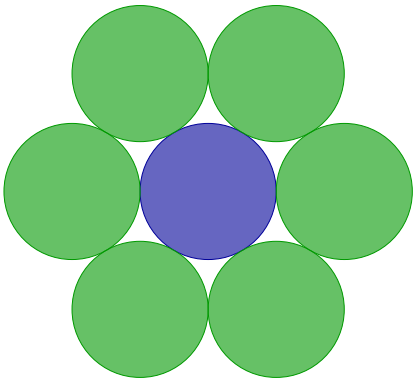


Figure 8.2: Geometrical interpretation of the Kissing number in 2D.

proved Newton to be correct. [237]

The Kissing number Kiss(m) can be interpreted as the maximal number of non-overlapping m-dimensional spheres $\in \mathbb{R}^m$ that can touch another same sized m-dimensional sphere. The geometrical interpretation in two dimensions is given in Figure 8.2, demonstrating that the Kissing number Kiss(2) = 6. Further the Kissing numer in one dimension would be Kiss(1) = 2. Many research has been done to solve this problem for higher dimensions. Beside the Kissing number of dimensions 1, 2 and 3, the exact Kissing number has been proven only for $m = 4, 8, 24$ [219, 228, 165]. For some further dimensions there have been calculated upper and lower bounds [176, 219, 44].

Formally the Kissing number can be defined as the maximum value for which points can distributed on a unit sphere, s.t. the angle between each pair of points is at least arccos(0.5):

Definition 11. *For a given $m \in \mathbb{N}$, we define the Kissing number Kiss(m) as*

$$Kiss(m) := \max_n \{n \in \mathbb{N} \mid \exists A \in \mathbb{R}^{n \times m},\ \|A_{i,:}\|_2 = 1,\ 2\langle A_{i:}, A_{j,:}\rangle \leq 1,\ i \neq j\}. \tag{8.2}$$

We take advantage of the Kissing number properties in our work, namely using the fact that there exists a set of vectors of unit length for which the cosine angle α between two vectors is bounded, and benefit from the additional literature that derives theoretical bounds on the number and dimension of those vectors.

8.3 Low-Rank Permutation Matrix Representation

A common approach to solve optimization problems with costs E over the set of permutation matrices $\mathcal{P}_n$ (including those arising from training neural networks for predicting assignments) is to relax the problem by replacing $\mathcal{P}_n$ by its convex hull conv($\mathcal{P}_n$), i.e., the set of doubly-stochastic matrices:

$$\min_{P \in \text{conv}(\mathcal{P}_n)} E(P). \tag{8.3}$$

Since P grows quadratically in n, has an unknown sparsity pattern, and the true solution is always full rank, such problems pose significant challenges for large n. In this work, we make the interesting observation that a non-linearity as simple as a rectified linear unit (ReLU, denoted by σ) is sufficient not only to restore a full rank, but to represent any permutation matrix exactly. More precisely, we propose to replace the set conv($\mathcal{P}_n$) in (8.3)

with the set $\mathcal{K}_m(\mathcal{P}_n) = \{\sigma(2VW^T - 1) \mid V, W \in \mathbb{R}^{n\times m}\}$ and use the *Kissing number* [31, 219, 344] (see (8.2)) to show that $\mathcal{P}_n \subset \mathcal{K}_m(\mathcal{P}_n)$ for a surprisingly small m.

Remind that the Kissing number can be interpreted geometrically as the maximum number of points that can be distributed on an m-dimensional unit sphere such that the angle formed between each pair of different points is at least $\arccos(0.5)$. This property quickly establishes $\mathcal{P}_n \subset \mathcal{K}_m(\mathcal{P}_n)$:

Proposition 1. *Let $P \in \mathcal{P}_n$ be an arbitrary permutation matrix, and let σ, $\sigma(x) = \max(x, 0)$ denote a rectified linear unit (ReLU). Then for every m such that $n \leq Kiss(m)$ there exist $V, W \in \mathbb{R}^{n\times m}$ such that*

$$P = \sigma(2VW^T - 1). \tag{8.4}$$

Proof. Let $V \in \mathbb{R}^{n\times m}$ be a matrix that satisfies the equalities and inequalities of (8.2), and let $W = PV$. Then it holds that

$$2\langle V_{i,:}, W_{j,:}\rangle \begin{cases} \leq 1 & \text{if } P_{i,j} \neq 1 \\ = 2 & \text{otherwise} \end{cases}. \tag{8.5}$$

Consequently

$$\sigma(2\langle V_{i,:}, W_{j,:}\rangle - 1) = \begin{cases} 0 & \text{if } P_{i,j} \neq 1 \\ 1 & \text{otherwise} \end{cases}, \tag{8.6}$$

which proves the assertion. □

To determine the minimal rank m that is required for representing a permutation of n elements, we rely on extensive studies in the past few decades which computed either exact values or lower and upper bounds for different values of m [44].

Using $P = \sigma(2VW^T - 1)$ for relaxing (8.3) yields a relaxation that requires only $2mn$ instead of n^2 parameters, with $m << n = \text{Kiss}(m)$. For instance, $\text{Kiss}(24) = 196560$ implies that matrices of rank $m = 24$ are sufficient for representing any arbitrary permutation matrix of up to $n = 196560$ elements, thus requiring ~ 4000 times less storage memory: $2 \cdot 24 \cdot 196560$ instead of 196560^2 parameters. Furthermore, $\mathcal{P}_n \subset \mathcal{K}_m(\mathcal{P}_n)$ ensures that – in stark contrast to direct low-rank factorization – *any* permutation matrix can still be represented *exactly*. Empirically, the optimization over parametrizations $\sigma(2VW^T - 1)$ turned out to cause significant challenges, likely due to the non-convexity and non-smoothness of the problem. To alleviate this problem, we resort to a smoother version of (8.4) which can still approximate permutations to an arbitrary desired accuracy:

Proposition 2. *Let $P \in \mathcal{P}_n$ and g denote an arbitrary permutation matrix and an arbitrary entry-wise strictly monotonically increasing function, respectively, and let s denote the row-wise Softmax function $s(A)_{i,j} = \frac{\exp A_{i,j}}{\sum_k \exp A_{i,k}}$. Then $\forall n \leq Kiss(m)$ and $\forall \epsilon > 0$ there exist $V, W \in \mathbb{R}^{n\times m}$ and $\alpha > 0$, such that*

$$\| P - s\left(\alpha g(VW^T)\right) \| \leq \epsilon. \tag{8.7}$$

Proof. Similar to the proof in Proposition 1, we start by choosing V satisfying (8.2) and setting $W = PV$ to obtain

$$(VW^T)_{ij} = \langle V_{i,:}, W_{j,:}\rangle \begin{cases} = 1 & \text{if } P_{i,j} = 1 \\ \leq 0.5 & \text{otherwise} \end{cases}. \tag{8.8}$$

Then $\forall i, j, k$ s.t. $P_{ij} = 1$ and $k \neq j$ (i.e., $P_{ik} = 0$) it holds that $g(VW^T)_{ij} > g(VW^T)_{ik}$. Finally, to yield the assertion we use the Softmax property of converging to the unit vector in the limit $s(\alpha A_{i,:}) \overset{\alpha\to\infty}{\to} e_j$ (with $j = \arg\max A_{i,:}$), by taking $\alpha > 0$ to be large enough. □

In practice, we use $g(x) = 2x$, in accordance with the representation in (8.4). We use this smoother version to validate the proposed low-rank representation for handling large matching problems in the experiments we report next.

8.4 Numerical Evaluation

The following experiments validate our efficient permutation estimation method for different applications, and they confirm the ability to scale to very large problem sizes. First, as a proof of concept, we demonstrate our approach on the application of point cloud alignment for the two non-linearities proposed in Section 8.3 and introduce our sparse training technique. We then valdidate the effectiveness of our approach in the context of linear assignment problems and show how to handle sparse cost matrices. We perform further experiments in the context of generic NP-hard quadratic assignment problems, and integrate our approach into a state-of-the-art shape matching pipeline, thus providing the same level of accuracy while enabling a higher spatial resolution.

8.4.1 Implementation Details

We use the PyTorch Adam optimizer [153] with its default hyperparameters in all our experiments.

Stochastic Optimization. Fully benefiting from our proposed compact representation requires the costs E (or an approximation thereof) to be evaluated *without ever forming the full (approximate) permutation matrix*, as this step would inherently return to necessitate n^2 many entries. To this end, we introduce the concept of *stochastic optimization*, which – for our softmax-based representation $s(2\alpha VW^T)$ arising from Proposition 2 – is not a stochastic training in a classical sense: we propose to fix all but two entries in each row of our approximate permutation. More specifically, in any supervised (learning-based) scenario where it is known that the y_i-entry of the i-the row of the final permutation P ought to be equal to one, each step of our optimizers merely computes the y_i-th and one randomly chosen (r_i-th entry) of each row, and computes the softmax s on these two entries only, i.e.,

$$P_{i,[y_i,r_i]} = s(2\alpha V_{i,:}(W_{[y_i,r_i],:})^T), \tag{8.9}$$

while implicitly assuming $P_{i,j} = 0$ for $j \notin \{y_i, r_i\}$.

In the above, we used $W_{[y_i,r_i],:}$ to denote the $2\times m$ matrix consisting of the y_i-th and the r_i-th row of W. Our stochastic approach requires the computation of $2n$ entries per gradient descent iteration only and – by randomly choosing the r_i – manages to still approximate the desired objective well.

Normalization of V and W. Since Proposition 1 and Proposition 2 rely on row-normalized matrices, we explicitly enforce this constraint whenever we compute the permutation P, by

using $V_{i,:} \leftarrow \frac{1}{\|V_{i,:}\|} V_{i,:}$, $W_{i,:} \leftarrow \frac{1}{\|W_{i,:}\|} W_{i,:}$. We omit this step from the presentations below for the sake of readability.

Softmax Temperature. Since the values of $\langle V_{i,:}, W_{j,:} \rangle$ are bounded by one following the aforementioned normalization, the *temperature* parameter α determines the trade-off between approximating a hard maximum (as required for accurately representing permutations, see Proposition 2) and favorable optimization properties (i.e., meaningful gradients). We specify the schedule (constant or monotonically increasing) in each of the experiments below.

8.4.2 Point Cloud Alignment

As a proof of concept, we demonstrate that our proposition is correct and the optimization process converges. We explore the different choices of non-linearity, starting with ReLU and continuing with Softmax, using the task of predicting a linear transformation over point clouds. In this task we aim to match a point cloud $X_1 \in \mathbb{R}^{n \times m}$ consisting of n m-dimensional points, uniformly distributed on the unit hyper-sphere, to its linearly transformed and randomly permuted version $X_2 \in \mathbb{R}^{n \times m}$.

To obtain X_2 we multiply X_1 by a randomly drawn matrix $\Theta_{\text{GT}} \in \mathbb{R}^{m \times m}$ and apply a random permutation. Then, we optimize over the estimated transformation matrix Θ which in this experiment defines our permutation matrix $P(\Theta)$:

$$P(\Theta) = \sigma\left(2VW(\Theta)^T - 1\right). \tag{8.10}$$

Note that in this case, our representation in (8.4) is fully parameterized by Θ, with $V = X_1$ and $W(\Theta) = X_2\Theta$, and V and W are row-wise normalized in each iteration. Here $P(\Theta)$ is equal to the correct permutation if the matrix Θ correctly aligns the point clouds, i.e. minimizes the angle between two corresponding points in V and $W(\Theta)$ while maximizing the angles between non-corresponding points.

We solve for the permutation by performing 20000 minimizing steps with a learning rate set to 0.01 over the negative log-likelihood loss

$$\hat{\Theta} = \arg\min_{\Theta} \; -\frac{1}{n} \sum_{i=1}^{n} \log\left(P(\Theta)_{i,y_i}\right), \tag{8.11}$$

where y_i is the index of the point in X_2 which corresponds to the i^{th} point in X_1. We experiment with different numbers of points n, each time choosing the dimension m to be just big enough to satisfy the Kissing number constraint from Proposition 1, i.e., $\text{Kiss}(m) \geq n > \text{Kiss}(m-1)$.

To check that we were able to find the correct transformation matrix Θ – and therefore the correct permutation matrix P – through optimization, we verify that the nearest neighbor (closest point) for each row i in V is located in row j of matrix W that satisfies $P_{i,j} = 1$. We find that this is indeed the case in all experiments with different number of points $n \in \{10, 100, 1000, 10000\}$, thus establishing that we could reach the correct representation through optimization. We achieve equally good results when replacing the point-wise non-linearity ReLU with Softmax $P(\Theta) = s\left(2\alpha VW(\Theta)^T\right)$.

Due to the quadratically growing size of the permutation matrix with an increasing number of points, we further propose to optimize for the permutation matrix stochastically, as described in Section 8.4.1. We ran experiments with similar settings as above,

wherein we gradually increased the value of the temperature parameter α linearly during optimization from $\alpha = 5 \cdot 10^{-5}$ to $\alpha = 1000$. In these experiments, we again found that each point was paired with its corresponding nearest neighbor, while reducing the memory consumption, as shown in Figure 8.3.

Accuracy values to the prediction of a linear transformation over point clouds are shown in Table 8.1. Here, we measure the distance between the true point clouds and their transformed counterparts, for each problem size (n).

nonlin.\n	10	100	1000	10000
ReLU	2.362×10^{-4}	1.0597×10^{-4}	4.115×10^{-4}	1.0387×10^{-4}
SoftMax	0.0829	0.026	0.0057	0.0012
Stoch. Softmax	0.0712	0.0164	0.00173	0.0002

Table 8.1: ℓ_2 norm distance between true and transformed point clouds in the point cloud alignment experiment across various non-linearities, and problem size (n).

8.4.3 Point Cloud Alignment on Spectral Point Representation

We conduct an additional experiment on point cloud alignment in the context of the functional maps framework [230]. Here, the goal is to extract a point-to-point correspondence between two shapes X, Y from a $m \times m$-dimensional functional map C [230] where m is much smaller than the number of vertices in X and Y. A possible interpretation of C is that it aligns the spectral point representations $\phi^X, \phi^Y \in \mathbb{R}^{n \times m}$ in which each point x is represented by the vector of values of the first m Laplace-Beltrami eigenfunctions at x such that $P \cdot \phi^X \approx \phi^Y \cdot C$ where P is the unknown permutation between X and Y. Given C, P can be retrieved by a nearest-neighbor query between $\phi^X, \phi^Y C$, as proposed in the original paper (see [230], Section 6.1), or by solving a Linear Assignment Problem (LAP) if a bijection is desired. This is exactly the same setting as in Section 8.4.2 with a small amount of noise in the point clouds. We use the FAUST registrations [26] with the original 6890 vertices, a downsampled version to 502 vertices for those experiments and C generated by the ground-truth correspondence. Then, $\phi^X, \phi^Y C$ can be directly used for V and W in our method.

We compare our method, which calculates correspondences the same way as described in Section 8.4.2, to a general LAP solver (specifically the Jonker-Volgenant algorithm from sklearn.linear_sum_assignment), nearest neighbor computation, optimal transport (as implemented in the python POT package), and stochastic matrices generated by Sinkhorn iterations. In Table 8.2 we show that our method outperforms all baselines in terms of geodesic error of the final matching and shows positive trends in terms of runtime and memory consumption.

8.4.4 Linear Assignment Problems

We next validate our method on balanced Linear Assignment Problems (LAPs), which typically involve assigning a set of agents to an equally sized set of tasks, e.g., when optimizing the allocation of resources. We show results on a collection of regularized LAPs

	502 vertices, $m = 20$			6890 vertices, $m = 50$		
Method	Error	Time	Memory	Error	Time	Memory
LAP	1.3×10^{-1}	0.023s	8.02 MB	3.1×10^{-1}	79.2s	565.31 MB
Nearest-Neighbors	8.2×10^{-1}	0.008s	5.22 MB	4.0×10^{-1}	2.6s	34.00 MB
Optimal Transport	3.8×10^{-1}	0.524s	12.58 MB	2.0×10^{-1}	182.7s	1862.27 MB
Sinkhorn iterations	1.4×10^{-1}	0.030s	9.4 MB	3.0×10^{-1}	12.0s	750.55 MB
Ours	2.2×10^{-3}	0.801s	18.18 MB	2.5×10^{-2}	77.6s	42.35 MB

Table 8.2: Comparisons of point-wise correspondence extraction from ground-truth functional map [30] for FAUST. The error is the mean geodesic matching error of all points. Please note that all code except ours and Sinkhorn iterations are from libraries that are likely more optimized in terms of runtime and memory consumption. The memory consumption is evaluated on CPU only.

in the form

$$\underset{V,W}{\arg\min} \quad \underbrace{\mathbf{tr}(A \cdot P(V,W))}_{\text{LAP term}} + \underbrace{\mu(P(V,W))}_{\text{regularizer}}, \tag{8.12}$$

where $P(V,W) = s\left(2\alpha VW^T\right)$ is a permutation and $A \in \mathbb{R}^{n \times n}$ is some given similarity matrix. While the Softmax non-linearity ensures all rows sum to one, $\mu(P(V,W))$ is a regularization term enforcing columns summing to one as well, to satisfy the permutation constraints:

$$\mu(P) = \sum_j \left(\sum_i P_{ij} - 1\right)^2. \tag{8.13}$$

Due to the row-wise Softmax all rows already sum to one but we incentive the columns to sum to one as well, as is necessary for permutations.

Dense Matrices. We evaluate on a set of LAPs based on descriptor similarities of 3D shapes from the FAUST dataset of human scans [26], with n randomly chosen vertices per object [106]. Let $D_X, D_Y \in \mathbb{R}^{n \times k}$ be two k-dimensional point-wise descriptors of the shapes X, Y corresponding to n points. We use the SHOT [261] ($\mathbb{R}^{n \times 352}$) and the heat kernel signature [283] ($\mathbb{R}^{n \times 101}$) with their default parameters as descriptors and stack them together to comprise $D_\cdot \in \mathbb{R}^{n \times 453}$ in total, then $A = D_X \cdot D_Y^\top$. Solving an LAP with this type of similariy matrix A is used e.g. in [303] as the initialization strategy.

We generate 100 problem instances by pairing each of the 100 shapes in FAUST with a random second shape to get the pair X, Y and evaluate the relative error of the energy (restricted to solutions that were valid permutations), and the average Hamming distance to the next valid permutation (namely, the number of rows or columns that violate the permutation constraint). We ran the experiments with $n = 100, m = 30, \alpha = 20$ and used a greedy heuristic to generate valid permutations from the results violating the permutation constraint (iteratively projecting the maximum value of the permutation to one, and the rest of the corresponding row and column to zero). Out of the 100 instances, 53 lead to valid permutations without the heuristic. The average relative error of immediately valid permutation is 1.8% and after pseudo-projection of all instances it is 2.0%. Due to the Softmax, every matrix has 100 non-zero entries that are all nearly equal to one. On average, the Hamming distance of invalid permutations to the next valid one is 1.38 (1.4%

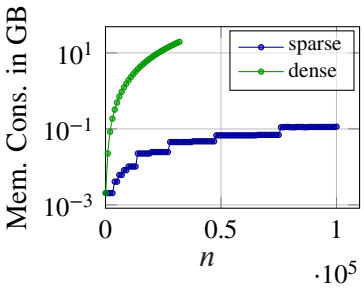


Figure 8.3: Memory consumption for point cloud alignment.

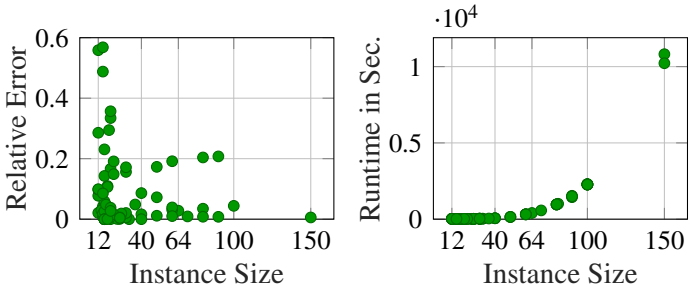


Figure 8.4: Relative error and runtime on the QAPLIB dataset.

of the problem size) which means in most cases one would have a valid permutation after only adjusting one entry.

Sparse Matrices. Given a matrix A , that is sparsely populated and only contains non-zero entries in a subset $S = \{(i, j)|A_{i,j} \neq 0\}$, we compute and optimize the permutation matrix sparsely in $(i, j) \in S$ by calculating the matrix factorization only at the required entries, similar to Section 8.4.1, but without restricting the number of entries per row of P to two. Also we take into account random entries $(q, r) \notin S$. We ran experiments for A with a matrix density of $|S| = 0.01n^2$ for $n = 1000, 5000$ and 10000 and $m = 20$ with increasing α from 1 to 20 iteratively and measure a Hamming distance of at most 0.28% of the problem size. To get a valid permutation matrix, we used the same heuristic as in the dense case and measured a relative error below 7.8%, compared to the Hungarian algorithm. Also, we could measure a memory reduction by over 65%.

8.4.5 Quadratic Assignment Problems

The QAP is a broadly employed mathematical tool for many real-life problems in operations research such as circuit design and shape matching. We demonstrate the application of our approach to non-convex QAPs of the form

$$\arg\min_{V,W} p(V, W)^T A\, p(V, W) \tag{8.14}$$

where $p(V, W) = \text{vec}(s\left(2\alpha V W^T\right))$ is the vectorized version of the permutation and $A \in \mathbb{R}^{n^2 \times n^2}$ is a cost matrix. V and W are normalized. The permutation matrix was optimized in a convex-concave manner, by optimizing the objective function

$$\arg\min_{V,W} p(V, W)^T (A - \beta I)\, p(V, W) + \mu(P(V, W)) \tag{8.15}$$

with β being iteratively increased from $-\|A\|_2$ to $\|A\|_2$ and with $\mu(P(V, W))$ being the same permutation constraint regularizer as in (8.13).

We show results on the QAPLIB [42] library of quadratic assignment problems of real-world applications which range between $n = 12$ and $n = 256$ and we choose $m = \text{ceil}(\frac{n}{3})$. The problems in the dataset are meant to be challenging and optimal solutions for some of the larger problems are not provided because they are unknown. Thus, we report the gap to optimality (when known) of our solution and consider a solution to be good if it falls within 10% of the optimum. We report the relative error and runtime in Figure 8.4. In 75 out of 87 instances the result was a proper permutation matrix.

Table 8.3: Geodesic errors and standard deviation (*std*) for noise-free and noisy data by Marin *et al.* [203] and our approach

	e_{prob}	$std\ (e_{prob})$	e_{emb}	$std\ (e_{emb})$	
[203]	0.051	17.4×10^{-4}	0.029	3.5×10^{-4}	noisy
ours	0.047	26.9×10^{-4}	0.026	29.2×10^{-4}	
[203]	0.043	16.3×10^{-4}	0.022	3.5×10^{-4}	noise-free
ours	0.041	8.1×10^{-4}	0.019	3.7×10^{-4}	

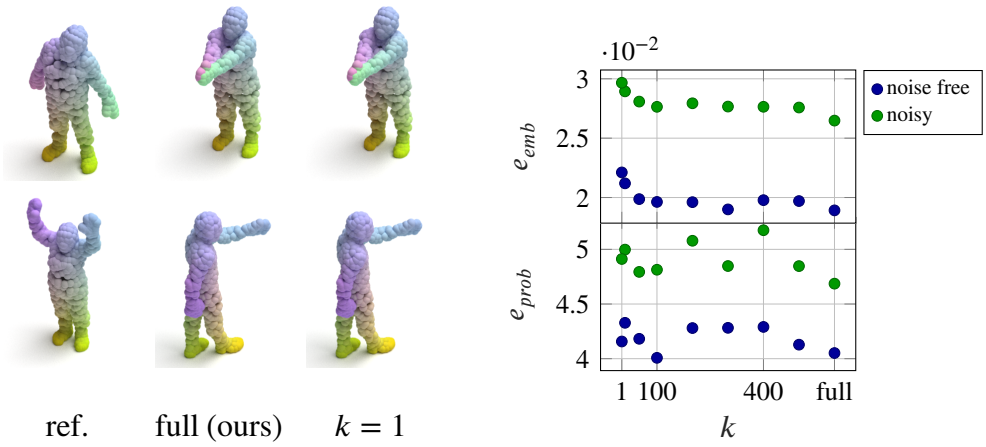
8.4.6 Shape Matching

Finally, we further assess the effectiveness of our approach for the application of non-rigid shape matching, a common task in computer graphics and computer vision. To this end, we incorporate our permutation matrix representation approach into the state-of-the-art shape-matching approach by Marin *et al.* [203], which learns the point correspondences using two consecutive networks $\mathcal{G}_\theta$ and $\tilde{\mathcal{G}}_\theta$, predicting shape embeddings and probe functions, respectively. We propose to replace the calculation of the permutation matrix based on the output of the first network $\mathcal{G}_\theta$ by $s\left(\alpha V W^T\right)$, with $\alpha = 40$. The network transforms the vertices of 3D objects X and Y into embeddings $\phi^x = \mathcal{G}_\theta(X)$ and $\phi^y = \mathcal{G}_\theta(Y)$, which are used to compute $V = \phi^x(\phi^{x\dagger} P_{gt} \phi^y)$ and $W = \phi^y$. V here replaces a transformed embedding. The network is trained on the modified loss function

$$\min_\theta \sum_l \| s\left(2\alpha V W^T\right)^l Y^l - P_{gt}^l Y^l \|_2^2 \tag{8.16}$$

for a given ground truth permutation P_{gt}, and V and W being normalized row-wise. Similar to Marin *et al.* , we train the networks over 1600 epochs on 10000 shapes of the SURREAL dataset [300] and evaluate our experiments on 100 noisy and noise-free objects of different shapes and poses of the FAUST dataset [26], that are provided by [203] in [202]. We follow the evaluation of Marin *et al.* [203] and calculate the geodesic distance between the ground truth matching and the predicted matching $match_1 = \mathcal{N}(\phi^x C_1^T, \phi^y)$ for $C_1 = ((\phi^{y\dagger}\tilde{\mathcal{G}}_\theta(Y))^T)^\dagger(\phi^{x\dagger}\tilde{\mathcal{G}}_\theta(X))^T$, whereby $\mathcal{N}$ is the nearest neighbor, $\dagger$ denotes the Moore-Penrose inverse and the calculation of C_1 arises from the following relation $C_1^T \phi^{y\dagger}\tilde{\mathcal{G}}_\theta = \phi^{x\dagger}\tilde{\mathcal{G}}_\theta$ of the alignment C_1. In the following, we refer to the measured geodesic distance as e_{prob}. A second error (e_{emb}) which only concerns the first network's predictions, is measured by the geodesic distance towards $match_2 = \mathcal{N}(\phi^x, \phi^y C_2)$ for $C_2 = \phi^{y\dagger} P_{gt} \phi^x$, which is, again, calculated following Marin *et al.* [203]. The results of our experiments are reported in Table 8.3, showing the average geodesic errors (over 10 runs for each experiment) for the approach presented in [203] and our method. The table reveals improved results compared to [203].

Stochastic Training. Given that the explicit calculation of the permutation matrix in (8.16) is memory-intensive for a large number of vertices, we employ stochastic training to avoid the need for computing the full permutation matrix. As we describe in Section 8.4.1 we only calculate the loss over a few entries where the final permutation ought to be equal to one and on k (here k can be ≥ 1) randomly chosen entries of each row of P in each iteration. This approach reduces the memory requirement and gives us the possibility to train



(a) Visualized results on FAUST

(b) Error values for different k

Figure 8.5: Visualized matching results (a) and error values (b) for the FAUST dataset for different levels of sparseness k during stochastic training.

with larger shapes consisting of more vertices. In our experiments, we applied the stochastic training technique on the SURREAL dataset, and then evaluated the performance on FAUST by measuring the error rates for varying values of k, as depicted in Figure 8.5b. We observed a small relative increase of less than 17% in e_{emb}, and also a small effect on e_{prob}, but with a less clear tendency as one could see for e_{emb}. For e_{prob} we measured an average standard deviation of 2.25×10^{-3} and for e_{emb} of 3.3×10^{-4}. Two noise-free examples of correspondences, visualized for full training and for stochastic training with $k = 1$, are shown in Figure 8.5a, with the reference image on the left and the corresponding shapes on the right.

Further details regarding the influence of the variable k, that determines the sparsity of the calculated permutation matrix, on the computation speed are shown in Figure 8.6. It shows the average computational speed per epoch (employing a batch size of 8) for the network $\mathcal{G}_\theta$.

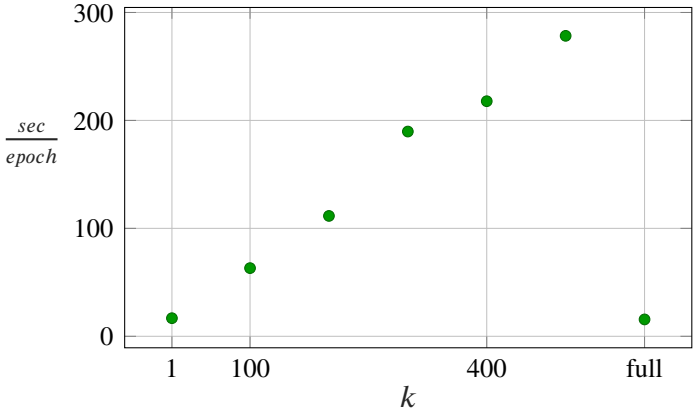


Figure 8.6: Average time (in seconds) needed to optimize the shape matching network $\mathcal{G}_\theta$ per epoch, depending on the stochastic variable k.

To evaluate the impact on the error and memory consumption when dealing with objects of larger size (consisting of more vertices), we ran further experiments using data from the TOSCA dataset [36].

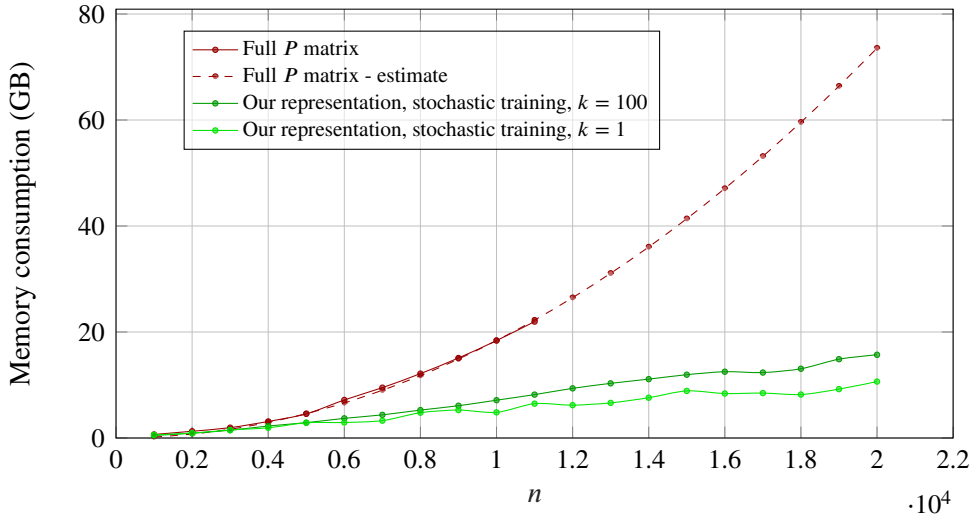


Figure 8.7: Memory usage when training the shape matching network with different permutation matrix representations: Using full matrices (red) vs. using our stochastic training scheme with different sparseness levels k (green).

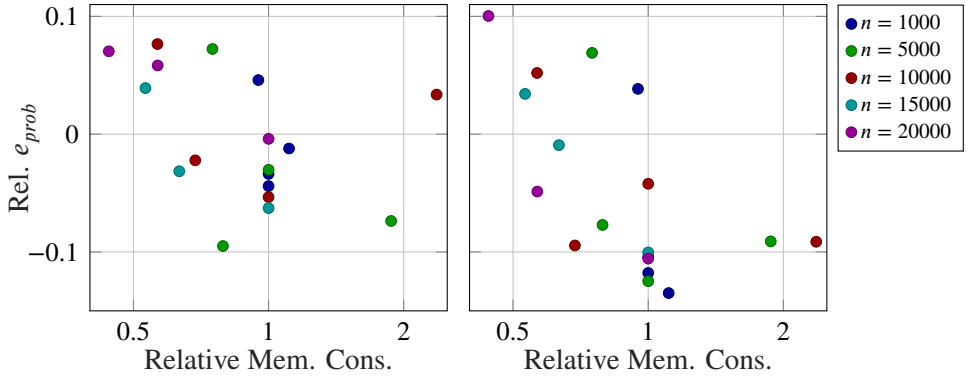


Figure 8.8: The relative memory-error trade-off for a varying number of vertices (n) and for varying sparseness (which causes the memory reduction), whereby the memory consumption is relative to $k = 100$ and the errors are relative to full training by [203] for $n = 1000$.

We trained for 400 epochs on the objects of the classes *victoria* and *michael* (32 objects in total) where up to 20000 vertices were sampled. These experiments revealed a reduction of memory consumption as shown in Figure 8.7. There, we further visualize how the proposed approach enables handling large problems that would have been infeasible otherwise. The dashed red curve added to this figure corresponds to the estimated memory that would have been required to accommodate full permutation matrices, as a function of problem size n. While our approach can accurately handle large problems with as much as $n = 20000$ vertices (green curves), running the equivalent experiments without it (red curves) would require prohibitively large amounts of memory ($\sim$ 73.6 GB, vs. 10.7 GB using $k = 1$). For estimating memory values (dashed curve) we assume memory usage follows a $c \cdot n^2$ curve, and estimate the value for c based on the full matrix experiments (solid red) we conducted for $n \leq 11000$. We evaluated the training on the class *david* of TOSCA and reported a relative memory-error trade-off for up to 20000 samples of each object in Figure 8.8. The graph indicates a correlation between higher memory usage and

lower error values for e_{emb}. The trends observed in the memory-error trade-off for e_{emb} are generally applicable to e_{prob} as well, although with some noticeable outliers.

8.5 Conclusion

In this chapter, we proposed a strategy to represent permutation matrices by a low-rank matrix factorization followed by a nonlinearity and showed that by using the Kissing number theory, we can determine the minimum rank necessary for representing a permutation matrix of a given size, allowing for a more memory-efficient representation. We validated this method with experiments on LAPs and QAPs as well as a real-world shape matching application and showed improvements in the latter. Additionally, we explored the potential of optimizing permutations stochastically to decrease memory usage, which opens the possibility of handling high-resolution data.

Our method offers a promising solution to contribute positively to the environment by reducing the computational cost of a variety of problems involving permutation matrices. However, it is important to acknowledge a limitation of our method. For certain problem formulations, such as the Koopmans and Beckmann form QAPs, stochastic learning may not be feasible because the double occurrences of the permutation matrix make the stochastic computation not applicable. Moreover, our method requires devising a non-trivial, problem-specific adaptation.

Adaptions to our method can involve the learning rate, as well as the selection of the α-parameter. If talking about (8.4), one can consider the adaptation of the thresholding (for the equation $2\sigma(2VW^T - 1)$ the threshold is set to 1). By decreasing the threshold, we simplify the optimization process, as fewer gradients are excluded from the experiment, while this could result in a less precise outcome. Additional adaptions might concern optimization techniques, such as fixing one matrix with descent characteristics (e.g. Gaussian random) in order to simplify the optimization. Moreover, it's possibly also necessary to adapt a network architecture that predicts the matrices V and W, and with further research in this direction, we believe to expand the potential to provide memory reduction benefits.

Part III

Closure

CHAPTER 9

Conclusion

In this work, we explored various methods for combining model-based and learning-based approaches and analyzed their individual behaviors. We also discussed how to harness the power of deep learning for inverse problems that are traditionally addressed using classical methods. Throughout this study, we observed a consistent trend: the combination of model-based and learning-based methods offers advantages across diverse applications.

9.1 Summary and Impact

We first considered a sparse data scenario, which is important in situations where a vast dataset is unavailable or impractical, and analyzed the model- and learning-based capabilities in *Chapter 5* for single image segmentation with user induced scribbles. Although deep learning has made significant steps in various domains, especially in specific scenarios such as single image learning, traditional model-based approaches are still of great value, as notably cleverly designed model-based techniques have shown to outperform standalone deep learning methods. However, the use of semantic data, which may be acquired through learning-based approaches on other data, supports further improvement of traditional methods, leading to a favor of combining modell- and learning-based methods.

The semantic information that comes from the data-driven methods can be of high value for classical approaches in multiple ways. In scenarios with enough available (training-) data, data-driven methods can offer support of classical approaches, that ensure strict result guarantees. Particularly in underdetermined reconstruction problems, classical approaches offer data consistency, but still introduce ambiguities in the result. Our findings indicate that underdetermined classical methods can be enhanced and better regulated using deep learning, especially when considering biases present in the training data. In the context of CT imaging, inaccuracies in data recordings can lead to misleading reconstructions. However in *Chapter 6* we have shown that by semantically guiding the reconstruction process, we can prioritize semantically relevant scenarios. Furthermore, in *Chapter 7* we demonstrated that underdetermined classical reconstruction challenges can benefit from semantic data-driven insights, even for non-differentiable problems like binarization or Poisson noise denoising, by extending energy dissipating networks to non-differentiable energies.

Another kind of task that can arise in the field of computer vision are problems where the challenge lies in the problem formulation itself, concerning model- and learning-based methods likewise. One kind of problem, where the challenges lie in their intrinsic formulation, are memory-intensive assignment problems. In *Chapter 8* we were able to largely solve the problem by changing the representation of the memory-intensive permutation matrix by representing large binary permutation matrices by nonlinear matrix factorization. This approach not only reduced memory requirements, but also showed better performance in shape matching.

In this thesis we have analyzed modell- and learning-based capabilities, discussed the value of deep learning for underdetermined problems, and studied intrinsic formulation challenges. As the challenges in computer vision continue to change, it is important to take advantage of the best of both worlds to achieve optimal results.

9.2 Limitations and Future Work

The results of this thesis open up several interesting directions for further research and technical challenges.

When comparing model-based methods, which are cleverly constructed, with *scribble-based single image segmentation* using neural networks, the former was found to be ahead in performance. An interesting question for further research is whether the underlying concepts of this model-based method, which uses both spatial and color information, can be integrated into learning methods. Two possible directions to do this are techniques such as unrolling or the learning of individual weights within this model-based framework.

The use of an adversarially pre-trained classification network for *semantically guidance of CT reconstruction* raises the question about how adversarial training influences the reconstruction results. Future research could explore how different adversarial training techniques for neural networks might yield different outcomes. Also, to prevent the development of visually implausible outcomes arising from the semantic guidance, the integration of learning techniques such as GAN and close collaboration with medical experts are important to ensure clinically reliable results. We have limited the guidance only to restricted regions with abnormalities in the target image – an application of the method to different areas where there has been no abnormality so far, as well as an extension of the guidance to the full reconstruction image by e.g. integration of detection networks are open and important research directions.

This approach, as well as the semantic application by *energy dissipating networks*, has the potential for other applications in different imaging modalities, such as 3D CT or MRI. In addition, for energy dissipating networks, future work on the extention of this approach to non-convex energies would be interesting to broaden the scope of applications.

In the context of *permutation prediction by nonlinear matrix factorization*, especially for shape matching problems, we have shown good results, but faced problems concerning the adaptability of this representation to different algorithms which remains an area for further research. For learning-based applications, a detailed analysis of the influence of network properties on permutation prediction performance could give information about inconsistent results observed in our experiments. Furthermore, while stochastic optimization has successfully reduced memory requirements, it has also increased computation time. Addressing this issue is less of a research question and more of a technical challenge. An efficient GPU implementation using custom kernels could significantly decrease the runtime.

Bibliography

[1] R. P. Adams and R. S. Zemel. “Ranking via sinkhorn propagation”. In: *arXiv preprint arXiv:1106.1925* (2011).

[2] J. Adler and O. Öktem. “Learned primal-dual reconstruction”. In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1322–1332.

[3] M. Aharon, M. Elad, and A. Bruckstein. “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation”. In: *IEEE Transactions on signal processing* 54.11 (2006), pp. 4311–4322.

[4] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. “Semantic soft segmentation”. In: *ACM Transactions on Graphics* 37.4 (2018), pp. 1–13.

[5] M. Al-Shabi, B. L. Lan, W. Y. Chan, K.-H. Ng, and M. Tan. “Lung nodule classification using deep local–global networks”. In: *International journal of computer assisted radiology and surgery* 14.10 (2019), pp. 1815–1819.

[6] A. H. Al-Shabili, X. Xu, I. Selesnick, and U. S. Kamilov. “Bregman Plug-and-Play priors”. In: *IEEE International Conference on Image Processing*. IEEE. 2022, pp. 241–245.

[7] G. S. Alberti, E. De Vito, M. Lassas, L. Ratti, and M. Santacesaria. “Learning the optimal regularizer for inverse problems”. In: *stat* 1050 (2021), p. 11.

[8] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. “Contour detection and hierarchical image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2010), pp. 898–916.

[9] S. G. Armato 3rd et al. “The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans”. In: *Medical Physics* 38.2 (2011), pp. 915–931.

[10] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. “Solving inverse problems using data-driven models”. In: *Acta Numerica* 28 (2019), pp. 1–174.

[11] M. Asim, F. Shamshad, and A. Ahmed. “Patchdip exploiting patch redundancy in deep image prior for denoising”. In: *NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks*. 2019.

[12] H. Attouch, J. Bolte, and B. F. Svaiter. “Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods”. In: *Mathematical Programming* 137.1-2 (2013), pp. 91–129.

[13] M. Awad. “An unsupervised artificial neural network method for satellite image segmentation”. In: *International Arab Journal of Information Technology* 7.2 (2010), pp. 199–205.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495.

[15] E. Bae, J. Yuan, and X.-C. Tai. “Global minimization for continuous multiphase partitioning problems using a dual approach”. In: *International Journal of Computer Vision* 92.1 (2011), pp. 112–129.

[16] Y. Bahat and T. Michaeli. “Explorable super resolution”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2716–2725.

[17] Y. Bahat and T. Michaeli. “What’s in the Image? Explorable Decoding of Compressed Images”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2908–2917.

[18] S. Bai, J. Z. Kolter, and V. Koltun. “Deep equilibrium models”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[19] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang. “Recent advances in adversarial training for adversarial robustness”. In: *arXiv preprint arXiv:2102.01356* (2021).

[20] S. Baluja and I. Fischer. “Adversarial transformation networks: Learning to generate adversarial examples”. In: *arXiv preprint arXiv:1703.09387* (2017).

[21] A. Beck and M. Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202.

[22] Y. Bengio, I. Goodfellow, and A. Courville. *Deep learning*. Vol. 1. MIT press Cambridge, MA, USA, 2017.

[23] F. Bernard, C. Theobalt, and M. Moeller. “Ds*: Tighter lifting-free convex relaxations for quadratic matching problems”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4310–4319.

[24] J. Bian, J. H. Siewerdsen, X. Han, E. Y. Sidky, J. L. Prince, C. A. Pelizzari, and X. Pan. “Evaluation of sparse-view reconstruction from flat-panel-detector cone-beam CT”. In: *Physics in Medicine & Biology* 55.22 (2010), p. 6575.

[25] G. Birkhoff. “Tres observaciones sobre el algebra lineal [Three observations in linear algebra]”. In: *Univ. Nac. Tucuman, Ser. A* 5 (1946), pp. 147–154.

[26] F. Bogo, J. Romero, M. Loper, and M. J. Black. “FAUST: Dataset and evaluation for 3D mesh registration”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3794–3801.

[27] M. Bojarski et al. “End to end learning for self-driving cars”. In: *arXiv preprint arXiv:1604.07316* (2016).

[28] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.

[29] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[30] Y. Y. Boykov and M.-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images". In: *IEEE International Conference on Computer Vision*. Vol. 1. IEEE. 2001, pp. 105–112.

[31] P. Boyvalenkov, S. Dodunekov, and O. R. Musin. "A survey on the kissing numbers". In: *arXiv preprint arXiv:1507.03631* (2015).

[32] G. Braun, A. Carderera, C. W. Combettes, H. Hassani, A. Karbasi, A. Mokhtari, and S. Pokutta. "Conditional gradient methods". In: *arXiv preprint arXiv:2211.14103* (2022).

[33] K. Bredies, K. Kunisch, and T. Pock. "Total generalized variation". In: *SIAM Journal on Imaging Sciences* 3.3 (2010), pp. 492–526.

[34] D. J. Brenner and E. J. Hall. "Computed tomography—an increasing source of radiation exposure". In: *New England Journal of Medicine* 357.22 (2007), pp. 2277–2284.

[35] A. Brifman, Y. Romano, and M. Elad. "Turning a denoiser into a super-resolver using plug and play priors". In: *IEEE International Conference on Image Processing*. IEEE. 2016, pp. 1404–1408.

[36] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[37] A. Buades, B. Coll, and J.-M. Morel. "A review of image denoising algorithms, with a new one". In: *Multiscale Modeling & Simulation* 4.2 (2005), pp. 490–530.

[38] M. Bucher, T.-H. Vu, M. Cord, and P. Pérez. "Zero-shot semantic segmentation". In: *Advances in Neural Information Processing Systems* 32 (2019).

[39] M. C. Buhler, A. Romero, and R. Timofte. "Deepsee: Deep disentangled semantic explorative extreme super-resolution". In: *Proceedings of the Asian Conference on Computer Vision*. 2020.

[40] M. Burger, A. Sawatzky, and G. Steidl. *First order algorithms in variational image processing*. Springer, 2016.

[41] R. E. Burkard and E. Cela. "Linear assignment problems and extensions". In: *Handbook of Combinatorial Optimization: Supplement Volume A*. Springer, 1999, pp. 75–149.

[42] R. E. Burkard, S. E. Karisch, and F. Rendl. "QAPLIB – A quadratic assignment problem library". In: *Journal of Global Optimization* 10 (1997).

[43] H. Cai, J. He, Y. Qiao, and C. Dong. "Toward Interactive Modulation for Photo-Realistic Image Restoration". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2021, pp. 294–303.

[44] F. Caluza Machado and F. M. de Oliveira Filho. "Improving the semidefinite programming bound for the kissing number by exploiting polynomial symmetry". In: *Experimental Mathematics* 27.3 (2018), pp. 362–369.

[45] D. Cao and F. Bernard. “Self-Supervised Learning for Multimodal Non-Rigid 3D Shape Matching”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023.

[46] N. Carlini and D. Wagner. “Towards evaluating the robustness of neural networks”. In: *IEEE Symposium on Security and Privacy*. IEEE. 2017, pp. 39–57.

[47] A. Chambolle, D. Cremers, and T. Pock. “A convex approach to minimal partitions”. In: *SIAM Journal on Imaging Sciences* 5.4 (2012).

[48] A. Chambolle and T. Pock. “A first-order primal-dual algorithm for convex problems with applications to imaging”. In: *Journal of Mathematical Imaging and Vision* 40.1 (2011), pp. 120–145.

[49] T. F. Chan and L. A. Vese. “Active contours without edges”. In: *IEEE Transactions on Image Processing* 10.2 (2001).

[50] A. Chaurasia and E. Culurciello. “Linknet: Exploiting encoder representations for efficient semantic segmentation”. In: *IEEE Visual Communications and Image Processing*. IEEE. 2017, pp. 1–4.

[51] D. Chen et al. “An interactive image segmentation method in hand gesture recognition”. In: *Sensors* 17.2 (2017), p. 253.

[52] G.-H. Chen, J. Tang, and S. Leng. “Prior image constrained compressed sensing (PICCS): a method to accurately reconstruct dynamic CT images from highly undersampled projection data sets”. In: *Medical Physics* 35.2 (2008), pp. 660–663.

[53] H. Chen et al. “Low-dose CT with a residual encoder-decoder convolutional neural network”. In: *IEEE Transactions on Medical Imaging* 36.12 (2017), pp. 2524–2535.

[54] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. “Semantic image segmentation with deep convolutional nets and fully connected crfs”. In: *arXiv preprint arXiv:1412.7062* (2014).

[55] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. “Rethinking atrous convolution for semantic image segmentation”. In: *arXiv preprint arXiv:1706.05587* (2017).

[56] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. “Encoder-decoder with atrous separable convolution for semantic image segmentation”. In: *European Conference on Computer Vision*. 2018, pp. 801–818.

[57] X. Chen, J. Liu, Z. Wang, and W. Yin. “Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[58] Y. Chen, M. Mancini, X. Zhu, and Z. Akata. “Semi-supervised and unsupervised deep visual learning: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[59] Y. Chen, X. Yin, L. Shi, H. Shu, L. Luo, J.-L. Coatrieux, and C. Toumoulin. “Improving abdomen tumor low-dose CT images using a fast dictionary learning based processing”. In: *Physics in Medicine & Biology* 58.16 (2013), p. 5803.

[60] Y.-C. Chen, C. Gao, E. Robb, and J.-B. Huang. “Nas-dip: Learning deep image prior with neural architecture search”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 442–459.

[61] Y. Chen and T. Pock. "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2016), pp. 1256–1272.

[62] Z.-H. Chen, J.-T. Kim, J. Liang, J. Zhang, Y.-B. Yuan, et al. "Real-time hand gesture recognition using finger segmentation". In: *The Scientific World Journal* 2014 (2014).

[63] Z. Chen, X. Jin, L. Li, and G. Wang. "A limited-angle CT reconstruction method based on anisotropic TV minimization". In: *Physics in Medicine & Biology* 58.7 (2013), p. 2119.

[64] J.-H. Choi, H. Zhang, J.-H. Kim, C.-J. Hsieh, and J.-S. Lee. "Evaluating robustness of deep image super-resolution against adversarial attacks". In: *IEEE International Conference on Computer Vision*. 2019, pp. 303–311.

[65] S. Y. Chun, Y. K. Dewaraja, and J. A. Fessler. "Alternating direction method of multiplier for tomography with nonlocal regularizers". In: *IEEE Transactions on Medical Imaging* 33.10 (2014), pp. 1960–1968.

[66] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. "3D U-Net: learning dense volumetric segmentation from sparse annotation". In: *Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 424–432.

[67] D. Ciregan, U. Meier, and J. Schmidhuber. "Multi-column deep neural networks for image classification". In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3642–3649.

[68] K. Clark et al. "The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository". In: *Journal of Digital Imaging* 26.6 (2013), pp. 1045–1057.

[69] R. Cohen, Y. Blau, D. Freedman, and E. Rivlin. "It has potential: Gradient-driven denoisers for convergent solutions to inverse problems". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18152–18164.

[70] M. Cordts et al. "The cityscapes dataset for semantic urban scene understanding". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[71] T. Cour, F. Benezit, and J. Shi. "Spectral segmentation with multiscale graph decomposition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. IEEE. 2005.

[72] K. Crane, M. Livesu, E. Puppo, and Y. Qin. "A survey of algorithms for geodesic paths and distances". In: *arXiv preprint arXiv:2007.10430* (2020).

[73] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. "Instance-sensitive fully convolutional networks". In: *European Conference on Computer Vision*. Springer. 2016, pp. 534–549.

[74] J. Dai, K. He, and J. Sun. "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation". In: *IEEE International Conference on Computer Vision*. 2015.

[75] T. Dai, Y. Feng, D. Wu, B. Chen, J. Lu, Y. Jiang, and S.-T. Xia. "Dipdefend: Deep image prior driven defense against adversarial examples". In: *ACM International Conference on Multimedia*. 2020, pp. 1404–1412.

[76] Z. Dai, H. Liu, Q. V. Le, and M. Tan. "Coatnet: Marrying convolution and attention for all data sizes". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 3965–3977.

[77] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 248–255.

[78] R. Dey and V. N. Boddeti. "Generating diverse 3D reconstructions from a single occluded face image". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1547–1557.

[79] A. Diallo, M. Zopf, and J. Fürnkranz. "Permutation learning via lehmer codes". In: *European Conference on Artificial Intelligence*. IOS Press, 2020, pp. 1095–1102.

[80] J. Diebold, N. Demmel, C. Hazırbaş, M. Moeller, and D. Cremers. "Interactive multi-label segmentation of rgb-d images". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2015.

[81] C. Dong, C. C. Loy, K. He, and X. Tang. "Image super-resolution using deep convolutional networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2015), pp. 295–307.

[82] J. Dong, J. Chen, X. Xie, J. Lai, and H. Chen. "Adversarial attack and defense for medical image analysis: methods and applications". In: *arXiv preprint arXiv:2303.14133* (2023).

[83] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. "Boosting adversarial attacks with momentum". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9185–9193.

[84] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu. "Efficient decision-based black-box adversarial attacks on face recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7714–7722.

[85] A. Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[86] H. Dröge, Y. Bahat, F. Heide, and M. Moeller. "Explorable data consistent CT reconstruction". In: *British Machine Vision Conference*. BMVA Press, 2022. URL: https://bmvc2022.mpi-inf.mpg.de/0746.pdf.

[87] H. Dröge, Z. Lähner, Y. Bahat, O. Martorell, F. Heide, and M. Möller. "Kissing to find a match: efficient low-Rank permutation representation". In: *arXiv preprint arXiv:2308.13252* (2023).

[88] H. Dröge and M. Moeller. "Learning or modelling? An analysis of single image segmentation based on scribble information". In: *IEEE International Conference on Image Processing*. IEEE. 2021, pp. 2274–2278.

[89] H. Dröge, T. Möllenhoff, and M. Möller. "Non-smooth energy dissipating networks". In: *IEEE International Conference on Image Processing*. IEEE. 2022, pp. 3281–3285.

[90] H. Dröge, B. Yuan, R. Llerena, J. T. Yen, M. Moeller, and A. L. Bertozzi. "Mitral valve segmentation using robust nonnegative matrix factorization". In: *Journal of imaging* 7.10 (2021), p. 213.

[91] M. Duff, N. D. Campbell, and M. J. Ehrhardt. "Regularising inverse problems with generative machine learning models". In: *arXiv preprint arXiv:2107.11191* (2021).

[92] N. Dym, H. Maron, and Y. Lipman. "DS++ - A flexible, scalable and provably tight relaxation for matching problems". In: *ACM Transactions on Graphics* 36.6 (2017).

[93] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun. "Convit: Improving vision transformers with soft convolutional inductive biases". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2286–2296.

[94] G. Easley, D. Labate, and W.-Q. Lim. "Sparse directional image representations using the discrete shearlet transform". In: *Applied and Computational Harmonic Analysis* 25.1 (2008), pp. 25–46.

[95] M. Eisenberger, Z. Lähner, and D. Cremers. "Divergence-Free Shape Correspondence by Deformation". In: *Symposium on Geometry Processing*. 2019.

[96] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi. "Neuromorph: Unsupervised shape interpolation and correspondence in one go". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021.

[97] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. "Deep shells: Unsupervised shape correspondence with optimal transport". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 10491–10502.

[98] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. "The pascal visual object classes (voc) challenge". In: *International Journal of Computer Vision* 88 (2010), pp. 303–338.

[99] L. A. Feldkamp, L. C. Davis, and J. W. Kress. "Practical cone-beam algorithm". In: *Journal of The Optical Society of America A-optics Image Science and Vision* 1 (1984), pp. 612–619.

[100] D. Feng et al. "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges". In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1341–1360.

[101] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam. "Adversarial attacks against medical deep learning systems". In: *arXiv preprint arXiv:1804.05296* (2018).

[102] M. Frank, P. Wolfe, et al. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110.

[103] D. Gabay and B. Mercier. "A dual algorithm for the solution of nonlinear variational problems via finite element approximation". In: *Computers & Mathematics with Applications* 2.1 (1976), pp. 17–40.

[104] Y. Gandelsman, A. Shocher, and M. Irani. ""Double-DIP": unsupervised image decomposition via coupled deep-image-priors". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11026–11035.

[105] K. V. Gandikota, P. Chandramouli, H. Dröge, and M. Moeller. "Evaluating Adversarial Robustness of Low dose CT Recovery". In: *Medical Imaging with Deep Learning*. 2023.

[106] M. Gao, Z. Lähner, J. Thunberg, D. Cremers, and F. Bernard. "Isometric Multi-Shape Matching". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021.

[107] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. "Inverting gradients-how easy is it to break privacy in federated learning?" In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16937–16947.

[108] S. Ghosh, N. Das, I. Das, and U. Maulik. "Understanding deep learning techniques for image segmentation". In: *ACM Computing Surveys* 52.4 (2019), pp. 1–35.

[109] D. Gilton, G. Ongie, and R. Willett. "Deep equilibrium architectures for inverse problems in imaging". In: *IEEE Transactions on Computational Imaging* 7 (2021), pp. 1123–1133.

[110] R. Girshick. "Fast R-CNN". In: *IEEE International Conference on Computer Vision*. 2015, pp. 1440–1448.

[111] S. Gold and A. Rangarajan. "A graduated assignment algorithm for graph matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (1996).

[112] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[113] K. Gregor and Y. LeCun. "Learning fast approximations of sparse coding". In: *International Conference on Machine Learning*. 2010, pp. 399–406.

[114] D. M. Greig, B. T. Porteous, and A. H. Seheult. "Exact maximum a posteriori estimation for binary images". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 51.2 (1989).

[115] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. "3D-CODED : 3D correspondences by deep deformation". In: *European Conference on Computer Vision*. 2018.

[116] A. Grover, E. Wang, A. Zweig, and S. Ermon. "Stochastic optimization of sorting networks via continuous relaxations". In: *arXiv preprint arXiv:1903.08850* (2019).

[117] J. Gu et al. "Recent advances in convolutional neural networks". In: *Pattern Recognition* 77 (2018), pp. 354–377.

[118] S. Gu, E. Holly, T. Lillicrap, and S. Levine. "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3389–3396.

[119] Z. Gu, S. Zhou, L. Niu, Z. Zhao, and L. Zhang. "Context-aware feature generation for zero-shot semantic segmentation". In: *ACM International Conference on Multimedia*. 2020.

[120] J. Guo and H. Chao. "One-to-many network for visually pleasing compression artifacts reduction". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3038–3047.

[121] Y. Guo, C. Zhang, C. Zhang, and Y. Chen. "Sparse dnns with improved adversarial robustness". In: *Advances in Neural Information Processing Systems* 31 (2018).

[122] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser. "CNN-based projected gradient descent for consistent CT image reconstruction". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1440–1453.

[123] O. Halimi, O. Litany, E. Rodolà, A. M. Bronstein, and R. Kimmel. "Unsupervised learning of dense shape correspondence". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4370–4379.

[124] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. "Learning a variational network for reconstruction of accelerated MRI data". In: *Magnetic Resonance in Medicine* 79.6 (2018), pp. 3055–3071.

[125] Y. S. Han, J. Yoo, and J. C. Ye. "Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis". In: *arXiv preprint arXiv:1611.06391* (2016).

[126] J. He, Y. Wang, and J. Ma. "Radon inversion via deep learning". In: *IEEE Transactions on Medical Imaging* 39.6 (2020), pp. 2076–2087.

[127] J. He et al. "Optimizing a parameterized plug-and-play ADMM for iterative low-dose CT reconstruction". In: *IEEE Transactions on Medical Imaging* 38.2 (2018), pp. 371–382.

[128] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask r-cnn". In: *IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969.

[129] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[130] H. Heaton, X. Chen, Z. Wang, and W. Yin. "Safeguarded learned convex optimization". In: 37.6 (2023), pp. 7848–7855.

[131] F. Heide et al. "Flexisp: A flexible camera image processing framework". In: *ACM Transactions on Graphics* 33.6 (2014), pp. 1–13.

[132] J. Hendrik Metzen, M. Chaithanya Kumar, T. Brox, and V. Fischer. "Universal adversarial perturbations against semantic image segmentation". In: *IEEE International Conference on Computer Vision*. 2017, pp. 2755–2764.

[133] J. R. Hershey, J. L. Roux, and F. Weninger. "Deep unfolding: Model-based inspiration of novel deep architectures". In: *arXiv preprint arXiv:1409.2574* (2014).

[134] S. Hong, J. Oh, H. Lee, and B. Han. "Learning transferrable knowledge for semantic segmentation with deep convolutional neural network". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[135] S. A. H. Hosseini, B. Yaman, S. Moeller, M. Hong, and M. Akçakaya. "Dense recurrent neural networks for accelerated MRI: History-cognizant unrolling of optimization algorithms". In: *IEEE Journal of Selected Topics in Signal Processing* 14.6 (2020), pp. 1280–1291.

[136] Z. Hu and H. Zheng. "Improved total variation minimization method for few-view computed tomography image reconstruction". In: *BioMedical Engineering OnLine* 13.1 (2014), pp. 1–10.

[137] P. J. Huber. "Robust estimation of a location parameter". In: *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[138] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi. "Oneformer: One transformer to rule universal image segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2989–2998.

[139] J. Jain, A. Singh, N. Orlov, Z. Huang, J. Li, S. Walton, and H. Shi. "Semask: Semantically masked transformers for semantic segmentation". In: *IEEE International Conference on Computer Vision*. 2023, pp. 752–761.

[140] Y. J. Jia, Y. Lu, J. Shen, Q. A. Chen, H. Chen, Z. Zhong, and T. W. Wei. "Fooling detection alone is not enough: Adversarial attack against multiple object tracking". In: *International Conference on Learning Representations*. 2020.

[141] B. Jiang, P. Sun, J. Tang, and B. Luo. "Glmnet: Graph learning-matching networks for feature matching". In: *arXiv preprint arXiv:1911.07681* (2019).

[142] M. F. M. Jimenez, O. DeGuchy, and R. F. Marcia. "Deep convolutional autoencoders for deblurring and denoising low-resolution images". In: *2020 International Symposium on Information Theory and Its Applications (ISITA)*. IEEE. 2020, pp. 549–553.

[143] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. "Deep convolutional neural network for inverse problems in imaging". In: *IEEE Transactions on Image Processing* 26.9 (2017), pp. 4509–4522.

[144] U. S. Kamilov, H. Mansour, and B. Wohlberg. "A plug-and-play priors approach for solving nonlinear imaging inverse problems". In: *IEEE Signal Processing Letters* 24.12 (2017), pp. 1872–1876.

[145] A. Kardoost, S. Müller, J. Weickert, and M. Keuper. "Object segmentation tracking from generic video cues". In: *International Conference on Pattern Recognition*. IEEE. 2021, pp. 623–630.

[146] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active contour models". In: *International Journal of Computer Vision* 1.4 (1988).

[147] A. Kattamis, T. Adel, and A. Weller. "Exploring properties of the deep image prior". In: *NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks*. 2019.

[148] I. Kezurer, S. Z. Kovalsky, R. Basri, and Y. Lipman. "Tight relaxation of quadratic matching". In: 34.5 (2015).

[149] V. Khryashchev, L. Ivanovsky, V. Pavlov, A. Ostrovskaya, and A. Rubtsov. "Comparison of different convolutional neural network architectures for satellite image segmentation". In: *Conference of Open Innovations Association (FRUCT)*. IEEE. 2018, pp. 172–179.

[150] H. Kim, R. Anirudh, K. A. Mohan, and K. Champley. "Extreme few-view ct reconstruction using deep inference". In: *arXiv preprint arXiv:1910.05375* (2019).

[151] J. Kim, J. K. Lee, and K. M. Lee. "Deeply-recursive convolutional network for image super-resolution". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1637–1645.

[152] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[153] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*. Ed. by Y. Bengio and Y. LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[154] E. Kobler, A. Effland, K. Kunisch, and T. Pock. "Total deep variation for linear inverse problems". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7549–7558.

[155] T. C. Koopmans and M. Beckmann. "Assignment problems and the location of economic activities". In: *Econometrica: Journal of the Econometric Society* 25.1 (1957).

[156] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems* 25 (2012), pp. 1097–1105.

[157] S. Kuanar, V. Athitsos, D. Mahapatra, K. Rao, Z. Akhtar, and D. Dasgupta. "Low dose abdominal CT image reconstruction: An unsupervised learning based approach". In: *IEEE International Conference on Image Processing*. IEEE. 2019, pp. 1351–1355.

[158] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quaterly* 2 (1955).

[159] A. Kurakin, I. Goodfellow, and S. Bengio. "Adversarial machine learning at scale". In: *arXiv preprint arXiv:1611.01236* (2016).

[160] A. Kurakin, I. J. Goodfellow, and S. Bengio. "Adversarial examples in the physical world". In: *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[161] J. Lafferty, A. Mccallum, and F. Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: Jan. 2001, pp. 282–289.

[162] E. Laude, T. Wu, and D. Cremers. "A nonconvex proximal splitting algorithm under Moreau-Yosida regularization". In: *International Conference on Artificial Intelligence and Statistics*. Proceedings of Machine Learning Research. 2018, pp. 491–499.

[163] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. "Handwritten digit recognition with a back-propagation network". In: *Advances in Neural Information Processing Systems* 2 (1989).

[164] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998).

[165] V. I. Levenshtein. "On bounds for packings in n-dimensional Euclidean space". In: *Doklady Akademii Nauk*. Vol. 245. 6. Russian Academy of Sciences. 1979, pp. 1299–1303.

[166] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. "Understanding and evaluating blind deconvolution algorithms". In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 1964–1971.

[167] B. Li, S. Liu, W. Xu, and W. Qiu. "Real-time object detection and semantic segmentation for autonomous driving". In: *MIPPR 2017: Automatic Target Recognition and Navigation*. Vol. 10608. International Society for Optics and Photonics. SPIE, 2018, pp. 167–174.

[168] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. "NETT: Solving inverse problems with deep neural networks". In: *Inverse Problems* 36.6 (2020), p. 065005.

[169] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen. "Medical image classification with convolutional neural network". In: *International Conference on Control Automation Robotics & Vision*. IEEE. 2014, pp. 844–848.

[170] X. Li et al. "Transformer-based visual segmentation: A survey". In: *arXiv preprint arXiv:2304.09854* (2023).

[171] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. "Fully convolutional instance-aware semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2359–2367.

[172] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar. "Efficient and interpretable deep blind image deblurring via algorithm unrolling". In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 666–681.

[173] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou. "A survey of convolutional neural networks: analysis, applications, and prospects". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[174] S. Liang, X. Wei, S. Yao, and X. Cao. "Efficient adversarial attacks for visual object tracking". In: *European Conference on Computer Vision*. Springer. 2020, pp. 34–50.

[175] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu. "Defense against adversarial attacks using high-level representation guided denoiser". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1778–1787.

[176] L. Liberti. "Mathematical programming bounds for kissing numbers". In: *Optimization and Decision Science: Methodologies and Applications*. Springer. 2017, pp. 213–222.

[177] C. Liguori et al. "Emerging clinical applications of computed tomography". In: *Medical Devices (Auckland, NZ)* 8 (2015), p. 265.

[178] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[179] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. "Efficient piecewise training of deep structured models for semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3194–3203.

[180] O. Litany, T. Remez, E. Rodolà, A. Bronstein, and M. Bronstein. "Deep functional maps: Structured prediction for dense shape correspondence". In: *IEEE International Conference on Computer Vision*. 2017, pp. 5659–5667.

[181] J. Liu, Y. Sun, C. Eldeniz, W. Gan, H. An, and U. S. Kamilov. "RARE: Image reconstruction using deep priors learned without groundtruth". In: *IEEE Journal of Selected Topics in Signal Processing* 14.6 (2020), pp. 1088–1099.

[182] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov. "Image restoration using total variation regularized deep image prior". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Ieee. 2019, pp. 7715–7719.

[183] N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, and T. Tan. "Accurate iris segmentation in non-cooperative environments using fully convolutional networks". In: *International Conference on Biometrics*. IEEE. 2016, pp. 1–8.

[184] R. Liu, S. Cheng, x. liu, L. Ma, X. Fan, and Z. Luo. "A bridging framework for model optimization and deep propagation". In: *Advances in Neural Information Processing Systems* 31 (2018).

[185] R. Liu, X. Fan, S. Cheng, X. Wang, and Z. Luo. "Proximal alternating direction network: A globally converged deep unrolling framework". In: *AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[186] R. Liu, L. Ma, Y. Wang, and L. Zhang. "Learning converged propagations with deep prior ensemble for image enhancement". In: *IEEE Transactions on Image Processing* 28.3 (2018), pp. 1528–1543.

[187] R. Liu, Y. Zhang, S. Cheng, X. Fan, and Z. Luo. "A theoretically guaranteed deep optimization framework for robust compressive sensing mri". In: *AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4368–4375.

[188] W. Liu, A. Rabinovich, and A. C. Berg. "Parsenet: Looking wider to see better". In: *arXiv preprint arXiv:1506.04579* (2015).

[189] Y. Liu, J. Ma, H. Zhang, J. Wang, and Z. Liang. "Low-mAs X-ray CT image reconstruction by adaptive-weighted TV-constrained penalized re-weighted least-squares". In: *Journal of X-ray Science and Technology* 22.4 (2014), pp. 437–457.

[190] Y. Liu, X. Chen, C. Liu, and D. Song. "Delving into transferable adversarial examples and black-box attacks". In: *arXiv preprint arXiv:1611.02770* (2016).

[191] S. Lohit, D. Liu, H. Mansour, and P. T. Boufounos. "Unrolled projected gradient descent for multi-spectral image fusion". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2019, pp. 7725–7729.

[192] E. M. Loiola, N. M. M. De Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. "A survey for the quadratic assignment problem". In: *European Journal of Operational Research* 176.2 (2007).

[193] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[194] A. Lugmayr, M. Danelljan, L. V. Gool, and R. Timofte. "Srflow: Learning the super-resolution space with normalizing flow". In: *European Conference on Computer Vision*. Springer. 2020, pp. 715–732.

[195] S. Lunz, O. Öktem, and C.-B. Schönlieb. "Adversarial regularizers in inverse problems". In: *Advances in Neural Information Processing Systems* 31 (2018).

[196] Z. Luo, W. Yang, Y. Yuan, R. Gou, and X. Li. "Semantic segmentation of agricultural images: a survey". In: *Information Processing in Agriculture* (2023).

[197] X. F. Ma, M. Fukuhara, and T. Takeda. "Neural network CT image reconstruction method for small amount of projection data". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 449.1-2 (2000), pp. 366–377.

[198] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083* (2017).

[199] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[200] M. Marcus. "Some properties and applications of doubly stochastic matrices". In: *The American Mathematical Monthly* 67.3 (1960), pp. 215–221.

[201] M. Mardani, H. Monajemi, V. Papyan, S. Vasanawala, D. Donoho, and J. Pauly. "Recurrent generative adversarial networks for proximal learning and automated compressive image recovery". In: *arXiv preprint arXiv:1711.10046* (2017).

[202] R. Marin. *Correspondence Learning via Linearly-invariant Embedding (PyTorch)*. `https://github.com/riccardomarin/Diff-FMAPs-PyTorch`. 2022.

[203] R. Marin, M.-J. Rakotosaona, S. Melzi, and M. Ovsjanikov. "Correspondence learning via linearly-invariant embedding". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1608–1620.

[204] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman. "Point registration via efficient convex relaxation". In: *ACM Transactions on Graphics* 35.4 (2016), p. 73.

[205] G. Mataev, P. Milanfar, and M. Elad. "DeepRED: Deep image prior powered by RED". In: *IEEE International Conference on Computer Vision Workshops*. 2019.

[206] J. N. McDonald and N. A. Weiss. *A course in real analysis*. Elsevier, 2004.

[207] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers. "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems". In: *IEEE International Conference on Computer Vision*. 2017, pp. 1781–1790.

[208] G. Mena, D. Belanger, S. Linderman, and J. Snoek. "Learning latent permutations with gumbel-sinkhorn networks". In: *arXiv preprint arXiv:1802.08665* (2018).

[209] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin. "Pulse: Self-supervised photo upsampling via latent space exploration of generative models". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2437–2445.

[210] C. Metzler, P. Schniter, A. Veeraraghavan, and R. Baraniuk. "prDeep: Robust phase retrieval with a flexible deep network". In: *International Conference on Machine Learning*. Proceedings of Machine Learning Research. 2018, pp. 3501–3510.

[211] A. Milioto, P. Lottes, and C. Stachniss. "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 2229–2235.

[212] F. Milletari, N. Navab, and S.-A. Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation". In: *International Conference on 3D Vision*. Ieee. 2016, pp. 565–571.

[213] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. "Image segmentation using deep learning: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2021), pp. 3523–3542.

[214] M. Moeller, T. Mollenhoff, and D. Cremers. "Controlling neural networks via energy dissipation". In: *IEEE International Conference on Computer Vision*. 2019, pp. 3256–3265.

[215] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2574–2582.

[216] V. A. Morozov. "On the solution of functional equations by the method of regularization". In: *Doklady Akademii Nauk*. Vol. 167. Russian Academy of Sciences. 1966, pp. 510–512.

[217] S. Mukherjee, S. Dittmer, Z. Shumaylov, S. Lunz, O. Öktem, and C.-B. Schönlieb. "Learned convex regularizers for inverse problems". In: *arXiv preprint arXiv: 2008.02839* (2020).

[218] D. B. Mumford and J. Shah. "Optimal approximations by piecewise smooth functions and associated variational problems". In: *Communications on Pure and Applied Mathematics* 42.5 (1989).

[219] O. R. Musin. "The kissing number in four dimensions". In: *Annals of Mathematics* (2008), pp. 1–32.

[220] Y. E. Nesterov. "A method of solving a convex programming problem with convergence rate O\bigl(k^2\bigr)". In: *Doklady Akademii Nauk*. Vol. 269. 3. Russian Academy of Sciences. 1983, pp. 543–547.

[221] A. Ng, M. Jordan, and Y. Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in Neural Information Processing Systems* 14 (2001).

[222] C. Nieuwenhuis and D. Cremers. "Spatially varying color distributions for interactive multilabel segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.5 (2012).

[223] C. Nieuwenhuis, S. Hawe, M. Kleinsteuber, and D. Cremers. "Co-sparse textural similarity for interactive segmentation". In: *European Conference on Computer Vision*. Springer. 2014.

[224] T. M. Nimisha, A. Kumar Singh, and A. N. Rajagopalan. "Blur-invariant deep learning for blind-deblurring". In: *IEEE International Conference on Computer Vision*. 2017, pp. 4752–4760.

[225] D. Nogneng and M. Ovsjanikov. "Informative descriptor preservation via commutativity for shape matching". In: *Computer Graphics Forum*. Vol. 36. 2. Wiley Online Library. 2017, pp. 259–267.

[226] D. Obmann, L. Nguyen, J. Schwab, and M. Haltmeier. "Augmented NETT regularization of inverse problems". In: *Journal of Physics Communications* 5.10 (2021), p. 105002.

[227] D. Obmann, J. Schwab, and M. Haltmeier. "Deep synthesis network for regularizing inverse problems". In: *Inverse Problems* 37.1 (2020), p. 015005.

[228] A. M. Odlyzko and N. J. Sloane. "New bounds on the number of unit spheres that can touch a unit sphere in n dimensions". In: *Journal of Combinatorial Theory, Series A* 26.2 (1979), pp. 210–214.

[229] S. Ono. "Primal-dual plug-and-play image restoration". In: *IEEE Signal Processing Letters* 24.8 (2017), pp. 1108–1112.

[230] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. "Functional maps: a flexible representation of maps between shapes". In: *ACM Transactions on Graphics* 31.4 (2012), pp. 1–11.

[231] C. Papazov and D. Burschka. "Deformable 3D shape registration based on local similarity transforms". In: *Computer Graphics Forum*. Vol. 30. 5. Wiley Online Library. 2011, pp. 1493–1502.

[232] N. Papernot, P. McDaniel, and I. Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples". In: *arXiv preprint arXiv:1605.07277* (2016).

[233] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. "The limitations of deep learning in adversarial settings". In: *IEEE Symposium on Security and Privacy*. IEEE. 2016, pp. 372–387.

[234] N. Parikh and S. Boyd. "Proximal algorithms". In: *Foundations and trends® in Optimization* 1.3 (2014), pp. 127–239.

[235] M. Paschali, S. Conjeti, F. Navarro, and N. Navab. "Generalizability vs. robustness: investigating medical imaging networks using adversarial examples". In: *Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 493–501.

[236] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. "Enet: A deep neural network architecture for real-time semantic segmentation". In: *arXiv preprint arXiv:1606.02147* (2016).

[237] F. Pfender and G. M. Ziegler. "Kissing numbers, sphere packings, and some unexpected proofs". In: (2004).

[238] U. Pinkall and K. Polthier. "Computing discrete minimal surfaces and their conjugates". In: *Experimental Mathematics* 2.1 (1993), pp. 15 –36.

[239] M. Prakash, A. Krull, and F. Jug. "Fully unsupervised diversity denoising with convolutional variational autoencoders". In: *arXiv preprint arXiv:2006.06072* (2020).

[240] G. Puy, A. Boulch, and R. Marlet. "Flot: Scene flow on point clouds guided by optimal transport". In: *European Conference on Computer Vision*. Springer. 2020, pp. 527–544.

[241] J. Radon. "On the determination of functions from their integral values along certain manifolds". In: *IEEE Transactions on Medical Imaging* 5.4 (1986), pp. 170–176.

[242] A. Raghunathan, J. Steinhardt, and P. Liang. "Certified defenses against adversarial examples". In: *arXiv preprint arXiv:1801.09344* (2018).

[243] K. Ramesh, G. K. Kumar, K Swapna, D. Datta, and S. S. Rajest. "A review of medical image segmentation algorithms". In: *EAI Endorsed Transactions on Pervasive Health and Technology* 7.27 (2021), e6–e6.

[244] B. Rasti, B. Koirala, P. Scheunders, and P. Ghamisi. "UnDIP: Hyperspectral unmixing using deep image prior". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–15.

[245] E. T. Reehorst and P. Schniter. "Regularization by denoising: Clarifications and new interpretations". In: *IEEE Transactions on Computational Imaging* 5.1 (2018), pp. 52–67.

[246] K. Ren, T. Zheng, Z. Qin, and X. Liu. "Adversarial attacks and defenses in deep learning". In: *Engineering* 6.3 (2020), pp. 346–360.

[247] J. Rick Chang, C.-L. Li, B. Poczos, B. Vijaya Kumar, and A. C. Sankaranarayanan. "One network to solve them all–solving linear inverse problems using deep projection models". In: *IEEE International Conference on Computer Vision*. 2017, pp. 5888–5897.

[248] T. R. Rockafellar. *Convex analysis*. Vol. 11. Princeton university press, 1997.

[249] T. R. Rockafellar. "Monotone operators and the proximal point algorithm". In: *SIAM Journal on Control and Optimization* 14.5 (1976), pp. 877–898.

[250] T. R. Rockafellar and R. J.-B. Wets. *Variational analysis*. Vol. 317. Springer Science & Business Media, 2009.

[251] E. Rodolà, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. "A game-theoretic approach to deformable shape matching". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012.

[252] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. "Partial functional correspondence". In: *Computer Graphics Forum*. Vol. 36. 1. Wiley Online Library. 2017, pp. 222–236.

[253] Y. Romano, M. Elad, and P. Milanfar. "The little engine that could: Regularization by denoising (RED)". In: *SIAM Journal on Imaging Sciences* 10.4 (2017), pp. 1804–1844.

[254] A. Rond, R. Giryes, and M. Elad. "Poisson inverse problems by the plug-and-play scheme". In: *Journal of Visual Communication and Image Representation* 41 (2016), pp. 96–108.

[255] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

[256] C. Rother, V. Kolmogorov, and A. Blake. ""GrabCut" interactive foreground extraction using iterated graph cuts". In: *ACM Transactions on Graphics* 23.3 (2004).

[257] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". In: 60.1-4 (1992), pp. 259–268.

[258] A. Ruszczynski. *Nonlinear optimization*. Princeton university press, 2011.

[259] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin. "Plug-and-play methods provably converge with properly trained denoisers". In: *International Conference on Machine Learning*. Proceedings of Machine Learning Research. 2019, pp. 5546–5557.

[260] Y. Sahillioğlu. "Recent advances in shape correspondence". In: *The Visual Computer* 36.8 (2020), pp. 1705–1721.

[261] S. Salti, F. Tombari, and L. Di Stefano. "SHOT: Unique signatures of histograms for surface and texture description". In: *Computer Vision and Image Understanding* 125 (2014), pp. 251–264.

[262] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould. "Deeppermnet: Visual permutation learning". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3949–3957.

[263] L. K. Saul. "A geometrical connection between sparse and low-rank matrices and its application to manifold learning". In: *Transactions on Machine Learning Research* (2022).

[264] L. K. Saul. "A nonlinear matrix decomposition for mining the zeros of sparse data". In: *SIAM Journal on Mathematics of Data Science* 4.2 (2022), pp. 431–463.

[265] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert. "Attention gated networks: Learning to leverage salient regions in medical images". In: *Medical image analysis* 53 (2019), pp. 197–207.

[266] U. Schmidt and S. Roth. "Shrinkage fields for effective image restoration". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2774–2781.

[267] K. Schütte and B. L. van der Waerden. "Das problem der dreizehn Kugeln". In: *Mathematische Annalen* 125.1 (1952), pp. 325–334.

[268] M. Seelbach Benkner, Z. Lähner, V. Golyanik, C. Wunderlich, C. Theobalt, and M. Moeller. "Q-Match: Iterative Shape Matching via Quantum Annealing". In: *IEEE International Conference on Computer Vision*. 2021.

[269] G. Seraghiti, A. Awari, A. Vandaele, M. Porcelli, and N. Gillis. "Accelerated Algorithms for Nonlinear Matrix Decomposition with the ReLU function". In: *arXiv preprint arXiv:2305.08687* (2023).

[270] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition". In: *ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 1528–1540.

[271] D. Shen, G. Wu, and H.-I. Suk. "Deep learning in medical image analysis". In: *Annual review of biomedical engineering* 19 (2017), pp. 221–248.

[272] L. A. Shepp and B. F. Logan. "The Fourier reconstruction of a head section". In: *IEEE Transactions on Nuclear Science* 21.3 (1974), pp. 21–43.

[273] W. Shi et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1874–1883.

[274] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang. "A comparative study of real-time semantic segmentation for autonomous driving". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 587–597.

[275] E. Y. Sidky, C.-M. Kao, and X. Pan. "Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT". In: *Journal of X-ray Science and Technology* 14.2 (2006), pp. 119–139.

[276] E. Y. Sidky and X. Pan. "Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization". In: *Physics in Medicine & Biology* 53.17 (2008), p. 4777.

[277] S. H. Silva, P. Rad, N. Beebe, K.-K. R. Choo, and M. Umapathy. "Cooperative unmanned aerial vehicles with privacy preserving deep vision for real-time object identification and tracking". In: *Journal of parallel and distributed computing* 131 (2019), pp. 147–160.

[278] R. Sinkhorn. "A relationship between arbitrary positive matrices and doubly stochastic matrices". In: *The annals of mathematical statistics* 35.2 (1964), pp. 876–879.

[279] R. Sinkhorn and P. Knopp. "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348.

[280] H. Sommerhoff, A. Kolb, and M. Moeller. "Energy dissipation with plug-and-play priors". In: *NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks* (2019).

[281] R. Strudel, R. Garcia, I. Laptev, and C. Schmid. "Segmenter: Transformer for semantic segmentation". In: *IEEE International Conference on Computer Vision*. 2021, pp. 7262–7272.

[282] M. Su and H Xu. "Remarks on the gradient-projection algorithm". In: *Journal of Nonlinear Analysis and Optimization: Theory & Applications* 1.1 (2010), pp. 35–43.

[283] J. Sun, M. Ovsjanikov, and L. Guibas. "A Concise and Provably Informative Multi-Scale Signature-Based on Heat Diffusion". In: *Computer Graphics Forum (CGF)* 28 (2009), 1383–1392.

[284] J. Sun, H. Li, Z. Xu, et al. "Deep ADMM-Net for compressive sensing MRI". In: *Advances in Neural Information Processing Systems* 29 (2016).

[285] R.-Y. Sun. "Optimization for deep learning: An overview". In: *Journal of the Operations Research Society of China* 8.2 (2020), pp. 249–294.

[286] Y. Sun, J. Liu, and U. Kamilov. "Block coordinate regularization by denoising". In: *Advances in Neural Information Processing Systems* 32 (2019).

[287] Y. Sun, B. Wohlberg, and U. S. Kamilov. "An online plug-and-play algorithm for regularized image reconstruction". In: *IEEE Transactions on Computational Imaging* 5.3 (2019), pp. 395–408.

[288] R. Sundararaman, G. Pai, and M. Ovsjanikov. "Implicit Field Supervision for Robust Non-Rigid Shape Matching". In: *European Conference on Computer Vision*. 2022.

[289] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe. "Fast exact and approximate geodesics on meshes". In: *ACM Transactions on Graphics* 24.3 (2005), pp. 553–560.

[290] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". In: *arXiv preprint arXiv: 1312.6199* (2013).

[291] S. Takabe and T. Wadayama. "Theoretical interpretation of learned step size in deep-unfolded gradient descent". In: *arXiv preprint arXiv:2001.05142* (2020).

[292] C. Tian, Y. Xu, and W. Zuo. "Image denoising using deep CNN with batch renormalization". In: *Neural Networks* 121 (2020), pp. 461–473.

[293] Y. Tian, D. Su, S. Lauria, and X. Liu. "Recent advances on loss functions in deep learning for computer vision". In: *Neurocomputing* 497 (2022), pp. 129–158.

[294] Z. Tian, X. Jia, K. Yuan, T. Pan, and S. B. Jiang. "Low-dose CT reconstruction via edge-preserving total variation regularization". In: *Physics in Medicine & Biology* 56.18 (2011), p. 5949.

[295] Y.-H. Tseng and S.-S. Jan. "Combination of computer vision detection and segmentation for autonomous driving". In: *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE. 2018, pp. 1047–1052.

[296] D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Deep image prior". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9446–9454.

[297] J. M. J. Valanarasu, P. Oza, I. Hacihaliloglu, and V. M. Patel. "Medical transformer: Gated axial-attention for medical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention*. Springer. 2021, pp. 36–46.

[298] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis. "Compressed sensing with deep image prior and learned regularization". In: *arXiv preprint arXiv:1806.06438* (2018).

[299] G. Vardi. "On the implicit bias in deep-learning algorithms". In: *Communications of the ACM* 66.6 (2023), pp. 86–93.

[300] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. "Learning from Synthetic Humans". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[301] A. Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[302] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. "Plug-and-play priors for model based reconstruction". In: *IEEE Global Conference on Signal and Information Processing*. IEEE. 2013, pp. 945–948.

[303] M. Vestner et al. "Efficient deformable shape correspondence via kernel matching". In: *International Conference on 3D Vision*. 2017.

[304] T. Vu et al. "Deep image prior for undersampling high-speed photoacoustic microscopy". In: *Photoacoustics* 22 (2021), p. 100266.

[305] N. J. Walkington. "Nesterov's Method for Convex Optimization". In: *SIAM Review* 65.2 (2023), pp. 539–562.

[306] G. Wang, W. Li, S. Ourselin, and T. Vercauteren. "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer. 2018, pp. 178–190.

[307] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, and A. K. Nandi. "Medical image segmentation using deep learning: A survey". In: *IET Image Processing* 16.5 (2022), pp. 1243–1267.

[308] R. Wang, J. Yan, and X. Yang. "Combinatorial learning of robust deep graph matching: an embedding based approach". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[309] R. Wang, J. Yan, and X. Yang. "Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), pp. 5261–5279.

[310] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, and L. J. Latecki. "Lednet: A lightweight encoder-decoder network for real-time semantic segmentation". In: *IEEE International Conference on Image Processing*. IEEE. 2019, pp. 1860–1864.

[311] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. "Deep networks for image super-resolution with sparse prior". In: *IEEE International Conference on Computer Vision*. 2015, pp. 370–378.

[312] K. Wilhelm. "Auf dem Highway ohne Hände am Lenkrad". In: *Tagesschau* (Jan. 10, 2023). URL: `https://www.tagesschau.de/wirtschaft/autonomes-fahren-kalifornien-tesla-mercedes-100.html` (visited on 09/10/2023).

[313] Z. Wu, Y. Sun, J. Liu, and U. Kamilov. "Online regularization by denoising with applications to phase retrieval". In: *IEEE International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[314] T. Würfl, F. C. Ghesu, V. Christlein, and A. Maier. "Deep learning computed tomography". In: *Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 432–440.

[315] W. Xia, Z. Lu, Y. Huang, Y. Liu, H. Chen, J. Zhou, and Y. Zhang. "CT reconstruction with PDF: parameter-dependent framework for data from multiple geometries and dose levels". In: *IEEE Transactions on Medical Imaging* 40.11 (2021), pp. 3065–3076.

[316] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. "Adversarial examples for semantic segmentation and object detection". In: *IEEE International Conference on Computer Vision*. 2017, pp. 1369–1378.

[317] H. Xie, H. Shan, W. Cong, X. Zhang, S. Liu, R. Ning, and G. Wang. "Dual network architecture for few-view CT-trained on ImageNet data and transferred for medical imaging". In: *Developments in X-ray Tomography XII*. Vol. 11113. International Society for Optics and Photonics. 2019, p. 111130V.

[318] J. Xie, L. Xu, and E. Chen. "Image denoising and inpainting with deep neural networks". In: *Advances in Neural Information Processing Systems* 25 (2012).

[319] S. Xie and Z. Tu. "Holistically-nested edge detection". In: *IEEE International Conference on Computer Vision*. 2015, pp. 1395–1403.

[320] L. Xu, S. Zheng, and J. Jia. "Unnatural l0 sparse representation for natural image deblurring". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1107–1114.

[321] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang. "Low-dose X-ray CT reconstruction via dictionary learning". In: *IEEE Transactions on Medical Imaging* 31.9 (2012), pp. 1682–1697.

[322] W. Xu, Y. Xu, T. Chang, and Z. Tu. "Co-scale conv-attentional image transformers". In: *IEEE International Conference on Computer Vision*. 2021, pp. 9981–9990.

[323] W. Xu, D. Evans, and Y. Qi. "Feature squeezing: Detecting adversarial examples in deep neural networks". In: *arXiv preprint arXiv:1704.01155* (2017).

[324] Y. Yang, J. Sun, H. Li, and Z. Xu. "ADMM-CSNet: A deep learning approach for image compressive sensing". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.3 (2018), pp. 521–538.

[325] Y. Yang, J. Sun, H. Li, and Z. Xu. "Deep ADMM-Net for Compressive Sensing MRI". In: *Advances in Neural Information Processing Systems* 29 (2016).

[326] S. Yoo and F.-F. Yin. "Dosimetric feasibility of cone-beam CT-based treatment planning compared to CT-based treatment planning". In: *International Journal of Radiation Oncology* Biology* Physics* 66.5 (2006), pp. 1553–1561.

[327] T. Yu, R. Wang, J. Yan, and B. Li. "Learning deep graph matching with channel-independent embedding and hungarian attention". In: *International Conference on Learning Representations*. 2019.

[328] T. Yu, R. Wang, J. Yan, and B. Li. "Learning deep graph matching with channel-independent embedding and hungarian attention". In: *International Conference on Learning Representations*. 2020.

[329] A. Zanfir and C. Sminchisescu. "Deep learning of graph matching". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2684–2693.

[330] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte. "Plug-and-play image restoration with deep denoiser prior". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2021), pp. 6360–6376.

[331] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising". In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155.

[332] K. Zhang, W. Zuo, and L. Zhang. "Deep plug-and-play super-resolution for arbitrary blur kernels". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1671–1681.

[333] K. Zhang, W. Zuo, and L. Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising". In: *IEEE Transactions on Image Processing* 27.9 (2018), pp. 4608–4622.

[334] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen. "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation". In: *NeuroImage* 108 (2015), pp. 214–224.

[335] Y. Zhang, J. Chu, L. Leng, and J. Miao. "Mask-refined R-CNN: A network for refining object details in instance segmentation". In: *Sensors* 20.4 (2020), p. 1010.

[336] Z. Zhang and W. S. Lee. "Deep graphical feature learning for the feature matching problem". In: *IEEE International Conference on Computer Vision*. 2019, pp. 5087–5096.

[337] Z. Zhang, X. Liang, X. Dong, Y. Xie, and G. Cao. "A sparse-view CT reconstruction method based on combination of DenseNet and deconvolution". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1407–1417.

[338] Z. Zhang, Z. Zhang, Y. Zhou, Y. Shen, R. Jin, and D. Dou. "Adversarial attacks on deep graph matching". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 20834–20851.

[339] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz. "Semidefinite programming relaxations for the quadratic assignment problem". In: *Journal of Combinatorial Optimization* 2 (1998), pp. 71–109.

[340] S. Zheng et al. "Conditional random fields as recurrent neural networks". In: *IEEE International Conference on Computer Vision*. 2015, pp. 1529–1537.

[341] T. Zheng, C. Chen, and K. Ren. "Distributionally adversarial attack". In: *AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 2253–2260.

[342] Y. Zhong and W. Deng. "Towards transferable adversarial attack against deep face recognition". In: *IEEE Transactions on Information Forensics and Security* 16 (2020), pp. 1452–1466.

[343] Q. Zhou, Q. Wang, Y. Bao, L. Kong, X. Jin, and W. Ou. "Laednet: A lightweight attention encoder–decoder network for ultrasound medical image segmentation". In: *Computers and Electrical Engineering* 99 (2022), p. 107777.

[344] C. Zong. "The kissing numbers of convex bodies—a brief survey". In: *Bulletin of the London Mathematical Society* 30.1 (1998), pp. 1–10.